Databases and Information Systems VIII
H.-M. Haav et al. (Eds.)

doi:10.3233/978-1-61499-458-9-173

A Study on User Click Behaviour for WIS User Interface Improvements

Jevgeni MARENKOV[1], Tarmo ROBAL and Ahto KALJA
Dept. of Computer Engineering, Tallinn University of Technology, Estonia

Abstract. User behaviour prediction and analysis have become one of the most important approaches in the research of adaptive user interfaces (UI). Web personalisation, automatic UI adoption, recommender engines, flexible and responsive frameworks to user actions and user behaviour prediction provide alternative automatic solutions to enhance sophisticated user interfaces and serve as an alternative to conservative approaches in UI development. Nevertheless, research of unjust and improper user behaviour and its analysis in sophisticated user interfaces are left partly without needed attention. The objective of this paper is to focus on this kind of user behaviour and error analysis. In this paper we show that the rate of user mistakes while exploiting graphical UI depends on whether advanced UI development techniques like prototyping techniques and user tests were applied during UI development or they were disregarded. Moreover, the results demonstrate that graphical user interfaces without additional support for tablets and mobile devices have higher rate of user mistakes. For the latter, user tests were performed in order to analyse the importance and the scale of problem of clicking around UI elements. The paper also delivers solutions to user click misbehaviour problem to significantly decrease the rate of studied user interaction mistakes.

Keywords. user interface development, user click behaviour analysis, human-computer interaction

Introduction

One of the most important components in information systems, web applications, mobile and desktop applications is understandable, easy to use and comfortable graphical user interface (GUI) that satisfies end user requirements. Every user as an individual has his/her own preferences, habits and comprehension concerning user interfaces, making it a complex task to reach good results in developing a user interface (UI) that would satisfy each and every users' requirement over all possible users. Web personalisation, automatic UI adoption, recommender engines, flexible and responsive frameworks to user actions and user behaviour prediction are the main approaches applied for the task providing a replacement or substantial complement to conservative approaches in developing sophisticated responsive user interfaces.

During the recent years, user behaviour prediction and analysis has become one of the most important approaches in the research of user interfaces, including adaptive user interfaces (AUI). User behaviour prediction is used in behavioural targeting systems [1], recommender systems [2], click prediction algorithms [3], simulation of

[1] Corresponding Author.

user behaviour for testing user interface [4], and in user interest modelling [5]. Traditional user behaviour research is concentrated on user behaviour prediction, user-based profiling and user grouping − according to their behaviour patterns. On the other hand, there are studies that explore another branch of UI related issues – namely user experience and usability. Those studies measure user satisfaction with user experience [6] and try to accurately predict the quality of it [7]. As a result, the outcome is a collection of best practices and guidelines (for example guidelines for planned and impulsive users based on behaviour, cognition and emotion) that could be applied in developing sophisticated UI.

Commonly, graphical user interfaces are developed according to a previously established design, where various aspects of the system as well as its probable users are taken into account. One of these is the presumable behaviour of a probable user. Yet, one can never foresee the actual UI usage and how real users will exploit it after launching the system, introducing a gap between planned behaviour and the actual behaviour. The latter we call incorrect user behaviour.

Herein, we define an incorrect user behaviour as any kind of deviation in user behaviour that is not covered by the established design. Incorrect user behaviour may consist of many possible branches: incorrect behaviour in navigating within the system, incorrect UI element click events, and incorrect user actions in searching the information. In this research we mostly concentrate on missed clicks performed by users – clicks users assume that they performed on a clickable UI element, however no system response will follow, making users to define the situation as irresponsive system. As a result, users loose valuable browsing time while trying to understand whether they pressed on the element or not, and why there is no reaction on the system's side. Another good example is that user clicks multiple times on a particular element in anticipation of a system reaction, which may lead to problems such as multiple form submission, users losing their valuable time due to page reloads, and other browsing inconveniences. It is well-known that it is hard and practically impossible to re-educate users not to misbehave, therefore we concentrate on what and how could be done in order to make systems to recognize such situations and respond to users incorrect behaviour, and also learn from it.

In order to understand the scale of the aforementioned problems, user tests were performed starting from self-service systems with experienced users and moving to public systems, where tests were performed on different devices like PC's, and portable devices such as Tablets. Based on the received data and its further analyses, the problem and hypothesis were formulated and possible solutions with detailed analysis were proposed.

In this paper we are focusing on incorrect user behaviour analysis and possible errors which users make while using sophisticated UI systems. We will describe conducted user tests and analyse the importance and the scale of the problem of incorrect user behaviour in the light of miss-clicking around UI elements.

The rest of the paper is organized as follows. In Section 1 we discuss related works of the research area. Section 2 concentrates on the formulation of the problem and its consequences. Section 3 provides an overview of research environment, the technical solution established for data collection and presents results of user click-behaviour studies, while Section 4 proposes solutions and their analysis for the most critical problems. Finally, Section 5 draws conclusions and presents ideas for future research.

