



HAL
open science

Assisted Interaction Data Analysis of Web-Based User Studies

Xabier Valencia, J. Eduardo Pérez, Unai Muñoz, Myriam Arrue, Julio Abascal

► **To cite this version:**

Xabier Valencia, J. Eduardo Pérez, Unai Muñoz, Myriam Arrue, Julio Abascal. Assisted Interaction Data Analysis of Web-Based User Studies. 15th Human-Computer Interaction (INTERACT), Sep 2015, Bamberg, Germany. pp.1-19, 10.1007/978-3-319-22701-6_1 . hal-01599645

HAL Id: hal-01599645

<https://hal.inria.fr/hal-01599645>

Submitted on 2 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution| 4.0 International License

Assisted Interaction Data Analysis of Web-Based User Studies

Xabier Valencia, J. Eduardo Pérez, Unai Muñoz, Myriam Arrue, Julio Abascal

EGOKITUZ: Laboratory of HCI for Special Needs, University of the Basque Country
(UPV/EHU) Informatika Fakultatea 20018 Donostia, Spain
[xabier.valencia, juaneduardo.perez, unai.munoz, myriam.arrue,
julio.abascal]@ehu.eus

Abstract. User behaviour analysis requires defining experimental sessions with numerous participants. In this context, the specification of experiments is a demanding task, as several issues have to be considered such as the type of experiment, the type and number of tasks, the definition of questionnaires and the user interaction data to be gathered. The analysis of collected data is also complex and often requires repeatedly examining recorded interaction videos. In order to deal with these tasks, we present a platform called RemoTest which assists researchers to specify and conduct experimental sessions as well as to gather and analyse the interaction data. This platform has been applied to define different formal user studies on the web and has assisted researchers in detecting the main interaction characteristics of different user profiles and settings.

Keywords: Web accessibility, user testing, user behaviour, accessibility in use.

1 INTRODUCTION

User behaviour when interacting with the Web has been extensively studied in the last decade. This significant research area requires the conducting of experimental sessions with large and diverse groups of users. Experimental sessions have to be carefully planned in order to obtain meaningful results because a minor fault in the design could lead to an erroneous interpretation of results. Researchers need to clearly define the objectives of the experiment, the type of experiment, the stimuli to be presented to participants, the tasks to be performed and the procedure of the experimental sessions. In addition, sometimes specific questionnaires are required in order to explicitly obtain certain data from the participants. The experiment design process is demanding and involves knowledge from different areas such as human factors, hypertext, web technology, etc.

The designed experiments are intended to gather significant interaction data. This data could be gathered through combining different methods such as the traditional ones of recording sessions with video cameras or using specific software components to conveniently collect and store it.

Interaction data analysis is a tedious task especially when only traditional recording methods have been used. Researchers are required to view the recorded videos repeatedly and annotate every meaningful interaction event and data. Among others, cursor movement events are of vital importance when studying the accessibility-in-use of websites or web navigation strategies. For instance, actions such as pointing to a target (buttons, scroll bars, check boxes and radio buttons, etc.), clicking on a target or providing accurate text entry could be very difficult for people with motor impairments due to their lack of dexterity [23]. Examining the cursor movements on recorded images is a hard task that can be alleviated through the application of software components. These components should be efficient in appropriately storing, presenting and preparing all this interaction data for analysis.

In addition, involving an appropriate number of participants for a specific experiment is also challenging. Frequently, this is due to the location and the rigorous timing of sessions. Nowadays, there is an increasing interest in using software tools for conducting experiments remotely. That means that participants are observed while they perform the tasks in their habitual daily environment. This particularly facilitates the conduction of experiments with disabled people, as their environment is already adapted to their needs. Moreover, this type of experiments gathers real interaction data without any obtrusive observation mechanism [2]. It also makes it possible to involve a larger number of participants, as they do not have to physically get to a specific location.

This paper presents a platform called RemoTest that can be applied in remote and in-situ experimental sessions. Its objective is to assist researchers when designing experiments, conducting experimental sessions and analysing data gathered in the sessions. A case study based on a real in-situ exploratory study with 16 participants (11 people with physical impairments and 5 able-bodied participants) is described in section 4. The objective of this experiment was to analyse the differing interaction characteristics of users with physical impairments and able-bodied users. The interaction of the 11 participants with physical impairments was manually analysed based on the video recordings and results are presented and discussed in [20]. The RemoTest platform was also applied to analyse the data of all the participants so the results of manual and automated analysis are contrasted in this paper as a proof-of-concept of the proposed platform.

