

Case Studies, Automatic Reporting, Evaluation and Optimization of Usability using Advanced User Behaviour Analysis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

David Neukam

Matrikelnummer 0826601

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.Prof. Mag.rer.nat. Dr.techn. Reinhard Pichler
Mitwirkung: Dipl.-Ing. Bernhard Kruepl-Sypien

Wien, 25.07.2013

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Case Studies, Automatic Reporting, Evaluation and Optimization of Usability using Advanced User Behaviour Analysis

MASTER'S THESIS

submitted in partial fulfilment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

David Neukam

Registration Number 0826601

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ.Prof. Mag.rer.nat. Dr.techn. Reinhard Pichler
Assistance: Dipl.-Ing. Bernhard Kruepl-Sypien

Vienna, 25.07.2013

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Acknowledgements As this work was dependent on user studies, only the support by Univ. Lektor Robert Baumgartner and the students of his course *Applied Web Data Extraction and Integration* (AEXIN) Summer Term 2013 rendered it possible to create this thesis in this way. Without the participation of these students, it would have been more difficult to convince a sufficient number of people to contribute to this study. Great thanks also to my supervisor Prof. Dr. Reinhard Pichler, Bernhard Krüpl-Sypien, Christoph Herzog and Ruslan Fayzrakhmanov for supporting this work with feedback and creative ideas. Also, I have to thank my study colleges Maximilian Aster and Sebastian Morgenbesser for this great teamwork on implementation of common components.

Abstract

Current approaches to track user behaviour in order to improve web pages are mainly focused on desktop users and do not consider the special group of mobile users. Because of the increasing domination of smartphones in the World Wide Web these days, it is very important to analyse and optimize web pages according to those devices. *Advanced User Behaviour Analysis* tries to improve the effectiveness and usability of web pages in general and for specific user groups, using information that is gained from tracking users utilizing mobile devices. The term *Advanced User Behaviour Analysis* points out the fact that new approaches are evaluated and used in order to achieve this goal. The main focus is to use mobile devices to track typical user interactions that can be archived and further interpreted, in order to analyse or adapt web pages. The key concept of this project is the limited viewport (visible area) of a website on a mobile device acting as a surrogate for the attention of a user. By tracking this small area, it can be reconstructed what a user could have read.

With this approach, studies are made which show possibilities of usage of such a tool. Also a prototype (see chapter 6) of an analysis-portal was created that can be used to analyze the recorded data. This tool was used to analyze the data from a study (see chapter 7) done with students of the course *Applied Web Data Extraction and Integration*. Metrics and indicators for web usability are identified that are not available with traditional 'inter-page-level' approaches like Google Analytics focusing on the navigation path between web pages. The analysis shows how people behave differently when accessing a desktop or mobile version of a website with their smartphone and which differences and similarities can be found when comparing different websites of one domain to each other. Some key performance indicators are theoretically discussed and some were implemented and their results are shown.

Kurzfassung

Aktuelle Herangehensweisen das Nutzerverhalten aufzuzeichnen um Webseiten zu verbessern zielen hauptsächlich auf die Benutzer von stationären Computern ab und berücksichtigen Nutzer mobiler Geräte nicht in ausreichender Weise. Aufgrund der Tatsache, dass die Verbreitung von Smartphones als Endgerät zur Nutzung des Internets signifikant ansteigt ist es notwendig geworden, Webseiten auch auf ihre Verwendbarkeit mit diesen Geräten hin zu analysieren und zu optimieren. *Advanced User Behaviour Analysis* versucht die Benutzerfreundlichkeit von Webseiten mit Hilfe der Daten die von diesen Geräten aufgezeichnet wurden zu optimieren. Hierzu bedarf es neuer Herangehensweisen die evaluiert und verwendet werden um dieses Ziel zu erreichen. Mobile Geräte werden verwendet um Benutzerinteraktionen aufzuzeichnen und weitergehend zu analysieren um Webseiten zu optimieren. Das Kernkonzept hierbei bildet der begrenzte sichtbare Bereich einer Webseite auf einem Smartphone welcher als Substitut für die Aufmerksamkeit eines Nutzers dient. Durch Erfassung dieses Bereiches kann eingegrenzt werden, welcher Bereich die Aufmerksamkeit eines Nutzers hatte.

Mittels dieses Projektes wurde eine Studie durchgeführt die die Möglichkeiten aufzeigen soll, die sich durch diesen Ansatz ergeben. Hierzu wurde ein Prototyp (siehe Kapitel 6) einer Auswertungsapplikation entwickelt der dazu verwendet wurde, die aufgezeichneten Daten der Studie (siehe Kapitel 7) durchgeführt mit Studenten der Lehrveranstaltung *Applied Web Data Extraction and Integration* zu analysieren. Kennwerte und Kennzeichen für Benutzerfreundlichkeit von Webseiten wurden identifiziert die mit herkömmlichen Herangehensweisen, die nur auf der Ebene von Webseitenbesuchen und den Pfaden zwischen Webseiten auswerten, nicht aber die Daten innerhalb einer Webseite verwenden wie beispielsweise Google Analytics nicht machbar sind. Beispielsweise wird das Verhalten von Personen beim Surfen auf einer mobilen Webseite mit dem Verhalten auf einer normalen Webseite bei Verwendung des eigenen Smartphones verglichen und Unterschiede und Gemeinsamkeiten werden aufgezeigt die auftreten, wenn verschiedene Webseiten eines Gebietes miteinander verglichen werden. Einige Kennzahlen werden hierfür theoretisch behandelt wohingegen andere auch praktisch implementiert wurden und deren Ergebnisse exemplarisch aufgeführt werden.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem statement	2
1.3	Aim of the work	2
1.4	Authorship	3
1.5	Methodology	5
1.6	Related project work	7
1.7	Relevance for Business Informatics	7
1.8	Structure of the work	7
2	Advanced User Behaviour Analysis	9
2.1	Introduction	9
2.2	Aim	9
2.3	Existing approaches	10
2.3.1	Eye Tracking	10
2.3.2	Mouse Tracking	10
2.4	Heatmaps	11
2.5	Outcome	13
2.5.1	Case Studies	13
2.5.2	Web Interaction Archiving	13
2.5.3	Content and process repackaging	13
3	Mobile Viewport Tracking	15
3.1	Introduction	15
3.2	Technical Basics	16
3.3	Data Model	20
3.4	Components	23
3.4.1	Architecture	23
3.4.2	Client-side script	23
3.4.3	Proxy Server	23
3.4.4	Web Service	24
3.4.5	Database Layer	24
3.5	Business Logic - Concepts	25

3.5.1	Client-side scripting	25
3.5.2	Action Type Classification	26
3.5.3	Important Content Identification	27
3.5.4	Heatmap Drawing	30
3.5.5	User Tracking	33
3.6	Possible integration into other projects	36
4	State of the Art	37
4.1	Web Usability	38
4.1.1	Important Terms for Web Usability	39
4.1.2	Web Usability Evaluation	39
4.2	Eye-Tracking	43
4.2.1	Important terms	44
4.2.2	Perception	45
4.2.3	Metrics and Characteristics	45
4.2.4	Problems	47
4.2.5	Eye-Tracking Web Analytics	48
4.3	Mouse-Tracking	48
4.3.1	Mouse- / Eye-correlation	49
4.3.2	Identified Movement patterns	50
4.3.3	Distraction	50
4.3.4	Existing / Commercial Products	51
4.4	Scrolltracking	56
4.5	Viewport Tracking on the Desktop	56
5	Setup of Mobile Viewport Tracking	59
5.1	Client Side Configuration	59
5.2	Server Side Configuration	61
5.3	Build and Run Mobile Viewport Tracking	64
6	Implementation of an Analysis Application	67
6.1	The Need of such an Application	67
6.2	Technical Background / Implementation	67
6.3	Features	68
7	Evaluation	73
7.1	Data source	73
7.1.1	AEXIN Task description	73
7.1.2	Data Amount and Quality	75
7.2	Studies	77
7.2.1	Intention of user	77
7.2.2	News Websites	80
7.2.3	Mobile Version vs. Desktop Version	83
7.2.4	Amazon Article Pages	85

7.2.5	Weather Pages	86
7.2.6	Orientation	86
7.3	Key Performance Indicators	87
7.3.1	Scroll Depth	87
7.3.2	Horizontal Scroll Depth	88
7.3.3	Visit Duration on Different Areas	89
7.3.4	Basic Statistical Data of Importance Matrix	90
7.3.5	Interaction Sequence of Zooms/Scrolls	90
7.3.6	The Sequence of Interactions	90
7.3.7	Homogeneity of the Distribution of Weights	91
7.3.8	Interactivity of a Web Page	92
7.3.9	Problems finding the right Zoom Level	92
7.3.10	Average Viewport Entry Count	93
7.3.11	Percentage of Horizontal Scrolls	93
7.3.12	Action Type	93
7.3.13	Other possible KPIs	94
8	Conclusion	95
8.1	Future Work	98
8.1.1	Other platforms	98
8.1.2	Use recorded data from Proxy	98
8.1.3	DOM-events / AJAX-aware	98
8.1.4	Include in websites as beacon script	98
8.1.5	Improve action classification	99
8.1.6	Transparent Proxy	99
8.1.7	Reconsider the concept of predefined heatmaps	99
8.1.8	Improve features of the analysis web application	99
8.1.9	Web page variant comparison	99
8.1.10	Evaluation of derived KPIs	99
8.1.11	Evaluation using eye-tracking system	100
8.1.12	Commercial usage of the project	100
A	Appendix	101
A.1	Mobile Viewport Tracking Sources	101
A.2	Heatmaps	122
	Bibliography	131
	List of Figures	139
	Abbreviations	143

Introduction

1.1 Motivation

The ultimate motivation for this project and the resulting theses is the increasing domination of smartphones in the World Wide Web these days. The rising capabilities of these modern devices enable their user base a lot of functions and possibilities to perform every day actions, such as booking flights, reading the news, looking up facts, even doing work and many more things on the web. Despite the possibilities users have got using their mobile devices, many web pages or web applications are not optimised for them. Of course lots of mobile websites have evolved these days, but many of them lack usability and do not provide important functions or information. A solution has to be found in order to increase the capabilities and usability of websites and web applications towards mobile devices.

How do website owners find out if their mobile appearances on the web are adopted by the users and used in a meaningful way? How do they know what could be improved? Of course one could make surveys or use statistic evaluations in order to find out how many people are using the mobile site and how happy they are with it. A much better solution would be to directly analyse the surfing behaviour of actual users that are visiting the particular sites. Many tools are available by now to track users on websites and to evaluate all kinds of data. The main problem all these tools are facing is that current approaches to track user behaviour in order to improve web pages are mainly focused on desktop users and do not consider the special group of mobile users directly. Why not focus on analysing the specific user group the website should be improved for? This is exactly what the idea of *Mobile Viewport Tracking* is about. It tries to improve the effectiveness and usability of web pages in general and for specific user groups, using information that is gained from tracking users utilizing mobile devices. The term *Advanced User Behaviour Analysis* points out the fact that new approaches are evaluated and used in order to achieve this goal. The main focus is to use mobile devices to track typical user interactions that can be archived and further interpreted, in order to analyse or adapt web pages.

The first step of a practical implementation is to track position data within a web page, and visually represent the collected data in the form of a so called heatmap. The next step is to take

the basic concept and develop a more sophisticated approach which uses the collected information in a broader view in order to enable semi-automatic or automatic analysis, repackaging of websites, and also archiving and visualising user surf sessions in a meaningful way. These tasks contain a lot of challenges that will be dealt with in this project and the resulting theses. The information that can be retrieved from tracking users is limited and has to be interpreted somehow in order to optimise web pages. This can either be done manually, using some kind of reporting tool, or automatically by a program which could be very complicated. For an automatic interpretation, the tracking system would have to be able to recognise patterns and similarities in the surfing behaviour of users and to derive conclusions from it. Another challenge is the repackaging of web pages where important parts have to be identified and rearranged to build an optimised version of the page. Archiving and visualising or replaying user interactions, requires tracking of all types of actions a user performs during a surf session, such as entering text, clicking links, hopping from one site to another, etc.