1. Related Works

A click or a tap, depending on the used device, is a method of communication between user and the front end of information system by the means of a graphical user interface. Due to the fact that it is the main method of communication between a system and its user, it is important to know and understand the "language" and the rules of communication between these two parties. User click studies are used in different areas, such as web personalisation, user behaviour and user click predicting and prediction of user navigation. In the following, each of these is discussed in more detail.

User clicks, as a characteristic of user interaction, have been studied as a measure of advertisement success. Cantú-Paz and Chenga [3] noticed that click probability is a factor to rank the ads in an appropriate order, filter out uninteresting ads, place the ads in different sections of the page, and to determine the price that will be charged to the advertiser if a click occurs. They created a model for personalised click prediction based on user demographic aspects, partitioning users into demographic segments based on age, gender, marriage status, interests, job status, and occupation. User profiling by demographic aspects is very effective approach in modelling user behaviour. Users can be profiled by locality [8], interests [9] or by combination of several parameters [10].

User click prediction is another brunch of studies for researchers. Personalisation is a solution exploited in sophisticated UI that makes the UI more user-specific based on many aspects like previous user activities, user interests, user demographic aspects and others [11], [12], [13], and [14]. Besides demographic aspects, there are other different critical aspects such particular user habits, previous experience, etc. that all affect users' decisions to click on one or another UI element. It is important for system owners to understand and to know what the most attractive UI section in their system is, how users exploit the system, and where they mainly perform clicks, in order to develop UI that is more responsive to user actions and more understandable to target users. To solve these problems, Zaragoza and Piwowarski [15] proposed a new framework for predicting user clicks based only on the formal algorithms and mathematic methods, and proposed models for predicting not only single clicks but clicks in multiple click sessions as well. Their model helps to predict the next user click in certain situations and they show that combined with other solutions the model could predict the user click with high accuracy (over 90%).

Besides marketing and advertising areas of research, there is another field of application of user clicks - user navigation research. These studies are concentrated on the users' path analysis [16], [17], [18], and [19]. This research branch covers structural patterns in user navigation, what strategies users follow while navigating, how the navigation structure affects user navigation behaviour strategies and patterns. Helic researched in his paper [20] user navigational behaviour as a complex process that is dependent on various factors, such as user background knowledge, current information need, information and navigation structures provided by system, network structure, etc. His research led to the conclusion that users are effective in navigation and that navigation is the process that consists of two phases: at the beginning users navigate from the periphery to the network core and in the second phase from the core to the final destination in the network periphery.

Sophisticated UIs and the UI issues were researched widely in [21], [22], and [23]. Various techniques and approaches are used for studying and solving problems of sophisticated UI. Thimbleby [24] researched the issues of UI using mathematical approaches. He showed that in some cases ordinary usability evaluation methods cannot reliably help in the development of UI, as some complex issues could be better discovered with a help of algebra and mathematic approaches. Rashid et al. [25] investigated how different configuration of input and output across displays could affect performance and preferences in map, text and photo search tasks.

Important part of UI development cycle is end user feedback analysis. Hedegaard's and Simonsen's [26] research is based on a major number of users reviews of products potentially containing information related to the usability. They concluded that UI may benefit from empirical studies of user-generated experience reports.

2. Problem Formulation

Generally, a user interface of a web information system (WIS) consists of different visual, navigable and clickable elements: buttons, hyperlinks, drop-down lists, combo boxes and other elements, such as design graphics etc. The design and style of concrete web page elements varies dependant on WIS architecture, target browsers and other parameters, e.g. cascading style sheets, but also on its users' habits.

On one hand, a lot of resources are invested into developing large scale WIS's, during which design and style of user interface and navigation structure is established. Different methods and approaches are used in designing and developing UI like user tests, prototyping, eye tracking studies, and prototyping. On the other hand, many web information systems are designed without massive and detailed UI tests, investments in prototyping, and full-fledged UI analysis. They are designed taking advantage of common patterns and practises. As a result, the quality of established UI can strongly vary, dependant on the approach applied during the process of development.

Users communicate with front-end mainly by moving the mouse device, finger, or by performing some actions, usually by means of clicking or tapping in case of mobile devices. Traditional UI research emphasizes the importance of user behaviour, click prediction, user-based profiling and user grouping. Therefore, the majority of research is focused on users' current behaviour with the purpose to predict their future behaviour.