2 RELATED WORK

In the last few years, the use of tools for remotely conducting user tests has come to the attention of researchers. These tools can be classified depending upon their architecture: server-side tools, proxy tools and client-side tools. Each one has its drawbacks and advantages. Server-side tools are the most unobtrusive. Participants are not required to install or configure anything on their systems. Among the HTTP requests from the servers, user interaction data could be also gathered by modifying the web pages with code to track the interaction data (for instance, JavaScript). One drawback is that the conducted tests could be only based on websites located on the servers

researchers have access to. Contrarily, proxy tools require participants to configure their web browser in order to access to the proxy. This type of tool enables more information to be gathered than via the server logs as well as permitting the tests to be conducted on any website but however, it is not possible to capture all the participants actions, such as the events on the browser. Finally, client-side tools are the most obtrusive ones since they require the installation of some add-on applications to the participant's system. Nevertheless, they are the best option for researchers because they capture all the user interaction data needed in a user behaviour study, such as cursor movements, scroll events, clicking actions, browser events (backward/forward button, print/save page, add bookmark), etc. without modifying the original web pages.

There are some commercial tools such as Google Analytics [11], Loop11 [15] and Morae [17]. Google Analytics is a server-side tool. Its objective is to obtain general connection data about the users of a specific web site but not to conduct formal experiments. It records information such as the network provider or the browser used by users accessing to a specific web site enriched with Google Analytics. Loop11 is a proxy tool developed for conducting user tests. It includes features that facilitate the definition of web tasks or questionnaires. Regarding the data analysis, it provides click stream reports to visualize the paths that the user followed, click heat maps and the option of visualizing the data in real time. There is a version of Loop11 using AccessWorks devoted to performing user tests with disabled people. Nevertheless this tool does not capture browser events. These events can be used as user disorientation indicators (for example, several clicks on the browser backward button in a task may sometimes imply user orientation problems). Morae is a client-side tool which stores user interaction data when interacting with either standalone applications or web sites. In addition, it provides tools to enable the observation and annotation while the user is performing the test. The tool is quite complex to configure due to the amount of available options provided. Sometimes, programmer skills are required in order to gather some interaction events.

There are many examples of tools developed in academic contexts. Webquilt [12] is a proxy-based tool that stores only information obtained from HTTP requests. NIST WebMetricsSuite [22], [5] is a server-side tool that provides methods to assess the usability of web pages by analysing the path followed by the user. These tools only gather events related to clicks. Therefore, no user interaction data is gathered. Other tools such as WET [8] or USAPROXY [3] are more comprehensive as they also enable the detection of problematic web elements by gathering user interaction data such as mouse movements and keyboard events by injecting Javascript code to the original web pages. Nevertheless, these tools do not assist in formal experiment definition, as they do not include features for task definition or the elaboration of questionnaires. WebRemUSINE tool [18] provides researchers with features for defining formal experiments. It is a client-side tool that uses both technologies Java Applets and JavaScript code to capture the interaction events. The tool is based on ConcurTaskTrees (CTT) task model annotation [19] for defining experiments. It detects usability problems analyzing the differences between the path followed by users to perform the defined tasks and the specified task model in CTT. It requires high knowledge levels

and expertise to define the task models, making this tool only usable by expert researchers. In addition, this tool does not provide features for defining questionnaires to fill in by participants during the experimental sessions.

On the contrary, Curious Browser [5] presents questionnaires to the user after every new visited page, with the aim of rating its interest. Nevertheless the tool does not provide methods to define formal experiments like defining tasks, pre/post task questionnaires and so on. Uzilla [7] is another comprehensive tool for defining formal experiments. It also provides features for defining questionnaires to be filled in by participants during the session. However, there is not much information about the suitability of the created questionnaires for people with disabilities.

Some other approaches can be found relating to the study of users' interaction performance in the wild. Gajos et al. [9] 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. [14]. This tool classifies users' characteristics based on their input events. Both tools provide valuable information about user behaviour or characteristics but do not allow performing formal experiments.

Power et al. list a number of requirements that a remote user tool should meet to be able to conduct experimental sessions with users with disabilities [21]. Some of them are related to participants and other to researchers. The requirements related to participants include the following: provide features to record demographic data (P1), specify the technology used (OS, browser, assistive technology) (P2), select the trials (P3). The ones related to researchers are the following: provide features to test customized and "real" websites (R1), define tasks for a set of users (R2), specify a set of questions to the user, before and/or after the task has been completed (R3), provide instructions and training documents for each trial (R4). Table 1 shows the requirements fulfilled by the academic context tools presented in this section.

The Remotest platform is a client-side tool that provides all the necessary features for defining formal experiments and fulfils the requirements proposed by Power and colleagues. In addition, the questionnaires created by the tool have proven to be accessible to people with disabilities in several evaluations.

Table 1. Classification of user testing tools according to the requirements proposed by Power et al. and their location.

