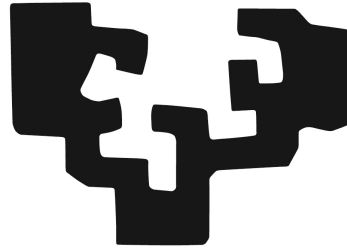


eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

Department of Computer Architecture and Technology

# A Web Transcoding Framework Based on User Behaviour Evaluation

by  
Xabier Valencia Parafita

Supervised by  
Dr. Julio Abascal and Dra. Myriam Arrue

Donostia, 2017



To my parents



# Agradecimientos

Quisiera aprovechar estas líneas para agradecer a todas las personas que me han ayudado durante el desarrollo de esta tesis.

Primero me gustaría dar las gracias a mis directores Julio y Myriam por guiarme por este largo sendero y por darme la oportunidad de formar parte del grupo de investigación Egokituz.

Cómo no, también he de agradecer al resto de compañeros de Egokituz, DGS, y Aldapa ya que sin su ayuda y apoyo esta tesis no habría sido posible. Una mención especial para Edu por todas las horas que hemos trabajado juntos en el laboratorio 302, 303, experimentos..., *ありがとう*. Mucha gracias también a Unai M., por todo el trabajo que sacó adelante durante el tiempo que estuvo con nosotros. No puedo olvidarme de los compañeros que han pasado por el grupo: Borja, Raúl, Idoia y Amaia. ¡¡Feliz año Borja!! Y cómo no, a los compañeros de *tuper txoko* y laboratorio, Unai B. (si está escrito en una tesis deben ser croquetas), Carlos y Ugaitz. Tampoco me puedo olvidar de Ana y su compañero de trabajo por todos los momentos que hemos vivido en el 302, ¿un té? Al recién llegado, Xabi (*ゴジラ!!!*) mucha suerte. Por último agradecer a Izaro y Elsa por todos los buenos ratos y alguno no tan bueno, la próxima vez seré mejor guía.

Gracias a Carlos y Luis por acogerme en vuestro grupo de la Universidad de Lisboa durante la estancia. Muchas gracias José por ayudarme durante toda la estancia a que me sintiera como si estuviera en casa. Tampoco puedo olvidar a Tiago, André y Hugo: muito obrigado.

Esta tesis tampoco hubiera sido posible sin la ayuda de diversas asociaciones de personas con discapacidad, tales como Elkartu (con Norma siempre ayudando). Y qué decir de los participantes en los experimentos, siempre dispuestos a aportar su inestimable granito de arena. Tampoco hubiera sido posible sin la colaboración con la universidad Carlos III de Madrid, con Lourdes siempre al pie del cañón.

Muchas gracias también a Jon Haitz, Gorka, Ane, Eider, Andoni, Beñat y los demás compañeros que tuve en Vicomtech, por iniciarme en el mundillo de la investigación: os espero en el Urederra, pero el bocata lo ponéis vosotros.

No podría dejar pasar esta oportunidad de agradecer a los músicos que me han acompañado durante horas, horas, horas y más horas. Sin duda han hecho que todo sea mucho más fácil. Deftones, NiN, Machine Head, Textures, While She Sleeps, Pendulum, Black Sun Empire o Camo & Krooked son algunos de los grupos que forman parte de la banda sonora de esta tesis.

Por último, quiero agradecer a mi madre, padre, hermanas, suegra, suegro, cuñados y cuñadas, que me hayan aguantado durante todo este trayecto y por ayudarme a seguir adelante. Especial mención a Judith que ha padecido más que nadie los diversos *deadlines* durante la tesis. Ahora estaré mas disponible ... creo.



# Acknowledgements

This PhD research was developed in the Egokituz Laboratory of Human-Computer Interaction for Special needs, which is part of the ADIAN research group, supported by the Department of Education, Universities and Research of the Basque Government under Grant IT980-16. It has been partially supported by the Ministry of Economy and Competitiveness of the Spanish Government and by the European Regional Development Fund (projects TIN2013-41123-P and TIN2014-52665-C2-1-R).





# Abstract

Accessibility to the web is mainly focused on improving design and evaluation methods (based on guidelines) to create new web pages accessible to all. However, this approach does not solve access to the pages that have been designed without taking into account its accessibility. In addition, given the diversity of users and access devices, even accessible pages can present barriers to specific users using certain devices. As an alternative, *transcoding* is a technique that automatically transforms inaccessible page code into accessible code. It is also possible to apply techniques of adaptation of the page to the needs of specific users in the process of recoding.

This thesis presents a framework that allows to convert into accessible web pages that initially are inaccessible, and to customize them to the needs of specific users. For this, we use *transcoding* techniques based on the evaluation of user behaviour.

Based on the proposed conceptual model, two components have been designed: a *transcoding* tool and a tool for conducting remote evaluations with users with disabilities.

The application of the *transcoding* technique requires:

- web pages marked with semantic meta-data, for which we have used an extension of the WAI-ARIA markup language.
- a set of adaptation techniques, for which we have made a selection from the literature and our own tests with real users.
- a system of reasoning that allows to decide which adaptation techniques to apply and to which annotated elements can be applied, for which we have designed a specific ontology.
- a model that contains the characteristics of the user (restrictions and preferences) and the device used to access the web.

From these elements, the *transcoding* tool modifies the code of the original page (not accessible) to make it a page accessible and adapted to the requirements of each specific user.

Both the design of adaptation techniques and the validation of the generated pages require user testing. To this end, a tool for remote experimentation with the participation of people with disabilities has been developed. This tool uses the Experimental Specification Language (ESL) and the Experimental Session Control Language (ESCL) that have been designed and specified using XML.

The different evaluations, carried out with people with reduced mobility and with vision problems, demonstrated how the proposed and developed framework can be used to gain a better knowledge about how the web is used. This information was applied to create new adaptation techniques that we implemented using the *transcoding* tool. The evaluation with actual users of the pages created showed that this technique is valid to improve its usability and accessibility.



# Resumen

La accesibilidad a la web se ha centrado principalmente en mejorar métodos de diseño y evaluación (a partir de pautas) que permitan crear nuevas páginas web accesibles para todos. Sin embargo este enfoque no resuelve el acceso a las páginas que han sido diseñadas sin tener en cuenta su accesibilidad. Además, dada la diversidad de usuarios y de dispositivos de acceso, incluso las páginas accesibles pueden presentar barreras para usuarios concretos que usan determinados dispositivos. Como alternativa, el *transcoding* es una técnica que transforma de manera automática el código de páginas inaccesibles en código accesible. En el proceso de recodificación, se puede además aplicar técnicas de adaptación de la página a las necesidades de usuarios específicos.

En esta tesis se presenta un marco de trabajo que permite hacer accesibles y adaptar a las personas con necesidades especiales las páginas web inaccesibles. Para ello utilizamos técnicas de *transcoding* basadas en métodos de evaluación del comportamiento del usuario.

A partir del modelo conceptual propuesto, se han diseñado dos componentes: una herramienta de *transcoding* y una herramienta para la realización de evaluaciones remotas con usuarios con discapacidad.

La aplicación de la técnica de *transcoding* requiere:

- páginas web hayan sido marcadas con meta-data semánticos, para lo que hemos utilizado una extensión propia del lenguaje de marcado WAI-ARIA
- un conjunto de técnicas de adaptación, para lo que hemos hecho una selección a partir de la literatura y de nuestros propios test con usuarios reales.
- un sistema de razonamiento que permita decidir qué técnicas de adaptación aplicar y a qué elementos anotados se pueden aplicar, para lo que hemos diseñado una ontología específica.
- un modelo que contenga las características del usuario (restricciones y preferencias) y del dispositivo que está usando para acceder a la web.

A partir de estos elementos, la herramienta de *transcoding* modifica el código de la página original (no accesible) para convertirla en una página accesible y adaptada a los requisitos de cada usuario específico.

Tanto el diseño de técnicas de adaptación, como la validación de las páginas generadas requieren la realización de test con usuarios. Para ello se ha desarrollado una herramienta de experimentación remota con participación de personas con discapacidad. Esta herramienta utiliza el Lenguaje de Especificación Experimental (ESL) y el Lenguaje de Control de la Sesión Experimental (ESCL) que han sido diseñados y especificados mediante XML.

Las diferentes evaluaciones, llevadas a cabo con personas con movilidad reducida y con problemas de visión, han servido para demostrar cómo puede ser utilizado el marco de trabajo propuesto y desarrollado para obtener un mayor conocimiento sobre cómo se usa la web. Esa información se ha usado para crear nuevas técnicas de adaptación que hemos aplicado mediante la herramienta de *transcoding*. La evaluación con usuarios reales de las páginas creadas demuestra que esta técnica es válida para mejorar su usabilidad y accesibilidad.



# Laburpena

Web irisgarritasuna arloko ikerketa lanak, denontzat irisgarriak diren webak sortu ahal izateko diseinu edo (gidaleroen arabera) ebaluazioen hobekuntzan oinarritu izan dira nagusiki. Oro har, irisgarritasun irizpideak kontutan hartu gabe diseinaturiko web orritan dauden arazoak ez dituzte konpontzen. Gainera, erabiltzaile eta gailuen aniztasuna kontutan izanda, nahiz eta web orria irisgarria izan, eragozpenak ager daitezke erabiltzaile eta gailu zehatz batentzat. Arazo hauek konpontzeko aukera bezala, webaren kodea automatikoki birmoldatuz, irisgarriak ez diren webak irisgarriak bihurtu ditzakeen teknika bat, *transcoding*-a izan daiteke. Irisgarria bihurtzeaz gain, birkodetze prozesuan egokitzapen teknikak aplikatu daitezke, web orria erabiltzailearen beharretara moldatu ahal izateko.

Tesi honetan, irisgarriak ez diren webak, behar bereziak dituzten pertsonen egokitu eta irisgarria bihurtzea ahalbidetzen duen lan esparru bat aurkezten da. Helburu hau lortzeko, erabiltzaile jokaeraren ebaluazio metodoetan oinarritutako *transcoding* teknikak erabiltzen dira.

Proposatzen den eredu kontzeptualetik bi osagai diseinatu eta garatu dira: *transcoding* teknikak aplikatzen dituen tresna bat eta pertsona desgaituekin urruneko ebaluazioak egiteko aplikazioa.

Transcoding teknikaren aplikazioak ondorengo beharrak ditu:

- Web orriak meta-data semantikoarekin markatuta egotea, horretarako WAI-ARIA etiketatze lengoaiaren luzapen bat erabili dugu
- Egokitzapen teknika multzo bat, literaturan aurkitutako eta gure benetako erabiltzaile testetan oinarritutako egokitzapen teknikak aukeratu ditugu
- Arrazoitze sistema bat, aplikatu behar diren egokitzapen teknikak eta zein anotatutako elementuen gain aplikatu behar diren erabakitzen duena. Sistema hau apropos diseinatu dugun ontologia batean oinarritzen da
- Erabiltzaileen ezaugarriak (murriztapenak eta lehentasunak) eta internet atzitzeko erabiltzen den gailua jasotzen duen modeloa

Elementu hauen arabera *transcoding* tresnak web orriaren kodea (ez irisgarria) moldatzen du erabiltzaile bakoitzaren beharretara egokituta eta irisgarria izan dadin.

Bai egokitzapen tekniken diseinurako eta baita lorturiko web orrien balidazioarako ere, erabiltzaileekin testak egitea beharrezkoa da. Horretarako desgaitasunak dituzten pertsonekin lan egiteko urruneko esperimendazio aplikazio bat garatu da. Aplikazio honek, tesi honetan diseinatu diren eta XML-en oinarritzen diren Esperimendazio Zehaztapen Lengoaia (ESL) eta Saio Esperimentalaren Kontrol Lengoaia (ESCL) erabiltzen ditu.

Mugikortasun eta ikuspen murriztasunak dituzten pertsonekin eginiko ebaluazio desberdinek, tesi honetan proposaturiko lanaren frogapenerako eta baita erabiltzaile hauen web erabilpenaren ezaugarri bereziak aztertzeko ere balio izan dute. Informazio hau erabili da, egokitzapen teknika berriak sortzeko eta *transcoding* tresnan erabiltzeko. Egokitutako web orriekin eginiko erabiltzaile ebaluazioek, aplikatutako tekniken balioztatze eta erabilgarritasuna eta irisgarritasuna hobetzeko balio izan dute.



# Contents

<b>Chapter 1 Introduction</b> .....	<b>1</b>
1.1. Accessibility, Usability and User Experience .....	1
1.2. How Can User Experience, Accessibility or Usability Be Ensured? .....	2
1.3. How Can Accessibility and Usability Be Measured? .....	4
1.3.1. Experimental Research.....	5
1.3.2. Experimental Designs.....	5
1.3.3. Moderated and Unmoderated Experiments.....	6
1.4. Research Motivation.....	7
1.5. Summary.....	7
<b>Chapter 2 Background</b> .....	<b>9</b>
2.1. Improving Accessibility and Usability Through Personalized Interfaces .....	9
2.1.1. Improving Accessibility from Design .....	9
2.1.2. Transcoding to Improve Accessibility .....	10
2.2. Measuring the UX, Usability and Accessibility .....	14
2.2.1. Architecture of Tools for Web User Testing.....	15
2.2.2. Web User Testing Tools in the Literature .....	16
2.2.3. Commercial Tools for Web User Testing .....	17
2.2.4. Tools Classification by Requirements for Including People with Disabilities in User Testing .....	18
<b>Chapter 3 Conceptual Model</b> .....	<b>21</b>
3.1. Definitions .....	21
3.2. Prerequisites.....	23
3.2.1. Annotation .....	23
3.2.2. Evaluation.....	23
3.2.3. Reasoning.....	23
3.2.4. Adaptation Techniques.....	24
3.3. User, Environment and Device Model .....	24
3.4. User Interface Transformation Process .....	26
3.5. Summary.....	26
<b>Chapter 4 Composing Accessible User Testing</b> .....	<b>29</b>
4.1. Evaluation .....	29
4.1.1. Empirical Evaluation.....	30
4.2. Introduction to XUL Interfaces .....	33
4.2.1. XUL.....	33
4.2.2. XUL Application.....	34
4.2.3. Preliminary Test.....	34
4.3. Data Gathering and Analysis with the RemoTest Platform.....	38
4.3.1. Experimenter Module.....	39
4.3.2. Coordinator Module .....	40
4.3.3. Participant Module .....	41
4.3.4. Results Viewer Module.....	41
4.3.5. Delimiting Pointing Trajectories.....	44
4.4. Assessing Accessibility, Usability and User Experience of RemoTest.....	46
4.4.1. Participants .....	46
4.4.2. Tasks.....	47
4.4.3. Materials & Equipment .....	47

4.4.4.	Procedure.....	47
4.4.5.	Results .....	47
4.4.6.	Discussion .....	50
4.5.	Analysing Validity of Data and Obtaining Characteristics from Cursor Measures .....	50
4.5.1.	Research Questions .....	50
4.5.2.	Participants .....	51
4.5.3.	Equipment .....	51
4.5.4.	Stimuli .....	51
4.5.5.	Tasks.....	51
4.5.6.	Procedure.....	52
4.5.7.	Results.....	52
4.5.8.	Discussion .....	56
4.6.	Case Study 1: Users with Low Vision.....	57
4.6.1.	Participants .....	57
4.6.2.	Materials & Equipment .....	57
4.6.3.	Stimuli .....	57
4.6.4.	Tasks.....	58
4.6.5.	Results .....	58
4.6.6.	Discussion.....	59
4.7.	Summary.....	60
<b>Chapter 5 Adaptation Techniques for Making the Web Suitable for User Needs..61</b>		
5.1.	Adaptation Techniques Classification .....	61
5.2.	Adaptation Techniques for People with Motor Impairments Using Touch Screen Devices 64	
5.2.1.	Perceivable - Information and User Interface Components Must Be Presentable to Users in Ways They Can Perceive .....	64
5.2.2.	Operable -User Interface Components and Navigation Must Be Operable .....	64
5.2.3.	Understandable - Information and the Operation of User Interface Must Be Understandable.....	66
5.3.	Adaptation Techniques For People with Low Vision.....	67
5.3.1.	Perceivable - Information and User Interface Components Must Be Presentable to Users in Ways They Can Perceive .....	67
5.3.2.	Operable -User Interface Components and Navigation Must Be Operable .....	67
5.3.3.	Understandable - Information and the Operation of User Interface Must Be Understandable.....	68
5.4.	Summary.....	69
<b>Chapter 6 Transcoding the Web to Enhance Accessibility and Usability .....71</b>		
6.1.	Model Access, a Transcoding System .....	71
6.1.1.	Architecture .....	72
6.1.2.	Design Considerations.....	78
6.1.3.	Discapnet Website Annotation Process.....	79
6.2.	Case Study 1: Adapting the Web for People with Upper Body Motor Impairments Using a Touch Screen Mobile Device.....	81
6.2.1.	Participants .....	81
6.2.2.	Positioning the Tablet.....	82
6.2.3.	Interacting with the Tablet. ....	82
6.2.4.	Tasks and Materials.....	82
6.2.5.	Procedure.....	85
6.2.6.	Results .....	86
6.2.7.	Discussion .....	92
6.3.	Case Study 2: Adapting the Web for People with Low Vision .....	93
6.3.1.	Participants .....	93
6.3.2.	Stimuli .....	94



- 6.3.3. Procedure..... 94
- 6.3.4. Materials & Equipment ..... 94
- 6.3.5. Results ..... 95
- 6.3.6. Discussion ..... 98
- 6.3.7. Feedback from Participants ..... 98
- 6.4. Summary..... 99

**Chapter 7 Conclusions & Future Work..... 101**

- 7.1. Conclusions..... 101
  - 7.1.1. RemoTest..... 101
  - 7.1.2. Model Access ..... 101
  - 7.1.3. Case studies ..... 102
- 7.2. Contributions ..... 102
  - 7.2.1. Publications ..... 103
- 7.3. Future Work..... 104

**References ..... 105**



# Figures

Figure 1. Interaction of the content transformation components .....	22
Figure 2. Device, enviroment and user models .....	25
Figure 3. Input and output devices .....	25
Figure 4. UI transformation process.....	27
Figure 5. Evaluation components.....	30
Figure 6. Empirical evaluation components.....	31
Figure 7. Demographic information form .....	35
Figure 8. General information about the performance of a participant in a visited web page .....	42
Figure 9. Comparing the cursor trajectory distance parameter of a participant .....	42
Figure 10. Distance to target” (Left) and “Distance from intention to target” (Right) ...	43
Figure 11. Speed chart (left) and acceleration chart (right) of a participant (U10 in the case study) .....	44
Figure 12. Beginning of pointing trajectories estimations for 3 physically impaired participants using different computer pointing devices: (a) joystick, (b) numeric keypad with a head pointer and (c) oversized trackball.....	45
Figure 13. Distribution by disability of the user group .....	46
Figure 14. Median values by user for motionless intervals along cursor trajectories .....	53
Figure 15. CI distribution boxplot by user calculated with an without estimation .....	53
Figure 16. Speed distribution boxplot by user calculated with an without estimation ...	54
Figure 17. Comparing chart about the speed parameter automatically obtained for each participant.....	55
Figure 18. Comparing chart about the curvature index automatically obtained for each participant.....	56
Figure 19. Comparing chart about the time for pointing a target and time for clicking on a target for each participant in the experiment.....	56
Figure 20. Discapnet web page with the target highlited .....	58
Figure 21. Slide gesture (Left) and pinch gesture (Right).....	65
Figure 22. Disambiguation User Interface for the “augmented tap” method of interaction.....	65
Figure 23. Discapnet web site .....	66
Figure 24. Discapnet web site after applying the suitable adaptation techniques.....	66
Figure 25. Discapnet web site adapted (left) and the original (right) .....	68
Figure 26. General architecture and workflow.....	72
Figure 27. Bidasoa Turism web site with the preference manager user interface .....	73
Figure 28. Coordinator module flowchart.....	74
Figure 29. Upper ontology categorization models .....	75
Figure 30. Web page annotation model (WAI-ARIA and added new roles and properties).....	75
Figure 31. Adaptation techniques information in the ontology .....	77
Figure 32. Areas element annotation in protégé. ....	80
Figure 33. a) Detected areas for the Discapnet homepage in the annotation process. b) Detected areas for the Bidasoa Turismo homepage in the annotation process .....	81
Figure 34. Tablet on the table with Velcro.....	82
Figure 35. Tablet on a lectern.....	82

Figure 36. Tablet on the table with a holder .....	83
Figure 37. Tablet mounted on the wheelchair .....	83
Figure 38. Target acquisition task with target links surrounded by other selectable elements with standard separation (TA2).....	83
Figure 39. Bidasoa Turism original (up) and adapted (down) with the navigation menu highlighted.....	84
Figure 40. CA task with the target link highlighted.....	85
Figure 41. ACT task with the three links highlighted.....	85
Figure 42. User selection time boxplots, by interaction method (links not surrounded by other selectable items).....	87
Figure 43. User selection time boxplots, by interaction method (links surrounded by other selectable items).....	88
Figure 44. User selection time boxplots, by interaction method (links surrounded with 20px separation).....	89
Figure 45. Boxplot for each NASA TLX questionnaire dimension.....	92

# Tables

Table I. 3x3 Latin Square example .....	6
Table II. Tools classification .....	13
Table III. Classification of user testing tools according to the requirements proposed ..	19
Table IV. XUL accessibility checklist .....	34
Table V. Cronbach for the items of Perceived Usability/Pragmatic Quality attribute....	48
Table VI. Mean, median and standard deviation for each of the pragmatic quality attributes .....	48
Table VII. Cronbach for the different scales that were used.....	48
Table VIII. Mean and standard deviation for the time required to complete the questionnaires by user group.....	49
Table IX. Participants information.....	51
Table X. Information about the total number of visited pages (VP) in Task 3 and the analysed pages (AP) for cursor movement characterization. ....	52
Table XI. Mean values of the time required for pointing to a target automatically obtained by RemoTest platform (APM), estimation with one second (AIS), with no estimation (WILD) and manually obtained based on video analysis (MPM).....	55
Table XII. Participants visited pages, Lostness, number of pre-goal pages, and time needed to complete the task .....	58
Table XIII. Number of cursor movements and distances, croll actions and distance, and number of pressed keys .....	59
Table XIV. Total number of techniques gathered, classified by group: content, presentation and navigation adaptation techniques.....	62
Table XV. Relation of subgroups defined in each general group of users: users with cognitive impairments (C), physical impairments (P) and sensory impairments (S).....	63
Table XVI. Number of techniques in each user type subgroup also classified by three main technique type groups (Cont.-Content , Pres.-Presentation and Nav.-Navigation) and the total for each subgroup (Tot.).....	63
Table XVII. Demographic data.....	82
Table XVIII. Users' characteristics from target acquisition tasks with standard interaction.....	87
Table XIX. Time needed to complete task in both conditions (Adapted and Original) .	90
Table XX. Number of tasks completed in search tasks in both contitions.....	91
Table XXI. Information of the participants with low vision.....	94
Table XXII. Results achieved for tasks 1 to 12 during both scenarios (NTC-number of tasks completed, ATT-average time per task, AVP-average number of visited pages, TTS-total time per scenario, TVP- total number of visited pages, ALN-average Lostness, ORD .....	96
Table XXIII. Calculated results for tasks 1 to 12 during both scenarios (TPT-total pixels travelled with cursor, NZM-total number of zoom magnify operations, NZD-total number of zoom demagnify operations, NZO-total number of zoom overall operation, NPV-number of visits to task objective previous page and destination link visible but not clicked, NPH-number of visits to task objective previous page and destination link hidden, not on screen). ....	97



---

---

# Chapter 1

## Introduction

---

---

The Web is constantly growing and the number of services provided through it is increasing enormously. As a consequence the Web is becoming an essential tool for everyday life [Wellman and Haythornthwaite, 2008]. Web services nowadays include informative web sites (newspapers, blogs, etc.), online shopping, e-government services, social networks, online multimedia, etc.

The Internet is an increasingly valuable tool for anyone but especially for people with physical, sensory or cognitive disabilities, because it allows them to perform numerous activities related to labour, leisure, learning, etc., that would be difficult or impossible in a physical environment. For this reason it is vital that digitally provided services are accessible to as many people as possible to reduce the digital divide [Dobrinsky and Hargittai, 2006]. For example, easier and more accessible web user interfaces enable the elderly to be more connected to the community via social networks [Coelho and Duarte, 2016].

But Web accessibility has to face some important challenges. The first one is the wide range of devices. The Web is accessed from different devices such as TV sets, mobile phones, tablets, laptops and PCs, etc. These computers have different input and output devices, such as keyboard and mouse, touch screen, joystick, special keyboards, etc. People with diverse needs require specific input/output devices, which can pose challenges to Web use.

Another challenge is the diversity of users [Shneiderman, 2000]. Each person is unique and has specific skills, preferences and expectations. For instance, blind people would prefer a lineal Web which they could access properly using a screen reader. People with low vision on the other hand, would prefer larger elements and suitable contrast. Not forgetting that the decisions made to enhance accessibility for some users may have the reverse effect for others [Theofanos and Redish, 2005].

This research work aims to provide solutions for accessibility in this scenario.

### *1.1. Accessibility, Usability and User Experience*

Many definitions exist for accessibility but regarding the Web, accessibility “means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web” [WAI Intro, 2016]

On the other hand, the ISO 9241 defines usability as “the effectiveness, efficiency and satisfaction with which users achieve goals in a specified context of use [ISO 9241, 2016]”, where these terms are defined as follows:

- Effectiveness: “Accuracy and completeness with which users achieve specified goals”

- Efficiency: “Resources expended in relation to the accuracy and completeness with which users achieve goals.”
- Satisfaction: “Freedom from discomfort, and positive attitudes towards the use of the product.”

Other factors such as learnability, memorability and error control, included in the concept of usability, have been also considered in addition to those proposed by the ISO 9241 [Nielsen, 1996]:

- Learnability: How easy is it to carry out a task for the first time?
- Memorability: How easy is it to recover the knowledge and skills after a period without using it?
- Error control: Does the system lead to errors being made? How severe are these errors? Is it easy to recover from the errors?

Regarding the User Experience (UX) ISO 9241-210:2010 defines it as: “person's perceptions and responses that result from the use or anticipated use of a product, system or service”. This definition also states that, “user experience is a consequence of brand image, presentation, functionality, system performance, interactive behaviour and assistive capabilities of the interactive system, the user's internal and physical state resulting from prior experiences, attitudes, skills and personality, and the context of use”.

User experience (UX) can be also defined as “the entire set of affects that is elicited by the interaction between a user and a product, including the degree to which all his or her senses are gratified (aesthetic experience), the meanings we attach to the product (experience of meanings) and the feelings and emotions that are induced (emotional experience)” [Law et al., 2009]. UX covers different aspects such as aesthetics, emotions, usability/pragmatic attributes, hedonic attributes, cognitive load, interactivity, social responses, persuasion and acceptability [Brajnik and Giachin, 2014]. Each of these aspects is evaluated with different metrics or established methods. Pragmatic attributes are connected to the users' need to achieve behavioural goals [Hassenzahl, 2003]. A product may be perceived as pragmatic because it provides effective and efficient means to manipulate the environment [Hassenzahl, 2004].

### ***1.2. How Can User Experience, Accessibility or Usability Be Ensured?***

Various organizations such as the United States Environmental Protection Agency, the European Commission or the W3C, promote standards and laws to ensure equal access to the web. These efforts have produced very useful guidelines and evaluation tools to further the progress towards Web accessibility, such as for instance, Section 508 [2016] and WCAG 2.0 [2016].

In the framework of this research work, another interesting tool to improve the accessibility of web pages is the WAI-ARIA annotation language standard [WAI-ARIA, 2016]. This language adds semantic meaning to the elements, using roles and properties, that are not native to HTML (Menus, treeviews, dynamic areas, etc.) so that assistive technologies and adaptation tools can handle them properly [Moreno et al., 2011].

The responsibility of applying accessibility standards usually relies on web developers and content providers. Some of them do not have the necessary time or



resources to fix the accessibility problems; they may lack the required knowledge or they find that repairing the accessibility of a web page is too expensive.

However, despite the existence of developers and content providers that create inaccessible content, the awareness of the different protagonists involved is increasing as are the number of accessible web pages.

Nevertheless, even when guidelines and standards are adequately applied, this does not guarantee that a web page is fully accessible. The skills of people using assistive technology to access the Web can also influence whether accessibility barriers are imposed or not [Vigo and Harper, 2013a].

In this situation, even if the number of accessible web pages is increasing regularly, the number of non-accessible pages is growing still faster. This is perhaps because there are fewer developers aware of accessibility than those that are not. In addition there is a huge quantity of non-accessible content that was created before accessibility guidelines became more established.

Transcoding appears to be a solution for adapting content to specific users, transforming inaccessible pages into accessible ones.

Transcoding allows the page code to be modified “on the fly” adding, changing and tagging the content without involving designers or developers [Asakawa and Takagi, 2008].

The tools that perform the transformations are called transcoders and are usually classified by location as a client, as a proxy or as server tools.

Server side transcoders are transparent to users because they do not need user intervention to configure or install the transcoder. The drawback being however, that only pages located in the same server can be transformed.

Proxy tools can handle any web page (as opposed to the web server side tool). It is possible to differentiate between two types of proxy tools, those that require the web browser to be configured and those that do not need any configuration. In the latter, the user has to enter the proxy URL to start the web navigation. Afterwards, besides transforming the page, the web transcoder modifies the URLs of the page links to be able to process the requests that follow. This kind of tool may encounter problems with dynamic content if the JavaScript code is not correctly analysed and modified. When the web browser is configured to use a proxy, there is no need to modify the URLs and therefore it can be more compatible with the dynamic content but less than client side tools.

Client transcoders are usually placed on the web browser as plugins but dedicated web browsers can be also found. Client side tools are probably the most obstructive tools because they have to be installed by users. Conversely, client side transcoders can handle the dynamic content better and have more control over the final results of adaptations.

Adaptations can be performed automatically or by means of annotations. Automatic transformations aimed to fix accessibility or usability problems are usually driven by an Information Retrieval system. They have limitations due to the difficulties such as assigning the alt text to images or identifying the page layout. In addition they are very dependent on the reliability of the Information Retrieval system.

To overcome some these limitations, annotation based transcoding systems are used. The annotations add the necessary semantic information, which allows the content to be transformed more accurately. Semantics adds some meaning to express the “intention” of some page elements. In addition, annotations can provide better access to the web by offering alternative content, presentation adaptations and navigation support. Annotations can be inserted in the HTML, or stored in an external location. Only the

owners of the web site can add the annotations to be included in the HTML. On the other hand, when annotations are externally stored, anyone can annotate a web page.

Annotations can be carried out manually, with the help of programs, with crowdsourcing techniques, etc. In the past, earlier tools required complex and time consuming web page annotations to add the semantic meaning needed to perform adequate adaptations. This was one of the reasons that the diffusion of transcoding methods was held back.

Nowadays the emergence of the Semantic Web (2016), HTML5 and the WAI-ARIA annotation, etc., has helped to alleviate the burden of the annotation process. In addition these languages already provide more information about the purpose of the elements than the bare HTML code. WAI-ARIA can describe application widgets, tree item, labels correctly associated to interaction elements, or the page layout, main content, navigation, etc. HTML5 also includes new tags to provide a better description of the elements. These languages are very helpful but not all the pages include HTML5, WAI-ARIA or other semantics, so in order to transform those pages, an additional annotation is still needed.

Considering that some developers and content providers are sometimes reluctant to provide accessible content, users themselves or a community of volunteers are the other actors that can collaborate to fix non-accessible content [Bigham, Ladner and Borodin, 2011]. This approach requires specialized tools to transform non-accessible code into accessible content without requiring programming skills.

To overcome the challenges imposed by the multiplicity of technology, the user diversity and the lack of knowledge about the users, new methodologies and tools are needed. Besides adapting new content, these tools have to help to fix accessibility barriers in web pages. This work aims to meet this challenge.

### ***1.3. How Can Accessibility and Usability Be Measured?***

Before adapting the content, it is important to detect which web page elements can cause problems or to observe the different behaviours of users. Thus adaptations can be created taking into account the users' conduct or the accessibility problems they faced. Various methods exist to measure the accessibility of Web sites. They are usually classified into three categories [Nielsen, 1994; Petrie and Bevan, 2009]:

- Semi-automatic accessibility/ usability evaluation tools
- Evaluation by experts
- Evaluations by users

Evaluations are carried out according to guidelines and standards, but guidelines are numerous and complex so this kind of evaluation is not a trivial matter and it requires the participation of experts. Semiautomatic tools such as Evalaccess [Abascal et al., 2004], Wave [Kasday, 2000], WebSAT [Martins et al., 2009], Design advisor [Faraday, 2000] and QualWeb [Fernandes et al., 2012] can be used to provide a first accessibility evaluation. Nevertheless, automatic evaluation should not be the only method used for accessibility evaluation. It must be complemented with the opinion of experts, who can check accessibility elements that automatic methods cannot find in the code (e.g.: correctness of the alt texts, complexity of the text, structure, etc.)

Experts can use evaluation methods such as heuristic evaluations or barrier or cognitive walkthroughs [Wharton et al., 1994; Yesilada, Brajnik and Harper, 2009].

These kind of methods can be carried out in the early stages of the development process and require skills and knowledge about users from the evaluator.

Evaluation with users is the most desirable method when budget and time are not an issue. User testing enables unpredicted barriers to be detected, as well as the understanding of the strategies used and the user behaviour, etc. [Vigo and Harper, 2013a; Vigo and Harper, 2013b]. In order to carry out evaluations with users, knowledge about experimental research is required. Tools for evaluation with users should help the researchers to design and carry out experiments more easily. In this way they would facilitate the design, creation and better modification of the adaption tools. The next section describes basic definitions of the experimental design that are considered to define the evaluation part of our conceptual model (Chapter 3).

### *1.3.1. Experimental Research*

Experimental research is a widely used discipline in psychology, biology, chemistry and more recently in Human Computer Interaction (HCI). The main purpose of experimental research is to analyse and explain how the manipulation of a known variable has direct influence on another variable. The variables that can be manipulated are “independent variables” and the variables where the effects of the independent variables are measured are “dependant variables”. A typical example is the measurement of the effect on someone’s health condition of a specific pill. Applying the treatment/factor (pill) is the independent variable and the health condition of the participant, the dependant variable.

There are also other kinds of variables such as “control variables”, that must be kept constant across the whole study in order to ensure the results obtained are really generated by the independent variable. Some typical control variables are noise, light condition, temperature, etc.

### *1.3.2. Experimental Designs*

We consider two main types of experiments: “Randomized Experiments” and “Quasi-Experiments”. In randomized experiments, participants are randomly assigned to every independent variable condition. On the other hand, in quasi-experimental experiments participants are assigned manually to experimental conditions, due to diverse reasons, but frequently because it is not possible to do it randomly.

Randomized experiments are categorized as between-subjects or within-subjects depending how participants are assigned to conditions/factors/treatments. In between-subjects experiments, participants are assigned to a group which is then exposed to a single condition. When the number of conditions is high the number of groups is increased and, as a consequence, the number of required participants.

Between-subjects experiments have the following advantages: participants are assigned to one condition only, consequently there is no learning effect and the duration of experiments is smaller. It also has drawbacks: when differences between groups are high (participants’ characteristics are not homogeneous) it is harder to detect the differences caused by the treatment. In addition, the number of participants required increases exponentially with the number of conditions.

In within-subjects experiments, participants are exposed to every experimental condition. Therefore, it is easier to find differences caused by the treatment since a participant also acts as a control participant (participant with no treatment). What is

more, the number of required participants is lower than in between-subjects experiments. Nevertheless, it also has some disadvantages: participant's performance in a condition can be influenced by the experience learned in previous conditions or by accumulated fatigue (carry over and order effects). With the aim of minimizing these effects, conditions are usually counterbalanced.

In the next paragraphs, two widely used counterbalancing methods are summarized: complete counterbalance and Latin square.

In the complete counterbalancing method, every possible condition sequence must be considered exactly the same number of times. For instance, if there are A, B and C conditions, all possible permutation must be considered (A->B->C, B->A->C...). This is not a problem when the number of conditions is not high, but with a high number of conditions the number of possible sequences is "n!". Consequently the number of required participants is huge.

The Latin square instead is a partial counterbalancing method. The aim of the Latin square is to minimize the number of possible sequences of conditions and, as a consequence, the number of the required participants. This is achieved by assigning each condition to each position the exact same number of times. In Table I an example of a Latin square can be found. Each condition A, B and C are on every position exactly once, A(1, 3, 2), B(2, 1, 3), C(3, 2, 1). Sudokus can be considered as a kind of Latin square with additional restrictions.

**Table I. 3x3 Latin Square example**

A	B	C
B	C	A
C	A	B

Another type of experiment is the Factorial Design, used when more than one independent variable has to be tested. Factorial experiments can be within-subject, between-subject and mixed. An example of a mixed design is to measure the effectiveness of a treatment in relation to participants' age and gender.

### *1.3.3. Moderated and Unmoderated Experiments*

Experiments can be supervised by a researcher (moderated test) or unsupervised (unmoderated test). In moderated user testing, one or more researchers are in charge of carrying out the experiment, preparing the set-up, explaining the procedures, taking notes, observing the participants, etc.

In unmoderated tests, participants without any supervision have to complete the tasks given. For this reason, in the case of unmoderated tests it is essential to provide clear instructions about the procedures and the tasks in order to avoid possible misunderstandings.

When economic resources or other limitations are not a problem, moderated testing is the best choice. Direct observation of participants provides valuable information about how each participant uses the system and the treatment. However, it is not always easy to recruit the appropriate number of participants, sometimes due to a lack of users with suitable characteristics, on other occasions caused by scheduling difficulties, etc. For this reason, remote user testing is becoming a valuable choice.

