Tracking User Actions for the Web-Based Front End of ExplorViz

Proposal for a Bachelor's Thesis

Maria Kosche

May 29, 2013

Kiel University
Department of Computer Science
Software Engineering Group

Advised by: Prof. Dr. Wilhelm Hasselbring
M.Sc. Florian Fittkau

Contents

Chapter 1

Introduction

The development of applications makes it necessary to perform usability tests. Developers often have difficulties empathising with their users. Since they are intensively employed with their software, it is hard for them to imagine what problems inexperienced users can have. Therefore, it is very important to understand the user's position. Tracking technologies are a useful method for this purpose. They investigate the user's behaviour by recording user actions, especially the user's mouse and keyboard interaction. It can also be used for logging the time it takes to perform certain tasks. Collecting this data helps to understand the user's view. The information can be used for locating usability problems and subsequently it is possible to optimise the structure and functionality of an application based on tracking data.

This raises the question what tracking methods and tools already exist, how they work, to what degree they are compatible with existing software, and how to integrate a new or given tracking method or tool into an existing application. While the thesis will shed light on these issues, this proposal introduces the relevant subjects.

1.1 Document Structure

Chapter 2 describes the goals of the thesis. Thereafter Chapter 3 gives an overview of the foundations and technologies, which will be investigated and used for the thesis. Chapter 4 clarifies the approach and after sketching the organisation in chapter 5, finally the planned schedule and work packages are described in chapter 6.

Chapter 2

Goals

The goal of the thesis is to implement a method to track user actions in ExplorViz.

G1: Identification of Tracking Methods and Tools in the Context of Web Sites

Therefor, we will identify existing tracking methods and tools. As mentioned in Section 3.1 Google Analytics and Piwik are two examples for such tools. Section 3.1 already indicates some tracking methods like recording mouse and keyboard input or logging the time spent on certain tasks. We will focus on the way of logging tracking data. A challenge will be the tracking of interaction with WebGL. This step is completed when at least five tracking methods and tools are identified.

G2: Evaluation of the Identified Tracking Methods and Tools

After the identification of methods and tools, it is necessary to evaluate them. The identified methods and tools will be compared and evaluated to find a convenient solution. For this purpose, evaluation criteria will be elaborated and applied to all identified tracking methods and tools.

G3: Implementation of a Method to Track User Actions in ExplorViz

The last step is the implementation itself. A tracking method or tool to track user actions will be integrated in ExplorViz. Furthermore, generating a log file containing a time stamp, the user, the action, and additional fields will be possible. An optional goal will be the generating of markov chains from the log files.

Chapter 3

Foundations and Technologies

This chapter gives an overview of foundations and technologies, which are used in the thesis. This can only be seen as a survey and each section will be investigated in more detail in the thesis.

3.1 Tracking

Tracking and especially web tracking is used to collect, store and connect user behavior records. Tracking is often associated with advertisement companies, which actively collect information such as age, sex and place of residence about users and accumulate it in user profiles. These profiles are used to show users individualised advertisements instead of random ads. Aside from this, tracking is also used for law enforcement. Tracking technologies provide spying on individuals and solving crimes such as identity theft and credit card fraud. Moreover a further major motivation for tracking are usability tests of applications. "By observing the steps an individual performs while trying to solve a certain task [...], usability problems can be discovered and fixed." [Schmücker 2011, p.1] There are possibilities to capture detailed records of user mouse and keyboard input. Recording cursor movements paths can be helpful for identifying problems about locating or using certain functionality. Furthermore, this data can be extended with a time component, so that tracking can be used to analyse how long certain tasks take, which tasks cause the main problems and in which order the user proceeds. In addition "web analytics, a related field, focusses less on the individual user, but more on the performance of a web site as a whole" [Schmücker 2011, p.1]. Tracking can help improve the structure of a web site by listing for example the number of visitors over time, the time visitors spend on one site and which pages they look at. [Schmücker 2011, section I. - III.] The most popular tools used for web tracking are Google Analytics and Piwik (Open Source), which will be investigated in more detail in the thesis. [Schmücker 2011]

3.2 Monitoring

Tracking is usally used for observing and recording user actions and user behaviour. On the contrary monitoring is used for observing the performance of an application.