Name	Location	P1	P2	P3	R1	R2	R3	R4
NISTWebMetrics Suite	Sever	No	No	No	Partial	No	No	No
WET	Server	No	No	Yes	Partial	No	No	No
WEBQUILT	Proxy	No	No	No	Yes	No	No	No
Curious Browser	Client	No	No	No	Yes	No	Partial	No
UsaProxy	Proxy	No	No	No	Yes	No	No	No
WebRemUSINE	Server	No	No	Yes	Yes	Yes	No	No
Uzilla	Client	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gajos et al.	Client	No	No	No	Yes	No	No	No
Hurst et al.	Client	No	No	No	Yes	No	No	No

3 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. This 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 and to analyse user performance improvement when interacting with adapted versions of original web pages and so on.

The architecture of the platform has been designed taking all these different types of experiments into consideration. In this case, we opted for a hybrid architecture model that includes some functions in a client-side module and the other ones in some server-side modules. The platform is split into four modules: Experimenter Module (EXm), Participant Module (PAm), Coordinator Module (COm) and Results Viewer Module (RVm).

3.1 Experimenter Module

This module provides a set of functionalities for defining all the components of the experiments. It is a server-side module which can be accessed by experimenters from any computer with an Internet connection. All the definition process is performed by the use of a web application and has been divided into five main steps:

Step 1: Specifying the type of experiment. This first step is intended to specify the type of the experiment (survey or navigation tasks on the web) and general characteristics of the experiment such as the stimuli to be presented in experimental sessions, number of questionnaires (demographic data questionnaire, 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 would ask to provide information about the tasks to be performed by participants and the stimuli to present. 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.

For each task some details have to be provided. For instance, for the “Fill in Questionnaire” task the questions, the type of questions (open-ended or closed) and possible options for answers (likert scales, ranges) have to be defined. There is some other information regarding title, id, etc., that has to be completed in order to ensure the accessibility of the created questionnaire. The “Web Navigation” task also requires some data. Currently, there are two types of tasks in this category: “searching target” tasks and “free navigation” tasks. The former one refers to tasks where participants are required to find a specific target (such as a specific button, link, form, etc.) whereas the latter one 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 so that, in the experimental sessions, an accessible task explanation web page is presented to participants before starting the task and an alert after performing the task.

Once all the tasks are defined, the dependencies between them have to be explicitly stated. This module provides features to define if any task should be presented just before or after another one. For example, in an experiment to measure satisfaction of users when interacting with a certain website, a “Fill in Questionnaire” task about satisfaction has to 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 has to be 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. The task sequence should be counterbalanced between participants and the experimenter is required to define one method (manual, latin square, rotation or random). In between-groups experiments, the tasks to perform by each group are manually selected by the experimenter. The task sequence for each group can be defined manually or automatically (using random method). In all cases, the EXm considers the dependencies between tasks and any other 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 automatically 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...
- 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 participants in a database through the module EXm. This tool includes options such as manual selection of participants in each group, randomly creating groups from the selected participants or establishing some 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.

3.2 Coordinator Module

The Coordinator Module is a server-side type module which has been developed as a Web Service. The objectives of this module are the following:

- Storing and managing the experiments defined by different experimenters applying features of EXm.

- Creating stimuli to present during the experimental sessions (questionnaires, task description web page, task completion alert, etc.).
- Defining the personalized experimental sessions specifications.
- Collecting and storing interaction data obtained in experimental sessions.
- Maintaining information about experimenters, experiments, participants in databases.

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. In addition, it prepares the participant 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 the obtaining of specific personalized experimental session specifications for each participant defined in an XML-based language developed for this purpose, the Experimental Session Controlling Language. The interaction data created in the sessions are managed and stored in a MongoDB database.

3.3 Participant Module

The participant module is a client-side type module. Therefore participants have to install this module in 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 JavaScript and XML. It processes the Experimental Session Controller Language for correctly conducting participants' experimental sessions. In addition, it gathers all the interaction data, as well as the XMLHttpRequest information during the whole session and asynchronously sends to COM by using AJAX technology.

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. Some statistics by pages, tasks or users can be visualized:

- Rapidity measures: Time on page, cursor average speed, cursor acceleration.
- Accuracy measures: Trajectory distance (cursor travel distance), curvature index (CI) relation between optimal path and path followed, distances to the centre of the target and to the last click and ratio of start-end position amplitude to start-target centre amplitude [1].

Fig. 1 shows the information presented 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

Fig. 1. General information about the performance of a participant in a visited web page.

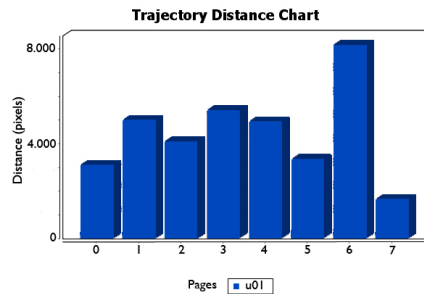


Fig. 2. Comparing chart about the trajectory distance parameter of a participant

This module includes a tool for comparing the performance of each participant in all the visited pages. Fig. 2 show the charts generated for comparing the trajectory distance of a participant in the visited pages.