Remote tests allow a wider range of participants to be reached who are able to do the tasks from home whenever they want. The advantages are even greater when people

with disabilities are part of the study. They are not required to travel, something which could be problematic for people with specific disabilities (e.g. people with severe motor impairments). In addition, the user's environment is already adapted to the participant's needs. The computer and the assistive technology is already configured and working as usual. When experiments are carried out in a laboratory, it is difficult to have the required assistive technology available and adequately configured to fit the participants' needs.

On the other hand, unmanned experiments also have drawbacks such as the lack of control over experimental sessions. Participants can be distracted during the session; they might not have the necessary motivation or they might not understand adequately how to complete the task or use the system.

To minimize these problems the following measures are usually taken:

- Allocating a large number of participants
- Providing clear instructions about the tasks and procedures
- Including training sessions

In addition, some cases require the extraction of specific data from cursor movements. To this end, suitable algorithms to extract data from cursor movements, usually related to target selection, are applied.

#### ***1.4. Research Motivation***

Accessibility is benefitting from important advances thanks to the guidelines and standards which have been proposed and the awareness of the various actors involved in the generation of content. However, there are still important challenges to be met before equal access is granted to the Internet for all people.

Personalized user interfaces can help to solve some of the existing challenges. Abstract user interfaces are used to generate personalized user interfaces that are later created depending on the user and device needs. However, these tools are devoted to generate new content and consequently do not solve problems with content that has been created previously without following these methodologies.

Diverse methodologies and languages exist to create abstract user interfaces. Nevertheless, even if numerous tools devoted to transforming content can be found, no methodologies or models were found which were dedicated to adapting existing content. Moreover, evaluation is often omitted in such methodologies or models. This is important as evaluation is a necessary step to detect needs that might inspire new adaptations or to evaluate the transformations carried out. Furthermore, evaluation can be used to observe the user behaviour, to decide which adaptations are needed to personalize the user interface, etc.

#### ***1.5. Summary***

This chapter establishes the basic concepts and the motivation of the research carried out. The next chapter summarizes the state of the art of tools to improve the accessibility of user interfaces and tools to evaluate the user interfaces or to analyse the user behaviour. In Chapter 3 a conceptual model for performing transformations is discussed. This conceptual model describes the factors that should be considered in

order to carry out the transformations by transcoding systems. In addition, it includes the evaluation as a part of the transformation process. Subsequently in Chapter 4, an implementation of the evaluation part of the conceptual model is described and tested. In addition, the behaviour of people with motor impairments and with low vision are analysed to gather the needs they might have to navigate the Web. In Chapter 5 some adaption techniques devoted to enhance the accessibility and usability of people with motor impairments and low vision are discussed and classified. In Chapter 6, an implementation of the conceptual model for transcoding is presented. This system implements the techniques discussed in Chapter 5 and is evaluated by users using the evaluation tool presented in Chapter 4. These implementations and tests support the composition of the evaluation and the transformation framework with which to adapt the web.

---

---

# Chapter 2

## Background

---

---

### *2.1. Improving Accessibility and Usability Through Personalized Interfaces*

Even when accessibility guidelines are adequately followed and met, some people still experience accessibility barriers due to their special needs, preferences or situational impairments. The need to solve specific problems for specific users can be resolved by means of personalized user interfaces. Moreover, this approach takes into account the challenge of having to deal with a diversity of devices and users as well as different knowledge levels. Personalized User Interfaces can be created at design-time, using specific languages or at run-time modifying the existing content through transcoding. In the following section both approaches are discussed.

#### *2.1.1. Improving Accessibility from Design*

A requirement of creating personalized user interfaces at design-time is to define the interface in an abstract way by means of a User Inter Description Language (UIDL). Abstract User Interfaces are converted into Final User Interfaces (FUI) taking into consideration various parameters such as the device used and the users' needs.

One of these languages is the User Interface Markup Language (UIML) [Abrams et al., 1999]. UIML was intended to be a simple language, independent of the emerging Internet technologies (XUL, HTML, JavaScript...) that would enable User Interfaces to be defined without having to take into account the technology or device that would be used by the user. UIML is based on the XML language and divides the User Interface into five components: description, structure, data, style and events. Since UIML is independent of the output, it can be used to provide personalized interfaces such as, audible output for blind people or personalized style sheets for the partially blind.

Maria is an UIDL based on the Cameleon Reference Framework [Paternò, Santoro and Spano, 2009]. It is divided into an Abstract User Interface and some Concrete User Interfaces allowing it to fine tune the abstract user interface to a platform. Abstract User Interfaces, describe the user interaction presentation models. Concrete User Interfaces set platform dependant characteristics, detailing how the interaction is represented in the given platform.

*USIXML* is another UIDL based on the Cameleon Reference Framework [2016], [Limbourg et al., 2004]. This language structures the interface development in four steps: Final UI, ConcreteUI, Abstract UI, Task&Concepts. With all this information, Personalized User Interfaces are built considering the task, the context, the device and/or the user needs in runtime.



Gajos and Weld [2004] presented *Supple*, a system tailored to people with reduced mobility. It automatically generates interfaces based on input device constraints, usage trace, functional specifications and a cost function. This function determines the interface with the lowest estimated cost that satisfies the device and user requirements. In the design process the developer produces a declarative description of the interface (interface specification), a device specification (device model) and a user model. In order to complete the user model, the authors created *ARNAULD*, a tool to elicit and capture user preferences [Gajos, Weld and Wobbrock, 2010]. The system displays two equivalent user interfaces and asks the user which one he or she prefers. The user's answers lead to the next options and questions until all the available options have been finished. After finishing the data collection the resulting interface is displayed and contrasted with the original one, to allow the user to introduce further changes or to reverse inadequate adaptations.

The authors later added an ability model to the system that elicits the user needs using a test with four tasks (pointing, dragging, list selecting and multiple clicking). With all these data the system is able to determine the best configuration for the user.

More recently a framework aimed at developing adaptive user interfaces for web applications to be used by people with disabilities has been developed in the *i2web* project [Ackermann, Velasco and Power, 2012]. This system takes into account the device and the context and in doing so, uses various models (Web technology, user, device and application models) to elicit the needs for adaptations. The user model is designed to map the device capabilities and user preferences and is based on the CC/PP framework [Reynolds et al., 2001]. The Web technology model allows some capabilities of the devices to be defined, such as hardware, software, user preferences, and adaptation rules. The device model defines the hardware and the software of the device used to access the Web. Finally the application model integrates the previous models to display the application adequately, using the HTML 5 or the WAI-ARIA semantics.

Summarizing, Abstract User Interfaces, can be adapted to different technologies, devices and users. In addition, some of these interfaces can be used to provide help to the users to accomplish the task. One of the requirements is that these languages are adopted by the developers.

Even if the technology to develop adaptive user interfaces is available, the development of these type of interfaces requires considerable training and is highly time-consuming. Unfortunately, most developers cannot devote enough time to these complex designs (due to company time constraints) or they lack the required training. This makes the adoption of these adaptive interfaces design methodologies quite improbable.

Even if this approach is adopted and personalized content is created, there is still a problem with the content that already exists. For this reason we adopted transcoding, a methodology that enables existing content to be made accessible. In the next section the basics of transcoding are discussed.

### *2.1.2. Transcoding to Improve Accessibility*

Transcoding is a technology that allows the content of a web page to be modified on the fly without requiring web designers or developers to add, change, or tag the code. Transcoder applications can work as a proxy, as a web server, or as a client application. Some transcoders require annotations in order to perform adaptations while others are based on information retrieval systems. Annotations add the necessary semantic



information, which enable the content to be transformed more accurately. Semantic tagging provides information about the designers' intention for specific elements that is not contained in the code. In addition, annotations provide better access to the web by means of alternative content and adaptation of the presentation and navigation support.

*Aurora* [Huang and Sundaresan, 2000] is a proxy based transcoding system that uses semantics to adapt the web content, taking into account the users' goals. Adaptations are made by means of a transaction model that characterizes the user goals in an abstract way for a concrete domain. *Aurora* only adapts the elements that are strictly relevant to accomplishing the goal, removing all the other elements. Adaptations are carried out by adaptors. The paper presents two adaptors, one for a text-only user interface and other for an icon based user interface.

Takagi et al., [2002] presented another transcoding system devoted to facilitate Web navigation to blind users. The system is placed in a proxy, and uses annotations that are inserted into the HTML code. They use XPATH expressions to annotate web page elements. Making annotations is a tedious process, carried out page by page. In order to alleviate the load the authors created another tool to propose annotations based on the similarity of previously annotated web pages.

A system to adapt the web for a wider range of users (the elderly and people with cognitive impairments) was presented by Richards and Hanson [2004]. The system works as a client and it uses functions that belong to the operative system, such as changing the font size or StickyKeys. The latter is a function that lets users type combination of keys such as *ctrl+alt+del* sequentially rather than simultaneously. This system allows users to select their preferred adaptations, by means of a settings button. The selected changes are immediately displayed to allow the user to figure out how each adaptation works. The adaptation techniques implemented do not require semantic information making this system valid for most web pages.

Bigham et al., [2006] presented a proxy system able to add the missing alt text of images in web pages. Three modules were created in order to get the missing text: context analysis, optical character recognition (OCR) and human labelling. When a user requests a page the system checks if the images have been previously annotated or not. If not, the proxy injects some JavaScript in order to allow the main content be loaded while the alt text is generated. It also injects a bar at the end of the page to inform the user that the tool is modifying the page.

*Dante* is an ontology created in by Yesilada et al., [2007] which aims to support web navigation for visually impaired people, adding travel information to web elements. In this way, the added semantic information helps to overcome the loss of data when visual information disappears. The system, among other adaptations, is able to fragment the page into more manageable chunks and to provide a table of content. Annotations are made with the help of the *COHSE* tool, which has been implemented as a Firefox add-on. The tool uses XPointer technology to point out the web page elements.

*SADIE* proposes an ontology that classifies the elements into removable, not removable or menu items [Harper and Bechhofer, 2007]. When elements are tagged as not removable, a priority can be added. Based on the annotation proposed, the system is able to remove unnecessary elements, reorder the web page or create toggle menus to avoid the use of skip links. Skip links, are links which allow screen reader users to jump to the main content, avoiding banners and other navigation items. The *SADIE* system uses the CSS attributes (id and class), to identify HTML elements and relate them with the annotation proposed. This reduces the effort required to annotate a web site since the CSS elements annotated are used across the whole web site. This advantage could be a handicap when the CSS elements are not used consistently within the website. This

system was later improved by adding behaviour related adaptation techniques [Lunn, Harper and Bechhofer, 2011]. From the strategies gathered from a study about coping strategies used by blind users when navigating the web they created some new adaptation techniques such as providing access to important chunks, summaries of the pages or quick access to menu items. To enable the new adaptations, they injected WAI-ARIA landmarks annotations.

*Hearsay* is a non-visual web browser able to interact with different input and output devices [Borodin et al., 2010]. Hearsay gathers the events generated by the browser or the user and sends them to the *Hearsay* application. This tool uses annotations to add the missing “alt” text to images, analyses the page layout to improve the navigation, associates labels with the web elements and carries out many more adaptations aimed at people with visual impairments.

*Sasayaki* provides a secondary voice, which augments the primary voice output of the screen reader software with contextual information [Sato et al., 2011]. In addition to the secondary voice, it also provides methods to jump to “semantically meaningful” locations. *Sasayaki* provides information, categorized as: spatial, contextual, social and analysis, to blind users in order to enhance their navigation. The spatial information helps the user to know where the cursor is. The contextual information can be used to provide appropriate support when the user is lost. The social information, is generated by volunteers who create summaries or locate the meaningful areas of the web page, such as the main content, advertisements, etc. Finally, analysis information such as providing summaries of user reviews about a product or suggesting links or buttons is automatically generated.

Another interesting tool is *SPRWEB* which adapts the colours of the web to people with colour vision problems [Flatla et al., 2013] with the aim of preserving the subjective responses to colours while improving the colour differentiability. *SPRWEB* uses a model to predict the perceived temperature, activity and weight of the colours in the web page in order to produce adaptations.

In addition to automatically adapting Web user interfaces another possibility is to involve users or web developers to modify web pages creating or selecting their preferred scripts. Even if transforming the web via script usually means that those scripts are tailored to a concrete web page, more general scripts can be also created.

*Accessmonkey* is a framework based on *Greasemonkey*, using scripting languages. It can be deployed in the client, in a proxy or even as a web server [Bigham and Ladner, 2007; Greasemonkey, 2016]. This framework has been implemented using JavaScript, so it can be run in different computer environments. As opposed to *Greasemonkey*, *Accessmonkey* provides methods to let developers edit, save or approve the adaptations made. Scripts can be created by developers and users with JavaScript skills.

*GAPforAPE* [Mirri et al., 2012] is a similar system. Like *Accessmonkey*, it is based on *Greasemonkey* but in addition it uses user profiles to store preferences. The user profile is stored locally and it follows the XML based IMS ACCLIP [2016] standard. Among other adaptations, such as CSS transformations or DOM manipulations, it also adds or modifies the scripts associated to a web page to improve its accessibility (to avoid automatic page refreshing, for instance). Every time a page is requested, the system checks for page specific scripts. If there are none, it applies a general script.

Another system intended to improve the accessibility and usability of web pages is *Chickenfoot* [Bolin et al., 2005]. This add-on allows repetitive operations to be automated or the web to be transformed via a scripting language. In contrast to *Greasemonkey* it is not restricted to XPATH. It provides a high level pattern language that allows the web page to be modified without knowing the page structure.

Finally, *AxsJax* is a framework that enables web developers to inject WAI-ARIA annotation into web pages [Google-AxsJAX, 2016]. Its main goal is to specify live regions (dynamic regions) in web sites to make users aware of them. Screen readers or adaptive systems can use this added information for better rendering of the page content. Annotations are added via *Bookmarklet*, *Greasemonkey* or *Fire Vox*. Table II shows a classification of the mentioned tools.

Table II. Tools classification

Tool	Users	Location	Information Retrieval (IR)/Annotation	Adaptation Selection method	How
SUPPLE	Motor impaired users	Client (App)	---	Live Preferences, ability model	Authoring
I2web	General	Client (App)	---	Preferences	Authoring
AURORA	General	Proxy	Annotation	---	Transcoding
Accessibility transcoding	Blind and low vision	Proxy	Annotation/Crowd	---	Transcoding
WebinSight	Blind	Proxy	Automatic/annotation/Crowd	---	Transcoding
Web Adaptation Technology	Older, Low vision	Client (add-on)	---	Live Preferences	Transcoding
Dante	Blind, low vision	Proxy	Annotation	Preferences	Transcoding
SADIE	Blind, low vision	Proxy	Annotation	Fixed	Transcoding
Hearsay	Blind, low vision	Client (add-on)	Annotation /IR/crowd	Fixed?	Transcoding
SPRWEB	Colour blindness	Add-on	---	Fixed	Transcoding
Accessmonkey	General, prototype for blind	Add-on, proxy, server	Scripts, annotation	Users script request	Scripting
GAPforAPE	General	Add-on	Scripts	Preferences	Scripting
Chickenfoot	General	Add-on	Scripts	User script request	Scripting
AxsJax	General (blind)	Add-on bookmarklet	Scripts	User script request	Scripting
Sasayaki	Blind	Addon	Annotation+IR	---	Transcodign
Model Access	General	Addon-proxy	Annotation	User model, live preferences	Transcoding

Proxy systems such as *Aurora*, *SADIE*, *Dante*, etc., can have problems when dynamic content or secure connections appear. In addition, the scope of server side systems is limited, since only pages that are located in the server where the adaptation tool is, can be adapted or corrected. On the other hand, client based systems are able to transform dynamic web pages, deal with secure connections and can adapt third party web pages. Their main drawback is that these kinds of systems have to be installed, but as mentioned in Section 4.4, the provision of clear instructions can alleviate the process, which may otherwise be cumbersome for some users.

Most systems use annotations to perform adaptations, which allows the production of more accurate adaptations than those made by systems based on information retrieval. There exist different annotation models and different methods to link the annotation

model with the interface elements. For instance, Aurora uses a complex annotation model. The annotator first finds and annotates the common elements for a domain (shopping, finances...). Secondly he or she defines which transactions are the steps required to accomplish a task. In addition, the elements required for the transactions are also identified.

Other systems, such as *SADIE* or the one developed by Takagi et al. [2002], use simpler annotation models. The tool developed by Takagi et al. [2002], uses XPATH to relate the interface elements to the annotation. This type of annotation is tedious because it requires the annotation of every page of the web site. On the other hand, *SADIE* uses CSS elements, IDs or classes to relate the annotation model to the interface elements. This procedure facilitates the annotation process, since CSS elements are shared within the web site. In this way, only a few elements are needed to annotate the whole web site. Conversely this annotation type encounters problems with specific web sites, especially when the semantic meaning of CSS elements is not shared across the whole site, or when the web site does not use CSS.

The annotation presented in this dissertation uses the CSS elements when possible and XPATH for the remaining cases. In addition, our system can benefit from pages which are already annotated with the standard WAI-ARIA, since the annotation used is based on an extension of that language. In fact, a great number of adaptations can be performed without adding any more annotations to pages with WAI-ARIA annotation language already included.

Systems based on scripting languages are highly versatile, but when more powerful adaptations are required, such as applying natural language processing techniques, more powerful programming languages must be used. In addition, these systems require considerable experience from the user in coding with scripting languages.

Most of the systems analysed are only devoted to cover the specific needs of specific user groups. This is the case of systems such as, *Dante*, *Hearsay*, *Sasayaki* and *SADIE* and *SPRWEB*, devoted to people with vision problems. Conversely, the transcoding system we present here is aimed to help users with diverse kinds of disability.

Some of the systems use user profiles or user models to perform adaptations. For instance, *Dante* or *GapForApe* store the preferences locally, which allow the adaptations to be personalized. But preferences are lost when the user accesses the Internet from a different device. On the other hand, in the system proposed in this work the preferences and the user model are stored in a server, so the user preferences can be used across different devices.

## **2.2. Measuring the UX, Usability and Accessibility**

In the previous sections we described several tools used to improve accessibility and usability. As mentioned in Chapter 1, there are several methods to measure accessibility, usability and user experience. Many of them are automatic, semi-automatic or carried out by experts, but definitive evaluations should be preferably carried out with real users to obtain feedback about the problems they face and their strategies to overcome accessibility barriers.

In the next section, a number of tools, mainly devoted to unmoderated user testing and their architecture advantages and disadvantages are reviewed. Finally, some of these are classified by the requirements that they have to meet so as to be easily used by researchers for experiments with people with disabilities.

### 2.2.1. Architecture of Tools for Web User Testing

Tools for remote user testing are usually located in one of three places: the server, a proxy between the server and the participant device or the client (device). In the following paragraphs the weaknesses and strengths of each approach are discussed and summarized.

*In the server.* When the user testing tool is located in a server, participants can carry out the experimentation transparently and comfortably. No installation or configuration is required. The only requirements are to access the test web page and to perform the pertinent tasks.

Early tools were only able to gather log data generated by servers [Extended Log File Format, 2016]. However, more recent technologies, such as JavaScript or Java Applets, enabled events generated by the cursor or keyboard to be gathered [UI Events, 2016] which provide more information about the participant than server logs [Atterer, Wnuk and Schmidt, 2006].

One drawback that cannot be overlooked is that only pages located in the same server can be analysed, hence no 3<sup>rd</sup> party web pages can be analysed. Another inconvenience is the lack of control over the experimentation. For instance, if the participant navigates to a page that is out of the server, large quantities of data are lost and frequently the experimental session must be stopped. The same problem arises if the participant closes the navigator tab, opens a new one, etc.

*In the proxy.* Proxy tools are located between the participants' device and the web page server. In order to perform the tasks related to the experiment, each participant has to use a web browser configured to access the Internet through the proxy. Alternatively researchers can provide a link to the proxy from which participants start the test and the next web page requests are made through the proxy. All the requests made by the web browser (texts, images, videos, etc.) are caught first by the proxy, which adds the necessary code to gather user interface events before reaching the participants' device.

As opposed to tools located on servers, proxy tools make it possible to analyze any existing web page. In this way researchers can analyse how the users navigate in different types of web pages, observe their interests, measure accessibility, etc.

Nevertheless, proxy systems also have some drawbacks. When the test is made through proxy without browser configuration, the data is lost when participants close tabs or when they type URLs manually. Adequately configuring the web browser avoids these problems but it makes the process more obstructive than accessing from a URL. In addition, proxy systems usually have restrictions with secure web pages and dynamic content.

*In the client.* These tools are located in the participants' device, usually as a browser add-on, but web browsers especially designed for user testing can be also found. With this approach all the web pages can be analysed. The number of events that can be gathered is larger since browser selections such as back, history, or the context menu produced by the right click can be gathered. These events are appropriated to learn the interest that a participant has for a web page, for instance if it has been saved, printed or added to the bookmarks.

As opposed to proxy or server systems with client tools there is no problem closing tabs or inserting the desired URL manually. Clients can handle dynamic content and



secure connections without problems. Nevertheless, data from secure connections has to be carefully handled, no keyboard events or information about web pages elements can be taken, as this information can be used to get personal data such as passwords, bank accounts, etc. The need to perform a previous software installation before starting the experimental session is a major drawback, proving to be rather obstructive for the participant. In all the three cases, when user interface events are gathered, a transmission mechanism is needed to send the data from the browser to the server.

### 2.2.2. Web User Testing Tools in the Literature

A number of web tools for in-situ or remote user testing can be found in the literature.

*NIST WebMetric* suite is one of the early server side tools. This suite comprises *WebVIP*, *WEB CAT* and *VISVIP* which are devoted to remotely evaluate the usability or accessibility of web sites [Scholtz, Laskowski and Downey, 1998]. With *WebSAT* the usability of a web page can be analysed by checking the HTML syntax. *WebCAT* is used to find problems related to page categories, sections or structure. *WebVIP* is a program that tracks the user path on websites. It adds a timestamp and an identifier to each visited link. It can also indicate if a link is internal or external. *VISVIP*, is the tool devoted to visualize data gathered from the *WebVIP* tool, such as the path followed by the participant.

*WET* is server tool based on a scripting language which is installed on the web site by modifying the HTML headers. It uses a file that adds the necessary functionalities [Etgen and Cantor, 1999]. *WET* captures browser events such as page loads, resize and also mouse or keyboard events. All the events are stored in cookies. In contrast to *WebVIP*, event handlers are added automatically. However, it requires that the events to be recorded are specified. Once the system is installed, the user clicks on the start/stop button to start/end the evaluation task.

*WebRemUSINE* [Paganelli and Paternò, 2002] is a more comprehensive server side tool based on *RemUSINE* [Paternò and Ballardini, 2000]. It uses task models to evaluate graphical applications. The tool is located in a server but it can gather interaction events since it is based on Java Applets technology. The tests are carried out in three phases:

- Creation of a task model for the web site
- Automatic analysis (errors, loading times, etc.)
- Evaluation

*ConcurTaskTrees* [Paternò, Mancini and Meniconi, 1997] is the language used to create the tasks model. With this task model, the tool can compare the path followed by the user with the optimal one. To facilitate the analysis, the system can display the data gathered graphically (time needed to complete tasks, visited pages, etc.). *Webquilt* is a usability proxy tool that captures some events such as page requests and clicked links or frames [Hong et al., 2001]. Like *VISVIP*, it provides a graph of the path followed by the user. The user does not need to install or configure anything to use *Webquilt*, participants are only required to start the experimental session from a specific URL.

*Usaproxy* is a very interesting proxy tool. It automatically adds JavaScript code to the requested page before it is presented to the user [Atterer, Wnuk and Schmidt, 2006]. In this way the system is able to track the mouse, keyboard and window events as well as the requests made to server.

*WUP* is a proxy tool devoted to carry out remote usability tests of mobile web apps [Burzacca and Paternò, 2013]. In order to find indicators of usability problems, the

designer records the optimal path and automatically compares it with the path followed by the user. In addition it also provides data visualization interfaces, events timelines and page timelines, to allow easier comparisons between the optimal path and the path followed.

Regarding client tools, the *Curious Browser* is a tool developed by Claypool et al. [2001] to study implicit interest indicators for the web. They have implemented a custom browser able to capture mouse events (click and movements), scroll and keyboard events. The browser can also present a questionnaire after every newly visited web page. Questionnaires were used to detect, in an explicit manner, the interest in the visited page and thereby analyse which gathered events could be then used to detect the interest implicitly..

*Uzilla* is a client tool developed as an add-on for Mozilla Firefox to enable web usability testing [Edmonds, 2003]. This tool automates the presentation of instructions and surveys (pre and post task) in addition to providing tools for the analysis. Like other tools, it gathers events from the mouse, the keyboard or the page but in addition it gathers browser events such as the back button or typing in the URL bar.

More recently, Gajos, Reinecke and Herrmann [2012] developed a system devoted to minimize the gap between the results of pointing performance in a laboratory and those in the wild. The tool distinguishes between deliberate and distracted mouse pointer movements. Another similar tool was proposed by Hurst et al. [2013]. This tool classifies user characteristics based on their input events.

Summarizing the strengths and weaknesses of the aforementioned tools, it should be pointed out that some of them, such as *NIST WebMetric* or *Webquilt*, do not gather events generated by the keyboard or cursor movements. Furthermore, the *NIST WebMetric* suite requires the whole web site to be previously downloaded before it can be tested. These tools are valuable for path related issues but not for detecting user characteristics or to analyse whether an element is causing problems to users or not.

Tools such as *WET*, *WebRemUSINE*, *WUP* or *Usaproxy* are able to gather a great number of events (for instance cursor, keyboard events or even common log requests). Conversely these tools are not able to gather events produced by the browser, such as print, save page or context menu). These events might be used to obtain metrics on user interest about the page or to infer the expertise of the user.

*Usaproxy* and *Curious browser* can be used to obtain valuable information about user behaviour or characteristics but do not allow formal experiments to be performed. Although *WET* enables the definition of tasks, it does not have the control of these tasks and it cannot automatically counterbalance experimental sessions. Being a server side tools it is only valid to test the sites hosted on the tool's server.

*WebRemUSINE* is a comprehensive tool for performing experimental studies, which requires creating a task model for each trial. Consequently, researchers need a thorough knowledge of *CurrentTaskTree* usage and enough time to create the appropriate task model for each task.

### 2.2.3. Commercial Tools for Web User Testing

*Loop11* [2016] displays the evaluation page on an HTML iframe element where controls are added to stop the ongoing task. It works as a proxy between the user and the analysed web page. It includes features that allow web tasks and questionnaires to be defined. Regarding the data analysis, it provides click stream reports to visualize the paths followed by the user, click heat maps and real time data. In conjunction with Knowbility [2016], a non-profit organization aiming to improve technology access for

people with disabilities, they created Access Works [2016] a database of people with disabilities that are available to perform web user testing with *Loop11*.

*Optimizely* [2016] is a tool able to perform A/B and multivariate tests easily, in order to judge the appropriateness of web elements for specific participants. It automatically calculates the statistical power needed to prove the significance of the changes made in the web page. The evaluation is made page by page. The fact that it cannot evaluate the whole web site makes it suitable for detecting elements that cause problems or to test new designs but hinders its ability to find problems related to navigation.

*Google Analytics* [2016] is a server-side tool aimed at obtaining general data about the use of a specific web site. It records information such as network provider, mouse events or used browser, etc. It provides a large number of visualization tools and options ranging from real time data (events, number of active users, etc.) to creating personalized variables to be visualized later. Testing different versions of a web page is also possible due to the *content experiments* module. This module enables the creation of different versions of the same page that are tested later by users. Using statistics, *Google analytics* automatically evaluates which is the most effective version.

The *Morae* system [Morae 2016] is an advanced usability software that can be used to carry out user testing of applications, web sites or products. It provides a wide range of features such as remote user observation, screen recording, video or audio annotation, etc. However, the tool is not easy installable by the user and in most cases an evaluator has to supervise the test even if it is performed remotely.

#### 2.2.4. Tools Classification by Requirements for Including People with Disabilities in User Testing

Power, Petrie and Mitchel [2009] listed a number of requirements that a remote user tool should meet to be able to conduct experimental sessions with users with disabilities. Requirements relating to participants include: providing features to record demographic data (P1), specifying the technology used (OS, browser, assistive technology) (P2), selection of trials (P3). The ones relating to researchers are as follows: provision of features to test customized and “real” websites (R1), defining tasks for a set of users (R2), specifying a set of questions to the user, before and/or after the task has been completed (R3), provision of instructions and training documents for each trial (R4).

In addition, the following requirements oriented to the experimenter were added: the tool should facilitate experiment design (R5), it must have counterbalancing methods (R6) and it should provide analysis methods (R7).

As it can be observed (Table III), only few tools like *Morae*, *Loop11* or *Uzilla* meets most of the requirements to design, carry out and analyse an accessible user testing. Even though, it was not possible to find information about the accessibility of the interfaces of tools like *Uzilla* or *Morae*. On the contrary, *Loop11* has been using with people with disabilities; but it has the problems of proxy tools commented before.

To conclude, it has to be highlighted that *Morae* would be probably the most comprehensive tool. The possibilities that brings *Morae* are huge but it requires a strong knowledge to make use of it adequately. This tool should be installed on the participant computer which could be troublesome due to its extend configuration. For this reason the participant would need a researcher to set up the system for the first time, which is not always feasible.



Table III. Classification of user testing tools according to the requirements proposed

Name	Location	P1	P2	P3	R1	R2	R3	R4	R5	R6	R7
NIST WebMetrics Suite	Server	No	No	No	Partial	No	No	No	No	No	Yes
WET	Server	No	No	Yes	Partial	No	No	No	No	No	Yes
WEBQUILT	Proxy	No	No	No	Yes	No	No	No	No	No	Yes
Curious Browser	Client	No	No	No	Yes	No	No	No	No	No	No
USAPROXY	Proxy	No	No	No	Yes	No	No	No	No	No	No
WebRemUSINE	Server	No	No	Yes	Yes	Yes	No	No	Yes	-	Yes
	(Applet)										
UZILLA	Client	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-	-	Yes
LOOP11	Proxy	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-	Yes
	(iframe)										
Optimizely	Server	No	No	No	Partial	No	No	No	Yes	No	Yes
MORAE	Client	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Google Analytics	Server	No	No	No	Partial	No	Yes	No	No	No	Yes
Gajos et al.	Client	No	No	No	Yes	No	No	No	No	No	Yes
Hurst et al.	Client	No	No	No	Yes	No	No	No	No	No	Yes



---

---

# Chapter 3

## Conceptual Model

---

---

In this chapter we present a conceptual model that describes the transcoding process and its prerequisites which we designed for transforming non-accessible web pages into accessible and personalized web pages.

### 3.1. Definitions

A Not-Accessible Web Page, **NAWP** is an annotated web page that is not accessible for a particular user **U** or that is not tailored to the specific needs of a particular user **U**. By “annotated web page”, we mean that the code has been previously enriched with adequate semantic tags or annotations using a suitable language (e.g. WAI/ARIA). By accessible web page we mean that the page meets the requirements of established accessibility standards, e.g. WAI/WCAG-2)

An Accessible and Personalised Web Page **APWP** is a web page that is accessible for a particular user **U** and meets the specific needs of this particular user **U**.

**transcoding** is a process that converts an **NAWP** web page that is not-accessible for a specific user **U**, into an accessible and personalised (for the same user) web page **APWP**, taking into account several parameters. **transcoding** transforms specific parts of the original code of an **NAWP**, producing a recoded **APWP**. To this end, it eliminates the inappropriate code and/or includes accessible code, following the instructions of a selected adaptation technique..

**transcoding (NAWP, parameters) produces APWP**

Where **parameters** are:

- An applicable adaptation technique **T**
- A tuple **model (U, D, E)** containing the characteristics of a specific user **U**, with a specific device **D**, in a specific environment **E**.

Therefore

**transcoding (NAWP, model (U, D, E), T) produces APWP**

There is a set of rules and facts used to assess the merits of the applicability of each adaptation technique.

**rules\_and\_facts (A, model (U, D, E))** produces a subset of rules and facts **AR** that can be applied to a specific set of annotations **A**, taking into account the

characteristics of the user, the device and the environment contained in the **model (U, D, E)**.

**rules\_and\_facts (A, model(U, D, E)) produces AR**

We use **deduction** in order to find an appropriate adaptation technique depending on the annotations:

**deduction** is the process of applying a subset of rules and facts that allow a suitable technique to be selected. **deduction** matches the set of applicable rules **AR** associated to the annotations **A**, with the subset of adaptation techniques associated to the same annotations, in order to select a suitable adaptation technique **T**.

**deduction (AR, adaptation\_techniques (A)) produces T**

where

**adaptation\_techniques (A)** is the set of all the available adaptation techniques associated to a specific set of annotations **A**.

Summarizing, let us assume:

- A user **U**
- A web page **NAWP** not accessible for this user:
- A subset of the adaptation techniques composed of the annotations **A** contained in the **NAWP** page.
- A model **model (U, D, E)** defining the characteristics of the user **U**, his/her device **D**, and his/her environment **E**.
- A subset of rules and facts **AR** that can be applied to the specific set of annotations **A** (contained in the **NAWP** page) taking into account the characteristics of the user, the device and the environment, contained in the model **model (U, D, E)**.

After applying the whole process, the **NAWP** page (not-accessible for user **U**) is converted into an accessible and adapted page **APWP**:

**transcoding (NAWP, model (U, D, E), deduction (rules\_and\_facts (A, model(U, D, E)), adaptation\_techniques (A))) produces APWP**

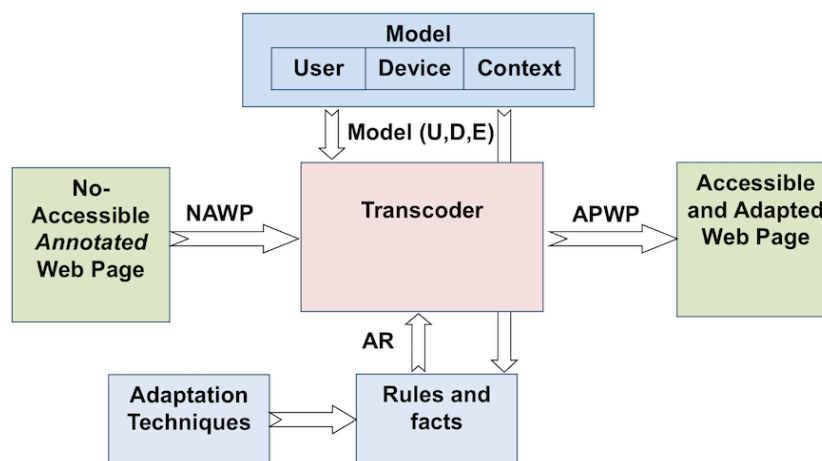


Figure 1. Interaction of the content transformation components

### 3.2. Prerequisites

#### 3.2.1. Annotation

Transcoding requires the initial web pages to be adequately annotated. Annotation is a process to automatically or manually include in the code, tags or annotations that provide meta-information about the page content, structure or function. Annotations are done by means of specific mark-up languages are such as **Accessible Rich Internet Applications Suite (WAI/ARIA)**, XHTML or HTML5.

Annotations for transcoding usually add semantic information in the form of the pair (Role, Property).

- **Role** indicates the purpose of the interface element. For example, whether it is help (tutorial, bread crumb, etc.), a landmark to identify a region of the interface, it belongs to the structure or hierarchy (structure), or is a widget (e.g., menu or alert).
- **Property**: establishes relations between the elements. E.g., “LABEL describes ELEMENT” allows to the correct tagging of the form elements, to provide alternatives texts, etc.) It is also possible to establish priorities to allow reordering or eliminating elements according to their significance.

#### 3.2.2. Evaluation

An evaluation is the development of a method focused on an objective and conducted by an agent.

- The **objective** is the aim that motivates the evaluation. Evaluations usually aim to measure either accessibility or usability.
- The **method**: is the set of procedures used to continue the evaluation process. E.g., empirical evaluation, functional inspection, formal inspection of usability, etc.
- The **agent** carries out a method with a predetermined objective. The agent can be usability or accessibility professional or a user. Automatic tools or virtual users can be agents in specific situations.

**(method, objective, agent) produces evaluation**

The evaluation process is the main source for:

- creating models
- selecting adaptation techniques
- and verifying the validity of the resulting user accessible web pages

#### 3.2.3. Reasoning

Decisions are made up of rules and facts extracted from the observed user behaviour, from available knowledge about user interactions, and from user preferences. The joint application of these rules allows a decision to be made regarding whether an adaptation

can be applied or not.

**rules\_and\_facts (knowledge, behaviour, preferences) produce decisions**

- **Knowledge** is the information about the usability and accessibility barriers that may be present in a web page. It is obtained in different ways: empirically, by literature review, by data-mining, etc.
- **Behaviour** depends on the use of the interface elements, the method of interaction, the device used, and the technology of support.
- The **preferences** of the user are expressed by means of certain parameters, which code a set of settings that can include methods of interaction and technical aids.

#### 3.2.4. Adaptation Techniques

An adaptation technique is composed of action and parameter. The **action** indicates the objective to be performed on the interface element that is pointed out by the **parameter**.

**action (parameter) produces adaptation**

- **Actions** are categorized according to the effect they have on the final interface: navigation, content, presentation or interaction.
- **Parameters** contain the role and priority of the actions performed by the adaptations.

### 3.3. User, Environment and Device Model

The information about the user, the environment and the device required for the transcoding is managed by means of an ontology. Figure 2 shows some of the parameters collected in the model.