The purpose of our research is to emphasize and pay more attention to incorrect user behaviour. In the context of our research we define incorrect behaviour as any behaviour when expected results of user action or a set of actions does not match its actual result. We discover incorrect user behaviour that is connected to user click behaviour, and perform user tests to analyse the importance and the scale of the problem. Our research covers the problems of clicking around UI elements, but not hitting the element itself, and the problems it raises. Incorrect user click behaviour could lead, for example to multiple form submission, lost or unsaved data, loosing of valuable browsing time or other inconveniences that could lead to no or unexpected response from the system. All these factors lead to a decrease of overall user performance and satisfaction. There could be different reasons, why users perform incorrect clicks e.g. sophisticated UI, not understandable target element style, blurred boundaries of UI elements, not thoroughly analysed positions of UI elements, inappropriate structure of navigation and so forth. Another aspect of the described problems is UI support for various types of devices with different screen resolutions (e.g. personal computers, tablets or mobile phones) and platforms (i.e. browsers) as it could be very difficult for users to use UI on mobile devices that is not properly developed for such devices. In that case the amount of incorrect user clicks could rise dramatically. Our aim is to show the extent of the problem of incorrect user clicks and to demonstrate and determine, how the problem importance varies dependant on devices (e.g. personal computers, tablets or mobile phones) that users exploit for communicating with WIS. To show the importance and the scale of the problem, we will perform user tests, collect data from users' interactions with system and analyse it.

3. Exploring User Click Behaviour

3.1. Capturing Users' Click Behaviour

As our purpose is to analyse user behaviour on client side, we cannot rely on widely used server side logging systems (e.g. web server logs) for that purpose as they log only server side requests; nor can we use any other server-side logging system. To study users' client side behavior (e.g. clicking on a design element) the interaction data must be collected also on client side. None of existing client side development frameworks (AngularJS, Backbone.js, JQuery) proposes a solution that would satisfy our requirements, i.e., to log user clicks on and around clickable UI elements and to save corresponding data (click coordinates, browser agent and others). As a result, we designed our own technical solution for client side event logging that satisfies our requirements and records all clicks users perform on clickable UI elements and around them. For this, special click-event listeners were added to clickable elements in both of the systems used for research (discussed in detail in Section 3.2).In case of a user incorrect behaviour, we wished to measure how close to the element user had clicked. Figure 1 outlines the general box model for clickable UI elements. A registered click can occur on clickable content inclusive the padding area around it together with border, constituting the clickable area. Thus, clicks outside the latter area are considered as incorrect user clicks, e.g. clicks in the margin area.

Figure 1. Box model and structure of a clickable UI element

In order to provide a starting point and a technical definition for incorrect user click, we analysed the UI structure of systems under study and concluded that an incorrect user click, in the context of our research, is a click around a desired clickable UI element in the region of 35 pixels. The area of 35 pixels was selected because mostly there are no other elements in that area. That is important, as other UI elements can affect the accuracy of our experiment.

The purpose of the established client side data collector was to handle user clicks on any clickable UI element or around it, and send the captured interaction data to server for storing into log database. The technical solution for collecting user interactions was based on the open source JQuery [27] JavaScript library, powerful in handling various client side events (e.g., click, hover, toggle, change and others), supporting various browsers (Internet Explorer, Chrome, Firefox, Safari, and Opera) through their different versions – thus providing the most flexible solution

The storage of collected data was implemented using the PostgreSQL database (version 9.2.6).The following data was captured and stored in the database for the purpose of user incorrect behaviour study: (a) remote hostname, (b) request URL, (c) username (to merely distinguish different users), (d) date and time of the request, (e) element unique id number, (f) element name, (g) click coordinates, (h) browser agent string, (i) click on UI element or around it – a flag detected by the script, and (j) click distance from clickable element in pixels.

3.2. Characterization of Web Information Systems Used for the Study

Two different web information systems were selected for the user incorrect behaviour study, each having a different strategy and approaches applied during their UI development. The first analysed system was a state self-service portal for entrepreneur documents system (EDS). The main purpose of the system is to provide services for entrepreneurs for submitting and managing law documents and for communication with Estonian e-government. EDS is a large scale web portal which UI was developed using massive UI tests, implementing prototyping techniques and with application target group analysis. The system UI design was developed following the W3C Web Content Accessibility Guidelines (WCAG).

The second analysed system was a web portal of an insurance company (ICS). Due to business reasons, we cannot outline the exact name herein at this point. The purpose of this system is to handle insurance offers and enable signing of contracts between the company and its clients via the system. The ICS is a relatively small WIS, and its UI was designed using UI designer expert knowledge and common patterns only.

The key difference in development of these two systems relies in the fact that EDS was designed by W3C WCAG guidelines and using real user tests, and implementing prototyping techniques, whereas ICS was designed based on best practices of UI development. These made the aforementioned WIS's suitable for user click behavior studies and testing the necessity of following WCAG. The fundamental difference between these systems UI development is also a strong reason why these systems were selected for our research.

3.3. Results of User Click Studies on EDS and ICS Web Information Systems

The purpose of the study was set to get for further analysis as much statistical data on user click-behaviour as possible, as every not strictly defined UI element or every wrongly chosen style could lead to incorrect user click-behaviour, which in turn could lead to erroneous actions taken by system, invoked by user.