The general information shown by the RemoTest platform assist researchers to obtain detailed information about the performance of each participant during the experimental session. In addition, this module also provides several graphs for each visited page. One is the “Distance to target chart”, where the distance to the target at every moment can be seen. With this chart researchers can easily identify users’ problems when aiming at the target, see Fig. 3 Left. As can be seen, once cursor is in the target nearby the user requires several attempts to place the cursor on it.

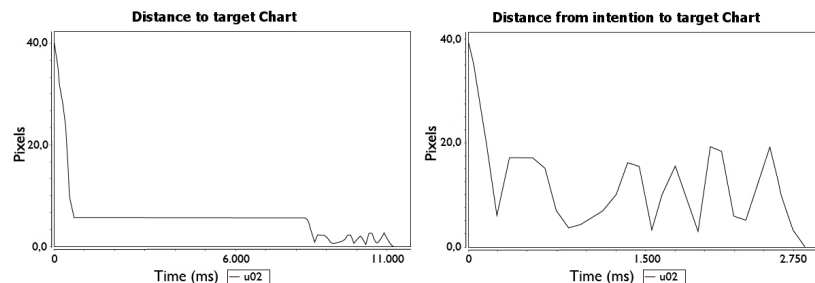


Fig. 3. “Distance to target” (Left) and “Distance from intention to target” (Right) charts

Furthermore, associated with this first chart the tool also provides another graph that shows the distance to the target but starting from users’ intention time to click the target, see Fig. 3 Right. The intention time to click is automatically calculated by the algorithm described in the section 3.4.1.

In addition, graphs of cursor’s movement speed and acceleration charts can be found. Through these graphs researchers can appreciate for instance, the different cursor movements taking into account the input device used by the user.

Fig. 4 shows one participant (who moves the cursor by pressing the numerical keypad with a head pointer) creates similar speed and acceleration peaks followed by short pauses, the time it takes the user to change direction to press another key with the head pointer.

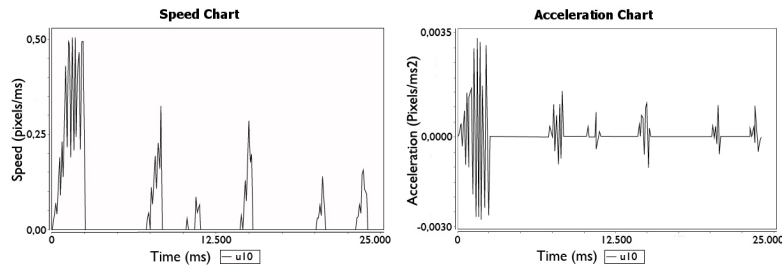


Fig. 4. Speed chart (left) and acceleration chart (right) of a participant (U10 in the case study)

Another important feature is the ability of the tool to make comparisons. In first place, researchers can view a comparison between different statistics about all visited pages in a task of a selected user. In the same way, the application also allows to compare these statistics to all participants for a given task. So, RVm is able to display automatically generated bar charts comparing the average or median, of the trajectory distances, curvature index, times to point and click or cursor speeds.

This straightforward visualization may assist experimenters in discovering relevant bits of users' interactions, and in case of also having complementary video recordings, fast locating the corresponding moments and save huge amount of time viewing and analysing video images.

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

3.4.1 Delimiting Pointing Trajectories

A pointing trajectory starts when a user resolves to move the cursor to reach an objective. Controlled laboratory experiments can specify restricted interactions to make the beginning of cursor movement explicit [9]. In contrast to those studies, more naturalistic settings with untagged web interactions do not permit the register of any explicit trace of the cognitive process behind the users' intention. In those cases, as in ours, heuristics are needed to estimate the beginning of the aimed movements [4], [10].

Among the other possible aimed movements within web GUIs (e.g., hovering over an element to see its tooltip), we have decided to focus on pointing trajectories that end with a navigational click and a posterior page load event. In our case we have considered the following bases for delimiting pointing trajectories.

Beginning of movement

A pointing to 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 provoke the need to analyse all the cursor trajectory throughout every page.

We have considered the first cursor move event recorded within every page as the beginning of movement candidates for pointing to navigational click trajectories. Each time a page scroll interaction occurs, actual cursor trajectory (if it exists) is rejected as a page candidate for pointing to navigational click trajectory, restarting a new pointing trajectory when the next cursor move event is triggered.

Additionally, pauses take place along cursor trajectories are useful to segment pointing trajectories. Aimed movements consist of several sub-movements separated by pauses [16], so trying to delimit a pointing trajectory as a cursor movement without pauses is not feasible.