1.2 Problem statement

Today, smartphones are getting more and more important but many websites are not optimized for them. Also, a lot of websites are optimized in a wrong way cutting away important information which would be available on the real website because it is hard to identify important content and areas. This project will provide tools to help identify important parts and show chances to optimize a website. Studies were made to compare and evaluate websites using identified KPIs (Key Performance Indicators). This thesis aims at answering the following questions and other questions that evolved during Case Studies:

- What knowledge can be derived from the data created by the prototype of the *Advanced User Behaviour Analysis Tool*?
- What is similar/different in the user experience when comparing web sites of different / same domain(s)?
- What is similar/different when using mobile / desktop version of a web site?
- What KPIs can be derived from the given data?

What is a “good” interaction sequence of zooms/scrolls?

Ratio of zooms/scrolls

What are the limitations for these KPIs?

1.3 Aim of the work

The main focus of this part is to analyse the data obtained through the *Advanced User Behaviour Analysis* tool in different ways. Studies with a defined count of users are used to derive knowledge from the data created by this tool. Examples for these studies are comparisons of web sites of different domains to identify common interaction patterns and answer general questions

like: “Is a website designed well if a user first zooms in to be able to read the text and then starts scrolling in different directions” or is this an indicator that the website is poorly designed for mobile devices. Manual studies like these are the base for this project. Through the usage of manual studies, patterns of usage behaviour are identified and classified. Main targets of this part are to identify metrics and indicators for usage behaviour and usability as well as answer the question how important content of a web page can be identified using this approach. Within this part, it was one goal to automatically identify areas of a web site that are important to many/very few people and use this importance of different elements of a website to help web designers, programmers and content providers identify ways to improve their web site as well as derive KPIs automatically from the given data. One example of such a KPI is the visit duration.

But not only focused on the visit duration of a whole page but on durations of the visibility on the viewport for different areas of a page. So a prototype of an analysis tool “beyond Google Analytics” was created. Possible KPIs could be the coverage that was read on the page as well as the “hotspots” on the website which were seen by most users (best area for important content that leads to further reading of additional articles or best areas to place advertisements) as well as the “poor dogs” on the website which are not seen by many people and could be removed by website administrators / designers or by some automated or semi-automated repacking.

1.4 Authorship

This thesis is based on a project work which was developed by a team of three students. Besides the individually implemented artifacts, the common basis of the project has to be extended to fit the requirements for all further tasks. Therefore, each of the theses will contain a common basis, but every work will answer different research questions including their own prototypical implementations. The main topic *Advanced User Behaviour Analysis* will be divided into the following sub-topics which have their own individual focus:

David Neukam: Case Studies, Automatic Reporting, Evaluation and optimisation of Usability using Advanced User Behaviour Analysis

Maximilian Aster: Repackaging Web Pages using Advanced User Behaviour Analysis

Sebastian Morgenbesser: Web Interaction Archiving and Replay using Advanced User Behaviour Analysis

Between those topics and the overall project several dependencies and relationships arise. Therefore, the chapters 2 and 3 explain the common basis by introducing the terms *Advanced User Behaviour Analysis* and *Mobile Viewport Tracking*. Those chapters were written as collaborative work. Also some minor parts within the introduction chapter such as the methodology were written in collaboration as the same methodology is used by all three theses.

In order to give an overview about the authors of the sections that were written in collaboration, the following table shows all chapters and sections that have different authors. It also includes the own chapters of each author on a top level.

Table 1.1: Author catalog

Chapter/Section	Author
1 Introduction	
1.1 Motivation	Collaborative
1.2 Problem statement	David Neukam
1.3 Aim of the work	David Neukam
1.4 Authorship	Sebastian Morgenbesser
1.5 Methodology	Maximilian Aster
1.6 Related project work	Collaborative
1.7 Relevance for Business Informatics	David Neukam
1.8 Structure of the work	David Neukam
2 Advanced User Behaviour Analysis	
2.1 Introduction	David Neukam
2.2 Aim	David Neukam
2.3 Existing approaches	David Neukam
2.3.1 Eye Tracking	David Neukam
2.3.2 Mouse Tracking	David Neukam
2.4 Heatmaps	David Neukam
2.5 Outcome	David Neukam
2.5.1 Case Studies	David Neukam
2.5.2 Web Interaction Archiving	Sebastian Morgenbesser
2.5.3 Content and process repackaging	Maximilian Aster
3 Mobile Viewport Tracking	
3.1 Introduction	Sebastian Morgenbesser
3.2 Technical Basics	Sebastian Morgenbesser
3.3 Data Model	Sebastian Morgenbesser
3.4 Components	Sebastian Morgenbesser
3.5 Business Logic - Concepts	
3.5.1 Client-side scripting	Maximilian Aster
3.5.2 Action Type Classification	Maximilian Aster
3.5.3 Important Content Identification	Maximilian Aster
3.5.4 Heatmap Drawing	David Neukam
3.5.5 User Tracking	Sebastian Morgenbesser
3.6 Possible Integration into other projects	David Neukam
4 State of the Art	David Neukam
5 Setup of Mobile Viewport Tracking	David Neukam
6 Implementation of Analysis Application	David Neukam
7 Evaluation	David Neukam
8 Conclusion	David Neukam

1.5 Methodology

The methodology describes how a certain type of problem is approached. As pointed out by [Hevner et al., 2004] there are in general two distinct paradigms applicable for Information Systems (IS), namely behavioural science and design science. The first one has its origin in natural science and tries to develop and proof a theory, e.g. why a certain human or organizational phenomena occurs when using a specific IS. Typical ways used to justify these theories are case studies, experimentation, or quantitative/qualitative analysis. But this approach is often not applicable, e.g. because the initial situation requires a creative problem-solving approach. In this case design science can support the research process. The difference can be aptly summarised by saying natural science tries to understand "what is", whereas design science helps to understand the "what can be". The output will be one or more artefacts which are iteratively evaluated to see if they fulfil the requirements defined by the environment. This artefacts can for example be models, methods, or implementations.

Design science seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished. [Hevner et al., 2004]

An important formulation in this sentence is that the created artefact is called innovation. The intention is to clearly point out that new ways to solve a problem have to be found, and not to use existing knowledge. The usage of existing "best-practices" would be profession design and would not result in a contribution to the knowledge base of the domain.

Another issue, which is also considered throughout this thesis, is that the results and development of artefacts may be interesting for technical and non-technical persons and should therefore target both audiences. The technical descriptions are therefore detailed enough to understand the implementations and their outcome, in order to allow to use concepts and components for other purposes. For non-technical persons the goal is to clearly define the importance of the problem statement and the utility of the solution. [Hevner et al., 2004]

Key aspects of design science are the three research cycles existing in every design science research project (see figure 1.1).

The relevance cycle shows that the environment opens opportunities and raises unsolved problem statements which have to be dealt with by the developed artefact. But the environment also defines criteria for the acceptance of the artefact. This means as long as they are not fulfilled additional design iterations may be needed. On the contrary, the design result may cause an adaption of the original requirements.

The rigor cycle shows that the incorporation of existing knowledge is essential for the artefact design. This does not only include state-of-the-art theories but also existing artefacts, models, or processes. But the research project should not only use knowledge, it should also contribute to the knowledge base. In this thesis a chapter describing the state-of-the-art in the application context is always preceding the description of the solution.

The last cycle is the design cycle itself. It shows that the creation of an artefact and the evaluation is an iterative process which requires the information gathered through the other

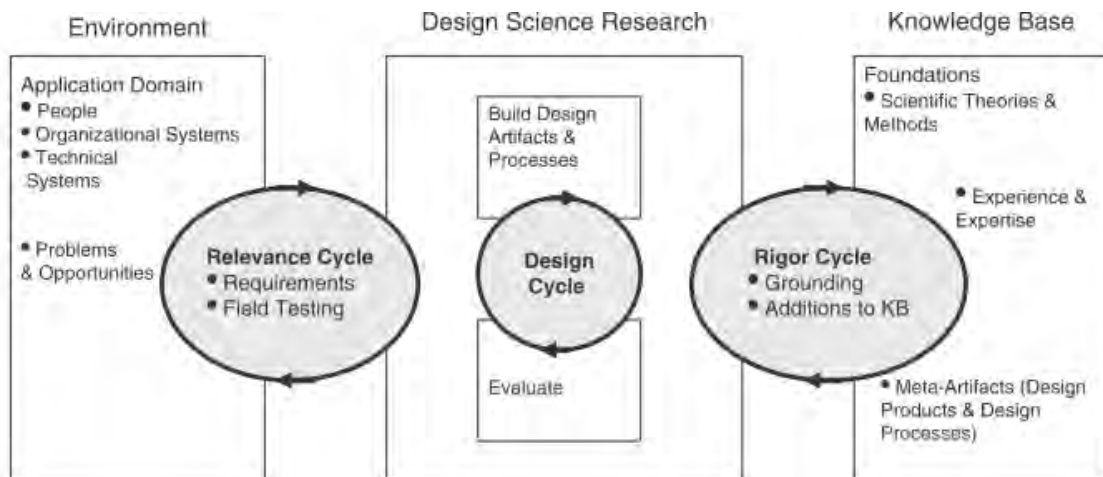


Figure 1.1: Design science research cycles [Hevner et al., 2004]

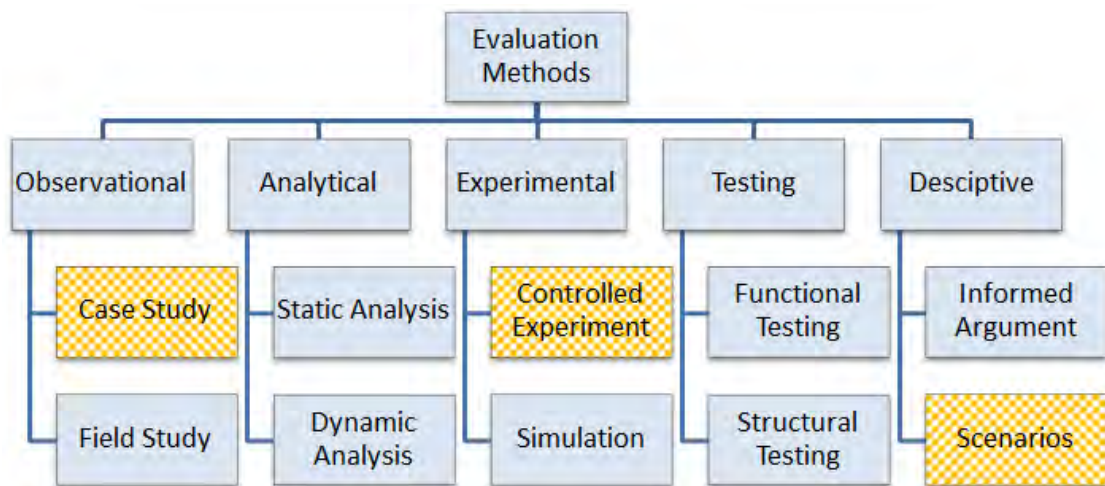


Figure 1.2: Design evaluation methods [Hevner et al., 2004]

cycles, i.e. the requirements and the existing theoretical and practical knowledge for the design and the evaluation. The solution part of the thesis and the evaluation are documented in separate chapters.

In [Hevner et al., 2004] various methods for the evaluation of artefacts are presented. Figure 1.2 shows which methods are used to evaluate the results of this thesis. Controlled experiments will be used to show the utility of the solution. In this kind of experiments the artefact is studied in a specific, well-defined environment, e.g. a specific set of web pages within a certain domain. Besides that, scenarios will be used to describe important aspects and additional application possibilities of the artefact. Parts of the solution will also be analysed within a case study to validate the artefact more detailed by a broader audience.

1.6 Related project work

The DBAI group at the Vienna University of Technology initiated several research projects in the area of Web Science up to now. The latest one is TAMCROW ¹ which has the goal to develop and propose a model for describing user behaviour of different crowds on the Web. The generation of the model will primarily be based on use cases from web accessibility, mobile browsing, web personalization, and automatic deep web traversal. This model will be applicable to a number of usage scenarios for different user agents. One of these groups are mobile agents. Each of these groups has their own strategies and therefore their own strengths and weaknesses in dealing with Web data. Our project will be used for the mobile browser data acquisition component, and our work contributes as well to the work package content and state/process repackaging. Web Interaction Archiving is a topic mentioned in TAMCROW as a further use case but not a focus of the project. Hence, this new perspective will enrich TAMCROW with additional insights. Finally, our contributions in user behaviour analysis will support the evaluation and help the TAMCROW project to elaborate the differences between the user groups.