User click-behaviour data, provided by actual system users, was collected during a period of 10 weeks at the end of 2013 from EDS and ICS systems. There were more than 70 unique participants for the EDS and about 65 participants for the ICS system, aged between 20 and 60 years. The data was collected from experienced and nonexperienced system users with different demographical criteria such as age and sex. During the test period, approximately 270 000 user clicks were gathered altogether from EDS and ICS WIS. Around 250 000 clicks were suitable for the analysis, the rest were disregarded due to data inconsistency (mainly due to issues with clickable element wrapper). The majority of clicks were collected from desktop devices (93%), and about 7% from mobile and tablet devices. The relative importance of collected data from the EDS and ICS WIS was respectively 56% and 44%.

The study revealed that only 3% of all collected clicks were classified as incorrect, which is reasonably low level. It can be taken as a threshold value for further evaluations. Another possible limitation is deliberate incorrect user behaviour. It could be defined as enormously high rate of incorrect clicks in limited UI section in a measured period of time. Performing the analysis of collected data no cases of deliberate user behaviour were detected. Therefore, the limitation is disregarded in further analysis. Although user demographic information was also captured, the analysis did not deliver any dependence on user age.

As was defined before (Section 1), the aim of our research is to point out the problem of incorrect user clicks, mainly for web sites and systems developed under conditions of limited resources for UI development. The results of first experiments confirm our suspicion that the problem of incorrect user behaviour importance is higher in the systems that were designed without high investments in UI. For confirmation we plan to launch a study involving different WIS's developed based on prior investments into UI study in comparison to systems established ad-hoc.

The analysis (Figure 2) showed the incorrect click rate for the EDS system, which was established according to WCAG, to be relatively and satisfactorily low (lower than 3%) for both desktop and mobile devices.

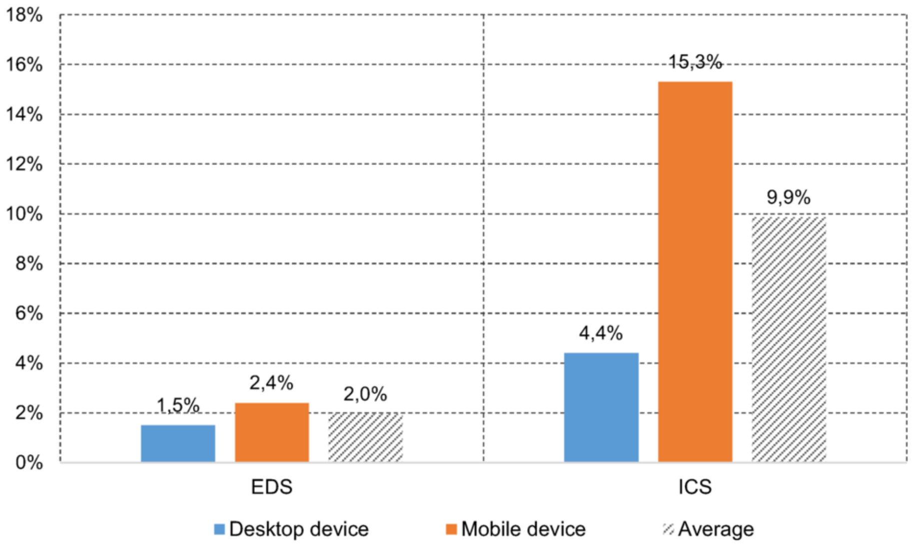

Figure 2. Incorrect click rate in the EDS and ICS WIS with calculated averages

However, for the ICS system UI, developed based only on best practices, the analysis indicated rather serious results of almost 3 times higher rate of incorrect clicks for desktop users and more than 6 times higher rate in the mobile device users category, compared to the EDS system. This clearly outlines the necessity of following UI development guidelines and the importance of testing and prototyping. It is to be noted that the ICS system UI was not developed to be used with mobile devices, which certainly has affected the results and is one of the reasons for such a high error rate (15%). The study results also showed that the average click rate for desktop devices is lower (2.8%) than in case of mobile devices (11.8%). In the following, let us consider a few examples of detected flaws in the UI design of the ICS system.

A high error-prone section identified in the ICS system with high rate of user incorrect click behaviour is depicted on Figure 3. Herein, we are dealing with a situation where clickable UI elements, namely icons, and the clickable regions around them are of size 16 x 16 pixels only, with a distance between them only 2 pixels. Both icons show high rate of mistakes. These icons are used to allow modifications to data presented in rows in a table of width 660 pixels. Obviously, the main reason of high user mistake rate relies on the size of the UI element. Hitting a space of 16x16 pixels needs good precision, especially on mobile devices, where users need to use their fingertips instead of a more precise mouse for performing operations. As a solution here to this problem we see to enlarge each UI element clickable area or icon size to 30 pixels, and also increase the distance between these elements to avoid misleading clicks.