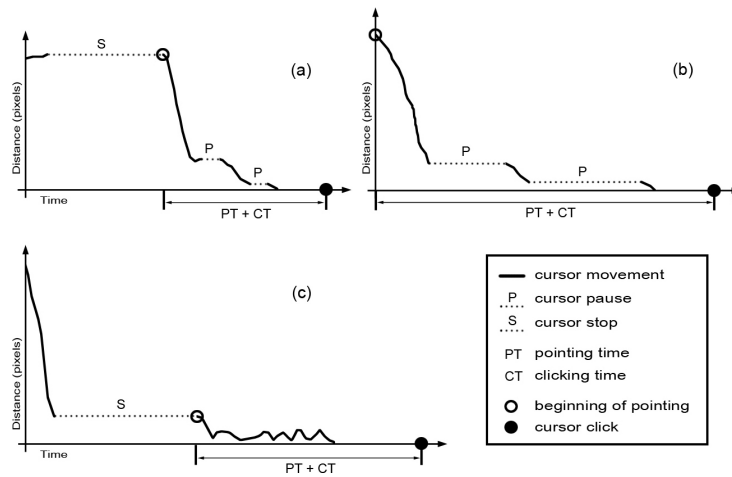


Fig. 5. Beginning of pointing trajectories estimations for 3 physically impaired participants using different computer pointing devices: (a) joystick, (b) numeric keypad with a head pointer and (c) oversized trackball.

Calculating Valid User Pauses

A pause during an aimed movement may correspond to a sub-movement transition, or in case of being long enough (movement stop) to cursor trajectory segmentation (Fig. 5).

Unlike other studies where a unique value for all users serves to distinguish between valid pauses during cursor aimed movements and movement stops [4], [10], we have followed an individual approach for this purpose. As we have observed from the interaction of the physically impaired participants of our study, valid pauses during aimed movements vary among users depending on the computer pointing device used and their ability with it (Fig. 5). For instance, a numeric keypad user needs more time

to change cursor trajectory (notable pauses between strokes) than a joystick user. We believe that calculating a movement stop threshold for every user will improve the quality of the segmentation and therefore the pointing trajectory-related measures.

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

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 have discarded all pages from our data logs without pointing to navigational click trajectories, for instance using keyboard shortcuts or browsing history navigation without moving the cursor.

4 CASE STUDY

This section is devoted to describing the benefits of using the RemoTest platform to define experiments, conducting experimental sessions and collecting and analyzing interaction data. This case study is based on a specific experiment carried out by the authors with 16 participants, 11 people with upper-body physical impairments (U01-U11) and 5 able-bodied people (U12-U16). The objective of the experiment was to analyse the different navigation strategies used by the participants. Web expertise varied among participants as well as the assistive technology and input devices used for accessing the Web. Table 2 shows information about the participants.

The experimental sessions were carried out locally in different settings (experimenters moved to the specific location and assisted participants during all the session) and used different computers: 7 experimental sessions were conducted at the Elkartu premises (a local association of people with physical disabilities), 6 participants carried out the experimental session in a laboratory of the Computer Science School at the University of the Basque Country and the remaining experimental sessions (3 of 16) were conducted at the participant's home. The platform used was similar in all cases: Windows operating system (Windows 7 except of U02 who used Windows XP) and Mozilla Firefox browser.

Participants were asked to install the PAm module on the local computer (their usual computer in some cases). To this end, the PAm Firefox add-on was placed on an URL and a web page with clear instructions of how to install it was created. This task was useful for detecting some minor issues in the instructions that needed to be fixed. Even though it was not the object of the study, this preliminary installation task was also useful in order to test the adequacy of using a client-side tool for conducting

experimental sessions. Despite only one participant (U02) having previous experience in installing this type of software, all participants were able to install the PAm module including the ones with low-level expertise in using the computer (U06, U07).

Table 2. Information about the participants

User	Expertise	Input Devices	Setting
U01	Medium	Reconfigured mouse and keyboard	Elkartu
U02	High	Oversized trackball and keyboard	Lab
U03	High	Joystick and keyboard with cover	Home
U04	Medium	Joystick and keyboard	Elkartu
U05	Medium	Joystick and on screen keyboard	Elkartu
U06	Low	Joystick and keyboard with handstick	Elkartu
U07	Low	Reconfigured touchpad and keyboard	Elkartu
U08	High	Numeric keypad and keyboard	Home
U09	High	Numeric keypad and keyboard	Home
U10	High	Head pointer and reconfigured keyboard	Elkartu
U11	Medium	Head pointer and reconfigured keyboard	Elkartu
U12	High	Standard mouse and keyboard	Lab
U13	High	Standard mouse and keyboard	Lab
U14	High	Standard mouse and keyboard	Lab
U15	High	Standard mouse and keyboard	Lab
U16	High	Standard mouse and keyboard	Lab

4.1 Experiment definition

The experiment consisted of three tasks: filling in a questionnaire about demographic data (Task 1), free navigation task with 5 minutes duration (Task 2) and searching for a target task with a maximum duration of 10 minutes (Task 3).