1.7 Relevance for Business Informatics

A major focus of Business Informatics lies on the development of solutions for business problems by applying new strategies and technologies. The Web has become an important factor throughout all kinds of business sectors, e.g. for advertising or as a distribution channel. But the target devices are changing from desktop browsers to smartphones, because the market of mobile devices grows, a trend which will hold on for some time. As a result companies have to react and adapt their web solutions in order to be able to compete. Business Informatics combines technical, economic, and social aspects, which makes the Web and mobile devices an ideal platform. The topic of this master thesis tries to integrate these social aspects by analysing user behaviour, and the technical aspect by creating artefacts, designed to solve problems relevant for businesses. Namely, the automatic re-packaging and optimisation of important content on web pages for mobile devices.

1.8 Structure of the work

Chapter 2 gives an overview of the project and its goals. The term *Advanced User Behaviour Analysis* is explained and the project target is defined. The following chapter *Mobile Viewport Tracking* (3) explains the solution created with some introducing technical basics as well as the data model used and an overview of the implemented components. Important business logic and heatmap drawing is explained and the possibilities of integration into other projects are discussed. Chapter *State of the Art* (4) gives an overview of the current State of the Art in Web Usability, Eye Tracking and Mouse Tracking technologies and helps categorise and assess the solution implemented. Chapter *Setup of Mobile Viewport Tracking* (5) gives an overview of the setup process of *Mobile Viewport Tracking* on the mobile devices and on the server. Chapter 6

¹<http://www.dbai.tuwien.ac.at/proj/tamcrow/>

gives an overview over the web-application that was created to help user studies based on the *Mobile Viewport Tracking* solution. Chapter 7 serves as an *Evaluation* of the solution created as it shows the functionality and findings deducted from user studies and proposes key performance indicators (KPIs) that were found useful during these studies. In the *Conclusion* (8), a brief overview over the whole work is given and key findings are resumed and results are discussed. Also, an Outlook for further project work is given.

Advanced User Behaviour Analysis

2.1 Introduction

The term *Advanced User Behaviour Analysis* is defined as an umbrella term for disciplines that focus on user behaviour analysis using sophisticated technical solutions and measurement techniques. Concrete solutions that fit under this definition are eye tracking and mouse tracking. Web analytics is typically based on static information and dynamic user behaviour [Albanese et al., 2004] of a user on web pages. Many studies using eye tracking technologies were done testing theories like the scanpath theory [Josephson and Holmes, 2002] on websites, identifying determinants and metrics of web page viewing behaviour [Pan et al., 2004], general user behaviour pattern identification [DeWitt, 2010] and specific optimisation of web pages [Rakoczi, 2010] targeting the dynamic user behaviour. Static information like server logs is used by [Khoo et al., 2008] to analyse web page viewing behaviour.

As opposed to common web analytics solutions such as Google Analytics¹ which focus only on the path the users followed on a website and lack information about the viewing behaviour on the pages itself, *Advanced User Behaviour Analysis* also analyses the user behaviour on single pages. An example how Google Analytics is used for well-founded web page analysis can be found in [Plaza, 2011]. By this definition, also other tracking-methods can be used for *Advanced User Behaviour Analysis* as long as they provide information that goes into the details of viewing behaviour of pages. A new approach that is proposed and implemented in this project is named *Mobile Viewport Tracking* (MVT). The details of the project are explained in chapter 3.

2.2 Aim

The goal of *Advanced User Behaviour Analysis* is to identify actual user behaviour on websites, find out how website visitors interact and identify why they behave in this way. For this, it serves as a basis to identify parts of web pages that are important and those which are not. This could be

¹<http://www.google.com/analytics>

done on a visual level or on the level of DOM² elements. The data collected could be visualised using so called heatmaps which are described in section 2.4.

One target of this project was to create an alternative to eye- or mouse tracking that does not have the drawbacks of both systems. Eye tracking needs expensive and time-consuming studies while mouse tracking suffers from low correlation between eye- and mouse-movement as further explained in section 2.3.2.

This approach could be used to create case studies or projects that build on top of it. Case studies could give hints on user experience on websites, show limitations and give the possibility to improve websites. The recorded data could also be used to build a new kind of web archive that is different from typical crawling approaches such as The Internet Archive³. Furthermore, the importance of different regions of a web page could be used to create a repackaged version of this page that shows only the relevant content and omits parts not of particular interest.

2.3 Existing approaches

There are multiple similar existing approaches. The two most common are mentioned below. Further detailed information is given in chapter 4.

2.3.1 Eye Tracking

Eye tracking is an approach to usability studies where the eye-movements and fixations are tracked. The aggregated movements are called scanpath, which is “defined as a sequence of fixations in Areas of Interest or lookzones [Pan et al., 2004]” which is heavily influenced by the stimulus [Josephson and Holmes, 2002]. These fixation points - defined as “relatively motionless gaze” [Pan et al., 2004] give a hint what parts of the website were read and which parts were ignored by the user. Website Analysis is typically done with stationary eye tracking systems sold commercially by different vendors^{4 5}.

2.3.2 Mouse Tracking

By tracking the movements and resting points of a participants mouse, a relative estimate of the participants attention can be made. There is a correlation up to 88% between eye and mouse movements [Chen et al., 2001]. This value is the one usually printed at the homepages of mouse tracking vendors to state the quality of data generated by mouse tracking, but this number is the percentage of “regions that the eye gaze didn’t visit” which also “were not visited by the mouse cursor, either” [Chen et al., 2001]. According to [Chen et al., 2001], the average correlation is about 58% in sections.

²Document Object Model <http://www.w3.org/DOM>

³<http://www.archive.org>

⁴<http://www.tobii.com/>

⁵<http://www.smivision.com/>

Mouse Tracking is offered by multiple vendors. Some solutions are free⁶, others^{7 8} cost from a few Euros a month⁹ up to multiple thousand Euro per Website Optimisation project¹⁰.

2.4 Heatmaps

Heatmaps are graphical representations of data with an overlay that shows the intensity of data of the specific problem domain like web analytics or meteorology. Every value of the data set is projected on a colour gradient from very high/hot (white or red), to very low/cold (blue).

Heatmaps originate from images created by thermographic cameras or the association of some colours to temperatures. They help identify remarkable data and gain a simple visual impression of a large amount of data in an intuitive way.

In usability study terms, heatmaps are typical representations of recorded data of eye-movements and especially their fixations which show "distribution of visual attention on the screen" [Buscher et al., 2010] or for clicks in mouse tracking systems.

As a heatmap is a simple visualisation that is very easy to understand, this approach is also used by *Mobile Viewport Tracking*.

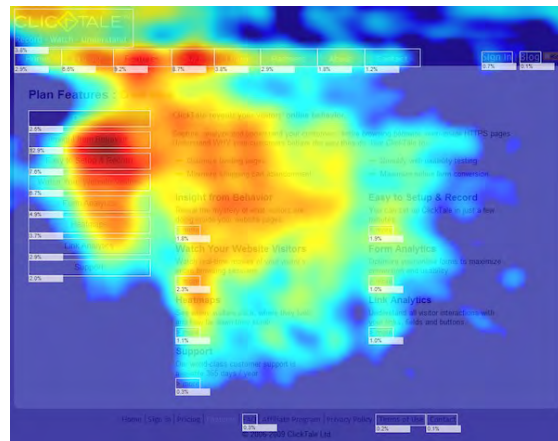


Figure 2.1: Heatmap: www.clicktale.com, Made by Clicktales Mouse Tracking Solution [Clicktale, 2013b]

Figure 2.2 shows a heatmap created with *Mobile Viewport Tracking*, while figure 2.1 is an example of a heatmap created with the commercial mouse tracking software Clicktale [Clicktale, 2013b]. As Clicktale tracks mouse movements that are precise to a single point, these heatmaps show a finer granularity than the heatmaps generated by *Mobile Viewport Tracking*

⁶<http://www.labsmedia.com/clickheat>

⁷<http://mouseflow.com/>

⁸<http://www.clicktale.com/>

⁹<http://www.crazyegg.com/>

¹⁰<http://www.m-pathy.com/>

where the whole screen of a mobile device is tracked, but the correlation of attention and mouse-movements is not very high [Buscher et al., 2010].



Figure 2.2: Heatmap: www.gmx.at, Aggregated data from 11 users created with *Mobile Viewport Tracking*

2.5 Outcome

The three different focal points that emerged from the basic idea of *Advanced User Behaviour Analysis* led to three different master theses. These different fields of application are shortly discussed in the following sections. The details can be found in the specific master thesis that covers the corresponding topic.

2.5.1 Case Studies

The first outcome are studies made with the *Advanced User Behaviour Analysis* approach which show possibilities of usage of such a tool. A prototype of an analysis-portal was created that can be used to analyse the recorded data. This tool was used to analyse a study done with students of the course “Applied Web Data Extraction and Integration” at the Vienna University of Technology. Metrics and indicators for web usability are identified that are not available with traditional ‘page-level’ approaches like Google Analytics. It shows how people behave differently when accessing a desktop or mobile version of a website with their smartphone and what differences and similarities can be found when comparing different websites of one domain to each other. Some Key Performance Indicators (KPI) are theoretically discussed and some were implemented and their results are shown.

2.5.2 Web Interaction Archiving

Another possible outcome of the data from *Advanced User Behaviour Analysis* is a new way to historicise the World Wide Web. A common approach to keep a history of the web is to store entire web pages in an archive at regular intervals. In the future you would be able to take a look at those old web pages and get a feeling how the web actually looked like in the past. The problem is that this approach is missing an important detail of the history which is the user experience. One could say that it is much more interesting to memorize how people actually perceived web pages at different times, rather than just storing the plain content itself. The idea is now to create a mechanism that enables us to archive actual user experience. This is done by using the data that is retrieved from *Mobile Viewport Tracking* which focuses on tracking actions of mobile devices as mentioned before. These “actions” are then combined to “interactions” on specific websites, and so it is possible to reconstruct whole surf sessions of users. There are several ways to visualise those sessions. The simplest form is to generate some kind of textual protocol that describes a specific user session on a specific site at a specific time. A more sophisticated approach, is the rendering of a video that shows the user session as it actually took place in the past in a visual way.

2.5.3 Content and process repackaging

A third outcome is to enhance content and process repackaging. Repackaging means the transformation of a single web page or a multi-page website into another representation. The first kind of transformation would be content repackaging, where parts of a web page are taken and rearranged, for a specific user group or target device, while others are omitted. The second one

is considered process repackaging. In this case the whole structure of multiple web pages and how they are interconnected is transformed, which also affects the underlying business process.

The user behaviour data collected within MVT cannot only support the way the repackaging is done but also leads to new use case scenarios. One of these scenarios is to use repackaging to directly draw a heatmap of the user behaviour within a specific web page. This approach, called client-side repackaging, uses browser functionality to adapt a web page. It is one of the implemented artefacts. The second solution, which was implemented on server side, is more complex and consists of several steps. These phases include the extraction of content from a web page, the segmentation into blocks, the classification and selection of some of these blocks, and the transformation of their content into a mobile version of the website. The clue is to use the user behaviour data to determine important content and to specifically choose this content within the selection phase of the repackaging process. By doing so the generated mobile version for a website makes it easier for users to find key information.

Mobile Viewport Tracking

3.1 Introduction

The last section introduced *Advanced User Behaviour Analysis*, which is the general term for analysing user behaviour especially focusing on web browsing behaviour. This section covers *Mobile Viewport Tracking*, which is the technical instrument we are using for user behaviour analysis. The idea behind *Mobile Viewport Tracking* is quite simple. Because of the pervasion of smartphones these days one could say that a very important part of web surfing behaviour is performed by users utilizing mobile devices. Therefore *Mobile Viewport Tracking* concentrated on tracking smartphone users. The main limitation of smartphones while surfing is still the small screen size, which has a maximum that a user can still handle. This disadvantage for surfing is turned into an advantage for tracking user behaviour. While surfing a user has to make changes to his viewport, which is the visible part of the screen, by zooming, scrolling, panning or rotating his device. The concept of viewport is further described in section 3.2.1. Therefore an assumption can be made, that the different states of the users viewport are in fact content, that the user looked at and with a certain probability also content that he read. A comparison can be made between tracking the viewport of mobile devices and previously introduced eye tracking systems. The aggregation of the recorded data can be an indicator of which parts of websites are more often viewed than others.