3.2.1 Kieker

Kieker is a tool, which provides Application Performance Monitoring (APM). It therefor collects and analyses monitoring data. It focuses on the application runtime behaviour. On the application- and service-level the operation response times, the user sessions and traces are investigated. On the system-level the collected data can include the CPU utilisation, memory usage etc. [van Hoorn et al. 2009; Rohr et al. 2008]

3.3 Experiments

Tracking user actions can be seen as a measurement method for experiments. There are different modalities like explanation of functionality or even the mood of the user. One has to deal with experiments for analysing the results of tracking.

An experiment is "a test under controlled conditions made to either demonstrate a known truth, examine the validity of a hypothesis, or determine the efficacy of something previously untried." [Shadish et al. 2002, p.1] Crucial here is the relationship between a cause and an effect, called causal relationship. The cause is "the producer of an effect, result, or consequence." [Shadish et al. 2002, p.1] The effect is the difference between what happened under the experimental condition as opposed to what would have happened without the experimental condition (called counterfactual). However, it is impossible to observe the counterfactual, because there is no facility for a given subject to both receive and not receive a treatment. [Shadish et al. 2002]

3.3.1 Threats to Validity

Researchers aspire to attain generalised causal knowledge, but causal generalisation conflicts with the fact that experiments have a very specific context. Each experiment consists of subjects, of the treatments themselves, of observations made on the subjects, and of the settings in which the experiment is conducted. Shadish defines 4 validity types, namely, internal validity, statistical conclusion validity, external validity, and construct validity.

Internal validity is closely related to statistical conclusion validity. Both focus on the relationship between treatment and conclusion. But since we are more interested in investigating causal-reasoning errors than in statistical inferences, statistical conclusion validity is not described further. External validity and construct validity are related as well, since they are both generalisations. Construct validity, however, focuses on questions about the definition of constructs to be measured, or about the units of measurement. These will already be answered by the identification and evaluation of tracking methods and tools. Therefore, only external validity will be described in more detail. [Shadish et al. 2002]

Internal Validity

Internal validity refers to the degree to which a causal conclusion is warranted. [Johnson 1997] Below there are given some reasons for questioning the internal validity, but there are only named threats standing in connection with web tracking.

Ambiguous temporal precedence There is uncertainty which variable is the cause and which is the effect.

Selection Differences between subject groups which may interact with the observed effect.

History A third variable may have influence on the observed effect.

Testing Repeatedly testing may influence the results, because practice and familiarity are relevant mechanisms.

External Validity

External validity refers to the degree to which experiments respectively their results are generalisable.

Interaction of the Causal Relationship with Subjects Subjects may have properties that interact with variables to analyse.

Interaction of the Causal Relationship Over Treatment Variations Different treatments may cause different effects.

Interaction of the Causal Relationship with Settings Environmental factors such as time, location etc. may have effect to results.

Context-Dependent Mediation If the explanation differs from time to time, the results may also differ.

3.4 WebGL

WebGL stands for Web Graphics Library and is a JavaScript API used to display 3D graphics in a web browser. It uses the OpenGL shading language GLSL and is based on OpenGl ES 2.0 (ES stands for Embedded Systems), which is the OpenGL specification version targeted at handheld and embedded devices such as cell phones, PDAs, consoles and vehicles. In general, OpenGL is a cross-platform standard 3D API for advanced 3D graphics. It is widely accepted and is used by games such as Half-Life, GLTron, Portal, Minecraft, StarCraft II and World of Goo. Furthermore, applications such as Blender or Google Earth include an OpenGL renderer. [Munshi et al. 2008; Shreiner et al. 2009]

As it became important to provide web-based and real-time rendering, WebGL was specified and it opened up possibilities for web-based 3D environments in web browsers

without any plug-in components. According to the distribution of browsers web-based 3D applications are also available for smart phones, tablets etc. WebGL also offers an integration with HTML content, so that, for instance, the interaction with other HTML elements and the use of standard HTML event handling mechanisms are available. Further, it is possible to use WebGL for 2D graphics as well. [*WebGL puplic wiki*; Cantor 2012; Anttonen and Salminen 2011; Seidelin 2011]

3.5 GWT

GWT stands for Google Web Toolkit and is a toolkit for developing web applications. It makes a development available, which enables the implementation of AJAX applications. Its characteristic is a Java to JavaScript compiler, so it allows developers to create complex JavaSript front-end applications in Java.