*Users* are described by means of their *Characteristics*. A *Characteristic* can be a *Disability* (*Sensorial*, *Physical*, or *Cognitive*) and/or a set of *Preferences* (uses trackball, uses buttons for scrolling, etc.)

The *Device* is described as a function of a number of features: *Operative System* OS), *Mode*, and *Assistive Technology*. Information about the *Operating System* provides valuable data with which to perform suitable adaptations, such as buttons' location, style, interaction paradigms, etc.

The model considers two *Modes* for the devices: *Input* and *Output*. There is a wide range of standard input and output methods: touch screens, mice, keyboards, etc. (see Figure 3). In addition, electronic devices used by people with disabilities have a larger range of input and output devices than standard devices. They include the specific assistive technology that users may require to interact with the computer. Assistive technology includes, for example, eye tracking systems, head pointers, Braille devices, glass magnifiers, etc.

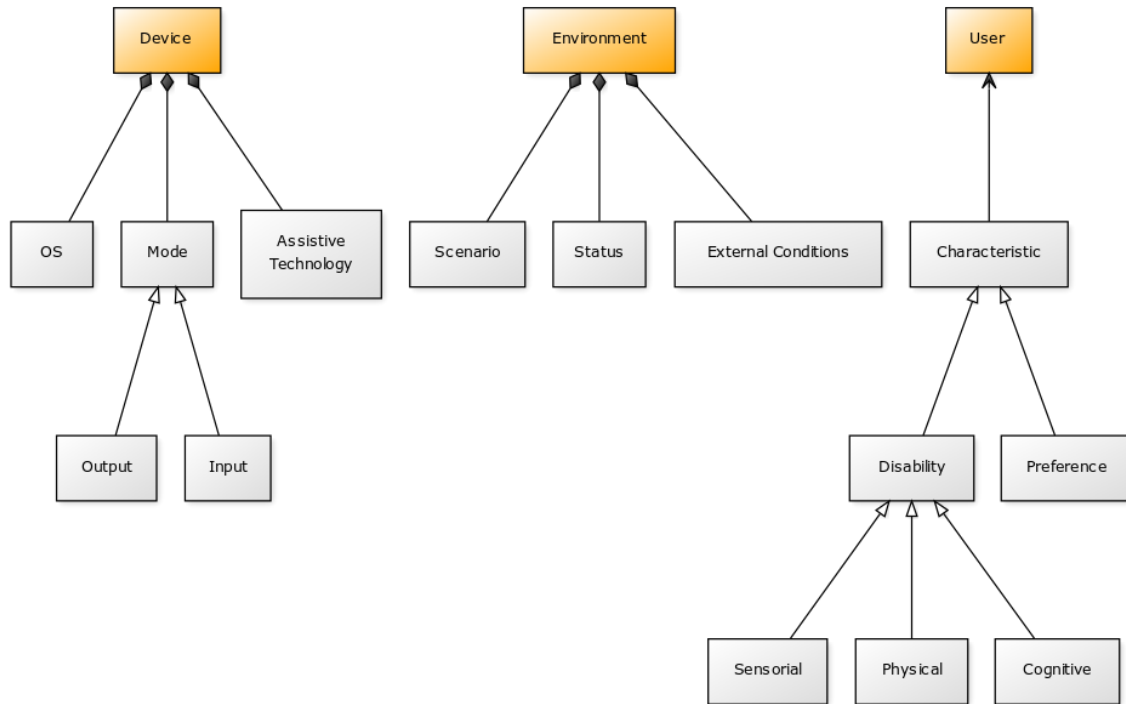


Figure 2. Device, environment and user models

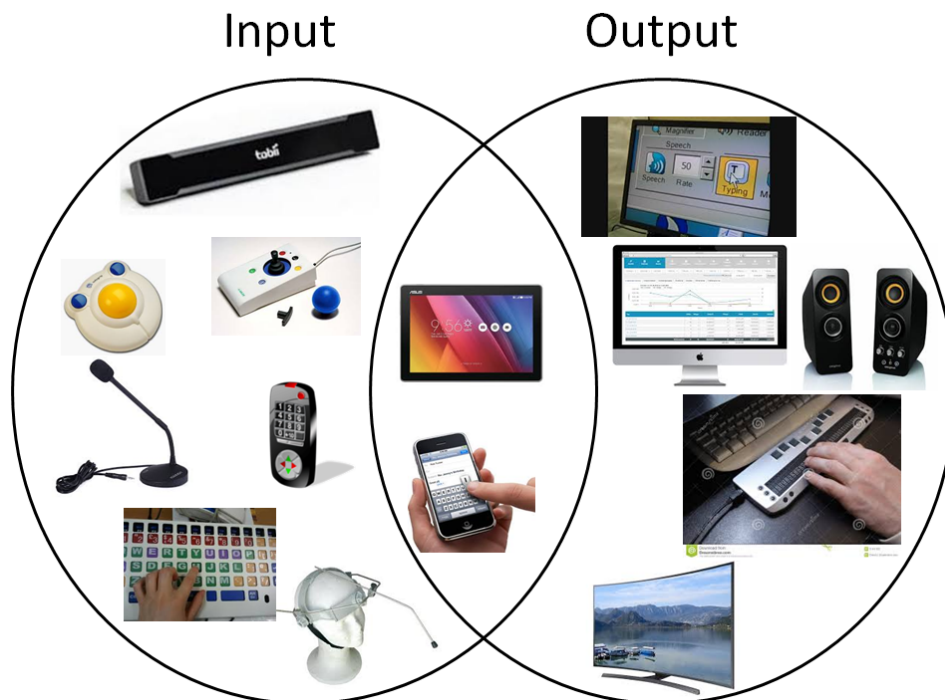


Figure 3. Input and output devices

Assistive technology can sometimes be integrated into the device (e.g. screen readers, software-based magnifiers, etc.). On other occasions the assistive technology is built into an external device (e.g. head pointers, eye trackers, Braille displays, hardware-based magnifiers, etc.).

Regarding the environment, *Status*, *External Conditions* and *Scenario* are considered. *Scenario* is the place where the user is located, such as “in the street”, “in a car”, “at home”, etc. *External Conditions* are those related to weather, noise, lighting, etc. While the *Status* is what the user is doing, e.g. walking, standing, sitting, driving, etc. Even though our model takes the context into account, in the implementation presented later this variable has a constant value. Future developments of the system will include creating an environment manager to set and update the user environment and to create or select the necessary adaptation techniques for the given environment.

### 3.4. User Interface Transformation Process

Figure 4 shows the transformation process we defined. It requires a model (of the user, the device interface), a knowledge base and a repository of adaptation techniques, in order to perform the transformations of web pages.

The knowledge base contains information about the users’ behaviour, their preferences, user stereotypes and the annotation language. In addition, it contains information about different models of devices, assistive technology, and adaptation techniques. Behaviour, preferences and stereotypes are used to group the rules, which decide the applicable adaptation techniques. Behaviour rules are used to select suitable adaptation techniques, depending on the assistive technology or the device used. Instead Preferences are used to allow the user to decide which adaptations to apply and also to customize certain parameters such as, size, colours or the position of helping elements. Stereotypes are used to create sets of rules applicable to users with specific characteristics.

A number of Adaptation Technique parameters are used to instantiate them, taking into account the roles and properties contained in the annotation model. Thereby the annotated interface elements are connected with adaptation techniques to subsequently carry out the required adaptations based on the behaviour, stereotype and preference rules. The Chapter 6 describes more details about annotation and how adaptations are performed.

### 3.5. Summary

This chapter presented the definition and the prerequisites that we established for obtaining accessible and personalized web user interfaces through transcoding.

The model proposed considers a number of parameters that are necessary to adequately modify the interface to suit user needs. Evaluation by users is a prerequisite to properly carrying out the transformations. These evaluations enable information to be obtained about the users themselves, to test the adequateness of the adaptation techniques, to validate the user interfaces and to feed the models.

The following chapters describe an implementation of the model. Firstly, the RemoTest tool, which implements the evaluation model based on the ESL and ESCL developed languages. RemoTest allows web interfaces to be evaluated and user behaviour to be observed through the events gathered from web navigation.

Secondly, an implementation of the transformation model is described. Model Access adapts the web to user needs taking into account the user, the device, the assistive technology he or she is using, the annotation model applied, and a set of valid adaptation techniques.







---

---

# Chapter 4

## Composing Accessible User Testing

---

---

Experiments involving people with disabilities are always needed to improve our understanding of the accessibility problems they have to face, to observe user behaviours or to evaluate techniques to enhance web accessibility.

But as discussed in chapter 2.2, carrying out experimental sessions with people with disabilities can be difficult and expensive. Although tools that allow remote experiments to be carried out can alleviate the problem, these tools must be accessible so people with disabilities can use them. Not only must the tool be accessible, the set-up, installation and configuration processes have to be as well.

In this chapter, first of all we present a formal description of the methodology we have developed. After that a preliminary study is described about the accessibility of interfaces created with the XML User Interface Language (XUL). The RemoTest platform, created taking into account the results gathered from this preliminary test, is subsequently presented. This platform implements the Experiment Specification Language and Experimental Session Controller Language.

Afterwards, data validity is examined with a study carried out with people with motor impairments. Finally, a case study conducted with people with low vision using PCs is presented. This case study shows how different measures can be used to examine the usability and accessibility in use.

### ***4.1. Evaluation***

The evaluation is started by an *agent* who carry out the evaluation, with a specific *objective*, using a *method* to achieve the desired *aim* (see Figure 5).

As far as the *Agent* and the *Methods* are concerned, numerous alternatives exist. Nevertheless to delimit the scope of the model, only the empirical evaluations carried out with real users were considered.

The main *objective* of the evaluations carried out in this model is to measure the accessibility and usability of a system during its use. In parallel, other evaluation tools can be applied, such as questionnaires (to measure User Experience), or observational studies (to study user behaviour during navigation). These kind of studies provides information about problems experienced by users and suggests possible solutions.

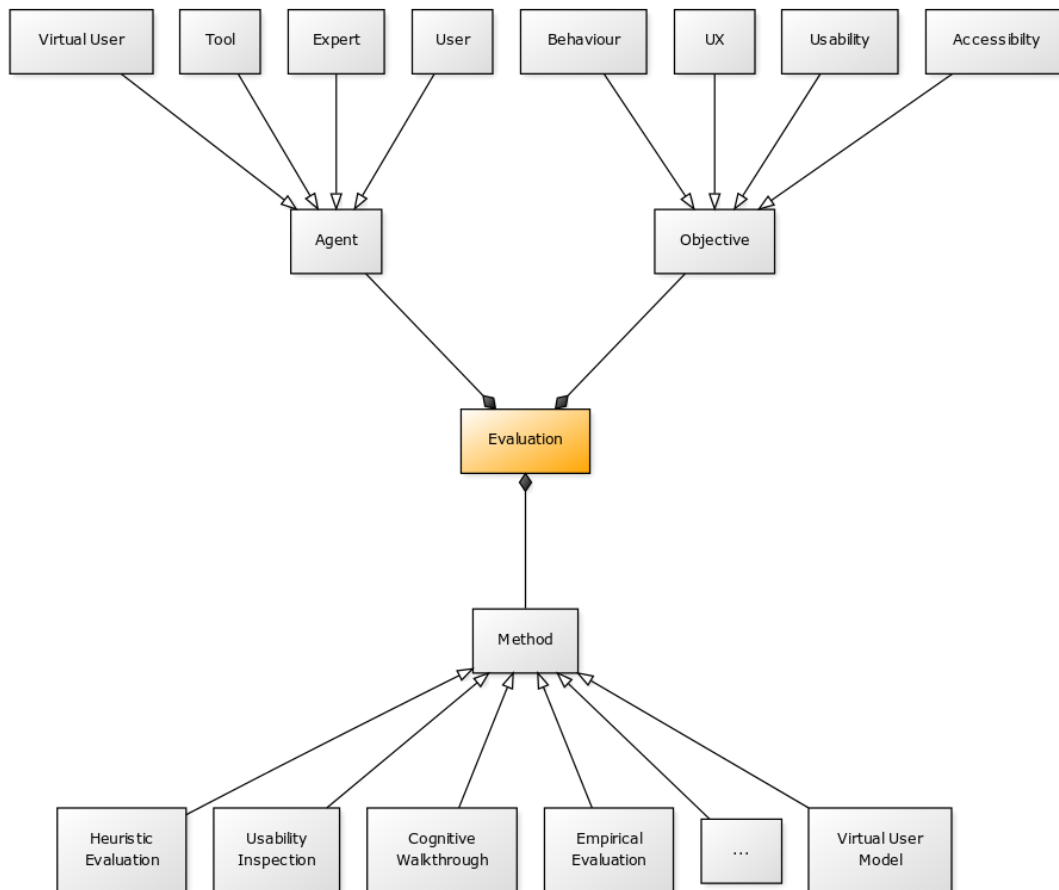


Figure 5. Evaluation components

#### 4.1.1. Empirical Evaluation

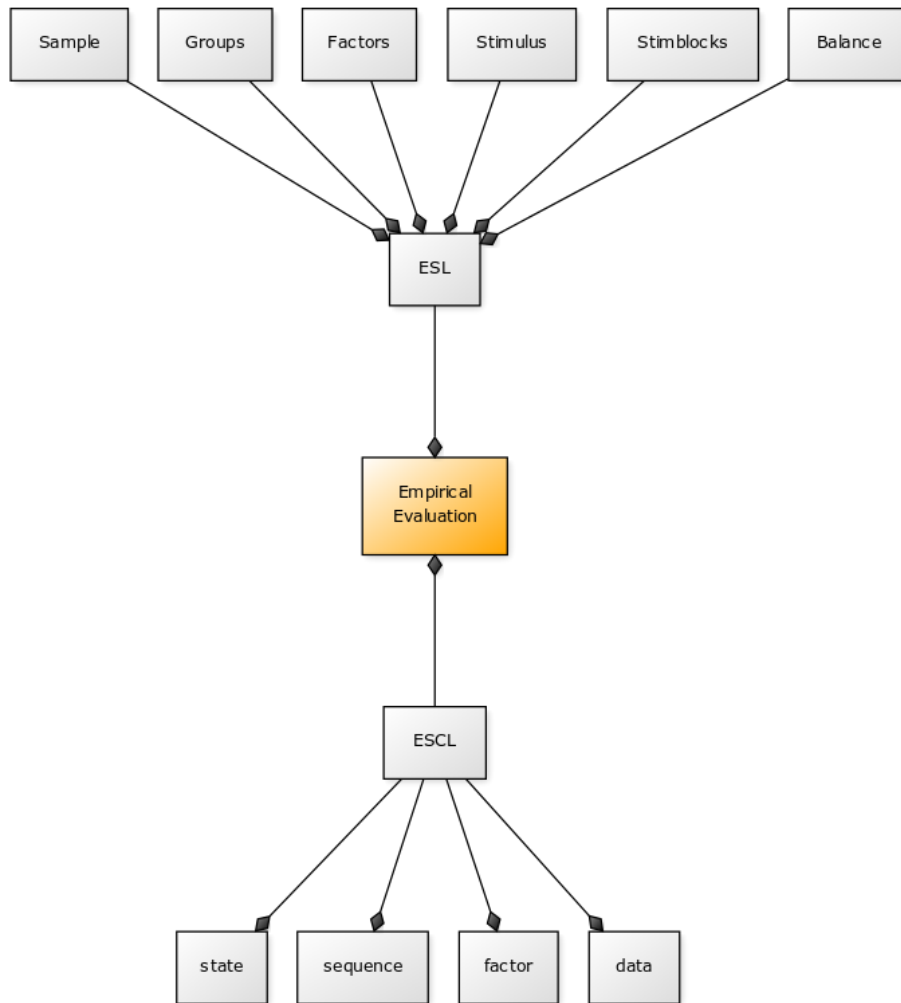
The empirical evaluation is split into two components: the experiment definition (Experiment Specification Language) and the experimental session (Experiment Session Controller Language) as can be seen in Figure 6.

*ESL*. The Experiment Specification Language (ESL), implemented in XML format, allows accessible web experiments to be defined. The vocabulary created includes the elements necessary for specifying different types of experiments.

In order to ensure that experimental sessions are accessible, a number of elements are added to provide additional help to participants. These elements correctly tag the questionnaire items, create accessible tasks descriptions, etc. The ESL is composed of six elements that define the experiment: *Sample*, *Group*, *Factor*, *Stimulus*, *StimBlock* and *Balance*.

The *Sample* element is mandatory and it must have at least one subject. Subjects' characteristics which are of interest for the study, such as gender, age, etc., can be added.

The *Group* element is included for “between subjects” studies. It is used to set the number of groups and the method to assign participants to groups (manual, random or based on their characteristics).



**Figure 6. Empirical evaluation components**

The *Factor* element is included in studies with a treatment. It has to state at least one *Level*. The *Level* has a *Treatment* and could include a *Target*, *Instructions* and *Questionnaire*. The *Instructions* element is used to provide information about the treatment or intervention. For instance, when a new interaction method is proposed, instructions explaining how to use it are added. Questionnaires can be used to gather participants' thoughts about treatment, user experience, etc. The *Treatment* can be a stimulus, to add/remove/alter elements in the user interface, or to add new functionalities. The *Questionnaire* is created with a *Block* of questions that can have a *Title* and a *Text* to explain the purpose of the block of questions. *Questions* have a *Text* element and can also include a *Tip*, image and audio used to provide additional help to participants. Furthermore, the type of question (open, liker scale, etc.) and if it is mandatory or not has to be defined. When required, the possible responses have to be included along with their text and additional images or audios in the *Response* element.

*Stimulus* and *Stimblock* elements are used to describe the stimuli of the study. *Stimblock* is used when a sequence of stimuli is required to be present as a block, one immediately after the other. The *Position* and *Condition* elements can be included in the *Stimblock* and *Stimulus* elements. The *Position* element sets when the stimulus or block of stimuli must be shown. The *Condition* element on the other hand, is used to set the dependencies between the different stimuli. In fact two types of stimuli are considered, web and surveys (questionnaires).

*Web* tasks, could have a *Start*, a *Goal* and a *TimeLimit*. The goal can be to reach a web page, but also to click a determined element, to read a document, to write a sentence, etc.

Finally the *Balance* element sets how the stimuli and factors are counterbalanced among participants or groups. Different counter balancing methods are considered: manual, rotation, Latin square, random and factorial.

```
<!ELEMENT ESL (Sample, Groups?, Factors, Stimulus+, Stimblocks *, Balance)>
  <!ELEMENT Groups (Group+, SampleAssig, FactorAssig)>
    <!ELEMENT Factor (Level+)>
      <!ELEMENT Level (Treatment, ?Target, ?Instructions, ?Questionnaire)>
    <!ELEMENT Stimulus (Title, Position?, Condition?, Instruction, (Web | Survey))>
      <!ELEMENT Stimblocks (Position?, Condition?, Stimulus+)>
        <!ELEMENT Web (Start?, Goal?, Condition?, TimeLimit)>
          <!ELEMENT Balance (Manual | Latin | Rotation | Random | Factorial)>
```

*ESCL*. The Experimental Session Controller Language (ESCL) is also specified in XML. It defines the participants' experimental session. The ESCL is composed of four elements, *State*, *Sequence*, *Factor* and *Data*.

The *State* element is used to define not only the stimuli, but also the interfaces needed to carry out the experimental session ( e.g. task instruction pages, farewell page, or login page). *State* can contain the *URL*, *Goal*, *Duration* elements. An attribute declares the type of state: login, survey, instruction, web or farewell.

The *URL* element sets the starting point page of the stimulus. The *Goal* element is used to set the conditions that finalize the task. The task can be finalized due to an event happening in an element of the web page, reaching a specific URL, or, manually by the users when they finalize the task.

The *Sequence* element sets how the *State* elements must be presented to the participant. Inside the *Sequence* element, the *Step* elements can be found. This element sets which *State* should be presented with which *Factor*. In addition, the *Condition* element is used to set the dependencies between tasks, such as having finished a previous task correctly.

The factor element defines the treatment that has to be performed following the stimulus. Adding/removing/altering elements or calling to functions that add new functionalities. Finally, the *Data* element can set which events have to be gathered, mouse, touch or browser events being some of the possibilities..

```
<!ELEMENT ESCL (State, Sequence, Factors, Data)>
  <!ELEMENT State(URL?, Goal, Duration)>
    <!ELEMENT Sequence(Step+)>
      <!ELEMENT Step(Condition?, State, Factor)>
        <!ELEMENT Factors(Factor)>
          <!ELEMENT Factor(Action, Target)>
        <!ELEMENT Data(Mouse?, Touch?, Browser?, HTTPRequest?)>
```

## 4.2. Introduction to XUL Interfaces

Before creating a tool to carry out experiments with people with disabilities, an application was developed with XUL [2016], to examine its accessibility and suitability as an interface for participants.

### 4.2.1. XUL

XUL (XML User Interface Language) is an XML based language to create User Interfaces (UIs), for the Mozilla platform. The main goal of the language is to allow easy development of cross-platform add-on applications which can run on any Mozilla integrated platform. It separates the program logic from the user interface components, facilitating the work of designers and programmers. Languages such as QML [2016] or XAML [2016] also apply this approach.

XUL is based on existing standards such as XML, HTML, CSS, DOM and JavaScript. In addition, the Cross-Platform Component Object Model (XPCOM) [XPCOM, 2016] technology can be applied when operating system functionalities are required.

Since the XUL language is based on web standards, its accessibility guidelines do not differ from established web accessibility guidelines. The XUL accessibility guidelines are classified in six sections: Keyboard Access, Assistive Information, Display, Human Computer Interaction, Media and Custom Widgets [XUL Accessibility Guidelines, 2013]. Each one has a set of verification checkpoints. For instance, the guideline Keyboard Access defines eight checkpoints related to tab order, keyboard shortcuts and so on.

All the guidelines and the related checkpoints can be seen as an accessibility checklist allowing the accessibility of a developed add-on application to be assessed. Thus, the XUL accessibility guidelines document provides a pass/fail statement for each checkpoint to be applied in order to elaborate a checklist that can be easily verified by developers at design time.

### 4.2.2. XUL Application

Following XUL accessibility guidelines, an XUL-based application was developed and its accessibility was evaluated using the XUL Accessibility Checklist. The add-on application for Mozilla Firefox created includes several pages with different types of web content.

For this study, two web pages containing forms were selected. The selection of this type of application was due to the following reasons: the diversity of elements included, the importance of being able to access all the questions and answer them appropriately, and the high interaction they require [Lazar, Kleinman and Malarkey, 2007].

The questions that users are required to answer are about demographic information, emotional aspects and issues related to the design of the visited web pages. These forms were presented to the user after they had spent some time browsing in a website. Figure 7. depicts the developed forms based on XUL.

The implemented application was verified by a sample of developers using the XUL Accessibility checklist Table IV in order to ensure the accessibility of the application according to the guidelines.

**Table IV. XUL accessibility checklist**

Guideline Id	Guideline
1.1	Logical tab order is provided
1.2	Keyboard functionality is provided for inaccessible features such as the column picker or added features such as column sorting
1.3	Keyboard alternatives are provided for toolbarbutton functionality
1.4	Keyboard shortcuts are present for important functionality
1.5	Context menus are triggered by the oncontextmenu event handler
1.6	All mouse operations have keyboard accessible equivalents
1.7	All scrollable elements are controllable with the keyboard
1.8	Keyboard focus is maintained and does not move unexpectedly
2.1	Alternative text is provided for meaningful images
2.2	All windows, including dialogs and wizards, have a descriptive title
2.3	Every form element has an associated label and radiobuttons are encapsulated in a groupbox
3.1	System settings are maintained
3.2	Color alone is not used to convey meaning and sufficient contrast exists between font color and background color
3.3	Visual elements and containers resize gracefully
4.1	Help documentation is provided including a description of keyboard shortcuts
4.2	Alerts are displayed using the alert scripting function or the notification box element
4.3	Interactive elements are sufficiently large and visible
4.4	Alerts are presented when the user initiates an error. The user has the opportunity and instruction to fix the error
4.5	User is informed of time limits and has control of response time when appropriate
5.1	Transcripts are provided for audio tracks
5.2	Video is captioned and a transcript is provided
5.3	User has control over animation and is warned about flashing content
6.1	Custom widgets provide accessible functionality

### 4.2.3. Preliminary Test

*Participants.* Two screen reader users with more than 6 years of expertise using the web were recruited for user testing: a woman (User 1) and a man (User 2) aged 30 and 40 respectively. User 1 considered herself as an intermediate Web browser user whereas User 2 considered himself as an advanced user.



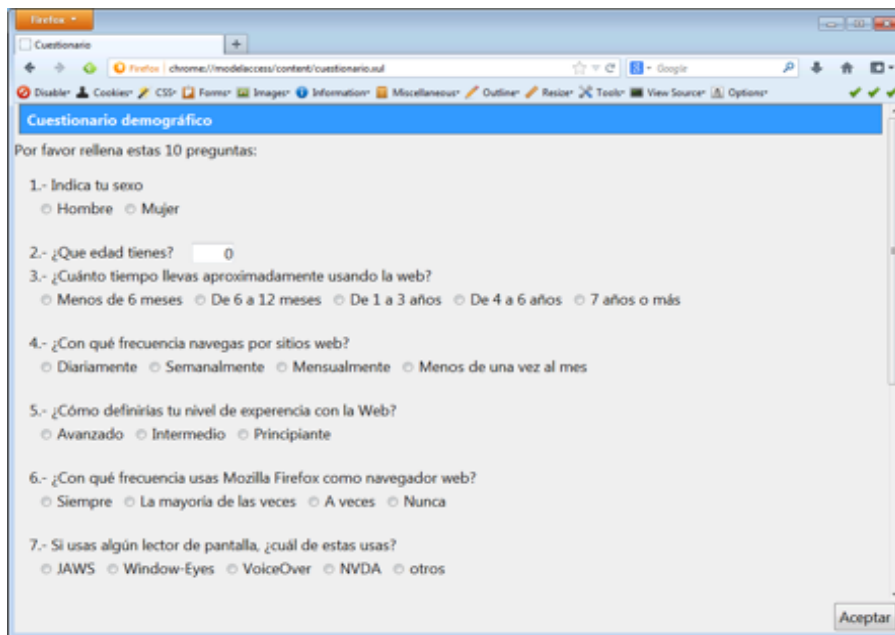


Figure 7. Demographic information form

Both of them use JAWS, but one of them (User 1) uses it infrequently as she prefers to use the VoiceOver screen reader on the Mac OS operating system. User 2 uses Windows and JAWS, but he does not usually use Firefox. Nevertheless, both are Mozilla Firefox sporadic users.

The experimental sessions were carried out in the same lab. They were asked to bring their laptop so they used JAWS configured with their personal preferences. The platform used was similar for both users: Windows operating system (User 1 used Windows XP and User 2 used Windows 7), JAWS 12 and Mozilla Firefox 22. Users were encouraged to report any barrier they detected when interacting with the XUL application. The sessions were recorded with a video-camera located behind the user in order to collect information about the interaction. The interviews were taped with a voice recorder.

In addition, an expert evaluation was performed by an expert screen reader user who has been working as an ICT accessibility consultant for the last five years at least. All the evaluation was carried on at her usual workplace and the findings were reported by email.

*Procedure.* First, the XUL application was installed on the users' laptops. Then, both users were asked to perform two tasks. The first one consisted in freely navigating for five minutes on a specific website ([www.discapnet.com](http://www.discapnet.com)). Subsequently, they were presented with the first form to be completed. The questions in this form were related to their navigation experience. The second task consisted in a search task on the same website to be carried out within a limited time interval of ten minutes. Finally, the application presented the second form containing, questions related to demographic data.

*Results.* The barriers found by the users and the related XUL accessibility guidelines are presented below. The evaluation carried out by the blind expert is also described.

*User 1*

This user experienced several barriers when filling in the forms developed with XUL. The barriers reported by the user with the related XUL guidelines are indicated in each case:

- Barrier 1.1: She was unable to know which answer option was checked in the multiple-choice type questions. This occurred when the user wanted to verify that the selected answer was the correct one. She navigated with the virtual cursor of JAWS that read the labels correctly. However, JAWS informed that the option was not checked, even though it was. Therefore, the experimenter had to tell her which option had been selected in order to ensure that it was the desired one. (Related guideline: 2.3)
- Barrier 1.2: She experienced navigation problems. Surprisingly, when she filled in a multiple-choice question, JAWS focus jumped to the first question of the form. Then, she had to navigate to the next question from the beginning of the form. This happened even though the program focus was at the correct position. (Related guidelines: 1.1, 1.8)
- Barrier 1.3: the JAWS navigation feature “shortcut” did not work properly. Due to the difficulties she was having, she tried to navigate through the form using the JAWS shortcuts, such as, for instance, forms or headings shortcuts but these did not work as expected. (Related guideline: 2.3)
- Barrier 1.4: Problems when clicking on a button. She was unable to find the button to continue. The button was located at the end of the form and was accessible using tab or arrows. However, JAWS did not detect this element correctly. (Related guideline: 2.3)

#### *User 2*

This user experienced similar problems to those reported by User 1 except for Barrier 1.3 (he did not try this mode of navigation). In addition, he reported several other barriers:

- Barrier 2.1: Problems with text input questions. JAWS was unable to detect a text area element even if the focus was on one. In addition, he reported sometimes hearing an incorrect label. (Related guideline: 2.3)
- Barrier 2.2: Alert messages were not adequately presented. JAWS detected the alert windows but not the text contained. Therefore, he could only hear the default sound of the alert and “OK” button but he missed the alert message. (Related guideline: 4.2)
- Barrier 2.3: Problems with numeric type inputs. Firefox adds special controls for this type of input. These controls are for entering the numeric values using two small buttons inside the element, one of them for increasing the value and the other for reducing it. These controls entered into conflict with the navigation controls. They were activated by keyboard arrows, which were the navigation mode used by this user. Consequently, he was unable to enter his age correctly. (Related guidelines: 1.6, 2.3)
- Barrier 2.4: JAWS read non-existent options in the UI. He reported to us that JAWS sometimes read text elements that were not in the UI. (Related guideline: 2.3)

*Blind expert evaluation*

Due to the different results obtained by the users and in order to obtain further knowledge of the situation, a review process was conducted by an expert blind screen reader user. She is a consultant on ICT accessibility.

She was asked to conduct a test with different versions of JAWS, Firefox and different operating systems. A summary of the testing can be found in Table 3. Results of the testing show the strong dependency that XUL accessibility features have with the user agent, the operating system and version of the screen reader. For instance, checkpoint 2.2 “All windows have different titles” only can be correctly detected if the user interacts with the application on a Windows 7, JAWS 13 and Firefox 23 platform. The fulfilment of checkpoint 2.2 is not so essential for user interaction (since the user can complete a task even if the title of the windows are not correctly presented). However, the same platform is required for the adequate fulfilment of checkpoint 2.3 “Labels are connected with their input element”. This checkpoint is essential if the user is supposed to access and fill in questions presented in a form.

The expert noted that Firefox generates inaccessible HTML elements for the application interface. Therefore, JAWS does not correctly read them to the user. This is due to the way JavaScript implements and interprets the DOM. Visually, the element appears in the interface, but the element does not seem to be in the DOM. Therefore, the screen reader just ignores it.

In conclusion, the latest versions of Firefox and JAWS (at least versions JAWS 13 and Firefox 23) seem to be the best combination for using the developed XUL application. However, it is worth pointing out that it is unusual for users to have the latest versions of screen readers. The same occurs with browsers. Most screen readers users use out-dated browser versions since the cursor mode of JAWS does not always work properly with the latest versions.

Some suggestions for improving the current version of the developed XUL application were indicated by the expert:

- The text for the questions should be defined as a header element
- The multiple-choice questions should define the possible answers as a list element
- Users should be provided with an input text element for introducing the answer. This increases the response time, making it less efficient and usable. However, it would be accessible for screen reader users

Implementing these suggestions allows users to overcome the most significant problems they are likely to have to face, principally when they do not use the latest versions of the browser and screen reader.

*Discussion.**Improvements to guidelines*

Most of the barriers are related to the guidelines referring to form elements, such as Guideline 2.3 or keyboard related issues, Guideline 1.1 and 1.8.

Tagging labels with the corresponding control does not seem to be enough to solve this problem. It is essential to correctly identify all questions and provide mechanisms in order to alert the user and the assistive technology about the existence of a list of questions or choices. Adding a new XUL element to tag the whole form would enable

assistive technologies to handle this information better and also to ensure the correct behaviour of the screen reader cursor.

The orientation of screen reader users considerably improves by applying a good simple practice: providing information at the beginning of the form about the total number of questions in the questionnaire and numbering each question.

Regarding the keyboard, the added controls and shortcuts can create conflicts with the browser's controls or shortcuts as well as with assistive technology controls. In this sense, information about the available shortcuts in the most used assistive technologies would be helpful. This would foster better and more efficient navigation and would avoid unexpected behaviour on the part of the technology.

Anyway, XUL accessibility guidelines should be reviewed in order to update issues regarding keyboard navigation, assistive technology compatibility, forms elements tagging, etc., not only to ensure that all elements are accessible but also to guarantee HTML-like behaviour, so users do not get confused or disoriented during the interaction.

#### *Improvements for the client*

The results obtained in user testing and the expert evaluation made clear that there are extremely important accessibility barriers in the XUL application developed. Problems arose in user testing that had not been previously detected.

Findings reported by the expert evaluation also highlight the importance of testing applications with different versions of Firefox and screen reader. There are numerous differences between versions and their possible combinations. Each combination has strengths and weaknesses. Using the latest versions of each one seems to be the best solution, but many people do not update their software, principally due to the high price of the updates. Moreover, users become accustomed to certain functionalities that they are unwilling to give up. For these reasons, it is crucial to ensure that the developed applications can be used in the widest possible range of versions.

Taking these results into account, the interfaces were developed in HTML, as described in Section 4.4, in order to make the participant module as accessible as possible.

### **4.3. Data Gathering and Analysis with the RemoTest Platform**

The RemoTest platform provides the necessary functionalities to assist researchers to define experiments, manage experimental remote/in-situ sessions and analyse the gathered interaction data. The platform admits a wide range of experiments. The objective can be, for instance, to study user behaviour when performing a task in different websites, to analyse and compare navigational strategies of different types of participants when interacting with the same website, to evaluate the accessibility-in-use of several websites, to gather significant information through surveys, to measure user satisfaction when using a certain web service, to analyse user performance improvement when interacting with adapted versions of original web pages, etc.

The architecture of the platform was designed taking all these different types of experiments into consideration. We adopted a hybrid architecture model that includes some functions in a client-side module and the remaining ones in some server-side modules. The platform is split into four modules: Experimenter Module (EXm) which creates the Experiment Specification Language, Participant Module (PAm) which

interprets the Experimental Session Controlling Language, Coordinator Module (COm), which interprets the ESL and creates the ESCL and Results Viewer Module (RVm).

#### 4.3.1. *Experimenter Module*

This module provides a set of functionalities with which to define all the components of the experiments. It is a server-side module, which can be remotely accessed by experimenters from any computer with an Internet connection. All the definition process is performed by means of a web application, divided into five main steps:

*Step 1: Specifying the type of experiment.* This first step is intended to specify the general characteristics of the experiment. The experiments defined should be based on web stimuli. These can be either a survey on the web for gathering explicit data from participants or web navigation experiments, which can also include questionnaires (e.g. demographic data, satisfaction questionnaire, etc.)

*Step 2: Determining the tasks and stimuli of the experimental sessions.* Depending on the type of experiment specified in Step 1 the platform requests information about the tasks to be performed by participants and the stimuli to be presented. Two main types of tasks can be defined: “Fill in Questionnaire” and “Web Navigation”. There is no limit to the number of tasks per experiment.

At this step, specific data for each task are provided. For instance, for the “Fill in Questionnaire” task the questions, the type of questions (open-ended or closed) and the possible answers (Likert scales, ranges) are defined. There is some other information regarding title, ID, etc., that is completed in order to ensure the accessibility of the questionnaire created. The “Web Navigation” task also requires specific data. There are currently two types of tasks in this category: “searching target” tasks and “free navigation” tasks. The former refers to tasks in which participants are required to find a specific target (such as a specific button, link, form, etc.) whereas the latter entails navigation without any concrete objective. Both require certain information from the experimenter, for instance, the starting URI, time limit for the task, etc. Searching target tasks also require the specific target URI to be provided. In addition, a title and a description can be added to each task. An accessible task explanation web page is also presented to participants before starting the task and there is an alert after performing the task.

Once all the tasks are defined, the dependencies between them are explicitly stated. This module provides features to define the order of presentation of the tasks. For example, in an experiment to measure user satisfaction when interacting with a certain website, a “Fill in Questionnaire” task about satisfaction will usually be presented just after a “Web Navigation” task.

*Step 3: Defining the procedure of the experimental sessions.* In this step, the number of groups of participants, the tasks presented to each group and the task sequence for each group is defined. Experimenters are asked to provide the number of groups in the sample. Specifying a unique group leads to a within-subject experiment. Therefore, each participant will perform all the defined tasks. Since the task sequence has to be counterbalanced between participants, the experimenter is required to select a method (manual, Latin square, rotation or random). In between-groups experiments, the tasks to be performed by each group is manually selected by the experimenter. The task sequence for each group can be defined manually or automatically (using the random method). In all cases, the EXm takes into account the dependencies between tasks. Any detected inconsistencies are notified to the experimenter.