Web navigation tasks (Task 2 and Task 3) were performed on the Discapnet website [www.discapnet.com] which is specialized in providing information to people with disabilities. The order of these tasks was predefined since the free navigation task was intended to familiarize the users with the website.

RemoTest platform was used for defining the experimental sessions. The definition process was as follows:

Step 1: Specifying the type of experiment. This experiment was a web navigation experiment.

Step 2: Determining the tasks and stimuli of the experimental sessions.

- Task 1: The EXm module guided the researcher in the process of defining the questions and possible options for responses.
- Task 2: This task was a “Web Navigation” type task with free navigation category. The EXm module asked for information regarding this task such as: duration, task description text, URL of the website, task completion text, etc.

- Task 3: This task was "Web Navigation" type task and search target category. The EXm module asked for similar information as in Task 2 and, in addition, the specific target has to be defined. In this case the target is a specific URL in Discapnet.

Step 3: Defining the procedure of the experimental sessions. In this experiment all participants had to do the same tasks and the order was the same in all cases. This information was inserted in the RemoTest platform.

Step 4: Specifying interaction data to be gathered. The EXm module presents all the interaction data PAm module is prepared to be gathered so the experimenter could select the most interesting in each case.

Step 5: Selecting the sample of participants. The RemoTest platform provides functions to select the participants of an experiment.

The experiment definition process leads to obtain an XML file containing the experiment definition based on the Experiment Specification Language.

4.2 Creating the experimental sessions

The COM module automatically created the necessary stimuli for the experiment based on its definition. In this particular case study, the stimuli to create were the following: the demographic data questionnaire, the free navigation task description web page, the free navigation task completion web page, the target searching task description web page and the target searching task completion web page.

In addition, the information included in the experiment definition XML file is applied to create the personalized experimental sessions. In this specific case study, all the experimental sessions will be similar.

4.3 Conducting the experiment

The PAm module installed on the client-side guided participants throughout the experimental session based on the XML file in Experimental Session Controlling Language. It managed the duration of tasks, the sequence, the presentation of stimuli, the gathering of data provided by each participant in questionnaires and the interaction data. All the information gathered by this module (that explicitly provided by the participant by answering questionnaires, and the interaction data implicitly obtained by the platform) were sent to the COM module in an asynchronous manner without interrupting the participant session. The experimental session completion web page was shown when all tasks were completed.

4.4 Interaction Data Analysis

The analysis presented in this section is focused on the cursor movement characterization features included in the RVm module and described in Section 3.4. These features were applied to the interaction data gathered in Task 3 (Searching a target task) allowing the researchers to identify navigation patterns and profiles between the participants. In this case, a total number of 323 web pages were visited by participants in

this task. Note that U11 had to be excluded from the analysis due to fact that she decided to leave the experimental session before finishing the tasks. Applying the pre-processing algorithm defined in Section 3.4.1, the data of 133 web pages were selected to perform the cursor movement characterization 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 due to some detected problems in the PAm module when gathering interaction data when the cursor was out of the web browser window, a new web page was loaded, cursor position errors with *iframe* components in the page). Table 3 shows the total number of the visited pages by each participant and the total number of pages to be considered for the cursor movement characterization.

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

The information in Table 3 allows researchers to select the participants for further analysis. For instance, some of the participants (U04, U05, U06, U07 and U13) have fewer than 5 pages for analysing. Some parameters (curvature index, pointing time, clicking time) may not be accurate enough due to this lack of data. At this point, it is possible to select the participants to be excluded for the cursor movement characterization process. In this case, we excluded those with fewer than 5 pages to be analysed.

Some parameters were calculated based on the interaction data obtained in Task 3 for the rest of participants. Median values of cursor speed were automatically calculated for each participant. Fig. 6 shows the resulting values. There is a considerable difference in speed parameter between participants. Able-bodied participants using standard mouse and keyboard (U12,U14,U15,U16), obtained the highest values.

The curvature index parameter allows problematic cursor movement patterns to be detected. It measures the relation between the optimal path and the followed path. The calculated values are shown in Fig. 7. U02 has the highest value meaning that there is a lack of precision in cursor movements. Actually, he was a trackball user with low upper-body movement precision. Users using numeric keypad (U08,U09,U10) are the ones with the best values (1.09, 1.01 and 1.02 respectively). Actually, the use of keypad for moving the cursor produces linear and more precise movements.

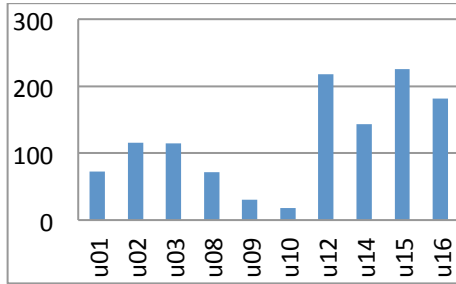


Fig. 6. Comparing chart about the speed parameter automatically obtained for each participant.