AJAX is an acronym for "Asynchronous JavaScript and XML" and describes a concept of transmitting data between client and server asynchronously. In this way it is possible to submit http requests and change a html site without reloading it. AJAX is platform independent and can be used by every browser, which supports JavaScript. Besides there is no need of installing a plug-in for using AJAX applications.

The GWT allows the developer to leverage his knowledge of Java and to use approved development environments like Eclipse. Furthermore, the GWT comes with different panel layouts and many widgets, which can be combined to a graphical user interface. Thereby creating a GUI is similar to creating it with Swing or SWT and the application gets more desktop-like. [Chaganti 2007; *Google Web Toolkit*]

3.6 Xtend

Xtend is a programming language, which syntax is similar to Java but provides more language elements like clojures. In fact it compiles into Java source code. It is still object-oriented and imperative, but also integrates some features known from functional programming languages. For example, Xtend supports lambda expressions, which can be used for implementing anonymous classes. Programmers, who are already familiar with Java, are able to leverage their knowledge and there is no big deal to learn Xtend. Some syntactical simplifications are made such as unmatched type inference and leaving out semicolons and empty parentheses. Finally Xtend is completely compatible to Java while being more readable and expressive. [*Xtend - Modernized Java: Documentation*; Lübbe 2012]

3.7 ExplorViz

ExplorViz is an online trace visualisation for large software landscapes, which can arise from the high number of software systems one company uses. It is intended as a support

for software engineer to comprehend software systems, for example, when creating new features. Therefore, it provides the functionality to disclose details such as the communication between programs, and the control flow in an application. Using an adequate and comprehensible abstraction for visualising, ExplorViz allows to structure the relevant information. Besides, performance is an important aspect, so the user experience is not restricted, while processing a huge trace amount. There is also attached great importance to the extensibility to enable other projects to visualise their data through a plug-in architecture.

The described technologies from Section 3.4 - Section 3.6), namely, WebGL, GWT, and Xtend are used for the development of the ExplorViz website. Thus, the integration of a tracking method or tool will require the use of these same technologies. [Fittkau 2013]

Chapter 4

Envisioned Approach

This section explicates our approach in more detail. Figure 4.1 shows an graphical overview of our approach. The following sections describe the different steps in more detail.

Section 4.1 to 4.4 describe the respective step.