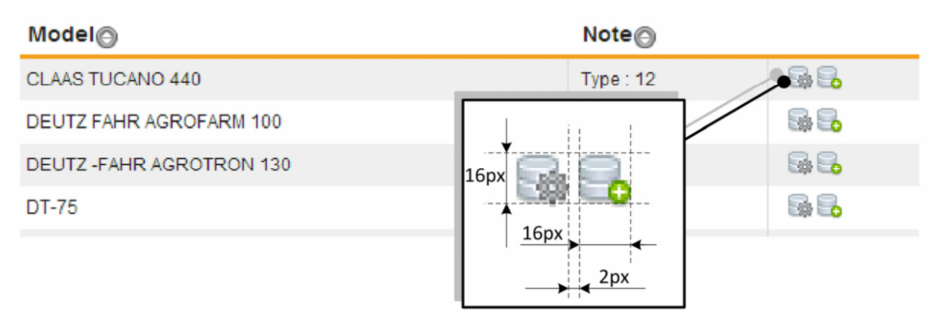

Figure 3. Example of UI elements that received a high rate of user incorrect click behaviour. Fragment of the ICS system UI

Figure 4 describes another example from the same system. The depicted clickable UI element also suffers from the problem of clickable area size. However, the analysis revealed yet another interesting finding. Do note the toggle button below the text 'Offer status' (Figure 4). As in-depth analysis revealed, the majority of users clicked instead of the toggle button on the label above it (Figure 4), believing this action to change the order of data representation. The error rate of user click-behaviour in this particular case was 8.1% on desktop and 34.5% on mobile devices. It is a clear example of poor UI design getting users confused. The most efficient fix could be to redesign that part of UI and make the label 'Offer status' clickable as well.

Similar situations to the aforementioned two examples, where users could not define the location of the clickable element, were found multiple times during the analysis for the ICS WIS.

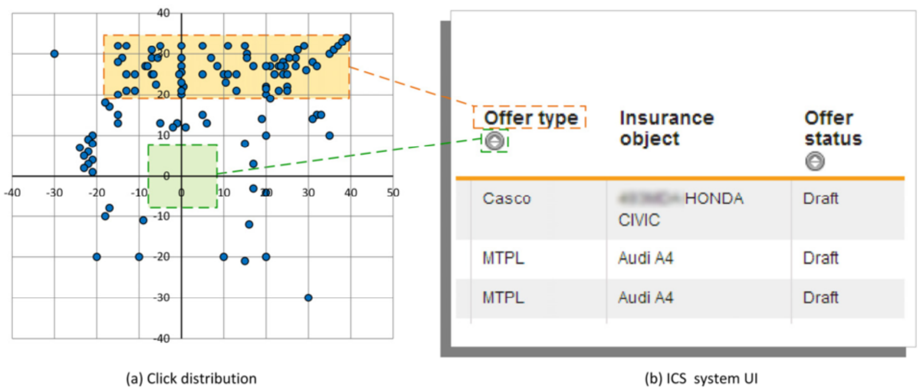

Figure 4. Excerpt from the ICS system UI (b) and click distribution (scatter diagram (a)) showing users clicking above the toggle button mainly on the label ‘Offer type’, which does not have any triggering action

The collected users’ interaction data showed that while exploiting WIS user interfaces that do not have specific UI implementation for tablet or mobile devices users make considerably more mistakes handling system UI and tapping elements than they do while using desktop computers and clicking. Obviously, a type of pointing device – mouse, pen, or fingertip – also plays a crucial role here.

4. Alleviating the Outlined Problems of User Clicking Behaviour

As shown, the results of our study emphasize the importance of well tested and analysed user interfaces in web information systems, and also the importance of supporting different device platforms in today’s mobile and technology-aware world. Regarding the latter, various approaches for designing WIS UI running on different devices exist. The most frequently exploited ones are creating separate UI for mobile and desktop platforms, and advantaging of responsive UI like Foundation 5 [28], Bootstrap [29] and HTML5 BOILERPLATE [30], where user interface adapts itself according to detected user device.

In case of the separate UI desing approach, devices receive targeted specific HTML templates, CSS (Cascading Style Sheet) style sheets and script files, depending on the media platform. There might be a desktop version, tablet version and also a separate mobile smart-phone UI version for a WIS. In practice it means developing and maintaining several different user interfaces. As a result, multiple UI maintenance needs additional resources, and surely increases costs as well. Also, users have to accustom to different UI designs on different devices.

In case of the responsive UI approach, all devices receive the same HTML templates with different CSS style sheets. Since the CSS version 2.1 standard, style sheets have the include ‘@media’ rule feature that defines how the document is rendered on different media based on available style declarations. This allows selection of suitable style according to client device screen resolution and type. The disadvantage of responsive UI approach however is the complexity of filtering of application content between the device types. This approach is not suitable if the functionality is dependent on a platform type. For instance, it is common to disable some system functionalities for mobile devices. Thereby, applying the responsive UI approach could lead to difficulties in system implementation.