So how does the tracking work? The answer is client-side scripting which is explained more detailed in the next section. A JavaScript snippet has to be inserted on the client side, in this case in the mobile device's browser. The script is able to track all the actions mentioned before such as zoom, scroll, pan, rotate etc., and transmits the recorded data periodically to the tracking server. The server receives the data, cleans it, does additional calculations in order to derive the exact action type etc. and then stores it in a database. Later on, calculations and an analysis can be done upon this data base.

How does the script get to the client's browser? There are actually two possibilities. First it can be directly embedded into a target website which should be analysed. The embedding of third party JavaScript components, as in this case, is called beaconing [Ganapathi and Zhang,

2011a]. This is fairly easy, and it can be done by the websites administrator. The advantage of this approach is that nothing has to be configured on the client side. The website owner has the power to turn on the tracking whenever he wants, of course he most likely would have to inform the user that he is tracked due to existing laws. The second possibility is to use a proxy server to modify the HTTP¹ response of each request returning HTML² documents and inject the script. The advantage of this solution is that multiple websites can be tracked, but the user would have to directly accept the tracking and configure the proxy in his mobile devices network settings. In this project the second approach was chosen in order to get a reasonable amount of data from several different websites to analyse and evaluate the project. The next section will cover some technical basics and after that the details of the architecture and abilities of *Mobile Viewport Tracking* are discussed in the following sections of this chapter.

3.2 Technical Basics

This section covers technical basics and shortly explains terms that are used in the following chapters.

3.2.1 Viewport

The viewport is the area that is actually visible during surfing. On a desktop computer the viewport is very large depending on the resolution. Mostly vertical scrolling is needed for surfing. A mobile device cannot display the whole page on the screen, therefore the viewport is always a small part of the website that is currently visible. A lot of scroll and zoom actions are necessary in order to navigate on a classic website designed for desktop computers. The coordinates and the size of the viewport can be retrieved by using simple JavaScript functions that read the following attributes:

- *pageOffsetX*: the X coordinate of the top left corner of the viewport
- *pageOffsetY*: the Y coordinate of the top left corner of the viewport
- *innerWidth*: the width of the viewport
- *innerHeight*: the height of the viewport

¹Hypertext Transfer Protocol - <http://www.w3.org/Protocols/>

²Hyper Text Markup Language - <http://www.w3.org/standards/>

The following image illustrates the viewport on a mobile device:

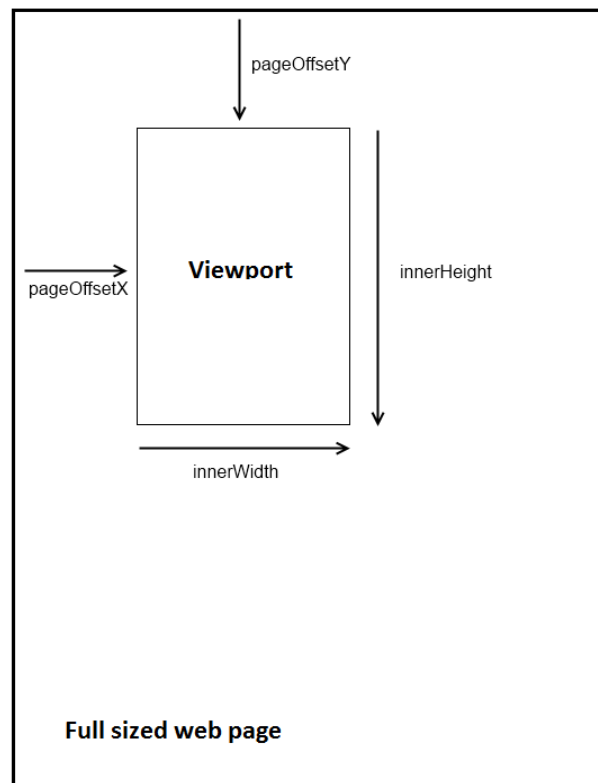


Figure 3.1: Illustration of mobile devices viewport in relation to the full size web page

3.2.2 Web Services

Web Services are used to provide an unified interface to a certain operation over the Web or in a closed network. They are uniquely identified with an URI³ which provides the access to the service. The communication is very likely performed over the HTTP protocol and the content is transferred in XML⁴ format. Basically there are two established types of Web Services. Standard Web Services, often using the SOAP protocol which is described below, and Restful Web Services using standard HTTP functions as an interface.

SOAP

SOAP stands for Simple Object Access Protocol⁵ and is an XML based protocol that is used as a wrapper for the communication with the Web Service. As common interface very often a so

³Unified Resource Identifier - <http://www.w3.org/Addressing>

⁴Extensible Markup Language - <http://www.w3.org/XML>

⁵<http://www.w3.org/TR/soap>

called WSDL⁶ is used. WSDL stands for Web Services Description Language and is also XML based. Every party interacting with the Web Service is able to derive a skeleton from the WSDL in the preferred programming language. The skeleton contains generated classes which make it easy to attach objects as parameters and call methods of the Web Service.

This is an example of the layout of an SOAP envelope:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
  </s:Header>
  <s:Body>
  </s:Body>
</s:Envelope>
```

The actual message is wrapped in the body tag, whereas the header information is packed into an own header tag. The header is optional and can contain meta information such as encryption or the ultimate recipient.

REST

REST stands for Representational State Transfer and defines a different type of Web Services. Restful Web Services are also available under a specific URI but they respond to standard HTTP GET and POST actions. This makes it very easy to call methods of the Web Service from any device that is able to invoke an URL. The protocol that is used to wrap the Web Services result is not defined and is up to the owner of the service. JSON which is described below, is often used to serialize objects in order to communicate with the service.

3.2.3 JSON

“JavaScript Object Notation (JSON) is a lightweight, text-based, language-independent data interchange format. [...] JSON defines a small set of formatting rules for the portable representation of structured data” [Network Working Group and Crockford, 2006]. The following block shows a simple example of a JSON object:

```
{
  "name": "MVT",
  "id": "1234567",
  "properties": {
    "number": 13,
    "data": [ "dataA", "dataB", "dataC" ]
  }
}
```

⁶<http://www.w3.org/TR/wsdl>

3.2.4 DOM

“The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page” [W3C, 2005].

3.2.5 Client-side scripting

Client-side scripting is a technique that describes computer programs that run on the client-side, more specifically in the clients browser. The most famous example is JavaScript⁷, which is supported out of the box by the majority of the available browsers. JavaScript and also frameworks based on JavaScript are mainly used to create dynamic HTML pages. The main advantage of JavaScript is, that it can be directly embedded into websites without the need to install client software or plugins to run it. Apart from creating dynamic websites, JavaScript can also be used to communicate with services in the background and also to access DOM properties such as current window size, scrolling position, referring URL etc., which makes it a good choice for this project.

3.2.6 AJAX

AJAX stands for Asynchronous JavaScript and XML and is a technology based on JavaScript, that gives the possibility to reload specific parts of websites without the need of reloading the whole page. This is very useful when creating web applications and dynamic web pages, because the user is not disturbed by page reloads and is able to interact with the web page like it would be a local application. Figure 3.2 shows the difference between the classic web application model and the AJAX technology [Garret, 2005].

⁷<http://www.ecma-international.org/publications/standards/Ecma-262.htm>

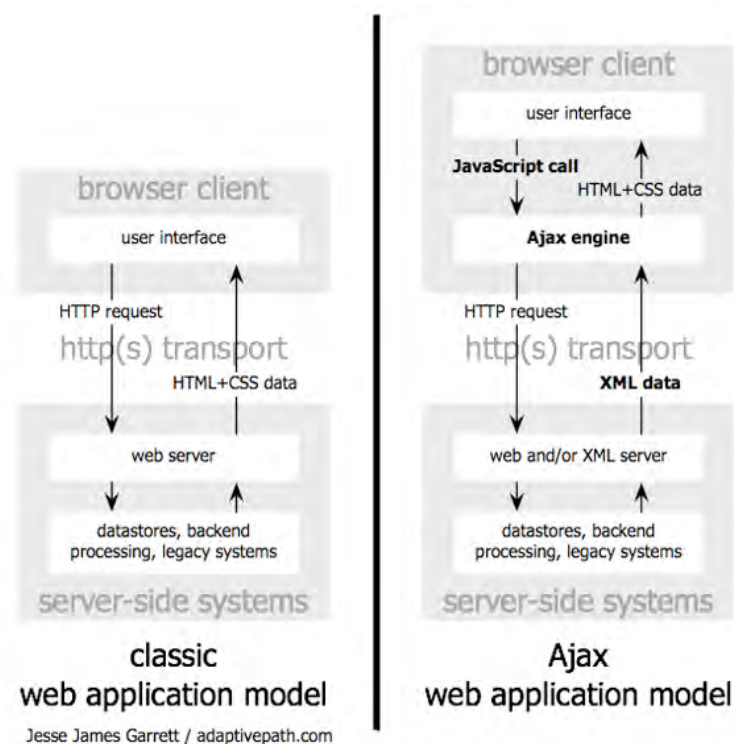


Figure 3.2: The traditional model for web applications (left) compared to the AJAX model (right). [Garret, 2005]

3.2.7 OR Mapping

OR Mapping stands for Object-relational mapping which is used in modern programs to map high level objects and entities to database tables. The layer or program in between is called the OR-Mapper and controls the communication from and to the database. In practice a programmer can e.g. call a save method of the OR-Mapper, pass an object as parameter, the OR-Mapper then translates the objects data to the databases query language and executes it. The result is then again parsed, turned into an object and returned to the caller. In the Java world the most famous OR-Mapper is called *hibernate*⁸.

3.3 Data Model

This section shows the basic data model used for storing and analyzing the recorded viewport data:

⁸<http://www.hibernate.org>

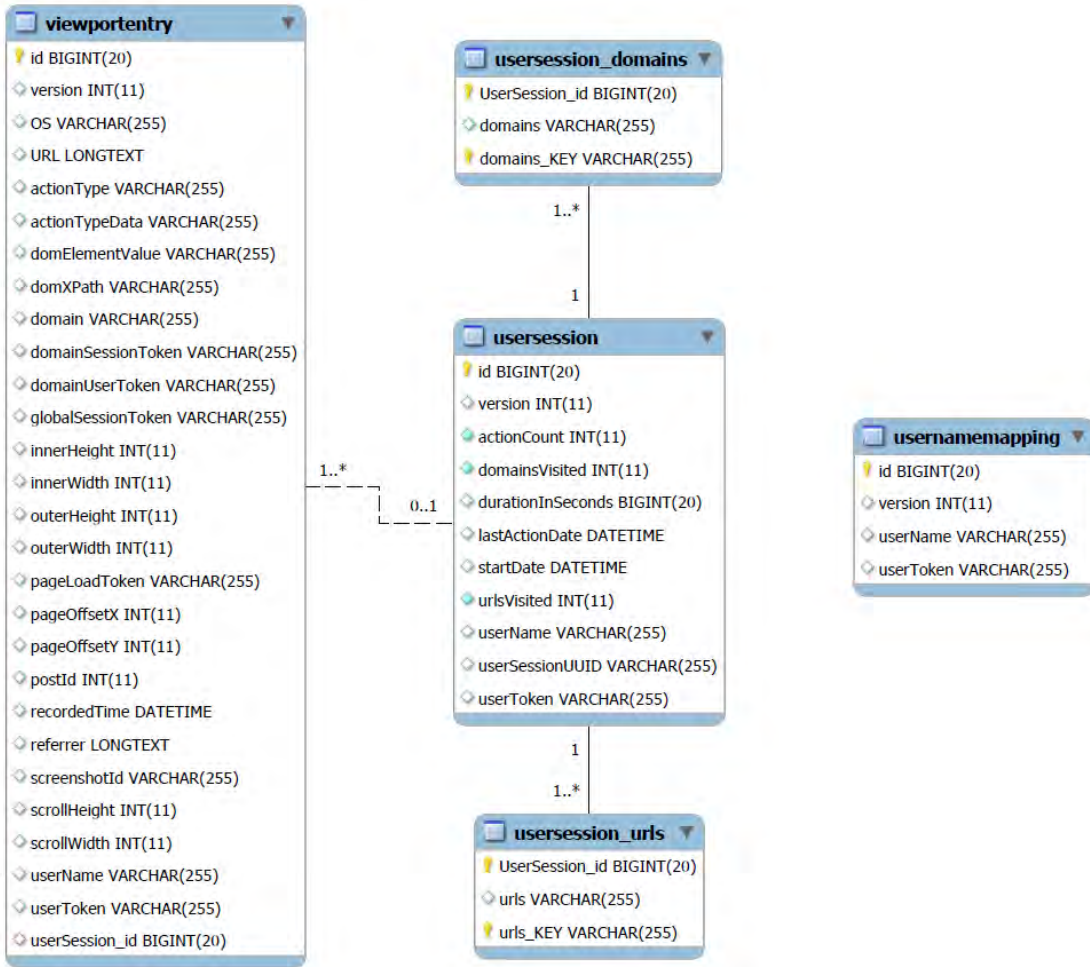


Figure 3.3: Data Model - part 1

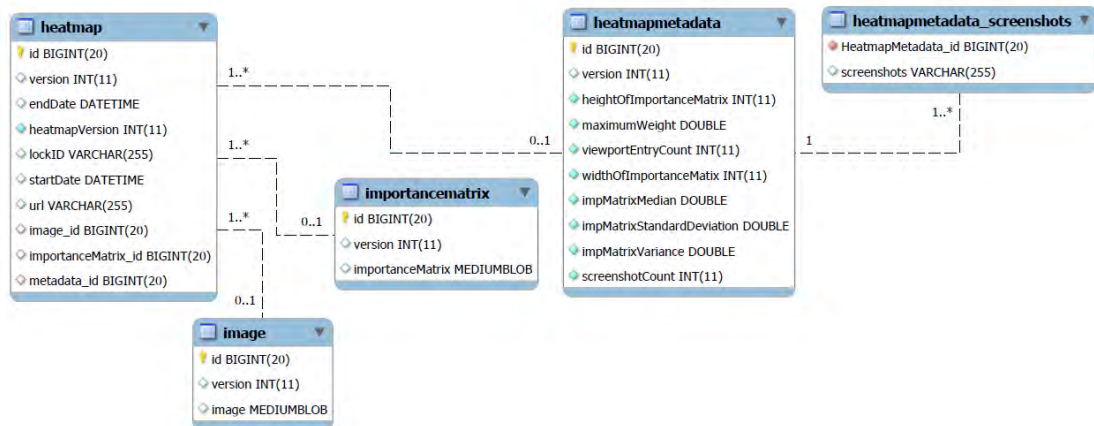


Figure 3.4: Data Model - part 2

The three main entities are the following:

- **ViewportEntry:** This is the main entity. A ViewportEntry represents one action on a mobile device for example a scroll, a zoom, a pan etc. The data is a flat table allowing the usage of data mining techniques.

Important fields are:

- *actionType*: Type of the action such as scroll, a zoom, a pan etc.
 - *domElementValue/domXPath*: If the action was key press or click the relevant DOM element is recorded together with the path and the actual value of the element
 - *domainSessionToken/domainUserToken/pageLoadToken/globalSessionToken*: Various session tokens in order to track the surfing user
 - *pageOffsetX/pageOffsetY*: The coordinates of the north west corner of the viewport. This represents the scrolling location on the website
 - *innerWidth/innerHeight*: The width and height of the viewport. This indicates the zoom level
 - *referrer*: Referring URL e.g. google.com
 - *recordedTime*: The exact date and time when the action occurred
 - *screenshotId*: If a screenshot was made for this action, the field contains an unique id which relates to the taken screenshot image
 - *userToken/userName*: If the user can be tracked (accepting cookies), the fields contain the unique session id and an optional user name if the user has registered himself
- **UserSession:** This entity is used to wrap up many viewport entries related to the same user session into one package. Additionally meta-data is added to the user session such as visited domains/URLs, the start time of the session, the end time of the session, the action count of the session or the duration of the session.

- *Heatmap*: This entity is used to store a heatmap based on the aggregated data of multiple sessions performed on one specific website. The heatmap contains an importance matrix that represents the raw data of the calculated heatmap itself. Together with a screenshot, a graphic representation of the heatmap can be generated from the raw data.

3.4 Components

This section will provide an overview of the architecture of the *Mobile Viewport Tracking* system, and will shortly describe each component and its purpose within the system.

3.4.1 Architecture

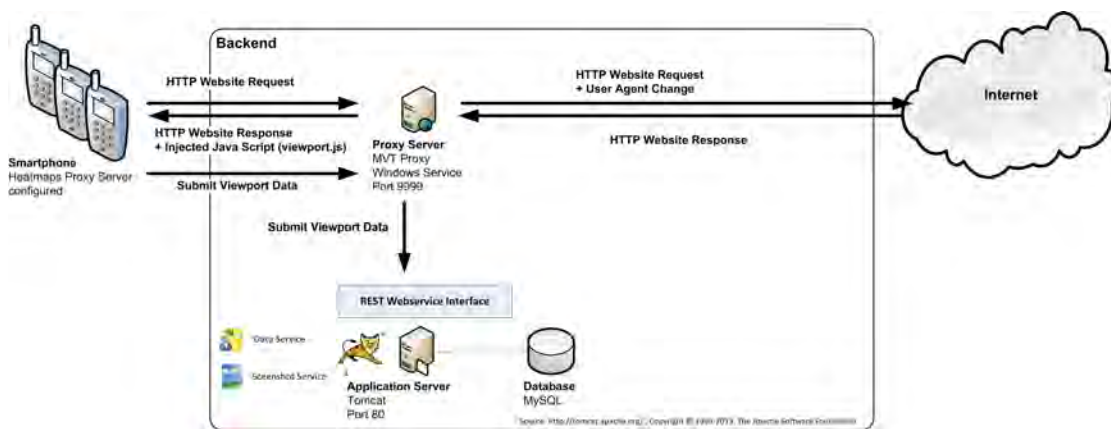


Figure 3.5: Architecture Overview

3.4.2 Client-side script

As can be seen in the architecture diagram the client side is most likely a smartphone running a mobile browser. To be able to track various actions related to the surfing behaviour of the user such as pan, zoom, scroll etc. client side code has to be injected somehow. The client-side logic basically consists of a JavaScript file containing functions for tracking surf related actions and sending the recorded data to the central application server which stores it in the database. The script can either be embedded directly in the target web page or has to be injected by a proxy server (see section 3.4.3).

3.4.3 Proxy Server

The proxy server is used for modifications of HTTP requests and also HTTP responses which are triggered by an user who is surfing on his mobile device. Therefore the proxy has to be configured on each mobile device that is going to be tracked. For HTTP requests the proxy server modifies the user-agent of all incoming requests, in order to avoid getting mobile websites

which are not targeted within this project. The current used user-agent is an iPad user-agent which proved to be the best suited one for this studies as technologies as Flash are also not available on an iPad but the screen size is mostly like a desktop device.

On the HTTP response the proxy modifies the HTML content and inserts a JavaScript tag that is importing the client-side tracking script into the head section of the HTML page. The proxy server is based on the Brazil Framework and is a fork of PAW Proxy⁹. The configuration of the proxy on the supported mobile platforms can be found in the appendix section 5.1.

3.4.4 Web Service

The Web Service component is the general interface of the application server. It serves as a data collector and provides a store data action as Restful Web Service method. The underlying business logic (see section 3.5) is responsible for cleaning, preparation, analysis and classification of the received data. The whole server side is written in Java¹⁰ and published as web application on a Tomcat¹¹ server. The responsibilities of the service basically split into the following two parts:

Data Service

As already mentioned, this service is responsible for storing and also processing the incoming data. The method is called directly from the clients (smartphones) as HTTP POST request. It also provides methods to retrieve heatmap data or to trigger the creation of heatmaps.

Screenshot Service

The second part of the service is responsible for taking screenshots of the visited web pages. This is done with Selenium Web Driver¹². This tool provides the possibility to open up a browser with a specified URL, waiting for the website to load and finally taking a screenshot of the entire web page which is later used to create heatmaps. For each user that surfs to an URL an screenshot has to be taken. Therefore the service is managed by a simple queuing mechanism with first-in first-out principle in order to prevent overloading of the server if to many people use the service.

3.4.5 Database Layer

The last component is the lowest layer which is responsible for storing the processed data and providing it for later analysis. For database connection and operations the OR-Mapper hibernate¹³ is used. Beneath this a MySQL database is attached, but because of the usage of an OR-Mapper, the database is easily exchangeable.

⁹<http://paw-project.sourceforge.net/>

¹⁰<http://www.java.com/>

¹¹<http://tomcat.apache.org/>

¹²<http://docs.seleniumhq.org/>

¹³<http://www.hibernate.org/>

3.5 Business Logic - Concepts

3.5.1 Client-side scripting

The essential component for the collection of user data on client-side is implemented in JavaScript. It can either be embedded in a web page by the content provider itself or by using a proxy server to inject it. The second approach has the advantage that data about any web page can be collected, the disadvantage is that the configuration of a proxy server is necessary, which may not be possible on all mobile devices.