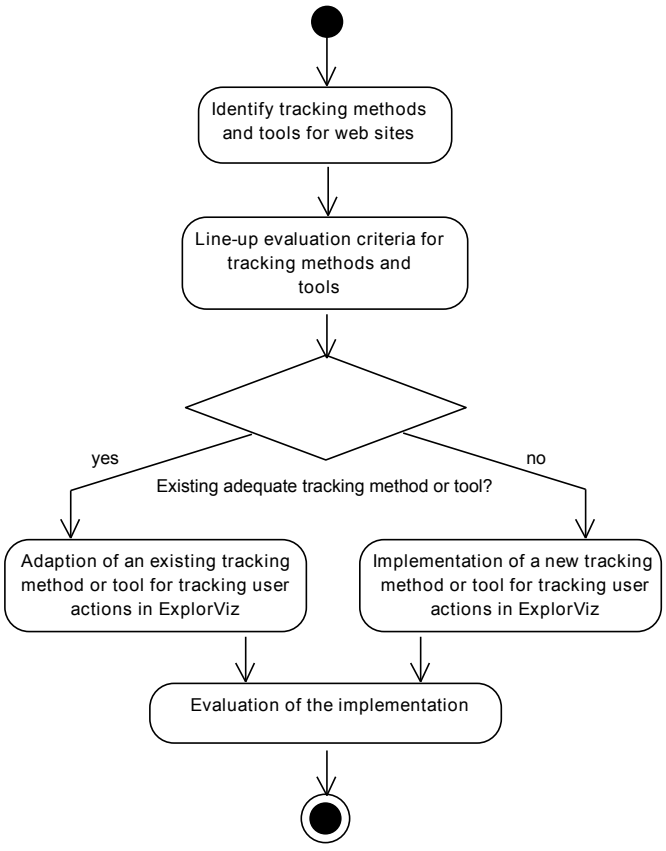

Figure 4.1. Graphical overview of the steps

4.1 Identify Tracking Methods and Tools for Web Sites

There exist several tracking methods which will be identified. For implementing a tracking method in ExplorViz, it is necessary to first have a look at existing methods and tools. There are several approaches for using tracking methods. This includes, for instance, both recording the user's mouse and keyboard interaction and recording the time an user needs for performing a task. In the thesis the tracking methods will be elaborated and existing tools will be investigated in regard to their functionality. Thereby the focus will be on web tracking.

4.2 Line-up Evaluation Criteria for Tracking Methods and Tools

The approaches from Section 4.1 will be compared and evaluated. Important criteria are the compatibility to the existing source code, the possibility of logging, and the requirement to track WebGL as well. The weak points of the selected methods and tools will be exposed and the considered methods and tools will be adapted to the requirements in agreement with the advisor.

4.3 Implementation of a New or Given Tracking Method or Tool for Tracking user actions in ExplorViz

This step depends on the outcome of Section 4.2 and is about integrating the resulting method or tool into ExplorViz. Therefor, the existing software will be analysed and an design for the software to create is made.

The implementation will be done with Java and Xtend. Furthermore, it must be clarified in which scope aspect-oriented programming has to be used. In this case an extension for aspect-oriented programming such as AspectJ is needed.

During the implementation the software will be documented and persistently tested for correctness.

4.4 Evaluate the Implementation

The last step is the evaluation of the developed software, which mainly refers to the correctness and usability. Furthermore, the logged data have to be comprehensible and analysable. Exact quality characteristics still need to be defined.

Chapter 5

Organisation

- Work packages: Described in Chapter 6
- Duration: One semester (2013-04-01 to 2013-09-30)
- Author: Maria Kosche
- Advisors: Prof. Dr. Wilhelm Hasselbring and M.Sc. Florian Fittkau
- Language: English
- Feedback: After each step of the approach, a feedback from the advisor is carried out. Thereupon, the feedback flows into the bachelor's thesis.

Chapter 6

Schedule

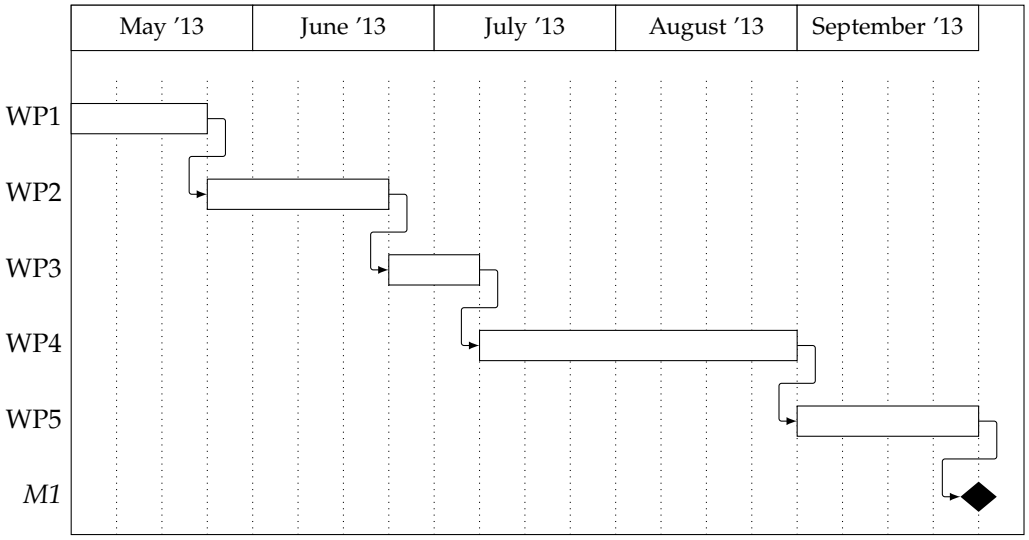

Figure 6.1. Schedule for the Bachelor's Thesis

6.1 Description of the Work Packages

6.1.1 WP1: Foundations

In this work package the foundations and technologies from Chapter 3 will be described more extensively. Especially the search for tools and methods takes place in this package.