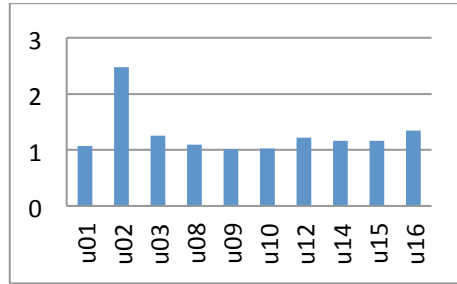


Fig. 7. Comparing chart about the curvature index automatically obtained for each participant

Other significant measures for characterizing cursor movements are the time required for pointing a target and the time required for clicking on a target. Both measures are considered for detecting problematic situations in cursor movements. For instance, a high value of time for clicking on a target may indicate that the user has problems trying to perform the click action due to the size of the target and lack of precision or some distracting content around the target. The algorithm presented in Section 3.4.1 is applied for calculating both measures for each participant. Fig. 8 shows the means of these measures for each participant automatically obtained by the RemoTest platform. It can be observed that participant U02 has the highest mean value for clicking on targets (3059ms) whereas U14 is the one with lowest mean value (536ms). U02 is one of the participants with physical disabilities using a trackball and U14 is an able-bodied participant. Able-bodied participants (U12,U14,U15,U16) obtained the better values for both measures (mean value for clicking is 855.32ms and mean value for pointing is 1931.41ms) than participants with physical impairments (mean value for clicking is 1872.53ms and mean value for pointing is 4351.27ms).

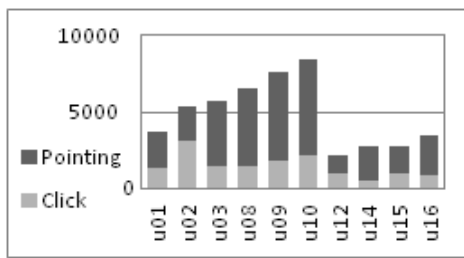


Fig. 8. Comparing chart about the time for pointing a target and time for clicking on a target for each participant in the experiment

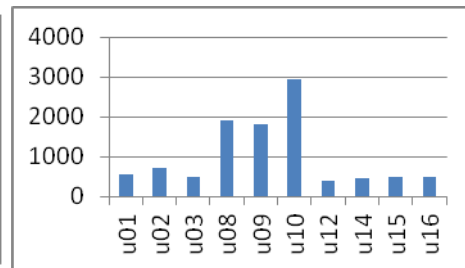


Fig. 9. Median values by user for motionless intervals along cursor trajectories

Recordings of participants with physical disabilities were manually analyzed and the time required for pointing to a target measure was annotated for each visited web page. The mean of automatically obtained values was compared with the mean of those obtained manually. These values can be found in Table 4. It can be observed that the values obtained automatically were generally lower than those manually gathered from the video analysis. Kendall's Concordance Tests was performed to analyse the agreement between rankings provided by both measures. The value obtained by the concordance coefficient was 0.73 ($p = 0.055$) meaning that there is some correlation between both rankings.

Table 4. Mean values of the time required for pointing to a target automatically obtained by RemoTest platform (APM) 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
MPM (ms)	6440	5080	3240	7190	8570	20480

Discussion

The results automatically obtained 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 clicking on a target assists 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, behaviour of participant U01 who uses a standard mouse differs in speed values for other participants using the same input device. It may be concluded that this participant requires some assistance by means of web interface adaptation mechanisms in order to improve his performance.

Observing the results in Fig. 9 it can be appreciated that participants U08, U09 and U10 obtained considerably higher values for their typical pauses between cursor sub-movements. These values may indicate some difficulties for starting the cursor movements as well as participants' fatigue during experimental sessions. In this case study, U10 is the one with the highest value (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 the 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 comparing with the values obtained manually. However, this could be due to lack of data used for automated analysis after the pre-processing. The current version of RemoTest platform will be more accurate in this sense as some of the errors when gathering coordinates of the cursor position have been fixed.

5 CONCLUSIONS

The RemoTest platform was conceived with a clear objective of assisting experimenters when dealing with the tedious task of carrying out remote/in-situ experiments with web systems. The functionalities included in the platform support experimenters throughout the entire process: designing the experiment, conducting experimental sessions, gathering interaction data and analyzing results.

The experiment definition features consider the specification of different kind of studies. For instance, user behaviour studies on concrete web tasks, comparative studies on navigational strategies when using different types of assistive technology, accessibility-in-use evaluations, web surveys for collecting specific information, user satisfaction measuring, user performance improvement analysis when applying adaptation techniques and so on.