One problem which occurs if the script is injected into every page that is loaded, is that it is also executed for IFrames within a page, which do usually just contain advertisement. In those cases the execution is aborted and no data is collected within this frame, because the data is not considered relevant for this project.

The script does not require that JavaScript frameworks like jQuery are included. The only dependency is to Hammer.js¹⁴, a JavaScript library to track multi-touch gestures, which is directly included in the main script. All the functionality of *Mobile Viewport Tracking* is defined in an own namespace *MVT*. This allows a clear structure and avoids conflicts with other client-side code may running in a web page.

The first action taking place after the initialization is to identify the user, for this purpose cookies on different levels are used. The lowest scope of user-tracking is per page, followed by domain, and global tracking. For a detailed explanation see section 3.5.5.

Afterwards the actual action tracking is started. The script always keeps track of the last web page state it sent to the server. This state basically includes the current position of the user on the web page, i.e. the current viewport. The next step is that this state is periodically compared with the current state, if a change occurred the data is queued to be sent to the server. Apart from this periodic data, data is explicitly collected e.g. for clicks. Some properties which are always collected are the URL, the operating system of the device, X and Y position, or the inner and outer width and height.

Table 3.5.1 shows a list of basic events that were identified and by which DOM events they are triggered. These events represent the explicit actions which are tracked and recorded, e.g. if a user submits a form (SUBMIT) or if text was entered into an input field (INPUT_CHANGED). Some of the events can also be specific for the device which is used (e.g. WEBSITE_DEACTIVATED). The table also shows one usage of the Hammer.js library, namely the tracking of DOUBLE_TAP and HOLD actions. Compared to the periodically collected data these events are delivering more information, e.g. the XPath of the HTML element which was invoked, the element value, or the position where the event was triggered.

When tracking user input security and privacy issues arise. For example when tracking the user behaviour on an online banking web page. To avoid possible fraud, password inputs are not recorded. But unfortunately it is not possible for a user to verify what is tracked and what is ignored. It has to be mentioned that the injection of the script is not possible if the connection is secured, i.e. if HTTPS using SSL or TLS is used.

¹⁴<http://eightmedia.github.io/hammer.js>

Table 3.1: Mapping of DOM events to MVT events

MVT event	DOM event
CLICK	click (a, area, span, div, p)
SUBMIT	submit (form)
INPUT_CHANGED	change (input, textarea, select)
KEY_PRESS	keydown (input, textarea)
WEBSITE_ACTIVATED	pageshow (window) focus (window)
WEBSITE_DEACTIVATED	pagehide (window) - iOS blur (window) - Android
ROTATION_TO_HORIZONTAL	orientationchange (window)
ROTATION_TO_VERTICAL	orientationchange (window)
DOUBLE_TAP	doubletap (hammer)
HOLD	hold (hammer)

As mentioned before, the collected data is not directly sent to the server but gets queued to reduce the server and client load. For asynchronously sending the encoded data a *XMLHttpRequest* object is used. It uses a HTTP POST request to transmit the data to the web server. An important aspect is that the same origin policy does not allow the access to the data returned from the MVT web server, because that server is not the host server of the web page. Nevertheless the data is received on the server. One way to circumvent this policy is the usage of JSONP¹⁵ which is a technique embedding a new *script* element into the HTML code with a specific request URL that usually returns a function call with the response data as parameter. For the purpose of tracking user actions it was not yet necessary to return data, but it could be a future extension. An alternative would be the proxy server which can redirect requests and manipulate the headers to circumvent the same origin policy.

Filters

Passwords were filtered on the client-side (see appendix section A.1.3 line 222ff). Only the first letter is transmitted in clear text, other letters are replaced by asterisks, in order to make aware of the capability to record these passwords. In fact code to record passwords could be injected by every node on the route between the webserver and the client.

Major explicit content websites were blacklisted and a filter for specific keywords was included on the server side to anonymize data recorded on this sites.

3.5.2 Action Type Classification

The next step after the client-side collection of data (section 3.5.1) is the classification of the collected data. A basic classification was already done within the client-side script. This classification, which was shown in table 3.5.1, will now be extended on server side. Every viewport

¹⁵JSON with Padding - <http://www.json-p.org>

Table 3.2: MVT events identified on server side

MVT event
ZOOM_IN
ZOOM_OUT
SCROLL_UP
SCROLL_DOWN
SCROLL_LEFT
SCROLL_RIGHT
SCROLL_UP_LEFT
SCROLL_UP_RIGHT
SCROLL_DOWN_LEFT
SCROLL_DOWN_RIGHT

change which was tracked has a unique, consecutive ID. This allows to easily find the previous viewport data and to compare the new one with it. To determine if a ZOOM_IN or ZOOM_OUT has happened the *innerWidth* and *innerHeight* are compared. To identify scrolling in one of the eight possible directions the X and Y position of the old and new viewport are compared. The source of the Action Type Determiner can be found in the appendix in section A.1.2.

3.5.3 Important Content Identification

After the data is collected (see section 3.5.1) and post-processed (see section 3.5.2) it is necessary to bring it into a format for further usage, this includes a machine-readable representation and a human-readable one. Unfortunately the raw data is not directly usable for further reasoning or evaluations because most actions on the user interface of a web page will create an event and the granularity of these actions is too low. The data is therefore summarized with the goal to identify the most important content on a web page. The approach chosen within the MVT project is described in the following section. The condensed data is then made available by a Web Service method returning a JSON object which can also be deserialized to a Java object. A human-readable version of the data are heatmaps, which can be automatically drawn (see section 3.5.4).

Weighted raster

Important content is identified by assigning weights to specific parts of the web page. For this purpose the web page is rastered, i.e. a virtual raster with the same height and width as the web page is used to represent the actual page. Every cell in the raster has a predefined width and height which can be configured. All cells are initialized with 0 weight. This raster will be further called *importance matrix*. For every user action which was recorded the weight of that part of the raster will be increased. Depending on the type of MVT event and other aspects a different weight will be assigned.

Weighting

The *importance matrix* is created by iterating all entries for a certain web page. The algorithm is first explained considering a simple scroll event to an arbitrary location on a web page. First of all, the cells which are affected are determined, i.e. the start column and row, and the number of spanned rows and columns. Two basic factors for the resulting weight of the event are readability of the text within the viewport, and the time spent at this specific location to study the content. The pressure representing the zoom level is calculated by looking at the longest dimension of the viewport entry, i.e. the inner width or height depending on the rotation. After controlled experiments with different devices, values for *minimum readability* and *maximum readability* were defined. If the longest dimension lies within the interval of these two values the pressure gets a high weight, otherwise not. Additionally the weight increases the closer the dimension value is to the *maximum readability*. To calculate the time spent on the specific part of the web page, the difference of the current and the previous viewport change is evaluated. Similar to the readability an interval with the highest weight was evaluated through controlled experiments. This interval ranges from 1 to 20 seconds and corresponds with the findings of [Josephson and Holmes, 2002] and [Pan et al., 2004], which stated that important information is processed during the first 15 seconds on desktop computers.

Afterwards the weight is low again, because the assumption is that the user may have put the smartphone away or closed the browser.

After calculating these basic weights, more specific calculations are performed depending on the action type of the previous and the current event. In general the weighting is divided into point-weighting and area-weighting.

Listing 3.1: Area-weighting