4.1. Enlarging Element Clickable Area

One solution to the researched problem of incorrect user click-behaviour while using sophisticated WIS user interfaces might be enlarging the elements' clickable area. To prove the concept, we studied how this enlargement would affect system response to users' actions. Table 1 outlines the results of the click distance study. As can be seen, enlarging the region around a clickable UI element where a click/tap event is captured, would dramatically decrease incorrect click-behaviour rate. In case of desktop devices, an increase of clickable area up to 15px would decrease the rate of incorrect clicks by a 25%, and for mobile devices by 14%; the difference becomes even larger with extending clickable area further due to pointing precision issues. The trends appearing in Table 1 are evident – for mobile devices the larger the target area is, the better is the hit; whereas for desktop devices, where the main pointer is a mouse arrow, the rate of incorrect clicks begins to drop when the clickable region is enlarged over 25px.

Table 1. Overall rate of incorrect user clicks and its dependency on the distance from clickable UI element.

Distance from Clickable UI element	Desktop Devices	Mobile Devices
0 – 15 px	25%	14%
15 – 25 px	44%	37%
25 - 35 px	31%	49%

This proposed solution of enlarging clickable area around UI element may seem rather simple at first, however it buries a new problem, namely intersection of enlarged clickable areas. Figure 5 represents this situation, when it is impossible to accurately define the object of user click (in practice the click on an element with higher z-index will get its action triggered) and thereby system response may not be accurate.

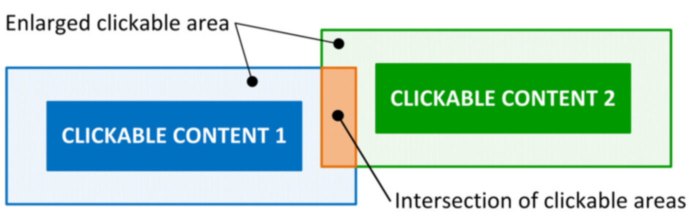

Figure 5. Intersection of clickable areas after extending UI element click region

As a solution to the intersection area problem, we see either to define that area as 'neutral', meaning clicks in that area do not trigger any events, or in case of overlapping clickable areas a confirmation dialog is invoked asking user to provide information on his/her intentions (Figure 6). The disadvantage of the aforementioned solution is the fact that user now needs to perform some additional operations. Also, this is not a common practice in UI exploitation, thereby users may not have previous experience with this kind of solution and can get confused of such confirmation dialogs.

The solution of 'neutral' area on the other hand is smoother in context of user interface exploitation and practically 'invisible' for users. The 'neutral' area does not affect proposed solutions, it disables predefined confirmation window solution in case of intersectional areas, and does not acquire any changes in user click behaviour. Still, we believe that altogether a solution for user incorrect behaviour would be to use a combination of the neutral area and action confirmation wisely together. Namely, if a system has identified user to make the same mistake repeatedly, it should provide help by asking about user intentions and memorize it for a period of time, e.g., session.

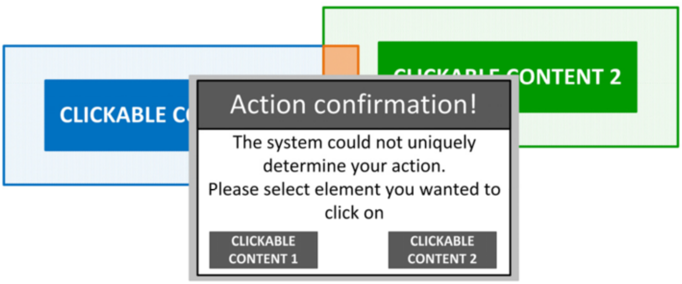

Figure 6. Confirmation dialog window appearing after user has clicked on intersection area

4.2. Using Web Storage for User Behaviour Correction

Another approach to the problem of clicking around UI elements could be to improve the UI implementation by exploiting a HTML 5 feature Web storage. Web storage is HTML API that provides the structure to store user data locally (in user browser) without additional load for the server side requests. Thus, this type of storage is being implemented only on client side and it could be used within the concrete browser. Web storage could not be shared among different browsers. From user perspective, it could however be considered also as a disadvantage of this approach.

Web storage consists of two implementations: local storage and session storage. The primary difference between these storage types is that the data stored within local storage does not expire; while session storage stores data only for a session.

The idea is to use Web storage for storing user reaction and decision in conflict situation. The conflict situation is a situation when user unwittingly clicks around the clickable element (Figure 7a). Generally, in case of conflict situation user receives no response from the system. After hitting the clickable element user reaction is being saved to the storage.

As a result, local storage contains various user reactions and corrections made after making incorrect action. Afterwards, if user hits around the clickable element the system requests the local storage for the similar situations. If similar situation is found, the system automatically corrects incorrect click based on the previous user reactions (Figure 7b).