*Step 4: Specifying interaction data to be gathered.* The experimenter selects the interaction data to be gathered in the experimental sessions. This data will be gathered by the PAm during the experimental sessions. Currently, the interaction data gathered by the platform are the following:

- Browser related events. Active tab, opening tab, closing tab, changing tab, backward button, forward button, vertical/horizontal scroll movements, screen resolution, window size, mouse context menu, favourites management, etc
- Cursor/Mouse related events. Clicking left button, clicking right button, using mouse wheel, size of clicked elements, hover events, size of hovered elements, tracking cursor movements
- Keyboard related events. Key down, key up, key pressed

*Step 5: Selecting the sample of participants.* The RemoTest platform provides functionalities to maintain information about the participants in a database through the module EXm. This tool includes options such as the manual selection of participants in each group, randomly creating groups from the selected participants or establishing a specific kind of criteria to select the sample, (such as gender, age, assistive technology used, etc.), applying filtering criteria to the query.

The information about the experiments is stored in an XML-based language specifically developed for this platform, the Experiment Specification Language (Section 4.1.1).

#### 4.3.2. Coordinator Module

The Coordinator Module is a server-side type module developed as a Web Service With the following objectives:

- Storing and managing the experiments defined by different experimenters, applying features of EXm
- Creating stimuli to be presented during the experimental sessions (questionnaires, task description web page, task completion alert, etc.)
- Defining the personalized experimental session specifications
- Collecting and storing interaction data obtained in experimental sessions
- Maintaining in databases the information about experimenters, experiments, and participants

This module creates all the necessary stimuli for the experiments (surveys, questionnaires, task description web pages, alerts, etc.) according to the definition provided by means of the Experiment Specification Language (Section 4.1.1). In addition, it prepares the groups and performs the counterbalancing methods, when necessary, in order to create personalized experimental sessions for each participant or group of participants. This leads to personalized experimental session specifications for each participant. These specifications are defined in an XML-based language developed for this purpose: the Experimental Session Controlling Language (Section 4.1.1). The interaction data created in the sessions are managed and stored in a MongoDB database.



### 4.3.3. Participant Module

The participant module is a client-side type module. Therefore, the participants need to install this module on their computers. It is currently an add-on for Firefox browser but it can be easily migrated to other platforms, since this module is mostly based on portable technologies such as JavaScript and XML.

The Participant Module processes the Experimental Session Controller Language for correctly conducting experimental sessions. It presents login procedures, questionnaires, task descriptions pages, etc., as described in the ESCL sequence.

It also tracks the participant during the entire experimental session. It checks, for example, whether the participant has found the target, clicked on the desired element or checks if time has run out, etc.

In addition, it gathers all the interaction data and generated events, as well as the HTTPRequest information during the complete session and asynchronously sends it to COM by using AJAX technology.

### 4.3.4. Results Viewer Module

The RVm is a server-side type module, which deals with the presentation of the interaction data gathered in experimental sessions. For this purpose, RVm implements functions for collecting the data from COM, structuring it in understandable blocks of events and presenting them to the experimenter as a web application. In addition several statistical values, grouped by pages, tasks or users, can be visualized.

These are some of the measures calculated by the RVm:

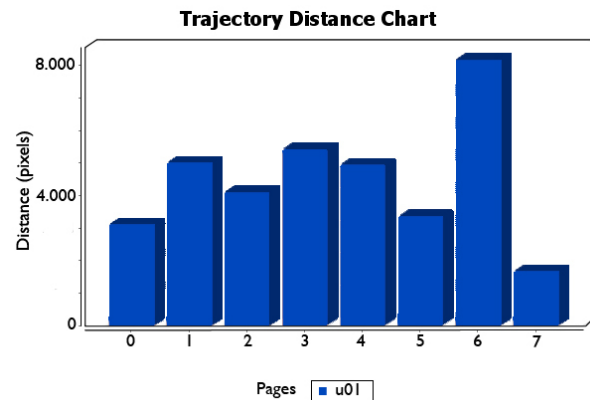
- Rapidity measures:
  - Cursor average speed
  - Cursor acceleration
- Accuracy measures:
  - Trajectory distance (cursor travel distance)
  - Curvature index (CI)
  - Distances to the centre of the target and to the last click and ratio of start-end position amplitude to start-target centre amplitude ( $A/A'$ )
- Navigation
  - Lostness
  - Time on Task
  - Task success
  - Time on page (III)
  - Scroll (III)
  - Areas
  - Zoom
  - Visited not seen
- Characterization
  - Number of acceleration and deceleration phases
  - Total distance travelled by the cursor
  - Duration of a click
  - Distance travelled by the cursor during a click
  - Speed and acceleration during a click
  - Number of accidental clicks

- Number of pauses
- Duration of pauses
- Time between trial start and cursor movement
- Time between click down and previous movement

Figure 8 shows the information displayed, regarding the performance of a participant in a visited web page.

Rapidity Measures	
Time on page:	00:00:16 (hh:mm:ss)
Average-speed:	91,92 Pixels/seconds
Number of ZA:	56
Number of acceleration to deceleration phases:	27
Number of deceleration to a acceleration phases:	22
Accuracy Measures	
Trajectory distance:	1470,78 Pixels
Curvature Index:	1,21
Distance between start to end point:	1213,66 Pixels
Distance between start point to target center:	1180,15 Pixels
Ratio of start-end position amplitude to start-target center amplitude:	1,03

**Figure 8.** General information about the performance of a participant in a visited web page



**Figure 9.** Comparing the cursor trajectory distance parameter of a participant

This module includes a tool for comparing the performance of each participant in all the visited pages. Figure 9 shows the charts generated for comparing the cursor trajectory distance of a participant in the visited pages.

The general information provided by RemoTest assists researchers to obtain detailed information about the performance of each participant during the experimental session.

The rapidity measures in conjunction with the trajectory distance, CI (optimal distance/cursor distance) and A/A' (measures if the click is made close to or far from target centre) can be used to detect a number of Usability problems: buttons too large or too small, interactive elements too far apart, etc. But also it serves to detect when the users have problems with the cursor: big CI and low Speed, lots of corrections when approaching the target, etc. [MacKenzie, Kauppinen and Silfverberg, 2001; Keates et al., 2002; Almanji, Davies and Stott, 2014],

Measures such as Lostness, time on task or success can be used in order to detect problems in the web page structure, Lostness is calculated by means of the formula proposed by Smith [1996]

$$L = \text{sqrt}[(N/S-1)^2 + (R/N-1)^2]$$

Where S is the total number of pages, N is the number of different web pages visited and R is the optimal number of page visits. 0 means not lost, from 0 to 0.4 not observably lost, from 0.5 to 1 the user is lost. A high value for time on tasks can also mean that the user is lost or that the action is too complex. Similarly, if the success is too low for a given task it means that there is a problem with the task.

Other interesting features are the area and the zoom. Every zoom change is measured. This information allow us to detect if a user changes the zoom when



interacting with a certain type of element or page. Several users repeating this behaviour may mean that the elements are too small.

Area, measures the percentage of the time that the cursor has been in a certain area of the web page. The possible areas are those specified by the WAI-ARIA landmark roles: main, navigation, banner, complementary, content info and search. This is automatically calculated when the WAI-ARIA annotation is present in the web page. However, although most web pages are not annotated, this information can be added via an external annotation system that allows the web page elements to be tagged with their landmark role. Other measures, such as time on the page or the scroll made on the page can be used as interest indicators [Claypool et al., 2001].

This module also provides several graphs for each visited page. One of them is the “Distance to target chart”, where the temporary evolution of the distance to the target can be seen. This chart allows researchers to easily identify any problems experienced by the users in approaching the target. As can be seen in Figure 10 Left, once the cursor is near the target, the user requires several attempts to place the cursor over it.

Furthermore, associated with this first chart, the tool also provides another graph that shows the distance to the target starting from the users’ intention time to clicking the target, see Figure 10 Right. The intention time to clicking is automatically calculated by the algorithm described in Section 4.3.5

In addition, graphs of cursor movement, speed and acceleration can be produced, Figure 11. Through these graphs, researchers can distinguish, for instance, different cursor movements, taking into account the input device used by the user.

These short pauses, when the speed and acceleration are zero, are usually related to the time needed by the user to move the head-pointer from one key to another in order to change the cursor direction

Another important feature of the tool is the possibility of making comparisons. In the first place, researchers can view comparisons between different statistical data obtained from all the pages visited by a selected user for a task. In the same way, the application also allows the statistical results obtained by diverse participants to be compared in a given task. Therefore, RVm is able to display automatically generated bar charts comparing the average values of the trajectory distances, curvature index, times to point and click or cursor speeds.

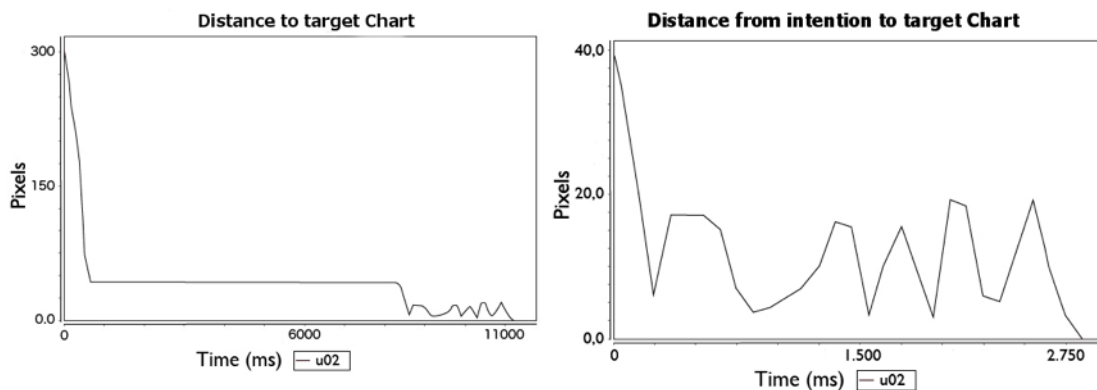


Figure 10. Distance to target” (Left) and “Distance from intention to target” (Right)

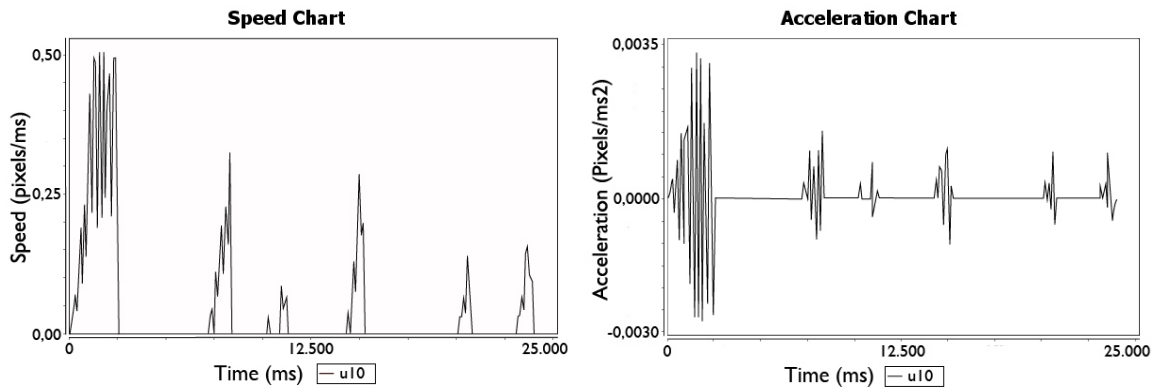


Figure 11. Speed chart (left) and acceleration chart (right) of a participant (U10 in the case study)

This straightforward visualization may assist experimenters in discovering relevant fragments of users' interactions, and in the case of also having complementary video recordings, speedy location of the corresponding moments, saving a huge amount of time viewing and analysing video images.

Another possible application of the RVm can be to distinguish users by the events gathered or to feed user models. For this purpose, characterization measures were added to the tool. The resulting data can be exported in CSV format in order to apply data mining techniques. The mongoDB database can be directly analysed with the Weka tool.

In order to extract the intended movements from all the cursor's kinematic data recorded by the RemoTest platform and to study pointing trajectory-related measures, we have processed the data as follows.

#### 4.3.5. Delimiting Pointing Trajectories

A pointing trajectory starts when a user resolves to move the cursor from the current position to a specific objective. Controlled laboratory experiments can specify restricted interactions to make the beginning of cursor movement explicit [Gajos, Weld, and Wobbrock, 2010]. In contrast to those studies, more naturalistic settings (with untagged web interactions) do not permit the registering of any explicit trace of the cognitive process behind the user's intention. In those cases, as in ours, heuristics are needed to estimate the beginning of the aimed movement [Chapuis, Blanch and Beaudin-Lafon, 2007; Gajos, Reinecke and Herrmann, 2012].

Among other possible movements of the cursor in a web graphic user interface (e.g., hovering over an element to see its tooltip), we decided to focus on pointing trajectories that end with a navigational click and a posterior page load event. To this end, we have considered the following bases for delimiting pointing trajectories:

*Beginning of movement.* The pointing of a navigational click trajectory may correspond to the complete cursor movement recorded along a page, however behaviours such as moving the cursor while reading the content of the page make it necessary to analyse the entire cursor trajectory on every page.

We considered the first cursor move events recorded within every page as a candidate for the beginning of a pointing movement for navigational click trajectories. Each time a page scroll interaction occurs, the actual cursor trajectory (if it exists) is

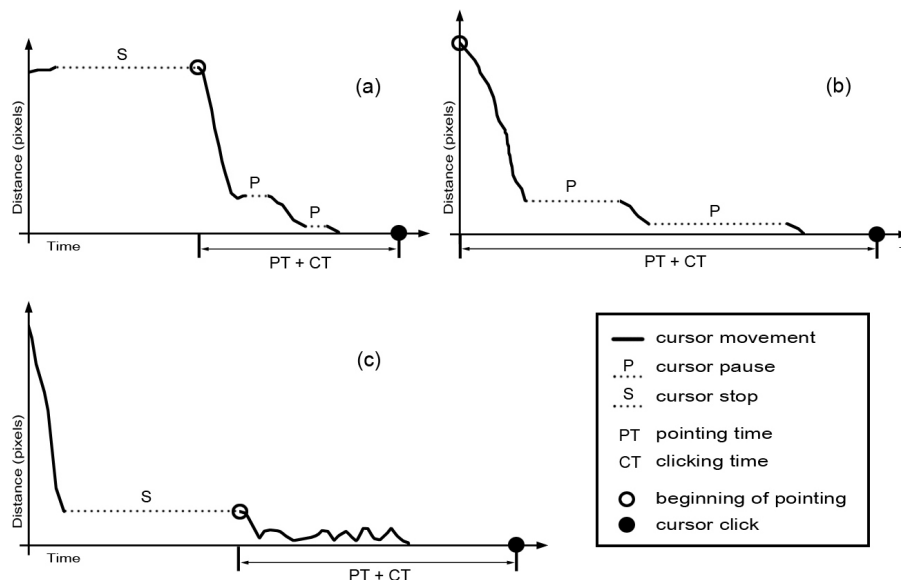
rejected (as a page candidate for indicating a navigational click trajectory), restarting a new pointing trajectory when the next cursor move event is triggered.

Additionally, pauses that occur along cursor trajectories are useful to segment pointing trajectories. Aimed movements consist of several sub-movements separated by pauses [Meyer et al., 1988], so trying to delimit a pointing trajectory as a cursor movement without pauses is not feasible.

*Calculating valid user pauses.* A pause during a movement may correspond to a sub-movement transition or, in the case it lasts sufficiently long (movement stop), to cursor trajectory segmentation (Figure 12).

Unlike other studies where a unique value for all users serves to distinguish between valid pauses during cursor aimed movements and movement stops [Chapuis, Blanch and Beaudin-Lafon, 2007; Gajos, Reinecke and Herrmann, 2012], we followed a different approach for this purpose. As we observed from the interaction of the physically impaired participants of our study, valid pauses during movements vary among the users depending on the pointing device used and on their ability to handle it (Figure 12). For instance, a numeric keypad user needs more time to change the cursor trajectory (there being notable pauses between strokes) than a joystick user. We assume that calculating a movement stop threshold for every user will improve the quality of the segmentation and therefore the pointing trajectory-related measurements.

To calculate each user movement stop threshold, we took into account all the intervals where the cursor velocity falls to zero within each page, registering these durations by user. From this data, we calculated the median value of all collected observations for each user, and so discarded stops duration that were two quartile deviations or more away from the participant's median. We used the median and quartile deviation values to reduce the importance of outliers within each distribution. Through an extensive observation of our data we concluded that values obtained this way were reasonable.



**Figure 12.** Beginning of pointing trajectories estimations for 3 physically impaired participants using different computer pointing devices: (a) joystick, (b) numeric keypad with a head pointer and (c) oversized trackball.

*End of movement.* As mentioned above, we only focus on pointing trajectories that end with a navigational click and a posterior page load event. Thus, we discarded from our data logs all pages lacking cursor trajectories pointing to navigational click trajectories (for instance pages navigated using keyboard shortcuts or browsing history navigation without moving the cursor).

#### 4.4. Assessing Accessibility, Usability and User Experience of RemoTest

In order to validate the accessibility and usability of the Participant Module, a comprehensive study was carried out with people with motor impairments, low vision and the blind. The study also covered the installation process. The installation process was considered because of its importance due to the fact that it is always the first step towards using a system. Nevertheless, little information about this issue (accessibility, usability, problems, etc.) can be found in the literature.

##### 4.4.1. Participants

Thirty-six users were recruited for this experiment. All of them had some experience in using computers and Internet browsing, which was a requirement of the study. Figure 13 shows the distribution by disability of the user group.

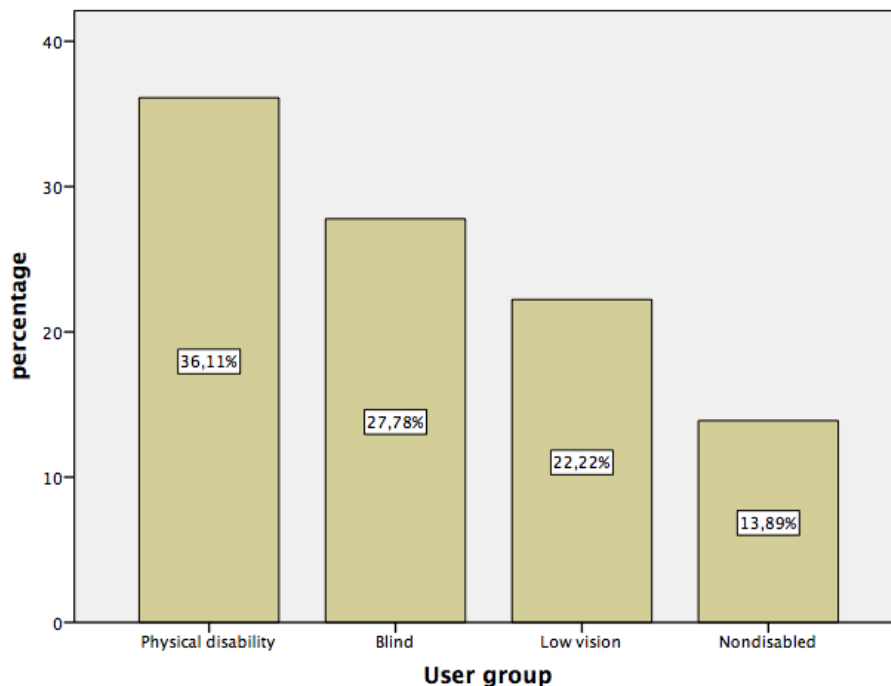


Figure 13. Distribution by disability of the user group

In order to control the cursor, participants with physical disabilities used a joystick, an adapted mouse, a head pointer, or specific configurations (such as switching right and left button functions in the mouse or the touchpad). All blind persons in the study, used JAWS screen reader (one of them used it jointly with a Braille display). Participants with low vision used the ZoomText screen magnifier software, browser zoom

functionalities or system settings to obtain high contrast interfaces.

#### 4.4.2. Tasks

- (1) Install the RemoTest Client (Firefox add-on)
- (2) Fill Questionnaire 1, pragmatic attributes, SAM
- (3) Fill Questionnaire 2, Demographic data

#### 4.4.3. Materials & Equipment

Participants were encouraged to use their own laptop and assistive technology, whenever possible, to evaluate RemoTest on different platforms and settings. Operating systems ranged from Windows XP to W7, and Firefox versions from 8 to 22.

Unlike the previous test, where the interfaces were created with XUL, in this case the interfaces were coded in HTML to ensure maximum compatibility with the different assistive technologies.

Questionnaire 1 was used to measure the perceived usability of the installation process and the emotions felt by participants during the installation. The first part of the form was based on the Attrakdiff questionnaire and consisted of a set of 7 word-pairs reflecting opposite adjectives to be rated on a 7- point Likert scale: technical-human, complicated-simple, impractical-practical, cumbersome-straightforward, unpredictable-predictable, confusing-clearly structured and unruly-manageable.

The second part of the questionnaire consisted of three questions to be rated on a 9-point scale based on the Self-Assessment Manikin (SAM) to measure the emotions felt during the installation process.

To measure the accessibility of the tool and installation process, videos and annotations were analysed, and a semi-structured interview was made to gather participants' opinions.

#### 4.4.4. Procedure

The sessions with participants were conducted one by one. A web page giving instructions (text and graphics) on how to install the add-on was provided to participants. After the installation, questionnaires 1 and 2 were automatically presented to users. Once the participant finished the tasks, the interview was carried out.

#### 4.4.5. Results

*Accessibility.* Even if some accessibility issues were encountered during the installation process, all participants were able to solve them and install the tool without significant difficulties.

The most critical issue was a popup window used by Firefox to inform users about the installation. Some screen reader users did not notice the alert and other users with low vision had some difficulties to find the popup since it was located in the upper left corner.

Chi-square and Fisher's exact tests were carried out in order to determine whether there were significant associations between variables of users' characteristics (user

group, assistive technology used, age, Internet usage experience, Internet expertise level, Internet use frequency, Mozilla Firefox browser experience, etc.) and the opinions of the participants about the complexity or simplicity of the installation process. No statistical evidence of associations was found.

*Usability and User Experience.* Each pragmatic quality (PQ) attribute, of the Attrakdiff [Hassenzahl, Burmester and Koller, 2003] questionnaire, was analysed separately. The results are presented in Table V. As can be observed, the Corrected Item-Total Correlation for each item is favourable or positive as all the values are above 0.4 except one: the technical-human item (0.387). This result confirms that this item had less consistency compared with the other attributes. However, the general Cronbach's Alpha value increment is low when removing this item (from 0.864 to 0.883). These results suggest that the item has moderate reliability. During an interview that was carried out after the installation task, some users commented that they had difficulties understanding the Technical-Human adjectives pairing.

**Table V. Cronbach for the items of Perceived Usability/Pragmatic Quality attribute**

Pragmatic quality (PQ)	Corrected Item-Total correlation	Cronbach's Alpha if item deleted
Technical—Human	.387	.883
Complicated—Simple	.649	.844
Impractical—Practical	.725	.831
Cumbersome—Direct	.681	.838
Unpredictable—Predictable	.774	.827
Confusing—Clear	.641	.843
Unruly—Manageable	.652	.842

The values obtained revealed a good perception of the pragmatic quality of the installation process of the RemoTest tool, as can be seen in Table VI. On the contrary, no reliable data can be gathered from the emotions scale. Nevertheless, the data obtained from questionnaires (Table VII) were positive for Pleasure and Dominance. Although Arousal had a lower score than others, it was still positive: 4.83 out 9, however, the standard deviation was too large

**Table VI. Mean, median and standard deviation for each of the pragmatic quality attributes**

	Technical—Human	Complicated—Simple	Impractical—Practical	Cumbersome—Direct	Unpredictable—Predictable	Confusing—Clear	Unruly—Manageable
Mean ( $\mu$ )	4.25	5.92	5.22	5.56	5.39	5.58	5.28
Median	4.00	7.00	6.00	6.00	6.00	6.00	6.00
Standard deviation ( $\sigma$ )	1.991	1.500	1.758	1.731	1.591	1.680	1.907

**Table VII. Cronbach for the different scales that were used.**

	Pleasure	Arousal	Dominance
Mean ( $\mu$ )	6.36	4.83	6.53
Median	7	5	7
Standard deviation ( $\sigma$ )	1.72	2.70	2.16

As can be seen in Table VIII, regarding effectiveness, all users groups were able to complete the questionnaires within a reasonable period of time. The time needed to complete Questionnaire 1 was higher than the time for Questionnaire 2 for all the user groups. This was because the questions in Questionnaire 1 (pragmatic attributes + SAM) were more complex than the ones in Questionnaire 2 (demographic data).

**Table VIII. Mean and standard deviation for the time required to complete the questionnaires by user group**

User group	Questionnaire 1		Questionnaire 2	
	Mean ( $\mu$ )	Standard deviation ( $\sigma$ )	Mean ( $\mu$ )	Standard deviation ( $\sigma$ )
Blind	349.50	205.69	295	200.76
Low vision	362.13	151.84	183.88	84.79
Physical Disability	610.77	430.36	293.69	129.29
Non Disabled	188.20	79.80	43.80	28.21

Participants with physical disabilities obtained the highest mean value (610.77 seconds) for filling in Questionnaire 1, although the standard deviation was also very high (430.36). Therefore, the high time required by participants with physical disability cannot be taken as a central trend. The data scatter is due to the fact that two participants took a considerably long time, as they used head pointers and also because they had problems to understand the pragmatic quality attributes.

*Satisfaction.* A final interview intended to gauge participant satisfaction and acceptance of the RemoTest tool was conducted. It was a semi-structured interview and participants were asked three questions in order to obtain their opinion:

- Question 1: What do you think about having a system for conducting inclusive experiments remotely?
- Question 2: Would you participate in remote experiments? Would you encourage friends to participate?
- Question 3: Would you feel more comfortable doing the experiment remotely (from home, office, etc.)?

All participants except one thought that it would be interesting to have such a system to conduct experiments remotely. One expressed concerns about security. He did not intend to install the add-on on his computer as he thought such a system could obtain personal data. Nevertheless, he would be prepared to try if the experimenters were people whom he trusted.

Most participants responded that they would participate in remote experiments and would encourage friends to do so. However, some participants thought they would participate only if it were not too time-consuming.

Three participants said that they would be more comfortable in remote settings as long as they were provided with a chat system to resolve any problem during the experimental sessions. Three other participants would prefer to carry out experiments in local settings. They claim that they are more comfortable when the experimenters are close to them so they can ask about any doubts concerning the tasks. The remaining participants responded that they would be comfortable in either a local or in a remote setting.



#### 4.4.6. Discussion

The results gathered served to explore the research questions defined for the experimental study. They revealed that the installation process of the RemoTest tool proved to be successful, irrespective of the disability or the assistive technology used by participants. All participants were able to install the tool even if some screen reader users and low vision users had to cope with problems when managing popup windows.

The usability of the installation process was tested according to the pragmatic quality attributes. All participants gave positive scores to the 7 pairs of adjectives in Questionnaire 1. Their feelings about the installation process were also positive. 82.35% of the participants rated the ease of the installation process with a score greater than 4 in a 7-point Likert scale. Statistical associations between variables of users' characteristics and their perception of the complexity of the installation were not found. This would support the assumption that the tool is accessible, simple and usable, irrespective of the participants' characteristics and the assistive technology used (visual impaired or physical disability).

Regarding the accessibility of the RemoTest itself, this proved to be accessible to all participants. All of them were able to fill in the questionnaires displayed by the tool. The main concern commented on by participants in the interviews were related to the semantics of the questions. This highlights the importance of using a clear and easy language.

The last interview revealed interesting data about user satisfaction and acceptance of the tool. All but one participant found these kind of tools interesting for conducting experimental sessions in local or remote settings. However, some of them showed some reluctance to perform remote experiments. Their main concerns were the duration of the experiments and the problem solving mechanism during the sessions. They would appreciate some kind of support such as chat systems in order to be able to ask for help if they were locked in any step or needed some clarification about tasks. Nevertheless, most of them showed a positive attitude towards participating in other experiments even if they were to be remotely performed.

Although the results are promising, more testing should be carried out so the results can be extended to a wider range of users, such as people with hearing loss, people with cognitive problems or the elderly.

### 4.5. *Analysing Validity of Data and Obtaining Characteristics from Cursor*

#### *Measures*

The objective of this study was first to analyse the adequateness of the proposed heuristic and to compare the different measures obtained from the participants that took part in the study.

The Curvature Index (CI) and the time needed to Point & Click where specifically analysed to check the heuristic correctness. Cursor speed was also analysed to compare the different users that took part in the experiment.

#### 4.5.1. *Research Questions*

- Q1: Are cursor measures more accurately described with the use of the heuristic?



- Q2: Are automatically obtained cursor measures helpful to compare users and devices and to detect problems?

#### 4.5.2. Participants

Sixteen participants took part in the experiment, 11 people with upper-body physical impairments (U01-U11) and 5 able-bodied people (U12-U16). Web expertise varied among participants as well as the assistive technology and the input devices used. Table IX shows information about the participants, the input device used and experiment location.

#### 4.5.3. Equipment

Users were encouraged to use their own computer when possible. In Elkartu<sup>1</sup>, the computers provided by the association where used. The platform used was similar in all cases: Windows operating system (Windows 7 except U02 who used Windows XP) and Mozilla Firefox browser with the RemoTest PAm add-on installed.

**Table IX. Participants information**

User	Expertise	Input Devices	Setting
U01	Medium	Reconfigured mouse	Elkartu
U02	High	Oversized trackball	Lab
U03	High	Joystick	Home
U04	Medium	Joystick	Elkartu
U05	Medium	Joystick	Elkartu
U06	Low	Joystick	Elkartu
U07	Low	Reconfigured touchpad	Elkartu
U08	High	Numeric keypad	Home
U09	High	Numeric keypad	Home
U10	High	Head pointer & Numeric keypad	Elkartu
U11	Medium	Head pointer & Numeric keypad	Elkartu
U12	High	Mouse	Lab
U13	High	Mouse	Lab
U14	High	Mouse	Lab
U15	High	Mouse	Lab
U16	High	Mouse	Lab

#### 4.5.4. Stimuli

The stimuli proposed was the Discapnet website. Discapnet is a website specialized in providing information to people with disabilities, organizations or relatives of people with disabilities. They provide news, documentary collections, information about the rights of people with disabilities, etc.

#### 4.5.5. Tasks

- Task1: Navigate for 5 minutes without any specific objective on the Discapnet website
- Task2: Find, within a 10 minute time limit, a list with addresses of day care institutions for people with cognitive disabilities on the Discapnet website

<sup>1</sup> Elkartu are the headquarters of the local Association of People with Physical Disabilities [<http://elkartu.org>]

#### 4.5.6. Procedure.

All users followed the same task sequence. First, they navigated without any specific target for 5 minutes, they were free to explore and read articles. The main idea of this task was to get a quick overview of the Discapnet website and its structure.

After this time, they were given a search task with a time limit of 10 minutes (Task2.). The reason for selecting a search task was that we expected that people would visit more web pages and as a consequence they would produce more clicks for analysis.

#### 4.5.7. Results.

Cursor measures were calculated from the interaction data gathered in Task 2. In this case, a total number of 323 web pages were visited by the participants in this task. The total number of visited pages analysed by user can be seen in Table X.

Data from 133 web pages were selected to perform the cursor movement analysis (23 web pages were excluded because of their lack of cursor movements as they were the result of repeatedly pushing the browser back button. 167 web pages were removed because the cursor was out of the web browser window, because the use of browser tabs, or because the cursor had been over an iframe).

**Table X. Information about the total number of visited pages (VP) in Task 3 and the analysed pages (AP) for cursor movement characterization.**

User	Visited Pages	Analysed Pages
U01	8	5
U02	32	10
U03	33	12
U04	14	1
U05	5	3
U06	5	2
U07	6	2
U08	31	15
U09	15	7
U10	13	6
U12	26	10
U13	9	4
U14	40	24
U15	50	15
U16	36	17

After pre-processing, the users with fewer than 5 pages visited were excluded (U04, U05, U06, U07 and U13). U11 was also excluded since she decided to leave the experimental session before finishing the task.

The first step in the heuristic was to calculate the median values of the cursor motionless intervals for each user. In Figure 14 the values calculated for each user can be seen.

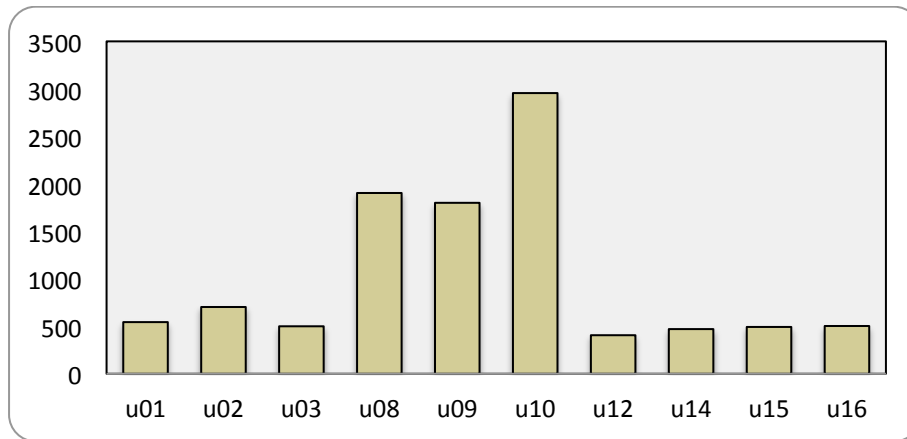


Figure 14. Median values by user for motionless intervals along cursor trajectories

A Kruskal Wallis test revealed a significant effect of the device used by participants on the motionless intervals,  $\chi^2(3) = 144.08$ ,  $p\text{-value} < 2.2e-16$ . A post-hoc test using Mann-Whitney tests with Bonferroni correction showed significant differences between Keypad – Joystick  $p=6.1e-16$ , Keypad – Mouse  $p=2e-16$ , Keypad – Trackball  $p=3.9e-11$  and Trackball - Mouse  $p=0.011$ . On the other hand, no significant differences were found between joystick - mouse or joystick - trackball.

As can be seen, U08, U09 and U10 had much longer pauses compared with the others. U10, who was using a keypad and a head pointer, needed even more time, around 1 second more, than the other two keypad (only) users.

Once the motionless intervals were calculated, the aforementioned heuristic was applied to estimate the aimed movements. According to this estimation, the Curvature Index (CI), point & click and the speed were calculated. Figure 15 shows the CI boxplot

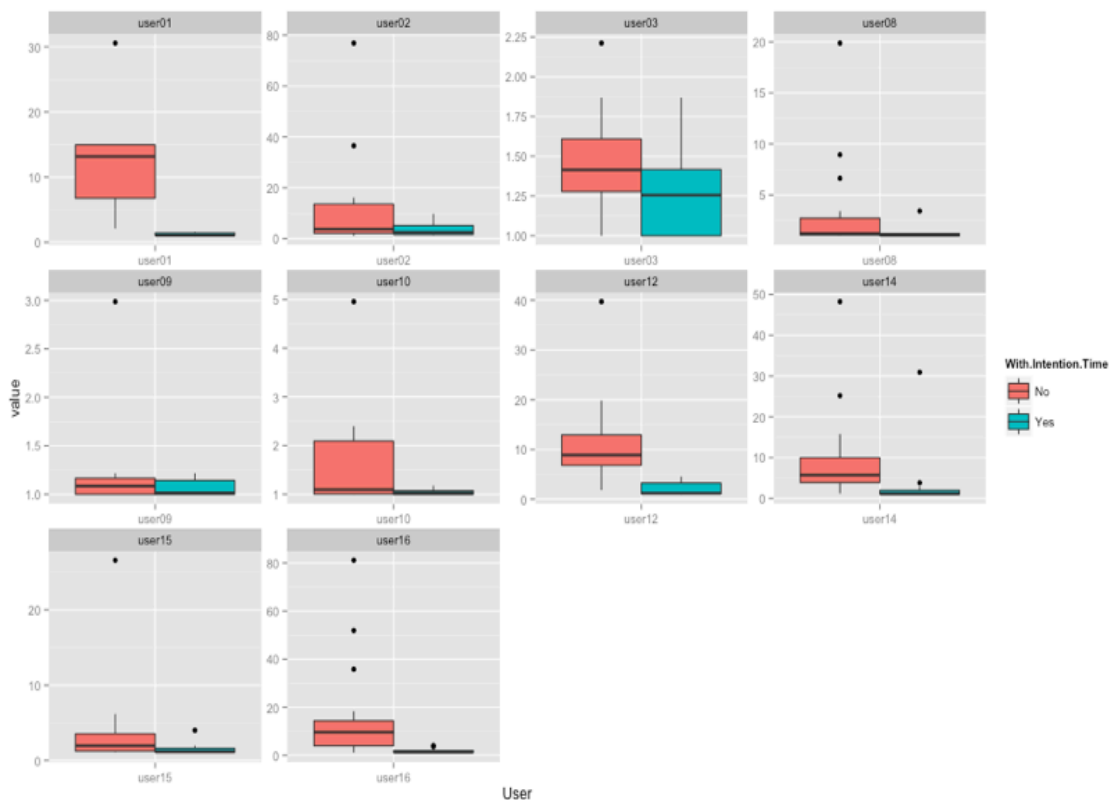


Figure 15. CI distribution boxplot by user calculated with an without estimation

calculated with and without estimation for each user.

Significant differences between the values obtained with and without the use of the heuristic were found with a Wilcoxon test,  $Z = 7.2571$ ,  $p < 0.01$ . Values obtained with the heuristic are closer to one, the highest value for the CI, for every user. The dispersion of the data is lower in most cases when it is calculated with the heuristic. Notable differences in the values, calculated with and without the estimation, can be found for mouse users (U01, U12, U14, U15 and U16) or trackball user U02. The influence of the estimation seems to have a lower impact for the other users.