6.1.2 WP2: Evaluation of Existing Tracking Methods and Tools

Foundations will be evaluated including the comparison of the different tracking methods and tools with regard to the specified criteria.

6.1.3 WP3: Specify or Choose Tracking Method or Tool for Implementation

In agreement with the advisor a tracking method or tool for the implementation is chosen. If no applicable tool or method is found, we will specify one.

6.1.4 WP4: Implementation

This work package includes the integration of the tracking functionality into ExplorViz. The specifications are iteratively updated according to a frequent feedback from the advisor.

6.1.5 WP5: Evaluation of Software

In this last work package the developed software will be evaluated in consideration of the specified quality characteristics and tested with a typical input sequence in ExplorViz.

6.1.6 Milestone 1: Submission

Finally the bachelor's thesis is revised and ready for submission.

Bibliography

[Anttonen and Salminen 2011] M. Anttonen and A. Salminen. Building 3D WebGL Applications. Technical report. Tampereen teknillinen yliopisto, 2011. (Cited on page 8)

[Cantor 2012] D. Cantor. WebGL Beginner's Guide. Packt Pub Limited, 2012. (Cited on page 8)

[Chaganti 2007] P. Chaganti. Google Web Toolkit - GWT Java Ajax Programming: A Practical Guide to Google Web Toolkit for Creating AJAX Applications with Java. Packt Pub Limited, 2007. (Cited on page 8)

[Fittkau 2013] F. Fittkau. Online Trace Visualization for System and Program Comprehension in Large Software Landscapes - PhD Thesis Proposal. PhD thesis. Kiel University, 2013. (Cited on page 9)

[Johnson 1997] R. B. Johnson. Examining the validity structure of qualitative research. 1997. (Cited on page 7)

[Lübbe 2012] K. Y. Lübbe. Improving a Transformation of Java Models to KDM. PhD thesis. Kiel University, 2012. (Cited on page 8)

[Munshi et al. 2008] A. Munshi, D. Ginsburg, and D. Shreiner. OpenGL ES 2.0 programming guide. Addison-Wesley Professional, 2008. (Cited on page 7)

[Rohr et al. 2008] M. Rohr, A. van Hoorn, J. Matevska, N. Sommer, L. Stoever, S. Giesecke, and W. Hasselbring. Kieker: continuous monitoring and on demand visualization of java software behavior (2008). (Cited on page 6)

[Schmücker 2011] N. Schmücker. Web Tracking. Available online: `http://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS11/snet-project/web-tracking_schmuecker.pdf` (last accessed on 2013-04-25). 2011. (Cited on page 5)

[Seidelin 2011] J. Seidelin. HTML5 Games: Creating Fun with HTML5, CSS3, and WebGL. Wiley, 2011. (Cited on page 8)

[Shadish et al. 2002] W. R. Shadish, T. D. Cook, and D. T. Campbell. Experimental and quasi-experimental designs for generalized causal inference. In: Houghton Mifflon Company, 2002. Chapter 1, 14. (Cited on page 6)

[Shreiner et al. 2009] D. Shreiner et al. OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1. Addison-Wesley Professional, 2009. (Cited on page 7)

[Van Hoorn et al. 2009] A. van Hoorn, M. Rohr, W. Hasselbring, J. Waller, J. Ehlers, S. Frey, and D. Kieselhorst. Continuous monitoring of software services: design and application of the kieker framework (2009). (Cited on page 6)

Bibliography

[*Google Web Toolkit*] S. Wargolet. Google Web Toolkit. (Cited on page 8)

[*WebGL puplic wiki*] WebGL puplic wiki. Available online: `http://www.khronos.org/webgl/wiki/Main_Page` (last accessed on 2013-04-25). (Cited on page 8)

[*Xtend - Modernized Java: Documentation*] Xtend - Modernized Java: Documentation. Available online: `http://www.eclipse.org/xtend/` (last accessed on 2013-04-25). (Cited on page 8)