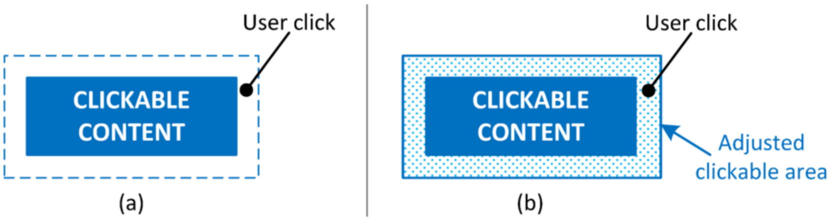

Figure 7. Concept of user behaviour correction: (a) incorrect user click remains uncaptured, and (b) subsequent to user action correction the click area is refined and user click captured

On the one hand, repository of user reactions is based on user previous incorrect behaviour and user's own corrections of his mistakes. It makes the collected data valid for current user and it is possible to apply previous behavioural patterns to further similar situations, although only within the limits of particular web browser

On the other hand, the solution of using web storage has some negative aspects. Web storage stores collected data on user's device. If the device is available to many parties under the same account the user personal data is available to many parties as well. Possible sharing of user personal data with other users could lead to erroneous system response and privacy issues. Another disadvantage of these solution is that Web storage is browser based solution. The data collected from user session in one browser cannot be shared not used within other browsers on device, due to security restrictions. In case user exploits several web browsers, the process of collecting data is not thoroughly efficient. Last but not the least, there is always possibility that user aimed to click another element and mistakenly clicked on wrong element. This could lead to stored data be compromised.

5. Conclusion and Future Work

In this paper, we have researched and presented the results of incorrect user behaviour study in sophisticated user interfaces in web information systems. Mainly, we researched the problem of incorrect user click-behaviour. The essence of the incorrect clicks relies on the fact that such clicks can lead to multiple form submission, users losing their valuable time, incorrect system response, and to other browsing inconveniences.

A secondary purpose of our study was to compare, how incorrect user behaviour differentiates on desktop and on portable devices such as tablets and mobile smart-phones. To analyse the main factors, that could cause the aforementioned problem and to measure the extent of the problem, we selected two different web information systems, varying by the amount of their users, approach applied during UI development, scale of the system, and the portability of UI to different platforms. We designed a client side logging system that captured user click-behaviour log for problem analysis. The developed user behaviour logger enabled us to track all user clicks in the area of clickable UI elements.

The collected click-behaviour log allowed us to evaluate users' actions on desktop and on portable mobile devices. The analyses revealed that firstly, there is a great difference in user experience and click-behaviour between systems developed following W3C Web Content Accessibility Guidelines (WCAG) and systems developed plainly on best practices without extensive testing, prototyping and analysis. The results were clearly in favour of standards and indicate the significance of following WCAG guidelines. Secondly, the size and position of any clickable UI element plays a crucial role in comprehending the user interface, and is even more important on mobile than on desktop devices, due to the pointing differences. Moreover, the results demonstrate that graphical user interfaces without additional support for tablets and mobile devices have a critically high rate of user incorrect behaviour while interacting with a system.

In addition we also analysed the general tendency of user click-behaviour errors and researched possible solutions to outlined problems. In practice, a combination of the two proposed solutions, i.e., web storage together with error detection and confirmation window could significantly decrease the rate of incorrect user click behaviour.

In terms of future research plans, there are several promising directions starting from a comparison of different approaches in developing user interfaces for mobile devices and investigating how each approach tries to handle incorrect user behaviour and what is the user reaction to that solution. Another promising further direction is the research of the problem of incorrect user clicks within people with disabilities. Moreover, it is necessary to analyse proposed solution and, if needed, to customise it to satisfy the requirements of UI for people with disabilities.

Acknowledgments

This research has been supported by IT Academy, Estonian Doctoral School in Information and Communication Technology (IKTDK), the European Regional Development Fund (ERDF), and partially by ERDF project No. 3.2.1201.13-0026 Model-based Java software development technology.

References

[1] Lavie T., Meyer J. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 2010, 68(8), 508–524.

[2] Nadi S. A Fuzzy Recommender System for Dynamic Prediction of User's Behavior. *International Conference for Internet Technology and Secured Transactions*. New York: IEEE Press, 2010. 1-5.

[3] Chen H., Cantu-Paz E. Personalized click prediction in sponsored search. *Proceedings. of the third ACM international conference on Web search and data mining (WSDM'10), 4-6 February, New York*. New York: ACM Press, 2010. 351-359.

[4] Roman P.E., Velasquez J.D. Stochastic Simulation of Web Users. *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 31 August-3 September, Canada*. Washington: IEEE, 2010. 30-33.