The values for the Point & Click boxplots calculated with and without estimation can be seen in Figure 16. Differences between measures were significant  $Z = 7.2021$ ,  $p < 0.01$ . Point & click calculated with the heuristic has notable lower durations and data dispersion. The CI is also lower.

In order to validate the automatically obtained results, recordings of participants with physical disabilities were manually analysed and the time required for pointing to a target measure was annotated for each web page visited.

Along with the values obtained with the proposed heuristic, calculated with a threshold value to distinguish the pauses for each participant, values were also calculated with a one-second threshold for the pauses for all participants. This threshold value, has been used in other studies such as [Gajos, Reinecke and Herrmann, 2012] or [Hurst et al., 2013]. The mean of automatically obtained values was compared with the mean of those obtained manually (Table XI).

It can be observed that the values obtained with the heuristic were generally lower than those gathered manually from the video analysis. On the other hand, the values obtained automatically without the heuristic were in general greater than the ones gathered manually. The values calculated automatically with the one-second threshold, were even lower, especially for the keyboard users (U08, U09 and U10).

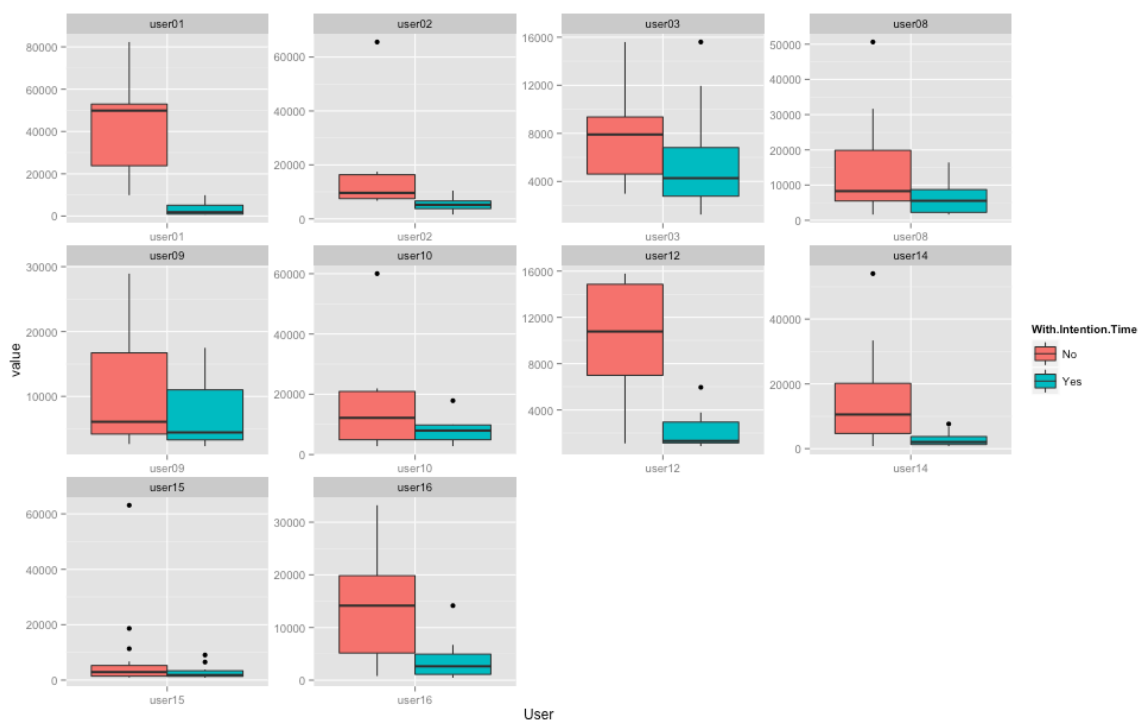


Figure 16. Speed distribution boxplot by user calculated with and without estimation

**Table XI. Mean values of the time required for pointing to a target automatically obtained by RemoTest platform (APM), estimation with one second (A1S), with no estimation (WILD) and manually obtained based on video analysis (MPM).**

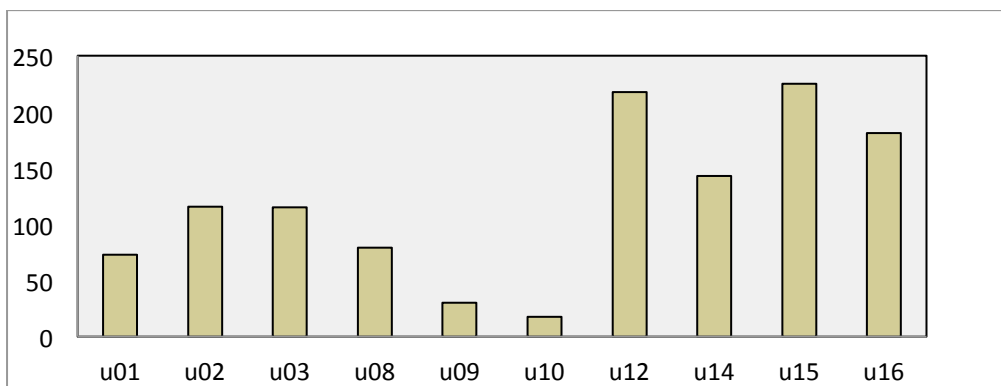
	U01	U02	U03	U08	U09	U10
APM (ms)	2318	2314.6	4289.5	5107.6	5728.7	6349.2
A1S(ms)	2038	1565.5	4289.5	447.3	670.9	251.3
WILD(ms)	42371.6	13129.4	6380.1	13110.7	9528.9	16782.8
MPM (ms)	6440	5080	3240	7190	8570	20480

Kendall's Concordance Tests was performed to analyse the agreement between rankings. The scores obtained for the concordance coefficient were 0.73  $T = 13$ ,  $p$ -value = 0.055 using the values calculated with the heuristic, indicating that there is some correlation between the values obtained manually and the ones obtained with the heuristic. As far as the values calculated with a one-second threshold are concerned,  $T = 2$ ,  $p$ -value = 0.055 with a coefficient of -0.733, these indicate that there exists a kind of negative correlation. On the contrary, non-correlation could be found with the values calculated without the heuristic  $T = 9$ ,  $p$ -value = 0.719 with a 0.2 coefficient.

With the data gathered automatically using the heuristic, the speed of the cursor was calculated. Figure 17 shows the median for each user. A Kruskal Wallis test was used to find differences. The test showed that significant differences existed for the cursor speed in the device used  $\chi^2(3) = 33.963$ ,  $p < 0.01$ . A post-hoc analysis showed differences between, keypad- mouse  $p < 0.01$ , keypad-trackball  $p = 0.031$ .

The CI median values for each user can be found in Figure 18. Significant differences can be found for the device used  $\chi^2(3) = 28.909$ ,  $p < 0.01$ . The post-hoc analysis revealed differences between keypad-mouse  $p = 0.00091$ , keypad-trackball  $p < 0.01$ , joystick-Mouse  $p < 0.01$ , joystick-trackball  $p < 0.01$  and mouse-trackball  $p < 0.01$ .

The time needed to click and point can be seen in Figure 19. Significant differences can be found for the time needed to point a target  $\chi^2(3) = 11.144$ ,  $p$ -value = 0.01, for the device used. The post-hoc analysis showed that only the keypad-mouse devices ( $p < 0.01$ ) had a significant difference. Regarding the time needed to click on an element since the previous mouse move, significant differences can be found:  $\chi^2(3) = 51.545$ ,  $p < 0.01$ . The post-hoc analysis shows differences between Keypad-mouse  $p < 0.01$ , keypad-trackball  $p = 0.02$ , Joystick-mouse  $p < 0.01$ , joystick-trackball  $p < 0.01$  and trackball-mouse  $p < 0.01$ .



**Figure 17. Comparing chart about the speed parameter automatically obtained for each participant.**

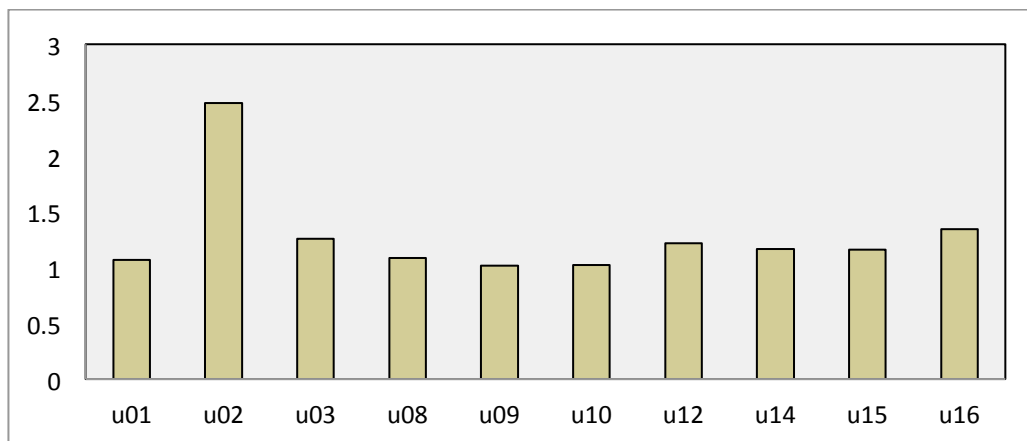


Figure 18. Comparing chart about the curvature index automatically obtained for each participant

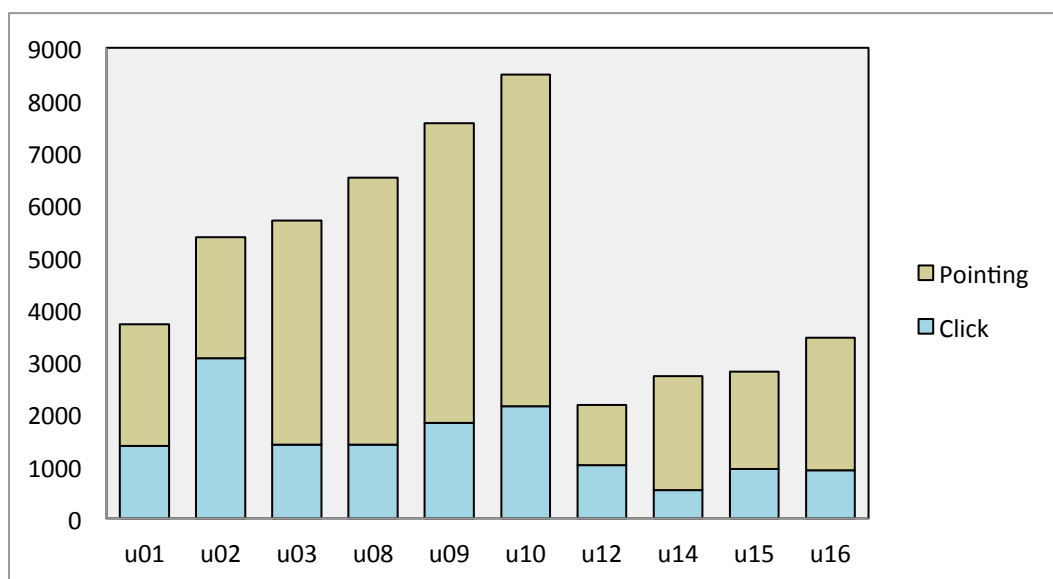


Figure 19. Comparing chart about the time for pointing a target and time for clicking on a target for each participant in the experiment

Finally if both measures are taken into account, significant differences can be found:  $\chi^2(3) = 29.514$ ,  $p\text{-value} = 1.746e-06$ . Nevertheless, the post-hoc analysis revealed significant differences between mouse-keypad  $p = 1.3e-05$  and mouse-trackball  $p = 0.017$ . The mouse-joystick was very close ( $p = 0.057$ ) but it was not significant.

#### 4.5.8. Discussion

The results obtained automatically by the RemoTest platform proved to be useful for characterizing the cursor movements and detecting different profiles between participants in a formal experiment. In fact, the observation of values obtained in curvature index, cursor speed and the time to click on a target can assist researchers in detecting problematic situations in experimental sessions due to lack of precision, inappropriate target dimension, features of assistive technology used by participants, etc. It would be possible to detect some problematic situations, even if the experiment were carried out in remote settings. For instance, the behaviour of Participant U01, who uses a standard mouse, differs in speed values to other participants using the same input

device. It may be concluded that this participant requires some assistance, for instance by means of web interface adaptation mechanisms in order to improve his performance.

Observing the results in Figure 14 it can be seen that Participants U08, U09 and U10 produced considerably higher values for their typical pauses between cursor sub-movements. These values may indicate difficulties for starting the cursor movements or participant fatigue during experimental sessions. In this case study, U10 is the participant who takes the longest (955.5ms). This participant used a head pointer that may cause fatigue and sometimes disorientation as the cursor is out of sight when pressing the key during cursor movements.

The algorithm for automatically calculating the required time for pointing to a target proved to be useful for ranking purposes. The values obtained are not accurate compared to the values obtained manually. However, this could be due to a lack of data for automated analysis after the pre-processing. The current version of the RemoTest platform is more accurate in this sense since most errors when gathering coordinates of the cursor position have been fixed.

#### ***4.6. Case Study 1: Users with Low Vision***

In the previous section, we gave details about the validity of data and about how to extract user characteristics from cursor related measures. In this case study, carried out with people with low vision, the RemoTest platform is used to detect accessibility and usability problems in use.

##### *4.6.1. Participants*

Six participants with low vision took part in the experiment. Three used a magnifier software such as ZoomText (U1, U3, U5) and the other three, browser zooms (U2, U4, U6).

##### *4.6.2. Materials & Equipment*

Participants were encouraged to use their own laptop and assistive technology. Operative systems were Vista, W7 and MacOs and Firefox browser with the RemoTest installed.

##### *4.6.3. Stimuli*

The Discapnet website [[www.discapnet.com](http://www.discapnet.com)], which focuses on providing information to people with disabilities, was selected as stimuli for the study. The website officially conforms to AA level as defined in the WCAG accessibility guidelines. However, an expert evaluation based on the Barrier Walkthrough Method [Yesilada, Brajnik and Harper, 2009] revealed some accessibility barriers for people with low vision: insufficient visual contrast, too long text lines, inflexible page layout and non-implemented skip links.

The insufficient visual contrast barrier was detected in the title section as well as in the navigation menu (maximum ratio 3.54:1 based on Colour Contrast Analyser measures). Some text lines may be too long depending on the zoom used by participants. As a consequence, they may be required to use horizontal scroll in order to reach the content. Another accessibility barrier can be caused by a floating button

implemented in JavaScript, located on the right hand side of webpages in the Discapnet website. It overlaps the web content when the zoom is increased.

#### 4.6.4. Tasks

- Task1: Find the “Canal Senior” (Senior Channel) web page which is one click away from the home page. In Figure 20 the link to access to the section is highlighted

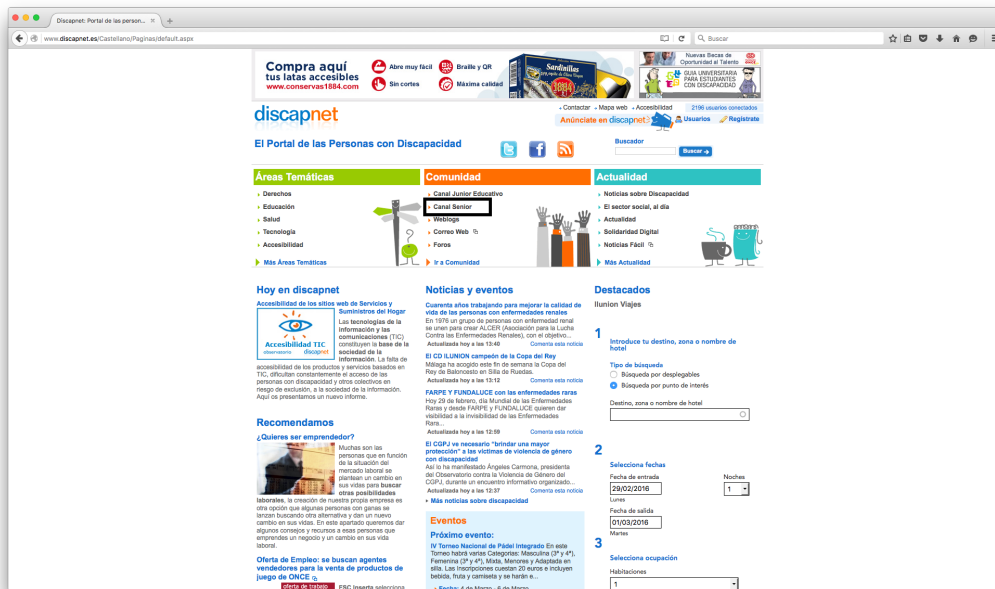


Figure 20. Discapnet web page with the target highlighted

#### 4.6.5. Results

Even if the target was not intended to be difficult, it was not an easy task for some users. U1 had several difficulties to find the target. He had a 0.807 value for Lostness and needed 2 minutes and 33 seconds to click on it. U5 could not finish the task in 4 minutes (Table XII).

The Number of pre-goal pages measures the times that a user has visited a web page with a link to the goal web page: three for U1, two for U4 and one for the other participants.

Table XII. Participants visited pages, Lostness, number of pre-goal pages, and time needed to complete the task

User	Visited Pages	Lostness	Number pre-goal pages	Time
U1	10	0.807	3	153
U2	1	0.0	1	18
U3	1	0.0	1	8
U4	3	0.417	2	44
U5	4	Not Found	1	240
U6	1	0.0	1	8



Analysing the pages visited, most of the users that had difficulties to complete the task thought that the target section was placed in the Health section or in a Thematic Area instead of in the Community section where it was actually placed.

A summary of the events gathered from the tool can be found in Table XIII. It can be seen how two users achieve the goal with only one cursor movement and completed the task in eight seconds. Even if the time was the same for both participants, the distance travelled by the cursor is much larger for U2 than for U6 (2955.5px>710,05px).

**Table XIII. Number of cursor movements and distances, croll actions and distance, and number of pressed keys**

User	Cursor Movements	Cursor Distance	Scroll actions	ScrollY distance	Pressed Keys
U1	73	17579.58	48	13914	11
U2	1	2955.5	0	0	0
U3	4	2860.36	3	428	1
U4	16	4954.55	7	1520	0
U5	25	19401.59	35	3623	42
U6	1	710.05	0	0	0

U1 was the user who used the most vertical scrolling followed by U5. The user that used the keyboard most was U5 (42 times). He used the “up” and “down” keys for scrolling, while the other users used the mouse wheel. U1, on the other hand, used the keyboard to perform changes in the magnifier software (ctrl, alt).

U5 provides an interesting case with regard to browser Zoom changes. He started with 300% zoom for the Discapnet domain. However, since Discapnet website is composed of different domains, when he entered the health section the domain was changed and in consequence the zoom value was changed automatically to 100% by the browser. U5 made a scroll action to the bottom of the page and then he came back to the top again (with a 100% zoom). After that, he increased the zoom value to 190%. Once again went down to the middle of the page. Then he changed the zoom level to 300% to navigate within a group of links in order to select one. He used the 300%, 190% and 100% zoom levels for 137, 17 and 9 seconds respectively throughout the task.

#### 4.6.6. Discussion.

The use of the Lostness measure with the pre-goal visited pages can be useful to detect problems. When Lostness is high but the pre-goal is low, it can mean that there is an error in the structure of the site or the item. Another possibility is that the section in which it is placed might be confusing or incorrectly set. On the other hand, when the Lostness and the pre-goal values are high, there may be a problem of visibility of the item. This can happen when the user is very close to the target, but he or she cannot see the element, or when the element has not been conceptualized adequately. This can happen because the text or the image is not meaningful, or because the interaction element does not look interactive.

The lack of responsiveness of the web site appeared to be another usability problem. A horizontal scrolling bar appeared when the size of the web site was increased. This can cause the user to lose information.

For example, due to the zoom used to visit the pages during web navigation, U5 should scroll horizontally a total of 2002px, in order to see the entire page. Since he only scrolled 125px horizontally on one page, he lost information and context.

Tracking the level of zoom applied by each user can be used to detect problems with element sizes but also to understand better the behaviour of the people that use browser zoom to magnify web pages. This allows us to study questions such as: When do they use zoom out and zoom in? With what purpose?

The distance travelled by the cursor is another interesting measure. In this preliminary study, participants that used magnification software made the cursor travel much more than the others. Comparing U3 and U6, who both took 8 seconds to complete the task, the travelled distances were 2860.36 and 710.05 respectively. Even if the time needed by U5 is much longer than the one needed by U1 (240 vs 153) the cursor distances were quite close (19401.59 vs 17579.58). This can be explained by the fact that users using magnifiers need to explore the web page moving the cursor up and down in order to displace the magnified window.

This suggest that this information can be useful for identifying users using magnifiers in order to propose to them adaptations that would require less effort when exploring a web page. For instance, adjusting the web page width to the magnified window without adding horizontal scrolling, so users are not required to move the cursor from left to right in order to explore the content.

#### ***4.7. Summary***

The RemoTest platform based on the theoretical model presented in Chapter Chapter 3 can be used to carry out experiments with people with disabilities.

During the development, some tests were carried out with people with disabilities. These tests showed the importance of including users during the development processes. Even if classic accessibility guidelines were followed, some accessibility issues were raised which would make the application very hard to use.

Taking into account the accessibility problems found in the first phase, the RemoTest platform was developed with HTML, avoiding XUL interfaces and implementing the concept model seen in Section 3. The RemoTest platform includes features for assisting researchers with the analysis of interaction data recorded during experiments. A straightforward visualization of each participant's interaction data in an understandable mode helps experimenters to discover, at a glance, valuable issues occurring during experiments and thereby to save a huge amount of time when analysing complementary video recordings if available. Additionally, the tool also performs heuristic estimations in order to obtain pointing trajectory-related measures that enable further understanding of participant behaviour.

In addition, RemoTest can also be used to test adaptive tools, new interaction paradigms, or to feed user models, as described in the following chapters.

---

---

## Chapter 5

# Adaptation Techniques for Making the Web Suitable for User Needs

---

---

An adaptation technique is the set of actions applied to a piece of code in a specific web page in order to enhance its accessibility, usability or its adjustment for the user. In order to be applied, adaptation techniques require information about the context, the function and the semantics of the element.

The quantity and quality of the techniques developed determines the capability of a system to adapt interfaces in order to meet user needs or suit the device capabilities. Some systems with few adaptation techniques are devoted only to solving specific problems for specific user needs. On the other hand, systems with a large range of adaptation techniques can be targeted at a wider range of users or devices needs.

Therefore, we have tried to gather and implement as many adaptation techniques as possible. These adaptation techniques were collected from the literature but also from the user needs detected in experiments that we had carried out previously.

In order to classify and order the adaptation techniques collected, a classification system was designed which is described in Section 5.1. This section also describes the characteristics taken into account when building the user model and the number of applicable techniques for each group.

We will describe only those techniques that have been tested in our user tests. The adaptation techniques described in the next section are those techniques devoted to improve accessibility for people with low vision and the ones for people with motor impairments using a touch screen device. The experiment carried out to extract the techniques are described in Chapter 6.

### *5.1. Adaptation Techniques Classification*

The survey of adaptive hypermedia methods and techniques by Knutov, De Bra, and Pechenizkiy [2009] was the basis for this work. In this document, adaptation techniques are classified in three main categories: content, presentation and navigation. This classification is quite standard and similar approaches can be found in other studies (such as those presented by Kobsa [2001] and Brusilovsky [2001]). Although there are terminology differences between these different works, the meaning is the same. In most cases techniques are classified into three main groups: the first group is for the techniques involving changes in the content displayed in the interface; the second group is for techniques involving style changes in the interface layout, and the third group is for those techniques dealing with interface structure and behaviour. Examples of techniques in each group include the following:

- Content adaptation technique: “1.1.6 Incorporate specific scrolling icons on each page”
- Presentation adaptation technique: “2.2.3 Place important areas of content near the top of the page”
- Navigation adaptation technique: “3.1.2 Create table of contents for the website”

We also have adopted this classification in our study. Table XIV shows the total number of techniques we identified for each adaptation classification.

**Table XIV. Total number of techniques gathered, classified by group: content, presentation and navigation adaptation techniques.**

Group	Total Number
Content (1)	39
Presentation (2)	37
Navigation (3)	23
Total	99

As can be seen, each technique has a notation for correct identification and classification. The numbering assigned to each technique identifies the group and the specific method to which the technique is related. For example, the number 1.5.3 identifies the third technique (3) that is related to natural language processing methods (5) and belongs to the content adaptation techniques group (1).

Different approaches have been considered for developing relationships between adaptation techniques and user characteristics: ISO/IEC TR 29138 [2016], Brajnik’s Barrier Walkthrough Method<sup>2</sup> and WCAG 2.0 [2016]. This analysis led us to define three general groups of users:

- Users with cognitive impairments (C)
- Users with physical impairments (P)
- Users with sensory impairments (S)

These user groups have been divided into subgroups based on the adaptation techniques we gathered. We defined 4 subgroups for cognitive impairments, 2 subgroups for physical impairments and 7 subgroups for sensory impairments.

The list of subgroups is shown in Table XV. More comprehensive user models can be found in the literature, but we decided to adopt a pragmatic user modelling approach. Therefore, only when we detected an adaptation technique related to a specific user characteristic did we define a subgroup of users. Note that this approach allows new subgroups to be incorporated when necessary.

Table XVI presents the number of techniques for each user subgroup classified by type (Content, Presentation and Navigation). It should be noted that some techniques can be replicated in different subgroups and groups. For example, the technique “3.1.2 Provide table of content for the website” is classified as necessary for the following subgroups in the cognitive impairments group: C.2 – Learning disabilities and C.4 – Reduced memory capacity. Furthermore, it is also included in some subgroups of the other two user groups: physical impairments and sensory impairments.

<sup>2</sup> Brajnik, G. Barrier Walkthrough Method, <http://sole.dimi.uniud.it/~giorgio.brajnik/projects/bw/bw.html>

**Table XV. Relation of subgroups defined in each general group of users: users with cognitive impairments (C), physical impairments (P) and sensory impairments (S).**

General Group	Subgroup	
C	C.1 Decline in maintaining attention	
	C.2 Learning disabilities	
	C.3 Language disabilities	
	C.4 Reduced memory capacity	
P	P.1 Limited movement	
	P.2 Inability to use mouse	
S	S.S.1 low vision	
	S.S.2 blindness	
	S.S Sight	S.S.3 colour blindness
		S.S.4 photosensitivity
		S.S.5 eye strain
S.H Hearing	S.H.1 hearing loss	
	S.H.2 deafness	

**Table XVI. Number of techniques in each user type subgroup also classified by three main technique type groups (Cont.-Content , Pres.-Presentation and Nav.-Navigation) and the total for each subgroup (Tot.).**

General Group	Subgroup	Number of techniques by group				
		Cont.	Pres.	Nav.	Tot.	
C	C.1	10	5	4	19	
	C.2	10	1	7	18	
	C.3	5	0	0	5	
	C.4	4	1	7	12	
P	P.1	5	3	7	15	
	P.2	2	1	6	9	
S	S.S.1	4	10	3	17	
	S.S.2	3	1	9	13	
	S.S	S.S.3	0	3	0	3
		S.S.4	3	0	0	3
		S.S.5	0	14	3	17
S.H	S.H.1	1	3	0	4	
	S.H.2	1	0	0	1	

Some techniques are not clearly classified in the literature, so we undertook their classification. In other cases, we assumed that some techniques were also of interest for different user groups which had not been considered by the source. These assumptions were made based on knowledge acquired in previous research work. For example, the technique “1.4.1 Stretchtext” was gathered from sources focused on general adaptation techniques and is not specific to any user group. We assume that it can be applied to users with decline in maintaining attention as when applied, less information is shown, only that which is most significant, and therefore the user has a clearer interface with fewer distracters.

In the next section the techniques that belong to the case studies described in the chapter are explained Chapter 6. These are devoted to improving accessibility and usability for people with motor impairments using touch screen devices. After that, we describe the adaptation techniques devoted to people with low vision, grouped according to the WCAG 2.0 criteria.

## 5.2. Adaptation Techniques for People with Motor Impairments Using Touch Screen Devices

### 5.2.1. Perceivable - Information and User Interface Components Must Be Presentable to Users in Ways They Can Perceive

Even if techniques related to this guideline are not strictly related to motor impairments, they also have other associated pathologies such as eyesight problems. For instance, they can benefit people with cerebral palsy or some elderly people. For this reason, a number of techniques for ensuring text legibility were also included.

Kane et al. [2009] found that some users may have problems because of low contrast or small font size. These barriers are often due to user characteristics, but on other occasions they are caused by environmental conditions. To ensure good contrast cream-yellow background, black text and blue links are used for the *main* content (which provides a contrast ratio of, at least, 9.41:1). In other sections of the page (*navigation, content-info, complementary, banner*) white background and black and blue letters are used (ensuring a contrast ratio of 9.65).

With respect to the font size, in the study carried out by Trewin, Swart and Pettick [2013], participants preferred font sizes ranging from 20pt to 56pt. For testing purposes we established a 24px font size, but in future versions the users themselves will be able to choose the font size that best suits their needs.

### 5.2.2. Operable -User Interface Components and Navigation Must Be Operable

In the study performed by Guerreiro et al. [2010], objectives that had a diameter of 12mm or longer were found to provide a good size/error ratio. For this reason we selected 12mm as the minimum size for the interaction elements (such as links or buttons). On the other hand, the document W3C Mobile Best Practices [2016] recommends a minimum separation, or inactive space, for small interaction elements. For this reason a 20px (4.46mm) space between links and buttons was added.

Regarding gestures, Trewin, Swart and Pettick [2013] found that the “tap” gesture was easily performed by 10 users out of 14. Three users had some level of difficulty and only one had serious difficulties. Yet they found that only in 48% of the interactions did the finger movement begin and end approximately at the same point, which is a necessary condition for a valid “tap”. They also found an average distance of 17.5mm between the target and the starting or ending point of the tapping for 28% of users. At the same time they noticed that actions such as “slide” (Figure 21 Left) and “pinch” (Figure 21 Right) were difficult for a large number of users. Similarly, Nicolau et al. [2014] verified that directional gestures were difficult, and that the most effective interaction technique was “tap” followed by “crossing”.

As a solution, Trewin, Swart and Pettick [2013] proposed new interaction techniques such as “steady tap” or “end tap”. “Steady tap” allows the user to select an item even if the finger moves away from the target within an established threshold. While “end tap” allows the activation of an element when lifting the finger from it. Both techniques allow the user to select an item even if there are uncontrolled finger movements during selection.

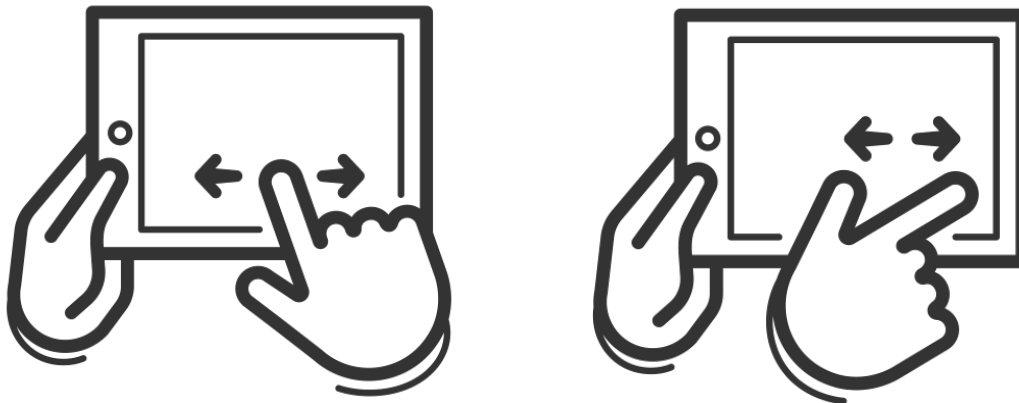


Figure 21. Slide gesture (Left) and pinch gesture (Right)

Since the “slide” gesture may be difficult or impossible for some users, as attested by Trewin, Swart and Pettick [2013] and Nicolau et al. [2014], we introduced buttons for scrolling in order to avoid making use of the “slide” gesture obligatory.

Bearing in mind that lack of precision can also be a problem, we decided to increase the activation area around the position where the finger landed: “augmented tap”. This reduces the precision requirements for selecting the desired item [Grossman and Balakrishnan, 2005; Findlater et al., 2010]. Nevertheless this can be problematic when there are other interactive elements close to the target. For these cases, a disambiguation list was added. The list is ordered by the distance from the finger to the targets, and from bottom to top as shown in Figure 22.

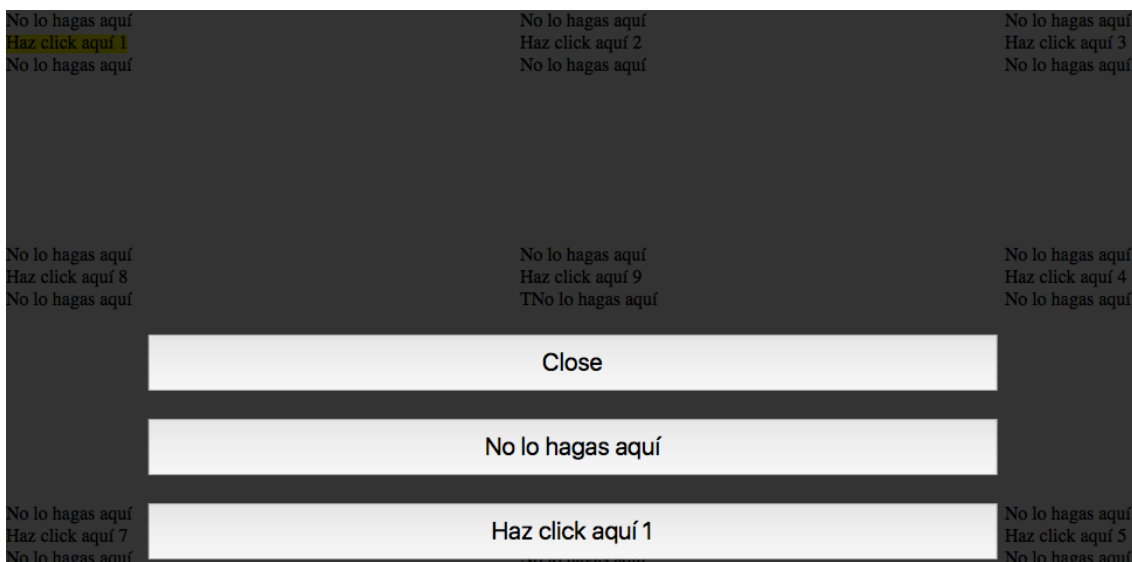


Figure 22. Disambiguation User Interface for the “augmented tap” method of interaction



### 5.2.3. Understandable - Information and the Operation of User Interface Must Be Understandable.

The layout for the adapted website is based on the WAI-ARIA landmarks (*banner, navigation, main, content-info and complementary*), grouping common elements as recommended by the W3C standard. The *banner* is located at the top, *navigation* elements on the left, *main* content in the middle and *content-info* at the bottom of the page. If there is any *complementary* content, it is placed on the right. In addition, *breadcrumbs* are inserted into the top of the *main* content.

Finally, a technique that eliminates non-essential page elements (such as advertising or unnecessary images) in order to make the interface clearer is also applied. Figure 23 shows a page before adaptation and Figure 24 shows the same page after applying all the adaptations techniques.



Figure 23. Discapnet web site

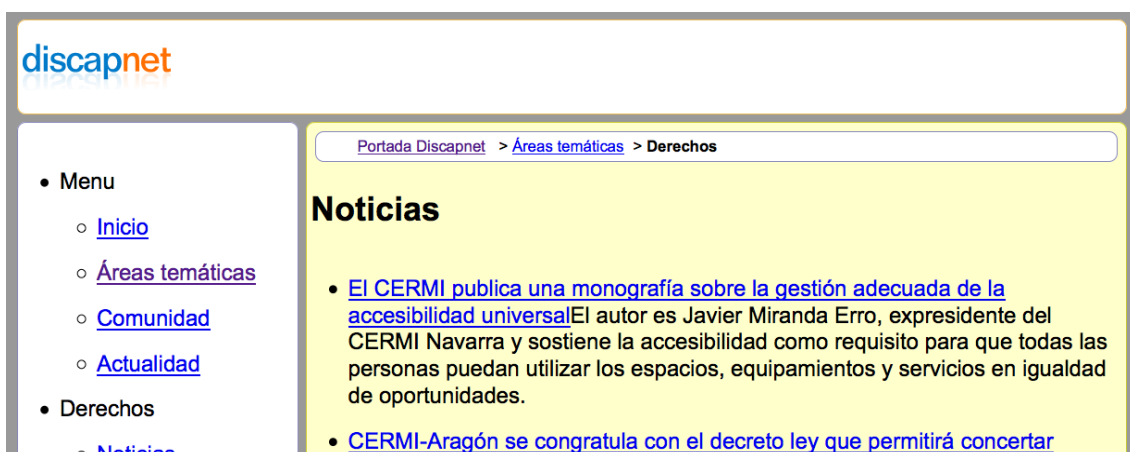


Figure 24. Discapnet web site after applying the suitable adaptation techniques



### 5.3. Adaptation Techniques For People with Low Vision

#### 5.3.1. Perceivable - Information and User Interface Components Must Be Presentable to Users in Ways They Can Perceive

Augmenting the text or the size of elements or providing an increased contrast are some basic techniques for improving web page accessibility for people with low vision [Kitchel, 2016; Theofanos and Redish, 2005]. As a starting point, the font size was set at a minimum of 14 points, in order to reduce the zoom increment/decrease operations required to explore the content. As a consequence, navigation becomes more comfortable “presentInformationInBigSize” [Kurniawan et al., 2006; Theofanos and Redish, 2005].