```
+-----+
|10|10|10|10|10|10|10|
+-----+
|10|10|10|10|10|10|10|
+-----+
|10|10|10|10|10|10|10|
+-----+
|10|10|10|10|10|10|10|
+-----+
| 5| 5| 5| 5| 5| 5| 5|
+-----+
| 5| 5| 5| 5| 5| 5| 5|
+-----+
```

Area weighting means that all cells within the viewport get the same weight added to their already existing weight. But there is an exception to this rule, namely in case the current rotation is vertical, in this case the weight of the last $\frac{1}{3}$ of the cells from the bottom of the viewport is halved. The reasoning for this exception is that a user usually starts to scroll after the first part of the page is read and does not continue reading until the end before scrolling. This was done because experiments showed that the last third is not read but people started scrolling after $\frac{2}{3}$ of the content was read. An schematic example is shown in listing 3.1. This method is applied for a simple scroll event.

Listing 3.2: Point-weighting

```
+-----+
| | | | | | | |
+-----+
| | 3| 5| 5| 5| 3| |
+-----+
| | 5|10|10|10| 5| |
+-----+
| | 5|10|15|10| 5| |
+-----+
| | 5|10|10|10| 5| |
+-----+
| | 3| 5| 5| 5| 3| |
+-----+
| | | | | | | |
+-----+
```

Point-weighting on the other hand always has one cell in the center which gets the full weight. The weight of the surrounding cells is steadily decreased. The outer corners for example are always $1/5$ of the original weight. The radius of the point-weighting could be adapted, but a value of 2 cells showed good results. An schematic example is shown in listing 3.2.

The following list shows what kind of weighting is performed for which event, or which combination of events:

- The current event is a `CLICK` or `HOLD`
In both cases a point-weighting is performed. The weight to be used is the pressure multiplied with a certain value for `HOLD` respectively `CLICK`.
- The current event is a `ZOOM_IN`
This event causes an area-weighting of the cells in the viewport. The weight is higher than when using `DOUBLE_TAP` to zoom because the zoom factor is multiplied with the basic weight of this action. The zoom factor is in this case not the absolute zoom factor of the web page but the relative one, compared to the previous viewport entry.
- The previous event was a `DOUBLE_TAP` followed by a `ZOOM_IN`
A `ZOOM_IN` caused by a `DOUBLE_TAP` is a common action resulting in point-weighting but with less weight than a `ZOOM_IN` using a multi-touch gesture. This is done because a `DOUBLE_TAP` is sometimes performed on a specific area of the web page which could be of interest but often also just to increase the zoom in general.
- The previous event was a `ROTATION_TO_HORIZONTAL`
In this case a area-weighting is performed because a rotation to horizontal orientation indicates that the content was interesting enough to give it more space on the display. A rotation to vertical on the other hand does not result in a weighting because the user most likely changed the orientation after finishing reading. The reason to check the event type of the previous event is, that the rotation events do still contain the previous width and height which makes it necessary to wait for the next event.

- The current event is a scroll event
If the event is one of the eight scroll events an area-weighting is performed with the pressure and time factor as basis.

Implementation

The Java class representing the heatmap in the MVT project is called *Heatmap* and contains information about the web page, i.e. the URL, information about the validity of the heatmap represented by a start and end date, a screenshot of the web page, metadata, and the *importance matrix*. If a heatmap is created, some of these properties are initialized directly, e.g. the URL, and some asynchronously, e.g. the image is set as after the screenshot service (see section 3.4.4) generated it. The creation respectively the update of the heatmap is performed as soon as new data is retrieved from the client-side script and the post-processing was performed. The default case is that the same *Heatmap* object is used for all users, but it is also possible to generate specific heatmaps, e.g. just with the data of a specific user. These objects can also be retrieved by invoking the Restful web service.

Besides that, heatmaps are versioned, i.e. the current version of a heatmap can be archived and if new data is retrieved a new heatmap is created. At the moment there is no automatic mechanism to archive heatmaps, but it could be a future extension. If a web page changes too much, the identified important content loses its validity.

3.5.4 Heatmap Drawing

For drawing heatmaps, two different algorithms were created that both have their benefits and drawbacks. While the Raster Drawer is closer to the implementation and more technical, the Colourful Drawer has its focus on a pleasant graphical representation. The sources of the concrete implementation of both drawers are available in the Appendix A.1.1.

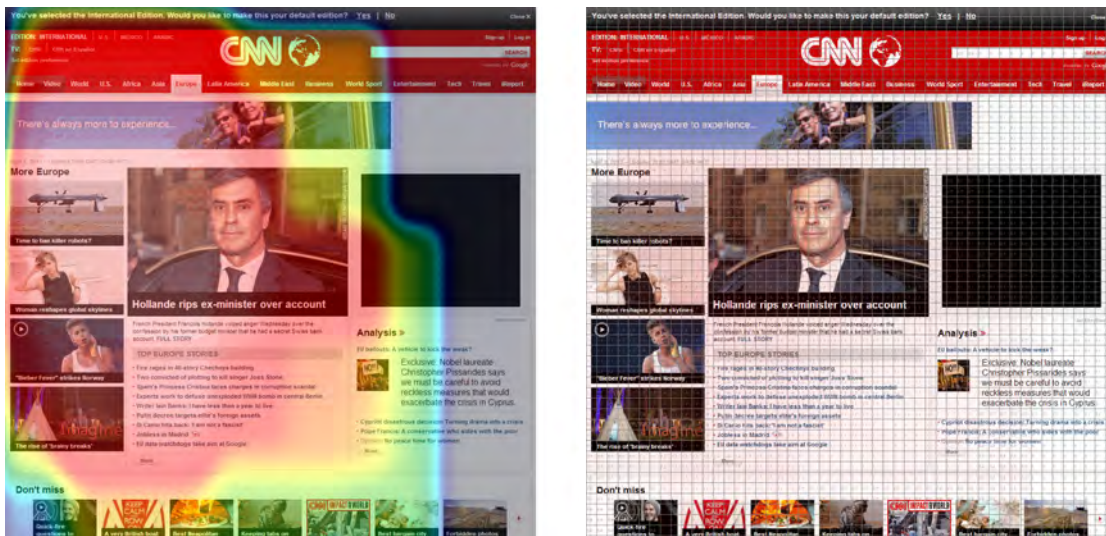


Figure 3.6: Comparing the Colourful Drawer (left side) with the Raster Drawer (right side)

Raster Drawer

As it was mentioned in section 3.5.3, *Mobile Viewport Tracking* uses a simple segmentation approach where the page is split into very small tiles (20px x 20px in this example). The Raster Drawer draws horizontal and vertical lines where the tiles are split. It also prints the logarithmized weight of this tile into the cell. The value is logarithmized with base 10, so the simple calculation of

$$weight = 10^{cellValue} \quad (3.1)$$

is needed to get the original value.

For visual representation, this drawer uses the opacity to represent the importance of a cell. It takes the ratio of the current cells value and the maximum weight of the heatmap to determine the opacity. A high opacity means a high importance, a low opacity means a low importance. This relationship is linear, so it is a simple percentage-value from the maximum.

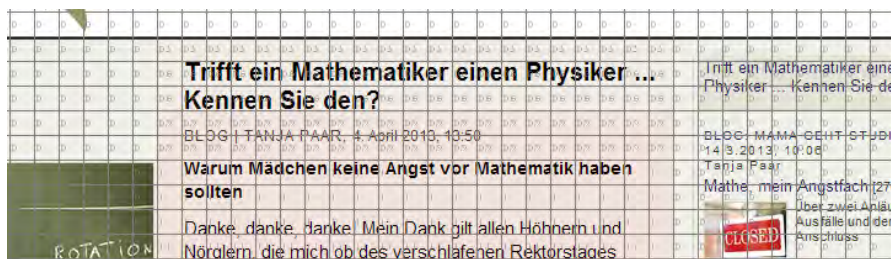


Figure 3.7: Raster Drawer in detail: the logarithmized weight and the opacity

As there is no projection of the values onto a colour gradient, it is not a heatmap according to the strict definition in section 2.4. The Raster Drawer can be used to see the real values of the heatmap.

Colourful Drawer

The Colourful Drawer has its focus on a pleasant graphical representation. It maps the value of the cells to a colour spectrum seen in figure 3.8.



Figure 3.8: The colour spectrum used to map the value to the resulting colour

The data is smoothed graphically by using a circle brush to create colour-gradients seen in figure 3.9.



Figure 3.9: The circle brush used to smooth data

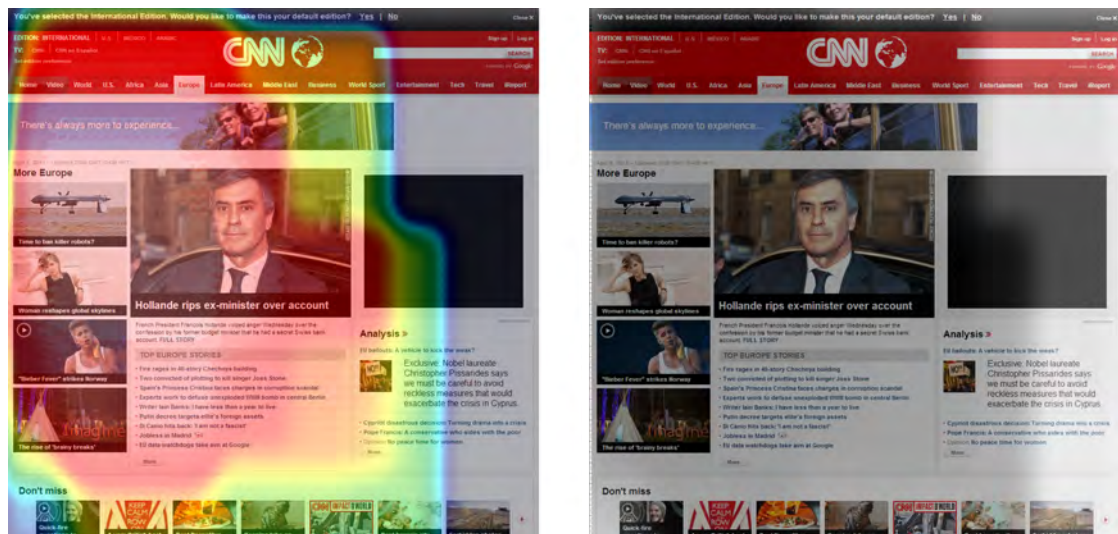


Figure 3.10: The colour scale is mapped from a transparent black and white image with different alpha-values (right side) to the colours of the colour-spectrum (left side)

First, an empty monochrome image is created (see figure 3.10). At the center of each tile, the centered brush is added with the relative weight (current weight / max weight) set as the opacity. In this manner, a smoothed version of the heatmap explained in section 3.5.4 is created. This

monochrome smoothed heatmap is mapped linearly onto the colour-spectrum shown in figure 3.8. The whole transformation is shown in figure 3.10. This leads to the final heatmap.

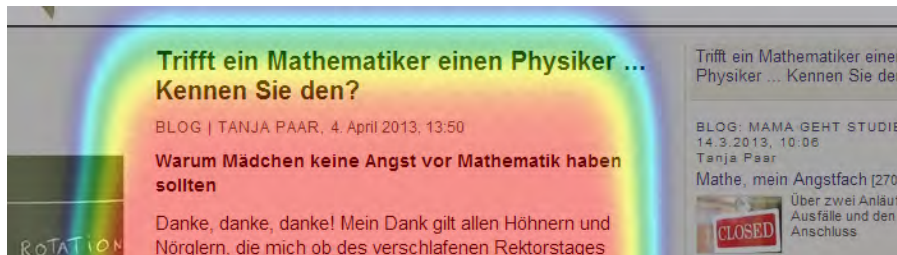


Figure 3.11: Colourful Drawer in detail

3.5.5 User Tracking

This section focuses on the business logic that handles the tracking of users. If the many viewport changes that occur could not be matched to a specific session even if it is only per page or domain, no useful analysis can be made. Therefore a tracking mechanism has to be used in order to group specific actions together and pin them to a specific session. The tracking of users is established using cookies. If users would not allow cookies than tracking would just be possible on the page level. The following section basically describes the different scopes of user tracking and their purpose within the tracking strategy.

Page

The first session identifier that has to be introduced is not a cookie. It is called page load token and it represents an unique id that is regenerated every time the page is loaded or reloaded. This means that also an F5 key press would trigger the creation of a new page load token. The simplest form of grouping actions together is by a single page load, which is exactly what the page load token is used for. This makes it possible to analyse all actions that took place on a specific page e.g. scroll down, zoom in, scroll, zoom out and so on. After the URL is changed by the user, or reloaded a new page load session begins.

Domain

The domain session token is actually a cookie. It is created for a specific domain to identify a user surfing multiple pages within a domain, including sub-domains. This ensures that different surfing sessions of the same user on the same domain can be tracked. Listing 3.3 shows the creation of a cookie which is available across sub-domains. In this example, the value of the cookie could be read on all pages within the "tuwien.ac.at" domain, e.g. `http://www.tuwien.ac.at` or `http://www.dbai.tuwien.ac.at`. All actions performed on these pages would belong to the same domain session. If the user changes the domain by e.g. clicking an external link, another cookie is created for the new domain if it does not already exist. Besides that, the session has a limited span of life, the default is 30 minutes. A special

case which always starts a new domain session, is if the referrer, i.e. the page the user is coming from, belongs to a different domain. The reason to create a new session is that the user may come to the page by following a direct link, e.g. from a search engine. In this case the actions performed by the user are seen as single sessions.

Listing 3.3: Domain cookie creation