[5] Lam W., Mostafa J. Modeling User Interest Shift Using a Bayesian Approach. *Journal of the American Society for Information Science and Technology*, 2001, 52(5), 416-429.

[6] Law E.L.-C. The measurability and predictability of user experience. *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, *13-16 June, Italy*. ACM Press: New York, 2011. 1-10.

[7] Liu W., Lv L., Huang D., Zhang Y. The research on the user experience of E-commercial website based on user subdivision. *Proceedings of the 6th international conference on Advanced data mining and applications, 19-21 November, China*. Berlin: Springer, 2010. 529-536.

[8] Robal T., Kalja A. A model for users' action prediction based on locality profiles. *The Inter-Networked World: ISD Theory, Practice, and Education*, 2008, 169-182.

[9] Godoy D. Learning user interests for user profiling in personal information agents: Thesis. *AI Communications*, 2006, 19(4), 391 – 394.
[10] Tang J., Yao L., Zhang D., Zhang J. A combination approach to web user profling. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2010, 5(1), 1-44.
[11] Mulvenna M.D., Anand S.S., Buchener A.G. Personalization on the net using web mining. *Communications of ACM*, 2000, 43(8), 122-125.
[12] Hofgesang P. I. Web personalisation through incremental individual profling and support-based user segmentation. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2-5 November, Fremont*. Washington: IEEE, 2007. 213-220.
[13] Zeng Y., Hao H., Zhong N., Ren X., Wang Y. Ranking and combining social network data for web personalization. *Proceedings of the 2012 workshop on Data-driven user behavioral modelling and mining from social media, 29 October-2 November, Maui Hawaii.* New York: ACM Press, 2012. 15-18.
[14] Reda A., Cutrell E., Noble B. Towards improved web acceleration: leveraging the personal web. *Proceedings of the 5th ACM Workshop on Networked Systems for Developing Regions, 28 June, USA*. New York: ACM Press, 2011. 57-62.
[15] Piwowarski B., Zaragoza H. Predictive user click models based on click-through history. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 6-10 November, Portugal.* New York: ACM Press, 2007. 175-182.
[16] Karanam S., van Oostendorp H., Indurkhya B. Towards a Fully Computational Model of Web-Navigation. *24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent System, 28-1 July, USA*. Berlin: Springer, 2011. 327-337.
[17] Kiseleva J. Context mining and integration into predictive web analytics. *Proceedings of the 22nd international conference on World Wide Web companion, 13-17 May, Brazil.* Republic and Canton of Geneva: International World Wide Web Conferences Steering Committee, 2013. 383-388.
[18] Dankar F.K., Emam K.E. A theoretical model for obfuscating web navigation trails. *Proceedings of the Joint EDBT/ICDT 2013 Workshops, Italy.* New York: ACM Press, 2013. 137-144.
[19] Robal T., Kalja A. Learning from Users for a Better and Personalized Web Experience. *Proceedings of the PICMET '12 Conference "Technology Management for Emerging Technologies", 29 July-2 August, Canada.* Washington: IEEE, 2012. 2179 - 2188.
[20] Helic D. Analyzing User Click Paths in a Wikipedia Navigation Game. *Proceedings of the 35th International Convention, 21-25May, Opatija*. Washington: IEEE, 2012- 374 – 379.
[21] Lopes R., Gomes D., Carrico L. Web not for all: a large scale study of web accessibility. *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A), 26-27 April, USA.* New York: ACM Press, 2010.
[22] Mesbah A., Mirshokraie S. Automated analysis of CSS rules to support style maintenance. *Proceedings of the 2012 International Conference on Software Engineering, 2-9 June, Zurich.* Piscataway, NJ: IEEE, 2012. 408-418.
[23] Ehmke C., Wilson S. Identifying web usability problems from eye-tracking data. *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it*. UK: British Computer Society Swinton, 2007. 119-128.
[24] Thimbleby H. User Interface Design With Matrix Algebra. *ACM Transactions on Computer-Human Interaction,* 2004, 11(2), 181-236.
[25] Rashid U., Nacenta M.A., Quigley A. The cost of display switching: a comparison of mobile, large display and hybrid UI configurations. *Proceedings of the International Working Conference on Advanced Visual Interfaces, 21-25 May, Italy*. New York: ACM Press, 2012. 99-106.
[26] Hedegaard S., Simonsen J. G. Extracting usability and user experience information from online user reviews. *Proceeding CHI '13 Proceedings of the SIGHI Conference on Human Factors in Computing Systems, 27 April – 2 May, France.* New York: ACM Press, 2013. 2089-2098. Hedegaard S., Simonsen
[27] JQuery [WWW] http://jquery.com.
[28] Foundation | The Most Advanced Responsive Front-end Framework from ZURB [WWW] http://foundation.zurb.com.
[29] Bootstrap [WWW] http://getbootstrap.com.
[30] HTML5 Boilerplate: The web's most popular front-end template [WWW] http://html5boilerplate.com.