However, the selected font size might not be enough for some people, so the text must be resizable to enable customization: “provideResizableFonts”. In addition it should be taken into account that serif type fonts are not adequate for people with low vision. Therefore, sans serif fonts should be used [Kitchel, 2016]: “provideSanSeriffFont”.

Another important issue regarding perceptibility is that of ensuring sufficient contrast in the interface [WCAG 2.0, 2016]. This is achieved using white colour for the background and black for the fonts. Navigation menus are coloured with a light-green background and black fonts. This combination of colours ensures the required contrast ratio for content to be readable for users with low vision or colour blindness: “provideHighContrast”.

#### 5.3.2. Operable -User Interface Components and Navigation Must Be Operable

Many people with low vision use screen magnifiers. They have one drawback however as they reduce the field of view. This forces them to move the field of view while navigating in order to be able to read the content. When the content is larger than the window, scrolling becomes necessary. In this situation the user with low vision has to move the field of view to the scroll bar and to come back to the text, in order to continue reading [Theofanos and Redish, 2005]. Vertical scrolling can be *easily* done with the mouse. On the contrary, horizontal scrolling requires the user to move the point of view to the page bottom to perform the scroll and then go back to the text, which is very time consuming and somewhat annoying [Ahmadi and Kong, 2012; Theofanos and Redish, 2005].

Furthermore, multi-column pages can also be confusing for people with low vision. Sometimes they cannot even notice the presence of more than one column in the page, and consequently they lose information [Theofanos and Redish, 2005]. These problems can be overcome with the application of adaptation techniques such as “pageLinearization”, “provideFlexibleLayout” or “presentInformationOnCenter” [Ahmadi and Kong, 2012; Theofanos and Redish, 2005; Kurniawan et al., 2006].

Linearized content provides a single column. In conjunction with the “provideFlexibleLayout” technique it allows the page to be resized without having to resort to the horizontal scroll. The technique “presentInformationOnCenter” is intended to minimize the movements within the field of view and mitigate the loss of context that usually occurs using magnifier tools.

However, the application of these adaptation techniques has a side effect: They generate pages which require considerable vertical scrolling. For this reason other techniques are needed to add links in the page, such as to include links to skip

navigation menus or to place the focus on the most important content on the page, “provideInternalLinks”, “provideSkipLink”.

Regarding the interaction with links, in order to make identification unambiguous and easier to operate, a hot area is added to enable the activation of the element even if the cursor is not placed directly over the text. This activation area is visualized when the cursor is around it by, for example, a change in the background colour to yellow (providing high contrast with the blue colour to the text of links) [Kurniawan et al., 2006] “addHotArea” “visualizeLinksWithContrast”.

### 5.3.3. Understandable - Information and the Operation of User Interface Must Be Understandable.

Different adaptation techniques were selected with the aim of producing a clearer and more understandable user interface: removing elements, reordering the page etc., with the main objective of simplifying it.

Kurniawan et al. [2006] found that floating elements can be troublesome. Therefore they proposed to remove them: “removeFloatingText”. Moreover, a technique to remove other possible distracters such as banners, non-essential information was also applied: “removeDistracters”.

With the same aim, a technique to locate the most important elements near the top of the page was also adopted: “putImportantContentOnTop”. In order to help the user to easily figure out the different sections of the page, a technique to identify navigation elements by colour was applied: “putNavigationColor”. Nevertheless some users reported this as troublesome [Theofanos and Redish, 2005]. For this reason, a line delimiting each section was also added.

Finally, the general recommendation by the WCAG to underline links was also adopted: “underlineLinks”. Figure 25 shows a page before (right) and after the application of all the adaptation techniques described in this section.



Figure 25. Discapnet web site adapted (left) and the original (right)

#### *5.4. Summary*

This chapter has described the classification adopted for adaptation techniques and the groups of users that can benefit from them. This classification is also the basis for the user model described in the next chapter.

The pragmatic approach followed allows the easy addition of new adaptation techniques when necessary. At the same time, if an added new adaptation technique is useful for another user group, not present in the existing classification, this new user group can be added. This approach allows the extension of adaptation systems from particular cases to wider ranges of user types. This approach can be applied to the adaptation techniques gathered from the literature and to the adaptation techniques arising from the experiments carried out previously (Chapter 4).

The next chapter describes an adaptation system that implements the transformation presented in the conceptual model. To this end we use the classification and the adaptation techniques detailed in this chapter.



---

---

## Chapter 6

# Transcoding the Web to Enhance Accessibility and Usability

---

---

In the last few years considerable research effort has been focused on improving user experience when accessing the Web. Various tools and techniques have been implemented in order to help users to perform tasks in an efficient and satisfactory way. However, there are still several groups of users that experience barriers in their interactions. Moreover, the rise of Web 2.0 technologies may create new types of barriers [Zajicek, 2007].

This chapter presents Model Access, a tool devoted to transcoding web pages to fit user needs while they are navigating. Model Access uses an ontology to store and process the available information. The system described here, together with the RemoTest platform, implements the complete conceptual model proposed in Chapter 3. Following its description, two case studies are presented, both carried out with RemoTest. The first experiment studies how people with motor impairments use touch screen devices in order to find helpful adaptations. The second case describes an experiment carried out with people with low vision.

### *6.1. Model Access, a Transcoding System*

The purpose of the Model Access transcoding system is to adapt annotated websites. For the annotations we defined a language based on- and extending- the roles and properties of WAI-ARIA.

The WAI-ARIA [2016] language allows the addition of semantic annotations to HTML code by means of roles (main, navigation widget, etc.) or properties (aria-describedby, aria-live, aria-label, etc.). These new roles and properties can be used to help assistive technologies to manage the interaction elements present in a web page in a better way. However, this is not a popular language in the designer community. Conversely, the popularity of HTML5 [2016] among developers is increasing. HTML5 corrected some of the lack of semantics in previous HTML versions. It defines the interaction elements by their roles. In this way, assistive technologies can provide better and more accessible use of the marked websites. Since this new HTML version has been launched fairly recently, most of the existing websites are formatted with previous versions. For this reason, website adaptation is a particularly suitable approach for many users, as it allows them to perform tasks and find information more quickly, with fewer interaction steps and in a more satisfactory manner [Brusilovsky, 2004].

We defined an ontology for modelling user characteristics, adaptation techniques, annotated websites, and the relationships between them. The selection of the applicable

adaptation techniques is inferred from this ontology, based on several rules that associate user characteristics with adaptation techniques for specific interaction elements in the website.

### 6.1.1. Architecture

Transcoding systems are usually classified as client, proxy or server tools depending on where they are placed. The system presented in this thesis has a hybrid architecture since one module is located in the client and others are in a server:

- Presentation Module runs on the user device (PC, mobile, tablet)
- The Adaptation and Coordinator Modules, and the Knowledge Base run on a server

Let us provide an approximate description of the adaptation process. A non-accessible page is cached by the Presentation Module and sent to the Coordinator Module. Based on the information collected from the Knowledge Base, the Adaptation Module carries out the pertinent adaptations. Subsequently, the Presentation Module presents the modified page to the user. Figure 26 shows the process for adapting a page previously annotated by us and another with the WAI-ARIA annotations already incorporated.

The logic of adaptations is in the Knowledge Base. It decides which adaptations can be applied to specific elements, according to defined rules. The Adaptation Module performs all the pertinent adaptations. This architecture enables the set of adaptations to be changed without touching the code. For instance, it is possible to add new types of users or devices by simply updating the Knowledge Base as the Adaptation Module is agnostic with respect to both the device and to the user. These modules are described in detail below.

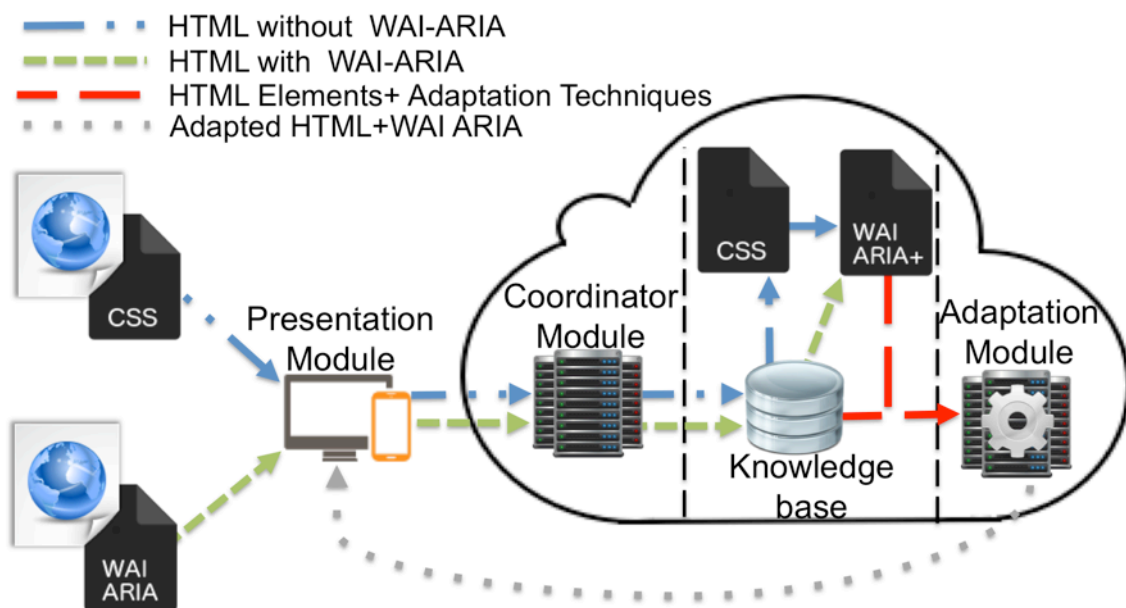


Figure 26. General architecture and workflow

*Presentation Module.* The Presentation Module is currently implemented as an add-on for the Firefox web browser and runs on PCs, mobiles and tablets. Despite being an add-on, this architecture enables easy migration to other platforms (chrome add-on, proxy, etc.) whenever the new Participant Module satisfies the following requirements:

- Identify the user
- Catch the web page
- Send the page to the Coordinator Module
- Get and present the modified page

In addition to these requirements, the developed Presentation Module manages preferences and collects user-generated events. These events can be used to feed specific data mining programs allowing them to detect changes in user skills, fatigue, deterioration, etc.

Regarding preferences, we implemented a preference manager able to make and display any changes made by the user in real time. This means that the user can immediately perceive the consequences of the chosen options. For example, Figure 27 depicts the preferences selection menu for enabling or disabling the scroll buttons.

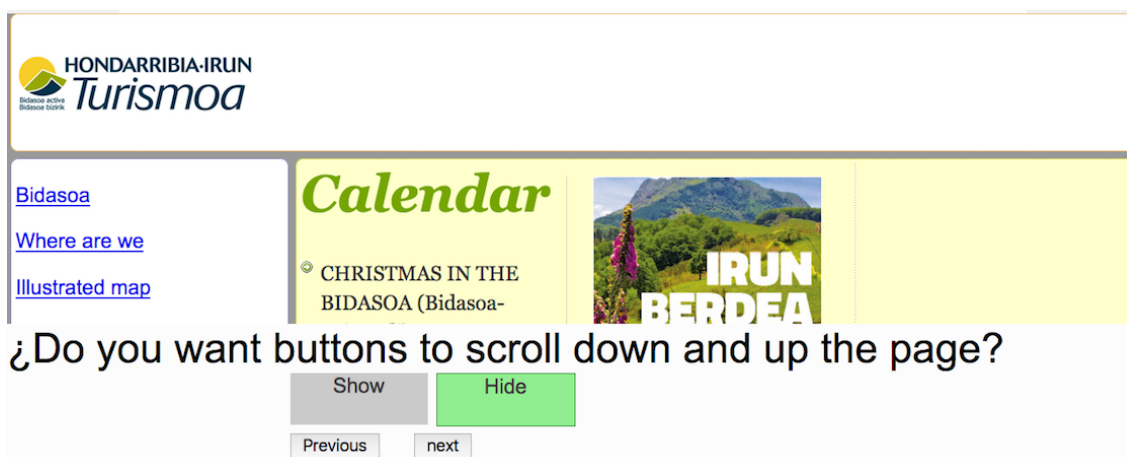


Figure 27. Bidasoa Turism web site with the preference manager user interface

*Coordinator Module.* The Coordinator Module has been implemented as a Web Service and is responsible for mediating between the various existing modules (Presentation Module, Knowledge Base, Adaptation Module). Its main tasks are:

- to establish communications with the Presentation Module
- to ask for the necessary adaptations to the Knowledge Base
- to update the Knowledge Base
- to communicate with the Adaptation Module

The Coordinator Module (see Figure 28) receives the page to be adapted from the Presentation Module along with the user credentials (username, password, device). Afterwards it obtains from the Knowledge Base the adaptation techniques suitable for the user, the device and the website.

Then the list of suitable adaptations and the original page are sent to the Adaptation Module, which performs the adaptations. Afterwards, the adapted page is returned to the Presentation Module.

When the user modifies his or her preferences a request is sent to the Coordinator Module, which updates the Knowledge Base. After that, it repeats the aforementioned process to adapt the page according to the new preferences.

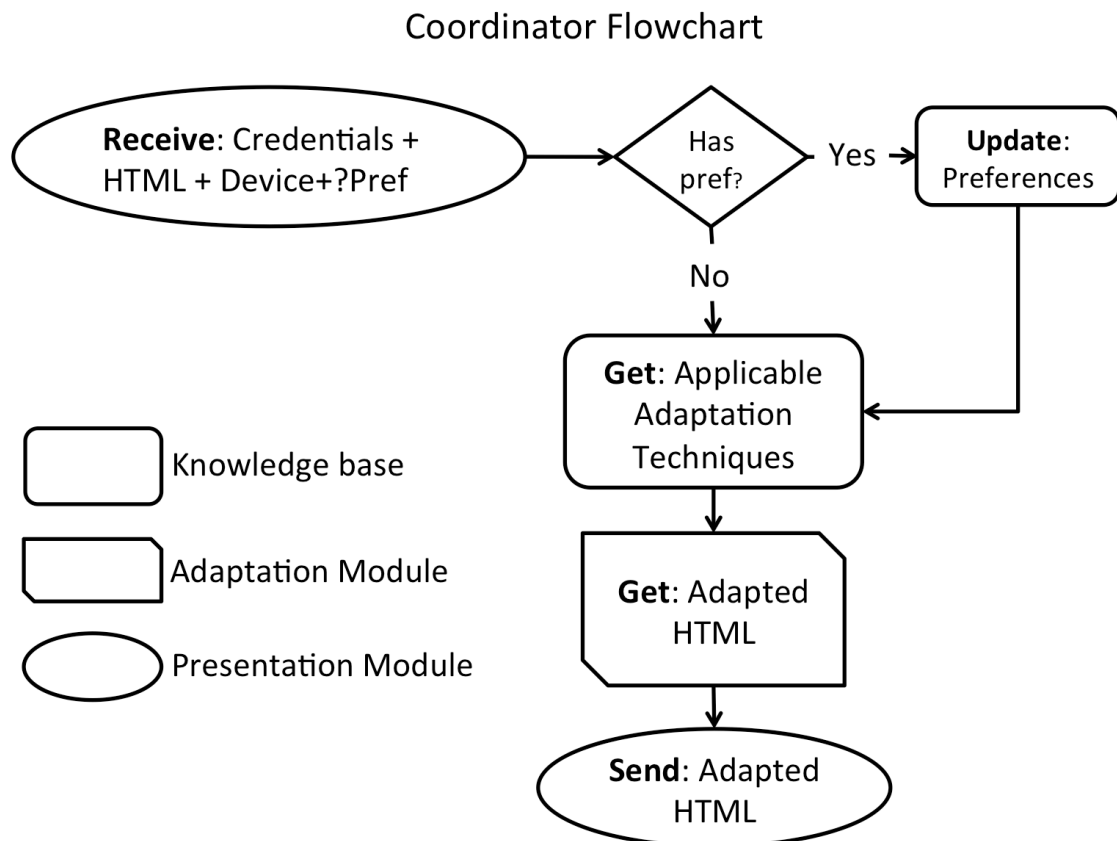


Figure 28. Coordinator module flowchart

*Adaptation Module.* This module is responsible for making the necessary adaptations. It receives the annotated web page and the list with the adaptation techniques that are applicable as input. Each adaptation technique to be applied has a number of parameters attached, composed of the web page elements identified by the annotation model. Some adaptation techniques do not require parameters. For example "provideEndTap" requires no parameters because it is an aid for interaction with no further configuration. Other techniques such as "removeDistractors" have as parameters those elements considered to be distractors.

After the application of all the suitable adaptation techniques the result is an accessible and adapted web page which is presented to the user.



*Knowledge Base.* The Knowledge Base is the core of the system. It was implemented in the OWL language [OWL, 2016]. It defines the user models, adaptation techniques, devices, assistive technology and the annotation model of the web pages. In addition, it includes the mechanisms necessary to select the most appropriate adaptations for a given user. Figure 29 shows the general concepts considered in the ontology and the following subsections describe each component of the ontology in detail.

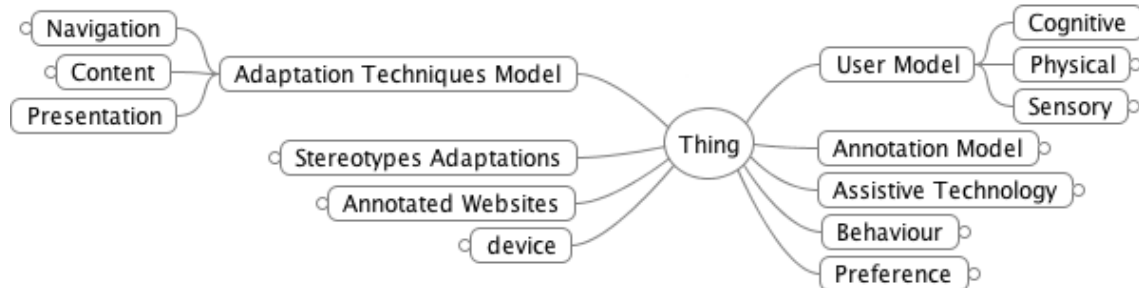


Figure 29. Upper ontology categorization models

#### *Annotation model*

Annotation adds semantic tags that enable the identification of the role of the elements to enable their adaptation. Using a mark-up language, the annotator can describe the role of interaction elements, taking into account the annotation model defined in the ontology. Annotations are stored on a different server to the original website and are made through the CSS elements and the DOM of the web page.

The annotation model is organized into two abstract classes: one for the roles and properties belonging to the WAI-ARIA mark-up and another for the new roles and properties created by us. The reason to extend the WAI-ARIA vocabulary is that the WAI-ARIA roles are sufficient to perform most adaptations but are not adequate for inserting new elements such as contextual help, tutorials, etc.

The new roles created are: *ContextInfo*, *Tutorial*, *FAQ*, *SiteMap*, *Captions* and *GeoMap* (see Figure 30). First, three elements were created to provide help to the user about an element, web page or even the entire site. The *SiteMap* role is used to identify an element that points to the sitemap or to create a new element if necessary. Captions are used to indicate where the captions are, and *GeoMap* is used to provide written directions, instead of a visual map.

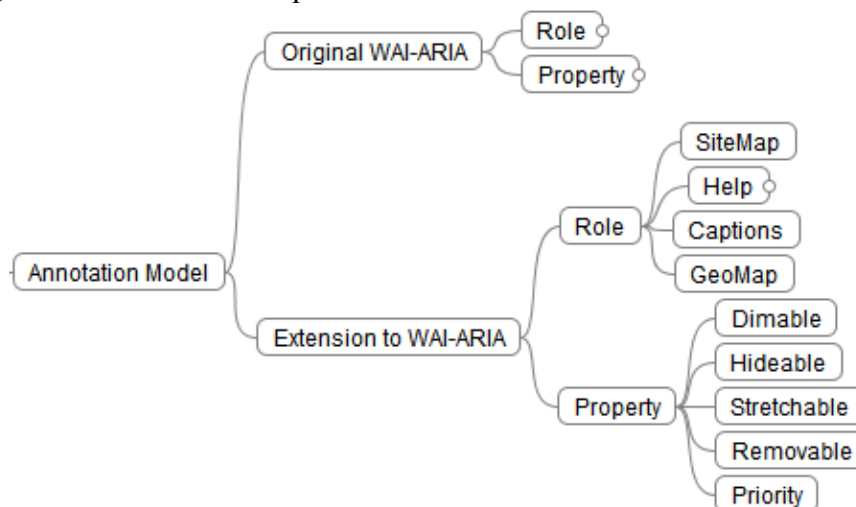


Figure 30. Web page annotation model (WAI-ARIA and added new roles and properties)

The new properties defined are: *dimnable*, *hideable*, *stretchable*, *removable* and *priority*. These properties are used to mark information or interaction elements as “optional” to simplify the interface for a given user. These properties are usually assigned to those interaction elements forming an intelligible chunk of information. Their annotation also includes a role, such as region, article, area, etc. Techniques such as “*dimming*” or “*stretchtext*” can be applied to these roles so that part of their content is hidden. For instance, an element tagged with “*removable=flashing*”, will be removed from the adapted interface for a user with photosensitivity or attention decline.

In the website annotation process, CSS elements, such as ids or classes, are annotated with semantic meaning by assigning the original WAI-ARIA or the defined new roles and properties. Nevertheless, CSS annotation is not enough for certain cases. For example, when the page does not use CSS, or when the ids or classes are not consistently used across the whole site. For those cases, XPATH annotation is used to assign a role or property to an HTML element.

This approach reduces the workload for the annotation process, as it requires less effort and maintenance work. In addition, pages with the original WAI-ARIA annotations can be adapted automatically even if they will not take advantage of the full capacity of the system.

#### *User model*

Different user characteristics were considered in building the user model. The ontology includes three general user interaction factors (see Figure 29):

**Cognitive:** Cognitive skills that are involved in human communication and in interaction with technological devices. Specifically, attention, learning, language processing and memory are considered.

**Physical:** Only mobility skills were modelled as physical user characteristics having an influence in the interaction with web pages. Limited movement and the physical inability to use a mouse are the two main disabilities considered in this model.

**Sensory:** Visual and hearing disabilities were included because of their relation to the interaction. Among the possible visual disabilities, blindness, low vision, colour altering, photosensitivity and eyestrain were included in the model. Among the hearing disabilities, deafness and hearing loss were taken into account.

#### *Adaptation techniques*

Although the adaptation techniques are also modelled in the ontology, our approach allows the extension of the adaptation knowledge and the easy integration of new techniques in a continuous process.

Adaptation techniques are organized into three main groups: content, presentation and navigation. In addition, each group is structured in methods and techniques. For instance, techniques relating to changing font style are grouped in the *FontStyle* method inside the presentation adaptations group (see Figure 31). For each technique the technique identification, name and an index to the implementation function are stored. Thus, there is direct matching between each technique stored in the ontology and the function which it implements in the system.



Figure 31. Adaptation techniques information in the ontology

In addition, each adaptation technique identifies the roles and/or properties of the interaction elements that it is related to. For example, techniques regarding navigation bar issues will identify the role navigation, whereas techniques regarding map display issues will identify the role *GeoMap*.

#### *Reasoning rules, constraints and stereotypes*

In order to infer adaptation techniques from the characteristics of a given user, a number of reasoning rules were included in the ontology. These rules can be classified into two groups: rules that infer adaptation techniques to apply to a given user and rules for defining constraints between techniques.

In fact, there can be inconsistencies when inferring adaptation techniques for a given user. This is the case when the user characteristics are a combination of different subgroup features. For example, let us consider the case of a user that experiences low vision and also limited movement. The adaptation techniques set for these types of users include the following two techniques: “2.3.2 Enlarge graphical buttons” and “2.7.2 Place content in web page to avoid scrolling”. The application of the first technique will enlarge the overall web page so that more scrolling will be needed as opposed to the previous adaptation technique oriented to avoid scrolling. In order to avoid these kind of inconsistencies a number of constraints were defined. In this particular case, applying

techniques such as stretchtext or hiding some optional information allows the web page elements to be enlarged while minimizing the scroll.

We decided to use stereotypes in order to overcome these situations. Using stereotypes, the inference rules were simplified enabling the selection of a predefined group of adaptation techniques (well-suited stereotype) instead of having to select the adaptation techniques one by one. Each stereotype is defined in the ontology as a set of adaptation techniques in which we previously detected and solved inconsistencies.

However, in our experiments [Pérez et al., 2014; Pérez et al., 2015; Valencia et al., 2015] we found an enormous variability of characteristics within a user group. For this reason, we decided to include the Assistive Technology used, the user Behaviour and user Preferences in the Knowledge Base in order assign the adaptation techniques to users more properly.

Behaviour includes the adaptive techniques that are beneficial to all users who use a particular assistive technology. Since users of the same assistive technology are also different, user preferences can be also defined in the Knowledge Base. Therefore, users can choose which adaptation techniques they want to apply. In addition to the matching technique other elements can be chosen. For instance, position of elements (up, down, etc.), colours, sizes, etc.

In addition to preferences or assistive technology, the “device” actually used was added to the ontology in order to be able to define the input/output methods or the device’s operative system. Along with these new classes and properties, matching new rules were included, in order to determine which adaptation techniques are applicable, depending on their features, screen size, if it is has a touch screen etc.

### 6.1.2. Design Considerations

Brajnik’s guidelines [Brajnik, 2004] were followed to design and develop the system. These guidelines provide the minimum requirements for a transcoding system.

**Efficiency:** Adapting web pages can slow down the interaction. Users must wait for the finalization of the adaptation process before accessing a web page. This process is not trivial because the Knowledge Base and the Adaptation Module have to collaborate. Nevertheless, our tests show that the delay in carrying out all the adaptations is about 300ms.

**Decoupling of servers:** the adaptation system is located outside the original web page server because the aim is to enable the system to adapt to any website. Currently, the Knowledge Base and the Adaptation Module are implemented in the same Web Service in order to minimize the overload. However, they can also work in different servers.

**Repurpose of content:** The original content is transformed with adaptation techniques, based on the annotations created to enable better access. In some cases, new content, with the purpose of providing assistance, could be added to the original web page, such as tutorials, alternative texts, etc.

**Appropriate output:** The current adaptation system is mainly based on stereotypes. However, in the future the objective is to produce personalized web interfaces based on each user’s specific characteristics.

**Customizability:** The annotator can specify which elements can be adapted by annotating them with the appropriate roles and properties. But in order to guarantee customizability, other requirements have to be satisfied, such as: decoupling of content, robustness, modularity, re-usability and content preservation. These issues are described in the following subsections.

- Decoupling of content: The system is able to store annotations in a different machine to the original web page server. When necessary, it refers to the original web page only through CSS elements, such as class or id and/or XPath
- Robustness: Since most website annotations are made using the CSS elements, it is possible to update the website and the system will still work correctly, unless specific class or id elements are changed
- Modularity: Different adaptation techniques can be applied to the same element. Techniques are applied one by one. The combination of a set of techniques produces the adapted version of a web page
- Reusability: Annotating class or id elements in CSS files allows the reuse of annotations because a number of annotated elements in a web page reappear across the whole web site
- Content preservation: The original web page is modified using its content. As stated before, the only new content added is for the purpose of providing help

**Adaptability:** Users can configure certain parameters such as the font size, or enable or disable some system functionalities such as specific buttons for navigation and scroll.

**Adaptivity:** The system was designed in order to allow user interaction data to be obtained by the Firefox Add-on. This feature is useful for updating user modelling and adaptation techniques to apply to specific users. Hence, adaptivity can be integrated in future versions.

**Interoperability:** The adaptation system architecture design maintains the usual operation of the adapted websites. The adaptation system can interoperate with other technology such as cookies, filling in forms, etc.

### 6.1.3. Discapnet Website Annotation Process

An HTML code and CSS style inspector (Firebug<sup>3</sup>) were used to analyse the source code of the website and to annotate the required elements with their roles and properties. The entire process was actually manual, which is quite tedious and difficult. This manual process led us to identify different areas in the web page that are outlined in Figure 33a. The areas identified and the role assignments are:

Area 1: this area is devoted to advertisements. The only annotation applied to elements in this area is to identify them as low priority elements; the priority property is set to 10. Therefore, if sorting adaptation techniques are applied these elements will be placed at the bottom of the web page.

Area 2: this area is identified as the banner of the website and is repeated in all pages in this web. We annotated it with the “banner” role. In this case, there is not a unique *div* for this element and, thus, the different elements composing the banner have to be identified (elements in the CSS files identified for this area are: classes: *logo* and *cabecera\_home* [*home heading*]; ids: *menu\_rapido* [*fast-menu*] and *registro* [*register*]). In addition, this area also includes two elements to annotate: the sitemap (2a) and the search functionality (2b). We annotated them with the “sitemap” and “search” roles respectively (elements in the CSS files identified for these roles are the second element of *menu\_rapido* [*fast menu*] which is expressed by XPath and id with value *buscador* [*finder*]).

Area 3: this area is composed of navigation elements. There are three main sections and subsections inside each element (see subareas 3a, 3b and 3c in Figure 33 a). Each

---

<sup>3</sup>Fire Bug, <http://getfirebug.com>

subarea is annotated with the “navigation” role (elements in the CSS files identified for this role are *divs: areas [areas]* (Figure 32), *comuni [communi(ty)]* and *actualidad [news]*). The property “labelledby” is used to point the title of each section (elements tagged as H2) and the properties “stretchable” and “dimnable” are used to facilitate the development of special navigation menus in adapted web pages.



Figure 32. Areas element annotation in protégé.

Area 4: the main content is placed in this area. It is annotated as “main” role. As can be seen in Figure 33a, this area is composed of two subareas, 4a and 4b. In this case, there is a class defined in the CSS files named *secciones [sections]*. However, this class also includes Area 5 (see Figure 33a). We assume that Area 5 is related to different content from 4a and 4b. Therefore, an XPath statement is necessary to identify 4a and 4b with the role “main”. These elements are also annotated with the “stretchable” and “dimnable” properties, pointing the H2 elements “*Hoy en Discapnet*” [*Today in Discapnet*] and “*Noticias y eventos*” [*News and events*] with the “labelledby” property.

Area 5: this area is composed of a set of images pointing to external locations and it is annotated with the “complementary” role. In this case, as stated previously for Area 4, an XPath statement is necessary to identify the third section element for the class *secciones [sections]* with the “complementary” role.

Area 6: this area contains some links, general information about the site such as, “*what is Discapnet*”, the last updated date, etc. We annotated it with the “contentinfo” role. The annotation is made for the CSS file element with *id* value equal to *pie[foot]*.

We have also annotated each chunk of information regarding the news. These are composed of a title (which is a link to the complete piece of information), several lines of text and a link (see Area 4b” in Figure 33a). This structure is annotated using the “article” role and it is defined in the CSS files with the class *noticiaPortada [newsFrontPage]*.

The annotation process described previously has been repeated for web pages in Discapnet having a different look and feel. Most elements were already annotated as the same types of elements are tagged with the same *div* and classes in the CSS files. However, some new elements were detected in this process and annotated correspondingly.





Figure 33. a) Detected areas for the Discapnet homepage in the annotation process.

b) Detected areas for the Bidasoa Turismo homepage in the annotation process

## 6.2. Case Study 1: Adapting the Web for People with Upper Body Motor Impairments Using a Touch Screen Mobile Device

A formal evaluation with three main objectives was conducted:

- To collect general characteristics of the users
- To test three new interaction techniques, augment, steady, end
- To measure the results of applying various adaptation techniques

With this purpose in mind the evaluation was divided into two parts: target acquisition tasks and navigation tasks. Target acquisition tasks were used to collect general knowledge about how participants select targets and evaluate the different interaction techniques. Navigation tasks, on the other hand, were used to compare the (adapted) user interface (generated by the application of the adaptation techniques) and the unadapted version (original).

### 6.2.1. Participants

Eight users with motor impairments in their upper limbs took part in the experiment. Half of them had tablets or smart phones, but only two use them regularly. Demographic data can be found in Table XVII.

A Firefox Web browser with the RemoTest add-on was installed to carry out the experimental session. Its aim was to present stimuli and to gather interaction data from participants for posterior analyses.

**Table XVII. Demographic data**

User	Age	Gender	Disability	Used Hand	Owens touch device?	Wheelchair
user1	41	Female	Cerebral Palsy	Right	No	Yes
user2	43	Female	Cerebral Palsy	Left	Yes (not frequently used)	Yes
user3	44	Male	Cerebral Palsy	Head pointer	No	Yes
user4	55	Male	Cerebral Palsy	Left	Yes	Yes
user5	55	Female	Glutaric Aciduria Type I	Left	No	Yes
user6	55	Male	Lack of Sensibility	Right	Yes	Yes
user7	47	Male	Cerebral Palsy	Right	No	Yes
user8	60	Female	Glutaric aciduria Type I	Left	Yes (not frequently used)	Yes

### 6.2.2. Positioning the Tablet

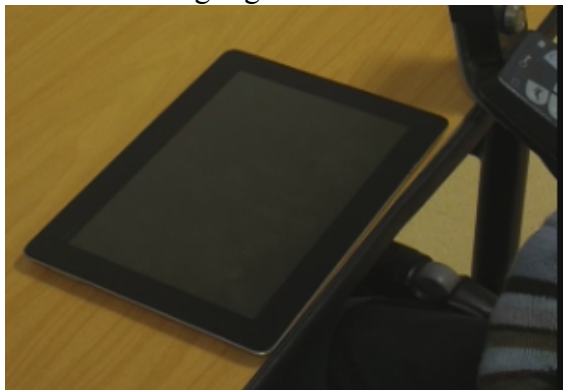
Different strategies were used in order to enable touch access to the tablet, depending on the needs of each participant. Two subjects used a tablet-mount to fix the tablet to their wheelchairs (see Figure 37). Four subjects placed the tablet on a table: one used a holder (Figure 36), two fixed it to the table with Velcro (Figure 34), and the remaining user did not use any additional help. The last two participants placed the device on a lectern (Figure 35).

### 6.2.3. Interacting with the Tablet.

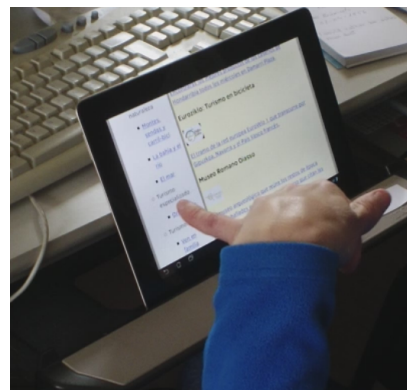
Three users needed specific adaptations to be able to interact with the touch screen. One used a glove with an attached touch pen. Another user used a glove with a finger cut out and the last one used a head wand with a touch pen.

### 6.2.4. Tasks and Materials

*Target acquisition tasks.* Target acquisition tasks were carried out to evaluate the implemented new interaction methods and to obtain details of how the users select targets. Three web pages were created, each one containing nine links to be selected, with the distribution shown in the Figure 38 (TA2). The target that had to be selected was highlighted with a yellow background. Once the highlighted link was selected, the next link was highlighted.



**Figure 34. Tablet on the table with Velcro**



**Figure 35. Tablet on a lectern**





Figure 36. Tablet on the table with a holder

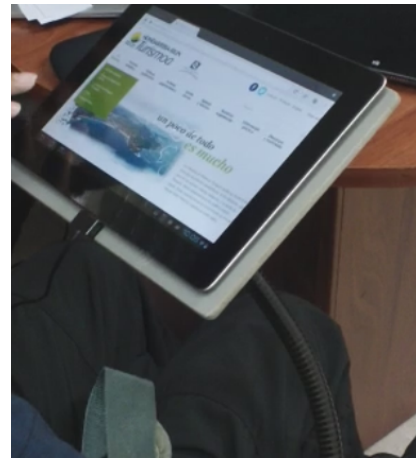


Figure 37. Tablet mounted on the wheelchair

In task TA1 the link to be selected had no other link around it. In task TA2 the link to be selected was surrounded by two links with a standard separation (Figure 38). TA3 is similar to TA2 but with a separation of 20px between links. Four interaction methods were tested: “standard”, “end tap”, “steady tap” and “augmented area”.

Do not click here Click here 1 Do not click here	Do not click here Click here 2 Do not click here	Do not click here Click here 3 Do not click here
Do not click here Click here 8 Do not click here	Do not click here Click here 9 Do not click here	Do not click here Click here 4 Do not click here
Do not click here Click here 7 Do not click here	Do not click here Click here 6 Do not click here	Do not click here Click here 5 Do not click here

Figure 38. Target acquisition task with target links surrounded by other selectable elements with standard separation (TA2)

*Navigation tasks.* . Users were asked to perform a set of navigation tasks in two websites: Bidasoa Turismo [2016] and Discapnet [2016]. All the tasks have two equivalent versions so each one could be assigned to each condition (adapted and original) indifferently, thereby avoiding learning effects.

The original “Bidasoa Turismo” website has a nine-category “toggle” menu with more than 50 selectable items. Figure 39 up. In the adapted version, all navigation items were displayed increasing the page size, Figure 39 down.



Figure 39. Bidasoa Turism original (up) and adapted (down) with the navigation menu highlighted

Therefore, in order to contrast the adequacy of the adapted version against the toggle menu, a number of item selections were set up. The selection of the tasks was based on the scrolling requirements to reach the target: NAV1 did not require any scrolling, NAV2 required some scrolling, and NAV3 was the task which required the most scrolling, since the target was located at the bottom of the web page.