Moreover, the RemoTest platform includes features for assisting with the analysis of interaction data recorded during experiments. A straightforward visualization of each participant interaction data in an understandable mode helps experimenters discovering valuable issues occurred during experiments at a glance, and saving 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 understandings within participants' behaviours. Future work will be focused on testing new performance measures to enrich users' characterization, searching for additional parameters that lead to better aimed movements estimations, and studying the application of the RemoTest tool for analysing interaction data of other groups of impaired users.

Acknowledgements. The authors gratefully acknowledge the support of the Spanish Ministry of Science and Innovation through Project ModelAccess (TIN2010-15549). The EGOKITUZ Research Laboratory is supported by the Department of Education, Universities and Research of the Basque Government (Grant# IT-395-10) and the University of the Basque Country (Grant# UFI11/45).

References

1. Almanji, A., Davies, T. C., and Stott, N.: Using cursor measures to investigate the effects of impairment severity on cursor control for youths with cerebral palsy. *Int. J. Hum.-Comput. Stud.* 72, 3 349-357 (2014)
2. Apaolaza, A., Harper, S., and Jay, C.: Understanding users in the wild. In : 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13). Article 13, 4 pa ACM, New York, (2013)
3. Atterer, R., Wnuk, M., and Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: 15th international conference on World Wide Web, pp. 203-212. ACM, New York, (2006)
4. Chapuis, O., Blanch, R., and Beaudouin-Lafon, M.: Fitts' law in the wild: A field study of aimed movements. *Tech. Rep.* 1480, LRI, Univ. Paris-Sud, France. 11 pages. (2007)

5. Claypool, M.L., P. Wased, M, and Brown, D.: Implicit interest indicators. In: 6th international conference on Intelligent user interfaces (IUI '01). pp. 33-40. ACM, New York (2001)
6. Cugini, J., & Scholtz, J.: VISVIP: 3D visualization of paths through web sites. In Database and Expert Systems Applications. In: 10th International Workshop on (pp. 259-263). IEEE (1999)
7. Edmonds, A.: Uzilla : a new tool for web usability testing. In: Behavior Research Methods, Instruments, & Computers., 35, 2, 194-20. (2003)
8. Etgen, M., Cantor, J.: What does getting WET (Web Event-logging Tool) Mean for Web Usability? In: 5th Conference on Human Factors & the Web, (1999)
9. Gajos, K. Z., Weld, D.S., and Wobbrock, J. O. Automatically generating personalized user interfaces with Supple. Artificial Intelligence, 174, 910-950, (2010)
10. Gajos, K., Reinecke, K., and Herrmann, C.: Accurate measurements of pointing performance from in situ observations. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). 3157-3166. ACM, New York (2012)
11. Google Analytics 2014, <http://www.google.es/analytics>
12. Hong, J., Heer, J., Waterson, S., and Landay J., A.: WebQuilt: A proxy-based approach to remote web usability testing. ACM Trans. Inf. Syst. 19, 3, 263-285. (2001)
13. HttpFox 2014, <https://addons.mozilla.org/es/firefox/addon/httpfox/>
14. Hurst, A., Hudson, S, E., Mankoff, J, Trewin, S., 2013. Distinguishing Users By Pointing Performance in Laboratory and Real-World Tasks. ACM Trans. Access. Comput. 5, 2, Article 5, 27 pages. (2013)
15. Loop11 2014, <http://www.loop11.com>
16. Meyer, D., Smith, J., Kornblum, S., Abrams, R., Wright., C.: Optimality in human motor performance: Ideal control of rapid aimed movements. Psych. Review, 95:340–370, (1988)
17. Morae 2014, <http://www.techsmith.com/morae.html>
18. Paganelli, L., and Paternò, F.: Intelligent analysis of user interactions with web applications. In: 7th international conference on Intelligent user interfaces (IUI '02). 111-118.3 ACM, New York, (2002)
19. Paternò, F., Mancini, C., Meniconi, S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In: IFIP TC13 Interantional Conference on Human-Computer Interaction. Steve Howard, Judy Hammond, and Gitte Lindgaard (Eds.). Chapman & Hall, Ltd., London, 362-369. UK (1997)
20. Pérez, J. E., Arrue, M., Valencia, X., Moreno, L.: Exploratory study of web navigation strategies for users with physical disabilities. In Proceedings of the 11th Web for All Conference ACM, New York, Article 20, 4 pages. (2014)
21. Power, C., Petrie, H., Mitchell, R., Stephanidis, C.: A Framework for Remote User Evaluation of Accessibility and Usability of Websites, Universal Access in Human-Computer Interaction. Addressing Diversity, Springer Berlin Heidelberg (2009)
22. Scholtz, J., Laskowski, S., Downey, L.: Developing usability tools and techniques for designing and testing web sites. In: 4th Conference on Human Factors & the Web. (1998)
23. Trewin, S., Physical Impairment. Chapter Book: Simon Harper, Yeliz Yesilada (Eds.): Web Accessibility - A Foundation for Research, pp. 37-46 Springer London (2008)