In the original “Discapnet” webpage the size of the elements are, in general, quite small and, in addition, selectable elements are often surrounded by others. By contrast in the adapted version, elements were larger and had more space between them. This also increased the page size, requiring users to a larger scroll more. In task CA, Figure 40, users had to select several elements while in ACT, Figure 41, participants were required to tap on small elements that were close to selectable elements. In the Adapted version of the same pages these elements were larger and had more separation between them. Nevertheless, the page became larger in the adapted version and more scrolling was required.

On the other hand, SD1, SD2, SD3, SD4 and SB1, SB2, SB3, SB4 were search tasks that allowed participants to use the adapted and the original versions more naturally. All the search tasks in Discapnet (SD1, SD2, SD3 and SD4) were three clicks away from the homepage. In the Bidasoa Tourism website (SB1, SB2, SB3, SB4), they were two clicks away from the home page.

*Subjective measures.* In order to collect subjective measures from participants, a NASA TLX questionnaire [Hart and Staveland, 1988] was used. The objective of this questionnaire was to analyse both conditions in each questionnaire dimension.

There was also a short final interview, which was carried out to ascertain which condition was their favourite and which one was more comfortable for reading and

selecting elements. In addition they were also asked whether they preferred the toggle menu or the open menu.

**Noticias**

Actualmente existen **217** noticias.

- ▶ [La pandilla de 'On Fologüers' protagoniza las campanadas de Nochevieja](#)
- ▶ [Las personas sordas podrán contactar con Iberdrola sin barreras de comunicación](#)
- ▶ [ILUNION certificó la accesibilidad de 23 sitios web y aplicaciones móviles durante 2016](#)
- ▶ [Mario Garcés preside el Patronato del Centro Nacional de Tecnologías de la Accesibilidad](#)
- ▶ [Abierta la inscripción para las XXVII Jornadas de la Escuela de Fisioterapia de la ONCE](#)
- ▶ [Más de 900 personas con discapacidad participan en el proyecto Webquest de COCEMFE y Fundación Vodafone España](#)
- ▶ [Plena inclusión reforzará acciones para reivindicar el derecho al voto de las personas con discapacidad intelectual](#)
- ▶ [Cada año más pacientes con alguna ER se benefician del Servicio de Atención Psicológica \(SAP\) de FEDER](#)
- ▶ [Un doctorado con discapacidad de la Universidad de Granada recibe una de las becas de Fundación Universia y Cermi](#)
- ▶ [El Tribunal Constitucional niega el voto a las personas con discapacidad](#)

página 1 de 22

Página 1 2 3 4 5 **6** 7 8 9 10 siguiente ➔

Figure 40. CA task with the target link highlighted

*¿Quieres darle vida a tu idea?* 2ª Edición Madrid 2016-2017

**discapnet**

Portada Discapnet | Actualidad | Discapacidad en los medios

**Actualidad**

- ▶ Noticias sobre Discapacidad
- ▶ El sector social, al día
- ▶ Actualidad general
- ▶ Noticias de fácil lectura
- ▶ Hemeroteca
- ▶ Solidaridad Digital **1**
- ▶ **Discapacidad en los medios**
- ▶ Agenda
- ▶ Boletines
- ▶ Equipo de redacción Actualidad

**Discapacidad en los medios**

Una selección de enlaces para conectar con especialistas en discapacidad. Sin olvidar los diarios y revistas de información.

**2 ▶ Radio**

- ▶ **Televisión**
- ▶ Prensa
- ▶ Publicaciones Fundación ONCE
- ▶ Publicaciones de ONCE

Lugar: **MADRID**  
Inscripción abierta hasta el 15 de septiembre  
Inicio del programa: 3 de octubre

Fundación **prevent** Con la colaboración de **ESADE** ESADÉ ENTREPRENEURSHIP INSTITUTE

**discapnet**

Portada Discapnet | Actualidad | Discapacidad en los medios | Enlaces a medios | Otros medios | TV

**Actualidad**

- ▶ Noticias sobre Discapacidad
- ▶ El sector social, al día
- ▶ Actualidad general
- ▶ Noticias de fácil lectura
- ▶ Hemeroteca
- ▶ Solidaridad Digital
- ▶ Discapacidad en los medios
- ▶ Agenda
- ▶ Boletines
- ▶ Equipo de redacción Actualidad

**Televisión**

En esta sección ofreceremos información sobre canales de tv especializados en discapacidad.

**3 ▶ Canales de televisión en España**

- ▶ **Programas de TV especializados en discapacidad**

**Destacados**

Programas de televisión con intérprete en Lengua de signos.

Figure 41. ACT task with the three links highlighted

### 6.2.5. Procedure

In the first part of the study, participants had a training session beforehand to get familiarized with the system and the task types. During the training session, participants

were asked whether or not they wanted specific buttons to scroll the page, in order to avoid the “swipe” gesture.

Each participant carried out nine tasks under each condition, adapted and original, in two websites Discapnet (4) and Bidasoa Turismo (5). Conditions were counterbalanced between users. For each equivalent task (ACT and ACT”, CA and CA”, etc.) one task was assigned randomly to one condition and the other task was assigned to the remaining condition. The order of tasks was randomly assigned but the search tasks were performed first. Once they completed each condition, participants rated it with the NASA TLX questionnaire.

Subsequently, three target acquisition tasks were carried out under four conditions: “standard”, “end tap”, “steady tap” and “augmented tap”. The conditions were counterbalanced between users and the task order was randomly selected for each user.

After the session, a semi-structured interview was conducted in order to gather more information about the users’ thoughts concerning the adapted or unassisted versions.

### 6.2.6. Results

First, general characteristics of the experiment participants, collected from the target acquisition tasks with the standard method of interaction are presented. Afterwards, the results obtained with different interaction techniques (“standard”, “end tap”, “steady tap” and “augment area”) are discussed. Finally, the adapted version is analysed and compared with the original, with regard to times, searches, preferences, and the results of the NASA-TLX questionnaire.

*Target acquisition tasks.* User characteristics. With the data obtained from the use of the standard method of interaction, we collected average distances from the centre of the target to the points where the finger touched the screen (TD) and left the screen (TU). The distance (D) travelled by the finger while selecting was also measured and its relation to the optimal distance (CI). In addition, the number of times (NF) that each user touched the screen with more than one finger was counted.

Table XVIII shows the results gathered from the standard method of interaction. Most users did not move their fingers substantially during the selection, except User8 who moved his finger 10.8 mm on average. Other users, such as User2, User5 and User7, also moved their fingers (between 3.65 mm and 4.9 mm) but notably less than User8.

User8 also had the largest CI, 2.07, followed by User7, User2 and User5. This indicates that their fingers travelled longer than optimal distances, indicating that they had problems of precision or control.

On the other hand, User1, User2 and User3 located their fingers quite far from the centre of the object (about 30mm), indicating difficulties to make the right selection.

Regarding the number of times they touched the screen with more than one finger, it is worth noting that User2 did this eleven times. Evidently, touching the screen with more than one finger hindered the user from making an accurate target selection.

The three different scenarios discussed above were considered to assess the four different methods of interaction. Since data was not normal, it was analysed with the nonparametric Friedman test. In all scenarios the null hypothesis H0 was “there are no differences between alternative methods and the standard one”.

Table XVIII. Users' characteristics from target acquisition tasks with standard interaction

	TD	TU	D	CI	CTU
User1	32.14	32.91	0.62	1.00	0
User2	29.21	28.56	4.90	1.50	11
User3	31.39	31.08	0.48	1.00	0
User4	3.51	3.51	0	1	0
User5	13.95	12.84	3.76	1.27	0
User6	5.28	5.28	0	1	0
User7	15.36	15.43	3.65	1.88	0
User8	18.54	19.02	10.80	2.07	0

*Target selection with nothing around (TAI)*. Friedman's test found no significant differences between the methods of interaction,  $\chi^2(3) = 1.819$ ,  $p = 0.610$ .

Although the differences were not significant, the “augmented” method achieved, on average, the lowest value (3516.78 ms), followed by “end” (3783.77 ms), standard (4329.38 ms), and finally “steady” (5055.23 ms). The data are more compact for “augmented”, as can be seen in Figure 42.

For all the users, except User6, at least one of the alternative methods of interaction produced a better average value. In some cases (User7, User4, User3, User1, User5) the difference was small, less than one second. Other users obtained higher differences on average: User2 obtained 3 seconds, and User8 7 seconds.

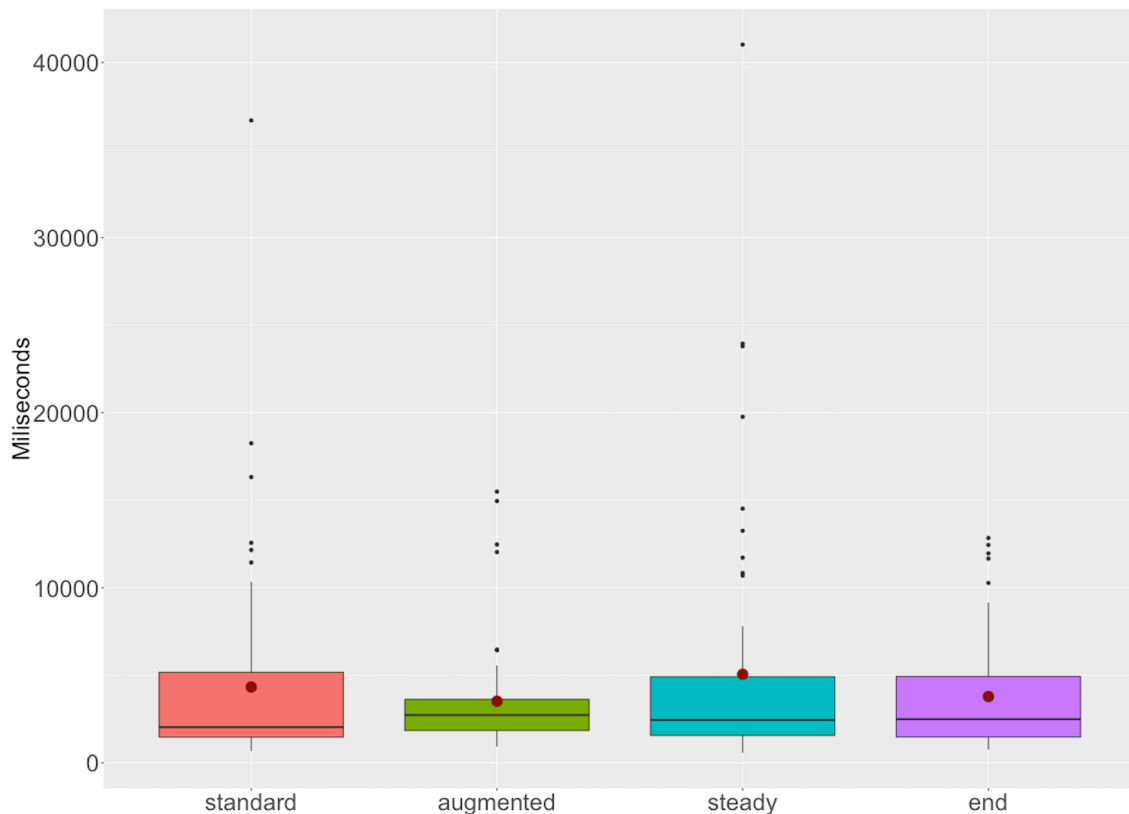
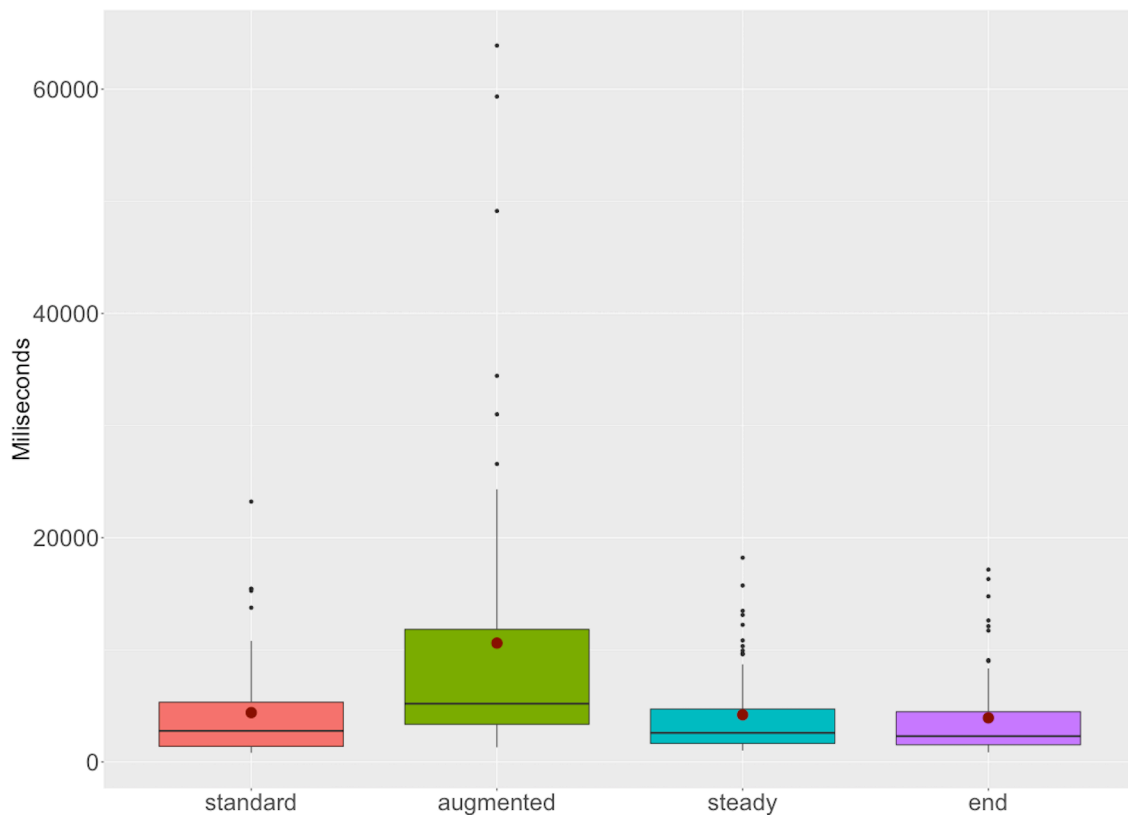


Figure 42. User selection time boxplots, by interaction method (links not surrounded by other selectable items)

*Target Selection with 2 surrounding links with standard separation (TA2).* In this scenario significant differences were found:  $\chi^2(3) = 41,732$ ,  $p < 0.01$ . A Mann-Whitney test with Bonferroni correction post-hoc test, showed differences between standard and “augmented” ( $p < 0.01$ ), but not with others (standard-end and standard-steady,  $p = 1$ ).

The “augmented” method, which in the previous case produced good results, is the worst this time. Having to frequently disambiguate between probable targets, it significantly increases the time required to select an item. In general, “end” (3930.08 ms) produced the best value on average, followed by “steady” (4202.85 ms), “standard” (4297.57 ms) and “augmented” (10597.92ms) as shown in Figure 43.

User1, User4, User5 and User6 obtained better values with the “standard” interaction method. “End” is similar to “standard” for User3, User4 and User6 ( $< 300$  ms). The subject User2 obtained an average difference of 6 seconds with “steady” and 3 seconds with the “end”. User7 obtained a difference of about 2 seconds with “end” and “steady”. Finally, User8 obtained a difference of almost 5 seconds with “steady” and nearly 3 seconds with “end”.

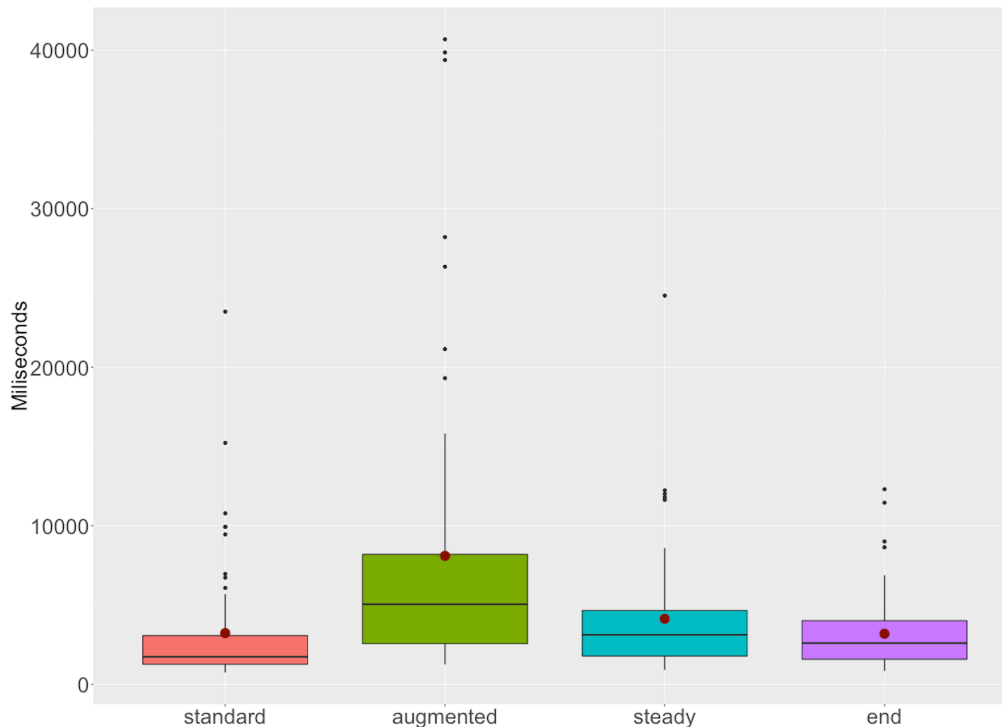


**Figure 43.** User selection time boxplots, by interaction method (links surrounded by other selectable items)

*Target Selection with 2 surrounding links and 20px separation (TA3).* In this case the Friedman test found significant differences:  $\chi^2(3) = 48,337$ ,  $p < 0.01$ . The Mann-Whitney post-hoc analysis with Bonferroni correction showed differences between “standard”-“augmented” ( $p < 0.05$ ), and “standard”-“steady” ( $p < 0.05$ ). In both cases, the differences were in favour of the standard interaction method. Overall “end” produced better average results (3192.83 ms) closely followed by “standard” (3232.52

ms). Slightly further behind were “steady” (4135.98 ms) and “augmented” (8105.34 ms) as can be seen in Figure 44.

On average, “standard” was better for User1, User3, User4, User7 and User8. However, “end” produced similar values to the “standard”, except for User8. User6 obtained a minimal difference with “end”. User5 reduced the time by 1 second with “steady” while User2 reduced the time by almost half with “end”.



**Figure 44.** User selection time boxplots, by interaction method (links surrounded with 20px separation)

*Was the interaction method helpful?* To find out whether, in any of the cases, the alternative methods were of any help, the users who were more likely to benefit were analysed. These were the users with the highest CI (User2, User7 and User8).

First, User2 was analysed, for whom significant differences were found in TA2:  $\chi^2(3) = 14.6$   $p < 0.01$ , although the post-hoc test could not clarify which pairs were implied (“standard”-“augment” 1, “standard”-“steady” 0.50, “standard”-“end” 1). Although for TA1 no significance was obtained it was quite close:  $\chi^2(3) = 6.6$   $p = 0.086$ . The same result was obtained with the post-hoc test.

For User7, no differences were found in TA1 and TA3 but there were differences in TA2:  $\chi^2(3) = 13.133$   $p < 0.01$  and in the post-hoc values: “standard”-“augmented” 1, “standard”-“steady” 0.291, “standard”-“end” 0.085. Although not significantly, “end” is very near to differentiating (0.085). Therefore, it seems likely that this user could benefit from “end” when the selected target is surrounded by other targets.

Finally, User8 obtained significant results in all three cases:  $\chi^2(3) = 13.93$   $p < 0.01$ ,  $\chi^2(3) = 17.4$   $p < 0.01$  and  $\chi^2(3) = 13.4$   $p < 0.01$ . The post-hoc test could not find differences between the pairs in TA1 (“standard”-“augmented”  $p = 0.767$ , “standard”-“steady”  $p = 0.291$ , and “standard”-“end”  $p = 1$ ). In TA2 the results were better for “steady” (“standard”-“augmented”  $p = 1$ , “standard”-“steady”  $p = 0.0488$ , “standard”-



“end”  $p = 0.3249$ ). In TA3 differences were found only for “augment”, worsening the time needed (“standard”-“augmented”  $p = 0.012$ , “standard”-“steady”  $p = 1$ , “standard”-“end”  $p = 0.182$ ).

#### *Design issues.*

*Buttons for scrolling.* During the training time, participants were asked whether or not they wanted to use the scroll buttons to avoid making the “slide” gesture. Three users decided to perform direct slide gestures without assistance, while another three preferred to use big buttons as an alternative to the slide gesture. For the remaining two participants (User1 and User3) this was essential because their own characteristics prevented them from making the “slide” gesture. For this reason, the intervention of a researcher was necessary to perform the scroll in the sessions with the original version.

*Enlarge elements and increase scroll.* Tasks ACT, CA, NAV1, NAV2 and NAV3, were analysed to know how much “enlarging the size of elements and therefore the site” benefits or harms the users. Table XIX shows the data obtained. These data were analysed using the Student t test for paired groups, since the distribution of the data was normal except for CA and NAV1 tasks. In such cases, results were transformed logarithmically to have a normal distribution. In all cases the null or H0 hypothesis was “the adapted and original versions produce similar results”.

**Table XIX. Time needed to complete task in both conditions (Adapted and Original)**

	Adapted				
	ACT	CA	NAV1	NAV2	NAV3
User1	141.222	47.035	13.829	63.872	77.931
User2	134.318	48.752	57.621	89.740	59.231
User3	58.409	53.252	22.828	42.022	72.244
User4	70.936	28.499	34.404	23.711	34.457
User5	110.420	36.945	22.834	74.991	82.144
User6	57.693	20.041	11.857	22.522	24.780
User7	92.537	35.906	31.940	62.048	46.654
User8	120.543	27.661	12.079	43.322	62.944
	Original				
	ACT	CA	NAV1	NAV2	NAV3
User1	113.285	29.644	87.696	72.818	89.030
User2	179.573	74.760	24.907	70.850	29.875
User3	76.017	128.293	39.488	43.429	23.940
User4	60.255	38.332	33.866	39.731	23.435
User5	131.105	69.609	37.552	63.170	29.981
User6	68.955	18.448	14.961	15.871	13.583
User7	66.066	56.716	50.999	21.533	15.268
User8	43.735	80.440	25.950	37.816	44.810

In the ACT task (several selections of elements) no differences were found:  $t(7) = 0.437$ ,  $p = 0.675$ . On the contrary, the CA task (selecting a very small element with scroll) is very close to having significant differences between the adapted one and the



original:  $t(7) = -2.2944$ ,  $p = 0.055$ . In this case the adapted version is better than the original -24.77 seconds on average, with a confidence interval from -0.11 to -49.42 seconds and an effect size of 0.81.

On the other hand, in the tasks implying comparison of the navigation menu (“toggle menu”) with the open menu, no significant differences were found in the case with little scroll NAV1:  $t(7) = -1.641$ ,  $p = 0.145$  and an effect size of 0.58. This supposes an average enhancement of 13.5 seconds for the adapted version over the original.

In the case with medium scroll demands, NAV2, no differences were found:  $t(7) = 1.151$ ,  $p = 0.287$  and effect size of 0.40. The original was 7.13 sec faster than the adapted version.

Finally, for the case with more scroll demands, NAV3, the adapted webpage, showed significant differences:  $t(7) = 3.22$ ,  $p < 0.05$  and an effect size of 1.14. The original was on average 23.81 seconds faster with a confidence interval ranging from 41.30 to 63.13 seconds.

*Search tasks.* In the original condition participants were able to find more targets, as can be seen in Table XX. Only one user (User4) finished more tasks in the adapted version. User2 did not find any and User6 found two in each version. The rest found more targets in the original, the most prominent difference being for User7 who found three in the original and none in the adapted version.

Table XX. Number of tasks completed in search tasks in both contitions

	Adapted	Original
User1	2	3
User2	0	0
User3	0	1
User4	2	1
User5	0	1
User6	2	2
User7	0	3
User8	1	3

*NASA TLX.* Each of the dimensions of the NASA TLX questionnaire was analysed with the Wilcoxon test. The resulting dimension distribution can be seen in Figure 45.

Unexpectedly, no significant differences were found for Mental Demand, even if in the adapted version both sites (discapnet and bidasoatourismo) had the same structure ( $W = 9.5$ ,  $Z = -0.428$ ,  $p = 0.688$ ). Neither were any differences found for the Effort dimension ( $W = 0$ ,  $Z = -1.720$ ,  $p = 0.25$ ). By contrast, significant differences were found in Physical Demand ( $W = 0$ ,  $Z = -2.40$ ,  $p < 0.05$ , effect size = 0.849) in favour of the adapted version. Regarding Temporary Demand ( $W = 6.5$ ,  $Z = 0.835$ ,  $p = 0.625$ ) and Performance ( $W = 22$ ,  $Z = 0.566$ ,  $p = 0.656$ ), users tended to value the original page higher, although not significantly. Finally, Frustration Level ( $W = 12.4$ ,  $Z = 0.567$ ,  $p = 0.625$ ) was very similar for both types of pages.

*Interview.* All but two users preferred the adapted version to the original one. To the question “Which one was more comfortable for reading and selecting links”, all but two selected the adapted version. Users were also questioned about their preferences

between open or toggle menu. Five preferred the “toggle” menu and three the open menu.

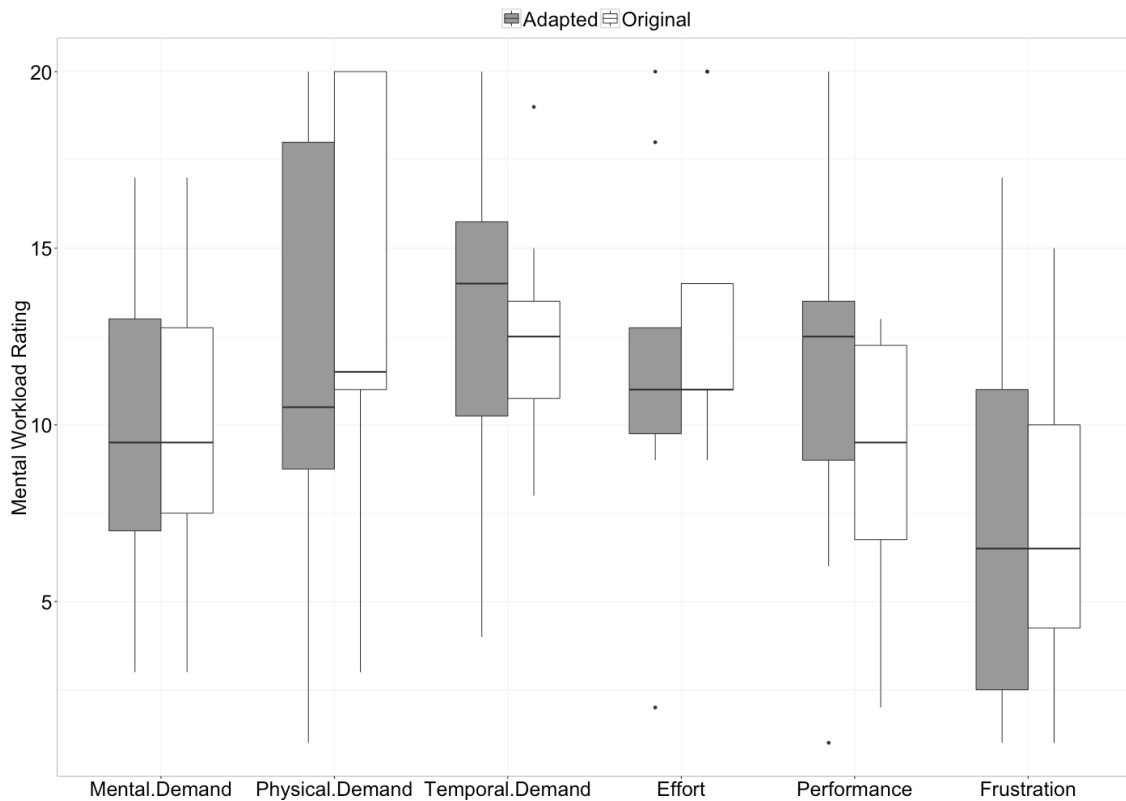


Figure 45. Boxplot for each NASA TLX questionnaire dimension

### 6.2.7. Discussion

In target acquisition tasks, interestingly most users needed, on average, more time for the task with only a link (TA1) than for the other tasks, using the “standard” interaction. This may be because when the links are positioned at the edges of the screen it can lead to unintentionally pressing or interacting with buttons on the navigation bar, back button, the watch, etc. On the other hand, in TA3 the best results were obtained, in contrast to TA2 and TA1, by the different methods of interaction except “steady”. This highlights the importance of active elements maintaining a minimum distance between each other.

The technique “augmented” applied to targets that were not surrounded, appears to provide some help to many of the users, though not significantly. For other cases, having to disambiguate increases the target selection time significantly, making it unsuitable for interaction elements which are surrounded.

It appears that some users can gain an advantage from alternative methods of interaction under certain conditions. Examples of this are the subjects User2 and User8, in the cases TA1 and TA2, with the steady method, or User7, with “end”, in TA2. In order to help other users, such as User1 and User3, it would be necessary to detect their pattern to select elements in order to preview the objective they want to click [Montague, Hanson and Cobley, 2012; Mott et al., 2016]. Remember that, even if they do not drag their fingers for selection, they set them down slightly away from the target (> 3 cm).

Regarding design issues, it seems clear that having to perform a lot of scrolling increased the time required to perform the task (NAV3). But the larger size of the links and the space between elements added in the adapted page might be helpful (NAV1). Not having significant differences in NAV2 is quite good, because performing the task in the adapted page entailed some scrolling. On the other hand, when the links are very small and they are surrounded (CA) better results are obtained with the adapted page, despite having to do more scrolling.

However, it should be pointed out that faster is not always better (a vision focused on productivity or business). Users do not always prefer to be faster, as they reported for our study: six out eight preferred the adapted page. Comfort, reduction in the number of mistakes or easier item selection can be the factors that lead to a preference for the adapted pages, even though task performance times may be longer.

The results in the search tasks can be explained because participants were probably already used to the structure of the sites. Another issue could be the use of the breadcrumbs as a method for identifying the page or section in the adapted page. Most of the users who took part in the study do not usually use –and are unfamiliar with– breadcrumbs. Therefore, a possible improvement would be the use of colours for the sections. Moreover, the section in which the users find themselves should be more evident (and not based on breadcrumbs).

Considering the NASA TLX questionnaire, adapted pages generated less physical demand, which is very important for people with motor disabilities. Results (although not significant) of the adapted pages in performance, frustration or time demand seem to be explained by the probable relation with the number of completed tasks in each condition (adapted and original).

Finally, from the results of the users' feedback regarding the menus, it would seem important to provide a customization option for the type of menu ("open" or "toggle"). Although the use of these types of menus for people with physical disabilities is not recommended, five users preferred it.

### ***6.3. Case Study 2: Adapting the Web for People with Low Vision***

Model Access was conceived as a tool to enable better access to the World Wide Web from different devices for people with different needs. In the previous study the tool was used by people with motor impairments by means of a tablet. In this case, users with low vision use the system from a PC and their preferred assistive technology.

The main objective of the study was to fine-tune the adaptation techniques to automatically produce personalized web interfaces for people with low vision. The Model Access system was arranged in order to generate user-tailored web interfaces by applying up to 20 personalization techniques for users with low vision, these are explained in Chapter 5.

#### ***6.3.1. Participants***

Nine people (4 females and 5 males) with low vision took part in this second study. All of them claimed to have a high level of Web expertise. Previous navigation experience in Discapnet varied among participants. Four participants used their own laptop (P5, P7, P8 and P9) whereas the rest used a laptop specifically configured based for the needs of each participant. Table XXI presents the information about the participants.

### 6.3.2. Stimuli

The stimuli presented to the participants come from the Discapnet website. There were two different stimuli based on Discapnet website: the adapted version with Model Access and the original or no-adapted version.

**Table XXI. Information of the participants with low vision**

User	Age	Discapnet Expertise	Assistive Technology	Screen Resolution
P1	39	High	Browser zoom	1600x900
P2	50	None	O.S. 125% magnification +high contrast cursor	1600x900
P3	37	Medium	O.S. 125% magnification	1600x900
P4	37	None	Mac O.S. Screen Zoom	1440x900
P5	46	Low	ZoomText10	800x600
P6	55	Low	Browser zoom	1366x768
P7	43	None	Browser zoom	1067x600
P8	45	Medium	Mac O.S. Screen Zoom	1440x900
P9	23	None	Mac O.S. Screen Zoom	2880x1800

### 6.3.3. Procedure

First, participants were informed about the study and were asked to sign a consent form. Each participant was asked to complete search tasks in both scenarios: the original version (Scenario 1) and the adapted version (Scenario 2). The total number of tasks was twelve, six in each scenario. Once the participants were informed and had given their consent, he or she was presented one scenario in which to complete the six search tasks. Then, the other scenario was evaluated. The order of the evaluation of the scenarios was counterbalanced.

The task assignment to each scenario was randomized as well as the order in which the six tasks were presented to participants. The time limit for each task was four minutes, with the maximum length of the experimental session being around an hour. A message was displayed when participants found the target or the time expired. Then, the next task was proposed and the homepage of the corresponding scenario was presented. After finishing the tasks assigned in both scenarios, participants were interviewed to gather information about their navigation experience in both versions of Discapnet.

### 6.3.4. Materials & Equipment

The efficiency, effectiveness and satisfaction of participants in both scenarios were measured using diverse parameters. The task completion time and Lostness formula were computed for efficiency. The task completion rate in each scenario was considered for effectiveness. Qualitative values gathered in the interviews were considered for satisfaction. In addition, participants' navigation strategies were also analysed.

Twelve search tasks were designed with the target located at different depths (number of links from homepage). In this sense, the target of the tasks was one, two or

three links from the homepage. Four different search tasks for each depth were defined. The study was designed for presenting two tasks of each depth in one scenario and the other two in the other scenario (a total number of six tasks per scenario).

The experiment was performed using two different tools: Model Access and RemoTest. Both tools were developed as Firefox add-ons. Therefore, participants were asked to use Firefox, and both add-ons were installed before starting the experimental sessions.

The automated adaptation system presents the stimuli to the participants. In Scenario 2, Model Access automatically applies the personalization techniques selected. RemoTest guides the entire session, presenting the information of different tasks to participants, managing the time for completing tasks, counterbalancing the order of scenarios, randomizing the tasks assigned to each scenario, etc.

### 6.3.5. Results

The results obtained from the data gathered by the RemoTest tool are shown in Table XXII. The data are ordered by scenario. Participants P1, P3, P5, P6 and P9 performed search tasks in Scenario 1 first. Participants P2, P4, P7 and P8 performed search tasks in Scenario 2 first.

Regarding the efficiency of participants, the average task completion time (TCT-AV) as well as the average Lostness value (LNS-AV) were computed for the completed tasks in both scenarios. Six participants (P1, P3, P4, P5, P6 and P8) improved the TCT-AVG value in Scenario 2, P4 and P8 performed the tasks in Scenario 2 first. P4 improved by 62 seconds (88 seconds in Scenario 1 and 26 seconds in Scenario 2). The average value of time improvement is 31.6 seconds for these six participants whereas the average value of time increment in Scenario 2 for the other three participants (P2, P7 and P9) is 28.6 seconds.

The maximum average Lostness (LNS-AV) value is 0.26 in Scenario 1 (P4) and 0.24 in Scenario 2 (P7). Comparing the values obtained by P7 in both scenarios, there is a low improvement in Scenario 1 (only 0.05 less than in Scenario 2). Participants with better TCT-AV values in Scenario 2 also improved their LNS-AV value for Scenario 2. The improvement of participant P4 merits special consideration (0.26 in Scenario 1 and 0.09 in Scenario 2). Participant P5 has to be also considered, as the obtained LNS-AV values are 0.21 in Scenario 1 and 0.09 in Scenario 2.

Regarding the effectiveness of participants, 39 tasks were completed in Scenario 1 (an average number of 4.33 tasks per participant) while 41 tasks were completed in Scenario 2 (an average number of 4.55 tasks per participant). Note that the total number of tasks was 54 in each scenario. The minimum number of completed tasks by participants is 2 in Scenario 1 (P5 and P9) and 3 in Scenario 2 (P3). There are two participants who completed all tasks in both scenarios (P2 and P3 in Scenario 1 and P1 and P6 in Scenario 2). Four participants (P1, P5, P6 and P9) improved the task completion rate in Scenario 2. Three participants (P4, P7 and P8) have a similar task completion rate in both scenarios. With regard to the depth of tasks, more tasks are completed in depth 1 and depth 2 task groups in Scenario 2 (the number of completed tasks per depth in this scenario are: all the depth 1 tasks, 14 tasks of depth 2 and 9 tasks of depth 3).

The data presented in Table XXIII is related to the navigation strategies applied by participants during the experimental sessions. The total distance (in pixels) the cursor went through on the web pages during the session (TPT) was computed by the experiment management tool. This data was not correctly gathered for two participants

(P8 and P9) due to technical problems. However, we included it for the other seven participants as we found significant differences between the two scenarios.

**Table XXII. Results achieved for tasks 1 to 12 during both scenarios (NTC-number of tasks completed, ATT-average time per task, AVP-average number of visited pages, TTS-total time per scenario, TVP- total number of visited pages, ALN-average Lostness, ORD**

Scenario 1 – Original Website												
T	NTC			AV	TCT (sec.)			AV	LNS			
	1	2	3		1	2	3		1	2	3	
P1	5	2	2	1	72	7	30	179	0.19	0	0.13	0.45
P2	6	2	2	2	33	4	26	70	0.09	0	0	0.28
P3	6	2	2	2	68	6	99	100	0.03	0	0	0.10
P4	4	2	1	1	88	191	11	62	0.26	0.79	0	0
P5	2	1	0	1	97	6	-	188	0.21	0	-	0.42
P6	5	2	1	2	51	9	30	115	0.06	0	0	0.17
P7	4	2	1	1	61	16	14	153	0.19	0	0	0.57
P8	5	2	2	1	56	12	99	58	0.12	0	0.37	0
P9	2	2	0	0	21	21	-	-	0	0	-	-
Scenario 2 – Adapted Website												
P1	6	2	2	2	55	11	103	50	0.16	0	0.33	0.17
P2	4	2	1	1	45	11	18	104	0.11	0	0	0.33
P3	3	2	1	0	15	13	17	-	0	0	0	-
P4	4	2	1	1	26	12	17	48	0.09	0	0	0.26
P5	5	2	2	1	66	12	22	164	0.09	0	0	0.26
P6	6	2	2	2	30	15	29	46	0.03	0	0	0.10
P7	4	2	2	0	78	14	142	-	0.24	0.17	0.32	-
P8	5	2	2	1	50	11	66	74	0	0	0	0
P9	4	2	1	1	78	26	42	165	0.14	0	0	0.43

Four of seven participants (P4, P5, P6, P7) obtained a lower value for this parameter. The gathered values for P4 and P5 participants merit attention as P4 reduces this value by 55,263 pixels (102,752 pixels in Scenario 1 and 47,489 pixels in Scenario 2) and P5 brings down this value by 147,535 pixels (184,243 pixels in Scenario 1 and 36,708 pixels in Scenario 2).

The total number of visited pages per scenario (TVP) significantly differs among participants with a minimum value of 16 pages in Scenario 1 for participant P9 to a maximum value of 65 pages in Scenario 1 for participant P4. The total number of visited pages in Scenario 1 is 299 and 295 in Scenario 2.

The total number of zoom increment/decrement actions was 152 in Scenario 1 and 82 in Scenario 2 (46.05% less). Considering the type of action in each scenario, 78 is the total number of magnification actions in Scenario 1 whereas 45 is the total number in Scenario 2 (42.3% less). The total number of zoom level decrement actions is 74 in Scenario 1 and 37 in Scenario 2 (50% less). Looking at the values gathered for each participant, all of them needed less zoom changing actions in the adapted version except for P8 (zero zoom increment/decrement actions in Scenario 1 and 5 zoom

increment/decrement actions in Scenario 2). The reason is that this participant utilizes ZoomText as assistive technology. However, he was asked to perform tasks in Scenario 2 first and required minimal support from ZoomText. The zoom actions collected are related to activation/deactivation of ZoomText software. In contrast, this participant used ZoomText during all the interaction with the original Discapnet version (Scenario 1) and kept the zoom to the same level.

**Table XXIII. Calculated results for tasks 1 to 12 during both scenarios (TPT-total pixels travelled with cursor, NZM-total number of zoom magnify operations, NZD-total number of zoom demagnify operations, NZO-total number of zoom overall operation, NPV-number of visits to task objective previous page and destination link visible but not clicked, NPH-number of visits to task objective previous page and destination link hidden, not on screen).**

Scenario 1 – Original Website							
User	TPT (pixels)	NZM	NZD	NZO	NPV	NPH	TVP
P1	20388	1	0	1	0	0	33
P2	12039	0	0	0	0	0	24
P3	7728	0	0	0	0	0	19
P4	102752	10	12	22	2	3	65
P5	184243	1	0	1	1	3	45
P6	29348	3	2	5	1	0	23
P7	64304	1	0	1	1	0	44
P8	-	0	0	0	0	1	30
P9	-	62	60	122	2	0	16
Scenario 2 – Adapted Website							
P1	26262	0	0	0	0	2	27
P2	20617	0	0	0	0	1	36
P3	36365	0	0	0	1	2	41
P4	47489	5	4	9	0	3	52
P5	36708	0	1	1	1	0	26
P6	18207	0	1	1	0	0	19
P7	21785	0	1	1	0	0	51
P8	-	3	2	5	0	0	22
P9	-	37	28	65	0	0	21

The number of times that participants accessed the webpage containing the target of the task were also analysed in order to obtain information. NPV column in Table XXIII corresponds to the number of visits to the web page containing the target link visible on screen when said target was not perceived by the participant. This situation was experienced by P4, P5, P6, P7 and P9 in Scenario 1 (the total number of such experiences was 7). In Scenario 2, only two participants (P3 and P5) experienced it once during the session. NPH column corresponds to the number of visits to the web page containing the target link although it did not appear on the screen (vertical scrolling was required to reach the target). This was experienced a total number of 7 times in Scenario 1 (3 times by P4 and P5 and once by P8) and 8 times in Scenario 2 (3 times by P4, 2



times by P1 and P3 and once by P2).

#### 6.3.6. Discussion

The results obtained in this study revealed that participants P1, P3, P4, P5, P6 and P8 improved their efficiency when interacting with the adapted version of the Discapnet website. This group of participants is composed of almost participants using specific screen magnification software (P4 and P8 used Mac O.S. Screen Zoom and P5 used ZoomText). Regarding effectiveness, only two participants reduced the task completion rate in Scenario 2 (P2 and P3). Participants P5 and P9 are the ones with more improvement (50% or more). Both participants utilized magnification specific software (P9 used Mac O.S. Screen Zoom).

The analysis of the navigation behaviour revealed that participants P4 and P5 significantly reduced the total distance the cursor went through on the web pages during the session in Scenario 2. We observed that they had to constantly move the cursor in Scenario 1 in order to place the software window on the part of the web page they wanted to explore. These cursor movements are significantly reduced in Scenario 2 due to the adapted narrow-page presentation. This leads us to think that these participants may be more comfortable in their interaction with the adapted version. In this sense, the adaptation techniques “Page linearization” and “Present important information as close to the centre of the screen as possible” seem to be adequate for this group of users. The users can focus their sight to the centre of the screen and require fewer cursor movements.

This group of users also gain benefits from the adaptation technique “Place important areas of content near the top of the page”. Scenario 2 required fewer zoom level changing actions. We think that locating all the navigational elements in a unique vertical navigation bar at the top of the page has had an influence. In Scenario 2, users utilizing specific magnification software can come back to it by directly moving the software window to the top of the page.

In contrast, the narrow-page style seems to confuse users utilizing browser zoom or operating system utilities. The significant increase in vertical scrolling due to the page linearization may be inconvenient for them. The adaptation techniques related to providing internal links and skip links have to be reviewed as they could improve the navigation when vertical scroll is otherwise required.

Although the personalization techniques selected for the adapted version of Discapnet website were initially thought to be for people with low vision in general, the results suggest us that this group of users is heterogeneous and the assistive technology used must be carefully considered.

#### 6.3.7. Feedback from Participants

Participants were briefly interviewed after completing all the tasks in both scenarios. We wanted to know about their interaction experience in each scenario and their preferences. Five participants preferred the adapted version of Discapnet website (P4, P6, P7, P8 and P9). All specific magnification software users preferred it except P5, even though the data analysis results indicated that his efficiency and effectiveness improved with the adapted version. On the other hand, the efficiency and effectiveness of participant P7 was better in Scenario 1 but he surprisingly preferred the adapted version. The only value which improved for this participant in Scenario 2 was the cursor



distance. This suggests that this is an indicator of the ease of use for people with low vision and should be considered when building personalized web interfaces. Participant P1 preferred the original version of Discapnet. The efficiency and effectiveness values obtained by this participant improved in Scenario 2. However, this participant confirmed having a high level of expertise in the Discapnet website, frequently accessing it to seek information. Therefore, she is used to the current Discapnet website interface and could be quite opposed to changes.

All participants stated that all texts presented on the adapted scenario were very easy to read due to an appropriate size, font and contrast with the background. Therefore, the adaptation techniques related to these issues showed good acceptance. Nevertheless, four participants (P1, P6, P8 and U9) thought that providing controls to customize these values would be useful for them. The adaptation technique “Add hot area around links” also had good acceptance.

Participants P1, P2, P4 and P5 proposed not centring the content and taking advantage of the entire width of the browser in order to reduce vertical scrolling. However, they insisted on the need to avoid horizontal scroll. Participant P4 suggested switching navigation menu and vertical main content distribution to horizontal and adding collapsible/expandable content. Applying the “Strechtext” adaptation technique could be considered in this case. It has not been included in the low vision user profile but it could have benefits for minimizing the vertical scroll. On the other hand, the remaining participants (P3, P6, P7, P8 and P9) stated they had no problem with the amount of vertical scrolling in the adapted version.

Participants P1 and P3, both preferred the original scenario and suggested improving the orientation of the navigation menu on the adapted page. Participant P1 proposed the inclusion of additional visual clues for this purpose.

Participants P4 and P5, both using screen magnifier software as assistive technology, suggested improving orientation by defining an appropriate colour scheme for the main sections. This idea contrasts with the claims of some participants (P6, P7 and P9) regarding the colour-overload on the Discapnet website. These comments suggest that “Identify the navigation with colours” adaptation technique should be reviewed and improved in future studies.

All participants appreciated the adaptation technique “Simplify the webpage removing distracters” applied for avoiding content such as banners, distracting information, etc.

#### **6.4. Summary**

The Model Access tool which implements the theoretical model of transformations is able to adapt the content to different kinds of devices or user needs.

Even though the annotation process can be quite tedious, the use of WAI-ARIA or HTML as a base facilitates the process. In addition, a great number of adaptations can be carried out in pages that already include these languages. The use of an ontology as a base for the system enabled new devices and assistive technologies to be added by simply modifying or updating the classes and reasoning rules.

Regarding the experiments carried out with people with motor impairments, due to the high variability of the characteristics and needs of users with motor disabilities, finding an optimal alternative method of interaction for everyone was not possible. However, some methods do work well for specific people under particular circumstances. For instance, participants who moved their finger more during the selection of targets could benefit from “end tap” or “steady tap” interaction methods.

The lack of more universal interaction methods can be mitigated by a more thorough longitudinal study that would enable us to determine when -and for whom- one interaction method is better than another. This knowledge could be used to provide a path towards dynamically adaptive interaction. For instance, in an adaptive system, “augmented tap” would be applied when a selectable element is alone and “steady tap” or “end tap” when the element is surrounded by other elements.

Nevertheless, it should not be overlooked that interaction methods must be changed with caution. Changes in the interaction techniques should not interfere with consolidated gestures, such as “slide”. Therefore, in addition to testing methods of interaction with target selection tasks, they should also be evaluated while surfing the web. This can help to determine how useful the alternative method of interaction really is.

The results of the study carried out with people with low vision indicates that objective data gathered in an experiment (such as the task completion time, completed task rate, etc.) is not always related to the participants’ perception of their navigation experience. One participant of the study preferred the automatically generated adapted version of Discapnet website even if his objective values were better in the original version. Another participant preferred the original version even though he completed more search tasks in the adapted version. Subjective satisfaction and perceived ease of use is influenced by several factors.

Some of the adaptation techniques considered in this work proved to be beneficial for some users while being inconvenient for others. For instance, linearizing and centring the information of a web page does not always facilitate content browsing. Apart from increasing the vertical scroll of the resulting page, these techniques also tend to spread the content and thus some of the visual structure of the original ad hoc design is lost. Users utilizing specific magnification software improved their experience but other users found navigating easier on the original version of the Discapnet website. They seemed to take advantage of visual information (structure, colours and icons) that enabled faster browsing for them. Combining different adaptations techniques with user customization features may increase their efficiency, effectiveness and satisfaction when browsing the Web.

---

---

# Chapter 7

## Conclusions & Future Work

---

---

### *7.1. Conclusions*

This thesis work presents a conceptual model for web transcoding based on user behaviour, an implementation of this conceptual model through the Model Access tool, and a testing tool, Remotest, used to collect the required data for the implementation and to evaluate the results.

The conceptual model defines all the elements necessary to perform the adaptations. Applications implementing this model allow the accessibility and usability of existing non-accessible pages to be repaired and for them to be adapted to the needs of individual users. The model includes the evaluation process as a prerequisite to producing more accurate transformations.

#### *7.1.1. RemoTest*

An implementation of the conceptual model led to RemoTest and Model Access tools. RemoTest collects data about user behaviour by means of different kinds of remote and local accessible experiments with real users. In addition, this tool is used to verify the validity and accessibility of the transcoded web pages. Given that user participation and evaluation is fundamental, RemoTest was designed to guarantee the accessibility of the experiments to a wide range of users.

The studies carried out with RemoTest served to:

- Assess the accessibility of the RemoTest tool itself and its installation
- Observe web navigation strategies used by people with diverse sensory and motor disabilities
- Detect usability or accessibility problems in web pages
- Evaluate the transformations performed by the transcoding tool Model Access
- Identify and characterize the assistive technology used (applying data mining techniques) [Perona et al., 2016]

#### *7.1.2. Model Access*

The Model Access tool implements the transformation model of the conceptual framework. Its aim is to fix the accessibility of pre-existing non-accessible web pages and to adapt them to the needs of specific users and to the requirements of their interaction devices.

Given that the conceptual model requires non-accessible web pages to be semantically tagged by means of a mark-up language, we designed an extension of the WAI-ARIA mark-up language in order to facilitate the web page's annotation. In this way, Model Access is compatible with pages tagged with original WAI-ARIA annotations.

Transcoding procedures use adaptation techniques to convert non-accessible code into accessible code. We collected a number of adaptation techniques which were well supported by the literature and others were obtained from our own experiments with users (through RemoTest). The granularity of the adaptation techniques facilitate the combination of adaptation techniques to create new adapted user interfaces for new devices or users.

Model Access uses models managed by an ontology. The use of an ontology as a base to model the users, devices, web annotations and adaptation techniques makes Model Access especially flexible and easily extensible. For instance, it permits the addition of new types of users or devices without affecting any other part of the system.

### *7.1.3. Case studies*

In this thesis a thorough analysis is carried out of diverse case studies to show how the Model Access and RemoTest tools can be used together for specific users with different devices. The results show that both systems work satisfactorily. In addition, they allowed us to learn about the participants' characteristics and strategies in combination with the adaptation techniques implemented. Furthermore, the data gathered from the experiments allow us to fine tune the adaptation techniques and to enrich the ontology. As a result, new sets of adaptation techniques were included and new preferences were added

## **7.2. Contributions**

The research work carried out in this thesis produced the following contributions:

- A conceptual framework that describes the necessary elements, user, device environment and adaptation techniques model, in order to transform inaccessible web pages into accessible and personalized web pages. It includes evaluation and annotation models needed to create suitable adaptations
- An implementation of the conceptual model by means of the Model Access and RemoTest tools
  - The Model Access transcoding tool uses an ontology that models the device, the user, the annotation language and the adaptation techniques in order to decide which adaptation techniques can be applied
  - RemoTest is a tool based on the Experimental Design and Experimental Session Control languages, to carry out user tests (remote or in situ) with people with disabilities. The platform provides the necessary elements to design, carry out and analyse accessible experiments
- Evaluations of the usability and user experience of the installation process of these tools, carried out with people with diverse types of disabilities
  - Evaluations with people with motor impairments
    - Validity of the algorithm for extracting the intended movements and the different navigation strategies used by participants

- Evaluation of the transcoded interfaces for touch mobile devices and four interaction methods for target selection
- Evaluations with people with low vision
  - Demonstration of how the events gathered can be analysed to find usability problems
  - Evaluation of the accessibility of transcoded pages for people with low vision

### 7.2.1. Publications

As a result of these works, the following research papers were published:

- Valencia, X., Arrue, M., Pérez, J. E., & Abascal, J. (2013). User individuality management in websites based on WAI-ARIA annotations and ontologies. In Proceedings of the 10<sup>th</sup> International Cross-Disciplinary Conference on Web Accessibility, (p. 1-10).
- Valencia, X., Arrue, M., Rojas-Valduciel, H., & Moreno, L. (2014). Interdependent Components for the Development of Accessible XUL Applications for Screen Reader Users. In WEBIST (2), (pp. 65-73).
- Pérez, J. E., Arrue, M., Valencia, X., & Moreno, L. (2014). Exploratory study of web navigation strategies for users with physical disabilities. In Proceedings of the 11<sup>th</sup> Web for All Conference, (p. 1-4).
- Valencia, X., Pérez, J. E., Muñoz, U., Arrue, M., & Abascal, J. (2015). Assisted Interaction Data Analysis of Web-Based User Studies. In Human-Computer Interaction, (pp. 1-19).
- Pérez, J. E., Valencia, X., Arrue, M., & Abascal, J. (2015). Elaborating a Web Interface Personalization Process. In Proceedings of the XVI International Conference on Human Computer Interaction, (p. 1-4).
- Pérez, J. E., Valencia, X., Arrue, M., & Abascal, J. (2015). Depicting the Keypad Mouse Interface: Exploring Pointing Behaviors of Three Input Devices. In Proceedings of the 17<sup>th</sup> International ACM SIGACCESS Conference on Computers & Accessibility, (pp. 413-414).
- Pérez, J. E., Valencia, X., Arrue, M., & Abascal, J. (2016). A usability evaluation of two virtual aids to enhance cursor accessibility for people with motor impairments. In Proceedings of the 13<sup>th</sup> Web for All Conference, (p.1-4)
- Perona, I., Yera, A., Arbelaitz, O., Muguera, J., Ragkousis, N., Arrue, M., Pérez, J. E., Valencia, X. (2016). Automatic device detection in web interaction In Proceedings of the XVII Conferencia de la Asociación Española para la Inteligencia Artificial (pp. 825-834).
- Perona, I., Yera, A., Arbelaitz, O., Muguera, J., Pérez, J. E., Valencia, X. (2017). Web elkarrekintzan erabilitako gailuen detekzio automatikoa Ikergazte (In press).
- Valencia, X., Pérez, J. E., Arrue, M., Abascal, J., Duarte, C., Carriço, L. (2017). Adapting the web for people with upper body motor impairments using a touchscreen mobile device Interacting with Computers. (Under Review).

### 7.3. Future Work

In addition to the work that has been already carried out, there are a number of challenges for future research.

- The Remotest platform is currently accessible for several types of disabilities (namely users with low vision, motor impairments, and blind people). Other users' needs should be tested in order to ensure the universal accessibility and usability of the tool. Specifically, studies should be done to better understand the needs of people with cognitive impairments or with hearing disabilities.
- Most of our experiments were carried out in controlled conditions with the supervision of a researcher. It would be interesting to develop un-moderated tests to check if the data gathered is useful for obtaining meaningful characteristics (e.g. cursor movements or unexpected web accessibility issues). In addition, longitudinal studies should be carried out to test new adaptation techniques or new interaction devices. Longitudinal experiments would inform us about user learning curve, effectiveness of adaptation techniques in natural usage, etc.
- Data mining techniques applied to the interaction data could be used to detect unusual user behaviour patterns, or to monitor health issues. The analysis of the events generated by the users would produce suitable information to feed the user model, infer the assistive technology used, or to detect changing abilities due to fatigue or deterioration of health conditions.
- Regarding Model Access, it would be interesting to detect and analyse incoherence among adaption techniques, in order to include this knowledge into the ontology. This would reduce the need for stereotypes and avoid inconsistencies between adaptations applied automatically.
- Even if the annotation process has been simplified compared with other similar systems, it can be improved. The current manual annotation process requires some time to be learned. With large sets of web pages annotation can become a bottleneck. There are two main approaches which can mitigate this:
  - automatic annotation using a procedure that ensures the adequateness of adaptations
  - semi-automatic annotation using methods where the users themselves, or volunteers, tag web pages with the help of suitable annotation tools. In this case, crowdsourcing and gaming methods can be used to enhance the motivation of users to keep annotating web pages

# References

- Abascal, J., Arrue, M., Fajardo, I., & Garay, N. (2006). An expert-based usability evaluation of the EvalAccess web service. In HCI related papers of Interacción 2004, (pp. 1-17).
- Abrams, M., Phanouriou, C., Batongbacal, A. L., Williams, S. M., & Shuster, J. E. (1999). UIML: an appliance-independent XML user interface language. *Computer Networks*, 31(11), (pp. 1695-1708).
- Access works (2016). <https://www.access-works.knowbility.org/>
- Ackermann, P., Velasco, C. A., & Power, C. (2012). Developing a semantic user and device modeling framework that supports UI adaptability of web 2.0 applications for people with special needs. In Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, (p. 1-4).
- Ahmadi, H., & Kong, J. (2012). User-centric adaptation of Web information for small screens. *Journal of visual languages & computing*, 23(1), (pp. 13-28).
- Almanji, A., Davies, T. C., & Stott, N. S. (2014). Using cursor measures to investigate the effects of impairment severity on cursor control for youths with cerebral palsy. *International Journal of Human-Computer Studies*, 72(3), (pp. 349-357).
- Asakawa, C., & Takagi, H. (2008). Transcoding. In *Web Accessibility* (pp. 231-260).
- Atterer, R., Wnuk, M., & Schmidt, A. (2006). Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In Proceedings of the 15<sup>th</sup> international conference on World Wide Web, (pp. 203-212).
- Bidasoa turismo (2016). <http://www.bidasoaturismo.com/index.php?lang=es>
- Bigham, J. P., & 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), (pp. 25-34).
- Bigham, J. P., Kaminsky, R. S., Ladner, R. E., Danielsson, O. M., & Hempton, G. L. (2006). WebInSight:: making web images accessible. In Proceedings of the 8<sup>th</sup> international ACM SIGACCESS conference on Computers and accessibility, (pp. 181-188).
- Bigham, J. P., Ladner, R. E., & Borodin, Y. (2011). The design of human-powered access technology. In The proceedings of the 13<sup>th</sup> international ACM SIGACCESS conference on Computers and accessibility, (pp. 3-10).
- Bolin, M., Webber, M., Rha, P., Wilson, T., & Miller, R. C. (2005). Automation and customization of rendered web pages. In Proceedings of the 18<sup>th</sup> annual ACM symposium on User interface software and technology, (pp. 163-172).
- Borodin, Y., Ahmed, F., Islam, M. A., Puzis, Y., Melnyk, V., Feng, S., & Dausch, G. (2010). Hearsay: a new generation context-driven multi-modal assistive web



- browser. In Proceedings of the 19<sup>th</sup> international conference on World Wide Web, (pp. 1233-1236).
- Brajnik, G. (2004). Achieving universal web access through specialized user interfaces In Proceedings of the 8<sup>th</sup> ERCIM UI4ALL Workshop, (pp. 28-29).
- Brajnik, G., & Giachin, C. (2014). Using sketches and storyboards to assess impact of age difference in user experience. *International Journal of Human-Computer Studies*, 72(6), (pp. 552-566).
- Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* 11, 1-2 (March 2001), (pp. 87-110).
- Brusilovsky, P. (2004). Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. *PsychNology Journal*, 2(1), (pp. 7-23).
- Burzacca, P., & Paternò, F. (2013). Remote usability evaluation of mobile web applications. In *International Conference on Human-Computer Interaction* (pp. 241-248).
- Cameleon Reference Framework (2016).  
[https://www.w3.org/community/uad/wiki/Cameleon\\_Reference\\_Framework](https://www.w3.org/community/uad/wiki/Cameleon_Reference_Framework)
- Chapuis, O., Blanch, R., & Beaudouin-Lafon, M. (2007). Fitts' law in the wild: A field study of aimed movements. Tech. Rep. 1480, LRI, Univ. Paris-Sud, France, (p. 1-11).
- Claypool, M., Le, P., Wased, M., & Brown, D. (2001). Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, (pp. 33-40).
- Coelho, J., & Duarte, C. (2016). A literature survey on older adults' use of social network services and social applications. *Computers in Human Behavior*, 58, (pp. 187-205).
- Discapnet (2016). <http://www.discalpnet.es/Castellano/Paginas/default.aspx>
- Dobransky, K., & Hargittai, E. (2006). The disability divide in Internet access and use. *Information, Communication & Society*, 9(3), (pp. 313-334).
- Edmonds, A. (2003). Uzilla: A new tool for Web usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(2), (pp. 194-201).
- Etgen, M., & Cantor, J. (1999). What does getting WET (web event-logging tool) mean for web usability. In *Proceedings of Fifth Human Factors and the Web Conference*.
- Extended Log File Format (2016). <https://www.w3.org/TR/WD-logfile>
- Faraday, P. (2000). Visually critiquing web pages. In *Multimedia'99*, (pp. 155-166).
- Fernandes, N., Costa, D., Duarte, C., & Carriço, L. (2012). Evaluating the accessibility of web applications. *Procedia Computer Science*, 14, (pp. 28-35).
- Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., & Wobbrock, J. O. (2010). Enhanced area cursors: reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23<sup>rd</sup> annual ACM symposium on User interface software and technology*, (pp. 153-162)



- Flatla, D. R., Reinecke, K., Gutwin, C., & Gajos, K. Z. (2013). SPRWeb: preserving subjective responses to website colour schemes through automatic recolouring. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (pp. 2069-2078).
- Gajos, K. Z., & Weld, D. S. (2004). SUPPLE: automatically generating user interfaces. In Proceedings of the 9<sup>th</sup> international conference on Intelligent user interfaces, (pp. 93-100).
- Gajos, K. Z., Reinecke, K., & Herrmann, C. (2012). Accurate measurements of pointing performance from in situ observations. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (pp. 3157-3166).
- Gajos, K. Z., Weld, D. S., & Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with Supple. *Artificial Intelligence*, 174(12), (pp. 910-950).
- Google Analytics (2016). <http://www.google.com/analytics/>
- Google-AxsJAX (2016). <https://code.google.com/p/google-axsjax/>
- Greasemonkey (2016). <https://addons.mozilla.org/es/firefox/addon/greasemonkey/>
- Grossman, T., & Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In Proceedings of the SIGCHI conference on Human factors in computing systems, (pp. 281-290).
- Guerreiro, Y., Nicolau, H., Jorge J., and Gonçalves. D. (2010). Towards accessible touch interfaces. In Proceedings of the 12<sup>th</sup> international ACM SIGACCESS conference on Computers and accessibility (ASSETS '10), (pp. 19-26).
- Harper, S., & Bechhofer, S. (2007). SADie: Structural semantics for accessibility and device independence. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(2), (p. 1-10).
- Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*, 52, (pp. 139-183).
- Hassenzahl, M. (2003). The thing and I: understanding the relationship between user and product. In *Funology*, (pp. 31-42).
- Hassenzahl, M. (2004). The interplay of beauty, goodness, and usability in interactive products. *Human-computer interaction*, 19(4), (pp. 319-349).
- Hassenzahl, M., Burmester, M., & Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In *Mensch & Computer 2003*, (pp. 187-196).
- Hong, J. I., Heer, J., Waterson, S., & Landay, J. A. (2001). WebQuilt: A proxy-based approach to remote web usability testing. *ACM Transactions on Information Systems*, 19(3), (pp. 263-285).
- HTML5 (2016). <http://www.w3.org/TR/html5/>
- Huang, A. W., & Sundaresan, N. (2000). A semantic transcoding system to adapt web services for users with disabilities. In Proceedings of the fourth international ACM conference on Assistive technologies, (pp. 156-163).

- Hurst, A., Hudson, S. E., Mankoff, J., & Trewin, S. (2008). Automatically detecting pointing performance. In Proceedings of the 13<sup>th</sup> international conference on Intelligent user interfaces, (pp. 11-19).
- IMS ACCLIP (2016). [http://wiki.cetis.ac.uk/ACCLIP\\_Briefing](http://wiki.cetis.ac.uk/ACCLIP_Briefing)
- ISO 9241 (2016). <https://www.iso.org/obp/ui/#iso:std:iso:9241:-171:ed-1:v1:en>
- ISO/IEC TR 29138 (2016). [http://www.iso.org/iso/catalogue\\_detail?csnumber=45161](http://www.iso.org/iso/catalogue_detail?csnumber=45161)
- Kane, S. K., Jayant, C., Wobbrock, J. O., & Ladner, R. E. (2009). Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In Proceedings of the 11<sup>th</sup> international ACM SIGACCESS conference on Computers and accessibility, (pp. 115-122).
- Kasday, L. R. (2000). A tool to evaluate universal Web accessibility. In Proceedings on the 2000 conference on Universal Usability, (pp. 161-162).
- Keates, S., Hwang, F., Langdon, P., Clarkson, P. J., & Robinson, P. (2002). The use of cursor measures for motion-impaired computer users. *Universal Access in the Information Society*, 2(1), (pp. 18-29).
- Kitchel, E. (2016). APH Guidelines for Print Document Design. <http://www.aph.org/research/design-guidelines/>
- Knowbility (2016). <https://www.knowbility.org/>
- Knutov, E., De Bra, P., & Pechenizkiy, M. (2009). AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, 15(1), (pp. 5-38).
- Kobsa, A. (2001). Generic user modeling systems. *User modeling and user-adapted interaction*, 11(1-2), (pp. 49-63).
- Kurniawan, S. H., King, A., Evans, D. G., & Blenkhorn, P. L. (2006). Personalising web page presentation for older people. *Interacting with computers*, 18(3), (pp. 457-477).
- Law, E. L. C., Roto, V., Hassenzahl, M., Vermeeren, A. P., & Kort, J. (2009). Understanding, scoping and defining user experience: a survey approach. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (pp. 719-728).
- Lazar J, Allen A, Kleinman J, Malarkey C. (2007) What Frustrates Screen Reader Users on the Web: A Study of 100 Blind Users, *International Journal of human-computer interaction*, Taylor & Francis, 22 (3), (pp. 247-269).
- Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., & López-Jaquero, V. (2004). USIXML: a language supporting multi-path development of user interfaces. In *International Workshop on Design, Specification, and Verification of Interactive Systems*, (pp. 200-220).
- Loop11 (2016). <http://www.loop11.com/>
- Lunn, D., Harper, S., & Bechhofer, S. (2011). Identifying behavioral strategies of visually impaired users to improve access to web content. *ACM Transactions on Accessible Computing (TACCESS)*, 3(4), (p. 1-35).

- MacKenzie, I. S., Kauppinen, T., & Silfverberg, M. (2001). Accuracy measures for evaluating computer pointing devices. In Proceedings of the SIGCHI conference on Human factors in computing systems, (pp. 9-16).
- Martins, W. S., Lucas, D. C. S., Neves, K. D. S., & Bertioli, D. J. (2009). WebSat—a web software for microsatellite marker development. *Bioinformation*, 3(6), (pp. 282-283).
- Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., & Keith Smith, J. E. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological review*, 95(3), (pp. 340-370).
- Mirri, S., Salomoni, P., Prandi, C., & Muratori, L. A. (2012). GAPforAPE: an augmented browsing system to improve Web 2.0 accessibility. *New Review of Hypermedia and Multimedia*, 18(3), (pp. 205-229).
- Montague, K., Hanson, V. L., & Cobley, A. (2012). Designing for individuals: usable touch-screen interaction through shared user models. In Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility, (pp. 151-158).
- Morae (2016). <http://www.techsmith.com/morae.html>
- Moreno, L., Martinez, P., Ruiz, B., & Iglesias, A. (2011). Toward an equal opportunity web: applications, standards, and tools that increase accessibility. *Computer*, 44(5), (pp. 18-26).
- Mott, M. E., Vatavu, R. D., Kane, S. K., & Wobbrock, J. O. (2016). Smart Touch: Improving Touch Accuracy for People with Motor Impairments with Template Matching. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, (pp. 1934-1946).
- Nicolau, H., Guerreiro, T., Jorge, J., & Gonçalves, D. (2014). Mobile touchscreen user interfaces: bridging the gap between motor-impaired and able-bodied users. *Universal Access in the Information Society*, 13(3), (pp. 303-313).
- Nielsen, J. (1994). Usability inspection methods. In Conference companion on Human factors in computing systems, (pp. 413-414).
- Nielsen, J. (1996). Usability metrics: Tracking interface improvements. *IEEE Software*, 13(6), (pp. 12-13).
- Optimizely (2016). <https://www.optimizely.com/>
- OWL (2016). <https://www.w3.org/TR/owl-features/>
- Paganelli, L., & Paternò, F. (2002). Intelligent analysis of user interactions with web applications. In Proceedings of the 7<sup>th</sup> international conference on Intelligent user interfaces, (pp. 111-118).
- Paternò, F., & Ballardini, G. (2000). RemUSINE: a bridge between empirical and model-based evaluation when evaluators and users are distant. *Interacting with computers*, 13(2), (pp. 229-251).
- Paternò, F., Mancini, C., & Meniconi, S. (1997). ConcurTaskTrees: A diagrammatic notation for specifying task models. In Human-Computer Interaction INTERACT'97, (pp. 362-369).

- Paternò, F., Santoro, C., & Spano, L. D. (2009). MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(4), (p. 1-30).
- Pérez, J. E., Arrue, M., Valencia, X., & Moreno, L. (2014). Exploratory study of web navigation strategies for users with physical disabilities. In *Proceedings of the 11th Web for All Conference*, (p. 1-4).
- Pérez, J. E., Valencia, X., Arrue, M., & Abascal, J. (2015). Depicting the Keypad Mouse Interface: Exploring Pointing Behaviors of Three Input Devices. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, (pp. 413-414).
- Perona, I., Yera, A., Arbelaitz, O., Muguerza, J., Ragkousis, N., Arrue, M., Pérez, J. E., Valencia, X. (2016). Automatic device detection in web interaction In *Proceedings of the XVII Conferencia de la Asociación Española para la Inteligencia Artificial*, (pp. 825-834).
- Petrie, H., & Bevan, N. (2009). The evaluation of accessibility, usability and user experience. *The universal access handbook*, (pp. 10-20).
- Power, C., Petrie, H., & Mitchell, R. (2009). A framework for remote user evaluation of accessibility and usability of websites. In *International Conference on Universal Access in Human-Computer Interaction*, (pp. 594-601).
- QML (2016). <http://doc.qt.io/qt-5/qtqml-index.html>
- Reynolds, F., Woodrow, C., Ohto, H., Klyne, G. (2001). Composite Capability/Preference Profiles (CC/PP): Structure and vocabularies. W3C Working Draft
- Richards, J. T., & Hanson, V. L. (2004). Web accessibility: a broader view. In *Proceedings of the 13th international conference on World Wide Web* (pp. 72-79).
- Sato, D., Zhu, S., Kobayashi, M., Takagi, H., & Asakawa, C. (2011). Sasayaki: augmented voice web browsing experience. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 2769-2778).
- Scholtz, J., Laskowski, S., & Downey, L. (1998). Developing usability tools and techniques for designing and testing web sites. In *Proceedings HFWeb, 98*, (p. 1-10).
- Section 508 (2016). <https://www.section508.gov/>
- Semantic Web (2016). <https://www.w3.org/standards/semanticweb/>
- Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, 43(5), (pp. 84-91).
- Smith, P. A. (1996). Towards a practical measure of hypertext usability. *Interacting with computers*, 8(4), (pp. 365-381).
- Takagi, H., Asakawa, C., Fukuda, K., & Maeda, J. (2002). Site-wide annotation: reconstructing existing pages to be accessible. In *Proceedings of the fifth international ACM conference on Assistive technologies*, (pp. 81-88).

- Theofanos, M. F., & Redish, J. G. (2005). Helping low-vision and other users with web sites that meet their needs: Is one site for all feasible?. *Technical communication*, 52(1), (pp. 9-20).
- Trewin, S., Swart, C., & Pettick, D. (2013). Physical accessibility of touchscreen smartphones. In *Proceedings of the 15<sup>th</sup> International ACM SIGACCESS Conference on Computers and Accessibility*, (p. 1-8).
- UI Events (2016). <https://www.w3.org/TR/uievents/>
- Valencia, X., Pérez, J. E., Muñoz, U., Arrue, M., & Abascal, J. (2015). Assisted Interaction Data Analysis of Web-Based User Studies. In *Human-Computer Interaction* (pp. 1-19).
- Vigo, M., & Harper, S. (2013 a). Evaluating accessibility-in-use. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, (p. 1-4).
- Vigo, M., & Harper, S. (2013 b). Coping tactics employed by visually disabled users on the web. *International Journal of Human-Computer Studies*, 71(11), (pp. 1013-1025).
- W3C Mobile Best Practices (2016). <https://www.w3.org/TR/mobile-bp/>
- WAI Intro (2016). <https://www.w3.org/WAI/intro/accessibility.php>
- WAI-ARIA (2016). <https://www.w3.org/TR/wai-aria/>
- WCAG 2.0 (2016). <https://www.w3.org/TR/WCAG20/>
- Wellman, B., & Haythornthwaite, C. (Eds.). (2008). *The Internet in everyday life*. John Wiley & Sons.
- Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In *Usability inspection methods* (pp. 105-140).
- XAML (2016). <http://msdn.microsoft.com/en-us/library/ms752059.aspx>
- XPCOM (2016). <https://developer.mozilla.org/en-US/docs/XPCOM>
- XUL (2016). <https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL>
- XUL Accessibility Guidelines (2013). [https://developer.mozilla.org/en-US/docs/XUL\\_accessibility\\_guidelines](https://developer.mozilla.org/en-US/docs/XUL_accessibility_guidelines)
- Yesilada, Y., Brajnik, G., & Harper, S. (2009). How much does expertise matter?: a barrier walkthrough study with experts and non-experts. In *Proceedings of the 11<sup>th</sup> international ACM SIGACCESS conference on Computers and accessibility*, (pp. 203-210).
- Yesilada, Y., Stevens, R., Harper, S., & Goble, C. (2007). Evaluating DANTE: Semantic transcoding for visually disabled users. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(3), (p. 1-30).
- Zajicek, M. (2007). Web 2.0: hype or happiness?. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, (pp. 35-39).