```
1 document.cookie = "MVT=12345; expires=...; path=/; domain=.tuwien.ac.at"
```

Another token used on the domain level is the domain user token, which is similar to the domain session token, except that it has no timeout. This means it is an eternal cookie that is only deleted if the user decides to delete it manually. The cookie is used to identify an user that has once surfed on a specific domain and then later comes back to it and visits it again. It offers the possibility to track a specific user forever but restricted to a specific domain.

Global

The main problem of all the approaches that were mentioned above, is the inability to track a specific user over multiple websites which would enable interesting evaluations and analysis. The reason why this does not work or should not work, is because of the limitation of cookies. They can only be created on domain level and not for different domains. Fortunately, the usage of the proxy server, which is configured on the mobile devices of all tracked users, opens other possibilities to track users in another way.

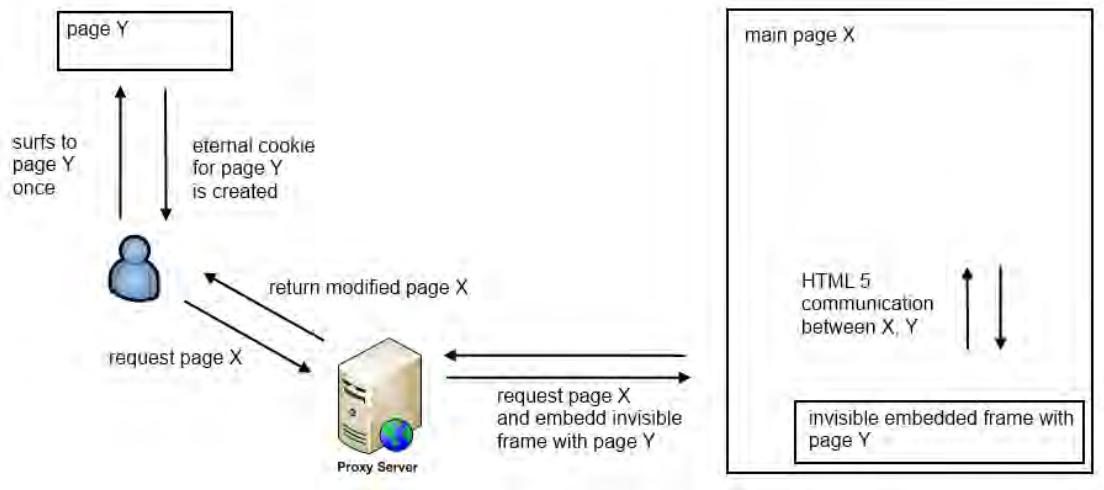


Figure 3.12: IFrame embedding trick using HTML5

Figure 3.12 shows a trick which enables the system to track users globally or across multiple websites. It uses the ability of the proxy server to change the content of requested websites. Mainly what the proxy does is, it requests the content of the desired web page and embeds an

invisible IFrame into the page that refers to our main registration page, which is called page Y in the figure. The user just has to visit the registration page Y once in order to create an eternal cookie for it, and after that he can be tracked. Every time the user visits a web page, the proxy server injects the invisible IFrame of page Y. In the client-side script it is now possible to communicate between the main page and the invisible IFrame using the HTML5 function *postMessage* and therefore retrieve the eternal cookie and identify the user uniquely.

This unique identification is a prerequisite for the server-side generation of the global session. As long as a user is performing an action on any website the same global session is used. Only if he becomes inactive and continues surfing at a later point in time a new session is created.

Registration Page

The registration page was already mentioned in the last section. It is used for creating an eternal cookie and also to give the user the possibility to register himself using a nickname. The registration page enables the user to map his eternal cookie to a self chosen name or number. This can be useful for later analysis of specific users or user groups. For Android users the global tracking cookie can be created automatically, without the users need to visit the registration page with the default system security settings. iPhone users have to visit it once, because otherwise the cookie is rejected by the devices operating system. Figure 3.13 shows a screenshot taken on an iPhone while surfing to the registration page.

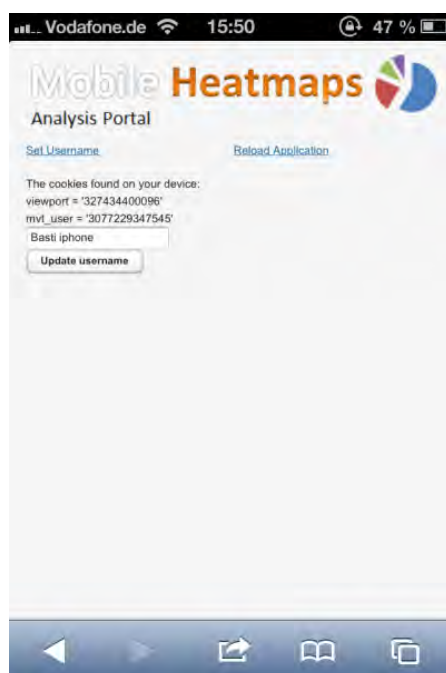


Figure 3.13: Registration page

3.6 Possible integration into other projects

Mobile Viewport Tracking was created under the premise of high integrability.

The core implementation is created as a Java library from which the drawing functions, business object design and persistence can be easily reused.

By using this library, a simple SOAP or REST based service interface could be created to expose data to other programming languages without the need of recreation of an underlying data access layer. There are innumerable usage scenarios for such a service.

Several REST methods are already provided by the current implementation of *Mobile Viewport Tracking*, e.g. a service method giving back the importance matrix of a given URL to be used for further website processing and repackaging.

As RDF¹⁶ and OWL¹⁷ are flexible data representation languages, a transformation from the given database to a triple store should be simple. With such a transformation, all benefits from using such a triple store could be used as exposing the data to the open data initiative or querying the data with powerful languages like SPARQL¹⁸.

The analysis web application offers the possibility to export all data to a simple CSV¹⁹ file. This CSV file is in flat data format which could be used as an input for common data mining tools. To ease usage, the transmitted data is already stored with enriched information (action type, session IDs) that could be used when the target is to improve such classifications (e.g. by a machine learning algorithm to determine the type of action).

¹⁶Resource Description Framework - <http://www.w3.org/RDF>

¹⁷Web Ontology Language - <http://www.w3.org/OWL>

¹⁸SPARQL Protocol And RDF Query Language - <http://www.w3.org/TR/sparql11-overview>

¹⁹Comma Separated Values - <http://www.ietf.org/rfc/rfc4180.txt>

State of the Art

Surfing the web does not only contain actions that are easy to observe like clicking a link or posting a message on a social network but also a lot of important unobserved actions [Punera and Merugu, 2010]. These actions could be e.g. reading some text further down the website and then coming back to the top and selecting the first result of a search.

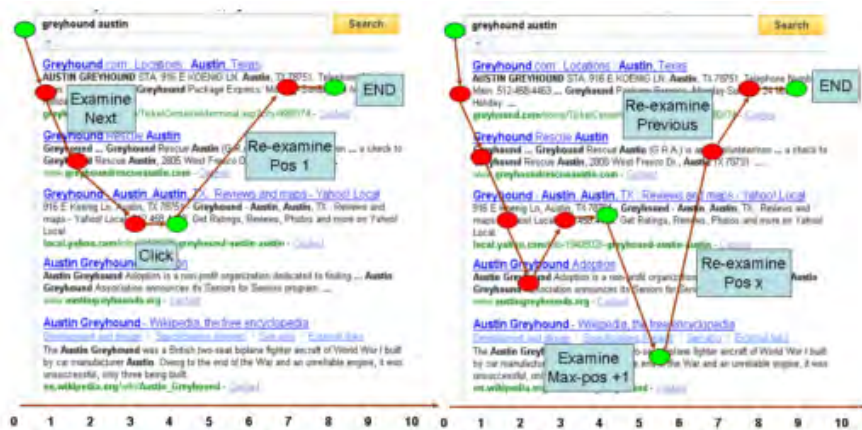


Figure 4.1: Unobserved User Actions [Punera and Merugu, 2010]

These hidden actions give additional information that is with usage of current technologies [Punera and Merugu, 2010] not very easy or even impossible to record. At the moment, the best tools available for studies of user interaction on websites which are capable of recording such unobserved actions are expensive eye-tracking-systems that use cameras to track eye movement. These systems are not used widely because of the high costs and complexity but the results of these studies seem very promising. According to Pan et al [Pan et al., 2004], web page viewing behaviour is driven by the gender of subjects, the order of web pages being viewed, and the interaction between site types and the order of the pages being viewed. Within this

study, a scanpath analysis was used to determine the user actions. Another approach is tracking mouse movements on desktop computers to generate heatmaps. According to Navalpakkam et al. [Navalpakkam and Churchill, 2012], movements of a mouse could be seen as a weak proxy for eye gaze mainly in the vertical direction but not in the horizontal direction. Also, this works only when users are not distracted [Navalpakkam and Churchill, 2012].

Tracking the small viewport of a mobile device can be seen as a scanpath analysis showing the movements of the viewport showing also the hidden actions and can be seen as an approach on what was relevant and what things where not. De Witts studies using eye tracking to determine impact of the Order Effect on website navigation items [DeWitt, 2010] showed some examples on what studies can be made using a system which is capable of scanpath recording like *Mobile Viewport Tracking*. As the approach shown in our prototype can be used without being recognized by the users of smartphones, also blind studies could be made which is not possible with multiple cameras focusing the user in eye tracking systems. As the user can only read what is shown on his display, we believe ¹ that our system does not show such a high deviation in areas viewed by the user as mouse tracking systems do.

Web beacon JavaScript tools like Google Analytics² or log file analysis or combinations of such tools [Ganapathi and Zhang, 2011b] can be seen as good tools to gain insight and optimize websites [Kumar et al., 2012]. But most of these systems focus on the inter-page-level. With the availability of data of user interaction within one page, a lot of possibilities emerge. Due to the mentioned limitations of existing systems like price of eye-tracking, no blind studies with eye-tracking systems, the high impact of distraction on mouse-tracking systems and the limitations of traditional analysis tools, it is believed that the system developed and its new approach on usability analysis can contribute to a better usability of the web.

4.1 Web Usability

Web Usability is an important field of study. If the first visit of a website is not satisfying, “40% of users will not visit the site again” [Nielsen, 2006]. This is an important change triggered by the technology change from desktop- to web applications. Web applications typically come with a try-before-you-buy model [Cloyd, 2001], whereas typical desktop applications were purchased before the user could use it. When the user already purchased the product, he will spend more time trying to find out how to solve his problems or get technical support [Cloyd, 2001]. This makes Web Usability design a must have nowadays [Cloyd, 2001].

The focus of Web Usability is a user-centered design [Yan and Guo, 2010]. This is defined by user characteristics, user types and levels of expertise [Badre, 2002] as well as researching the user goals [Yan and Guo, 2010] and general principles of navigation design [Yan and Guo, 2010].

¹It was not possible to get access to a eye tracking system to prove this statement as the equipment owned by the Vienna University of Technology was broken at the time this thesis was created

²<http://www.google.com/analytics>

4.1.1 Important Terms for Web Usability

A *visit* is the viewing of one or more pages on a web site by one visitor in sequence. The end of a visit is detected as a certain time of inactivity. Typically, this time is set to 30 minutes. After this time, it is assumed that the browser was closed or the visitor stopped interacting with the website [Khoo et al., 2008]. The terms session or user session are used conterminously. “A *unique visit* (as identified by IP address and/or persistent cookie) aggregates all visits made to a web site during a specified time-frame” [Khoo et al., 2008]. *Page views* count the number of times pages of a web site are accessed [Khoo et al., 2008]. “A *hit* is a request for a page or page element (image, frame, movie, etc.)” [Khoo et al., 2008].

4.1.2 Web Usability Evaluation

”Web evaluation is a measuring means for iterative process of Web usability design” [Yan and Guo, 2010]. This is typically done by the use of metrics.

Metrics

“Web metrics record and analyze visitor traffic to and through a web site” [Khoo et al., 2008]. Metrics are good mechanism for controlling the quality of a website [Kumar et al., 2012] as they help estimating if the visitors goals are being achieved [Khoo et al., 2008] leading to more quantitatively oriented decision-making [Weischedel and Huizingh, 2006].

Metrics are a core of usability and web design studies [Khoo et al., 2008]. This information can be used to improve websites. Typical data sources are:

- server logs / clickstream data
- customer surveys
- external data collected by web analytics firms

[Weischedel and Huizingh, 2006]

Incomplete data and interpretation problems occur more often, than one would expect [Weischedel and Huizingh, 2006]. Because of that, careful analysis and selection of measures is “necessary to ensure appropriate and meaningful use of this information” [Weischedel and Huizingh, 2006].

Metrics should have some essential properties to be useful:

- Simple
- Relevant
- Timely
- Instantly Useful: The user should find first insights as soon as he looks on it.

[Weischedel and Huizingh, 2006]

Some typical metrics were already mentioned in section 4.1.1 (visit, unique visit, page view, hit) - these are probably the most simple metrics to find. Other important metrics are upon others:

- *Bounce Rate*: The percentage of single page views (visits) of people not coming back to the website (this metric can also have different definitions, such as a visit that last less than 5 seconds).
- *Average Time*: Time on which the users stays on the website.
- *Percentage of New Visits*: The count of the number of sessions which new users visited website first time.

[Weischedel and Huizingh, 2006]

The usage of this data is critical to determine “priorities and needs related to the further development of websites” [Weischedel and Huizingh, 2006] and should be a base measurement for web analytics.

Web Quality Model (WQM) Web Quality Model (WQM) divides web metrics into three dimensions [Calero et al., 2005]. WQM proposed by [Calero et al., 2005] mentions the following dimensions which help classify possible metrics.

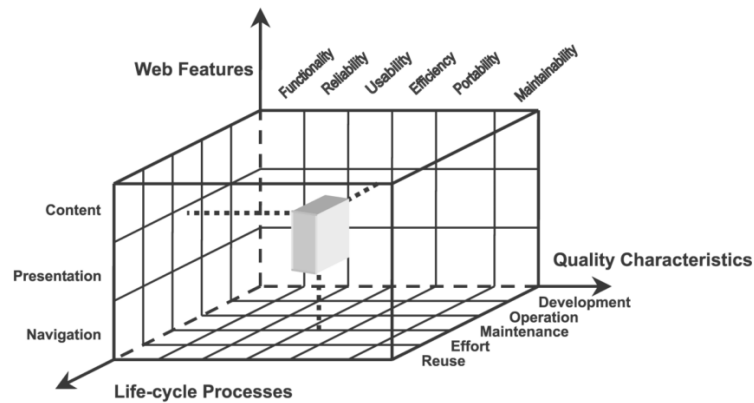


Figure 4.2: The Web Quality Model Cube [Calero et al., 2005]

- Web features dimension
 - content,
 - presentation
 - navigation
- Quality characteristics dimension
 - functionality
 - reliability
 - usability
 - efficiency
 - portability
 - maintainability

- Life-cycle processes dimension
 - development process
 - the operation process
 - and the maintenance process

[Calero et al., 2005]

The authors of the paper proposing the Web Quality Model identified 385 metrics [Calero et al., 2005] that fit into this model. Examples of such metrics are the 'Number of Navigational Links', 'Word Count', 'Ad Word Count', 'Horizontal scrolling', 'Invalid Links Count' or the '% of images with ALT text'.



Figure 4.3: The Web Quality Model Metrics Chart [Calero et al., 2005]

This model can help identify key metrics to be reported on a regular basis [Khoo et al., 2008].

Google Analytics Google Analytics is the most widely used web analytics tools with a market share of over 80% [W3Techs - Web Technology Surveys, 2013].

It uses a JavaScript beacon script that has to be placed into the website by the website administrator. Google Analytics provides a simple-to-use web interface to get an overview and typical reports. Many of the given metrics in the dashboard are meeting the criteria for useful web metrics from section 4.1.2.

It provides time series data [Plaza, 2011] that is possible to export, but it is not possible to import external data (e.g. server logs) [Fang, 2007]. With the use of exported data, it is possible to create statistically well-founded analyses like in [Plaza, 2011].

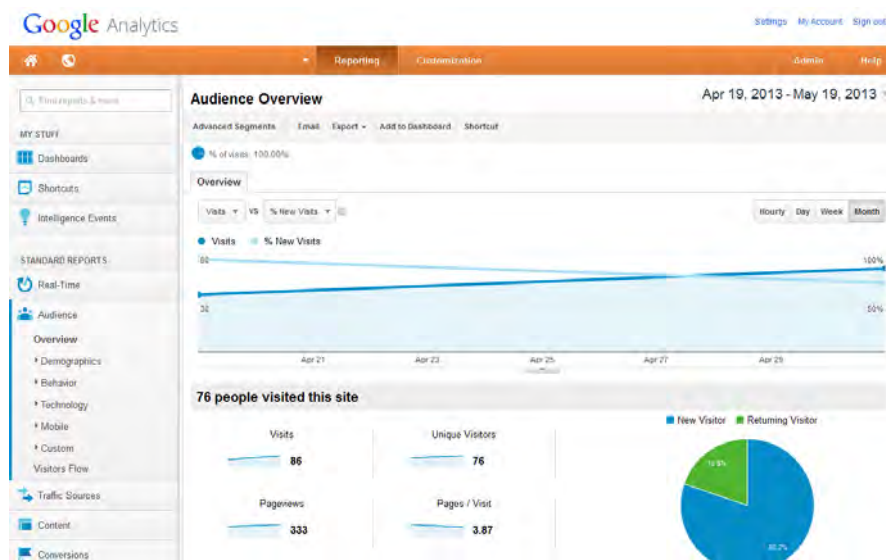


Figure 4.4: Google Analytics Dashboard

Google Analytics can be used in combination with other tools using custom variables [Clicktale, 2013a]. Google Analytics uses “aggregated non-personal information” to share with third parties [Fang, 2007] which could be a reason for website owners not to use Google Analytics.

One major drawback of Google Analytics is that it lacks support of tracking and analysis of page-level interactions.

Page-level interactions

Page-level interactions, which are behaviors on web search pages that do not hit the web server, are a useful supplemental source of search behavior data [Huang, 2011].

This could be cursor movements, scrolling, browser tab usage, text highlighting, or duration of a page view. These page level interactions can be recorded with minimal intrusion using JavaScript, which is built into all modern web browsers [Huang, 2011]. Typical interactions are:

- Cursor Activity
- Parallel browsing behavior
- Web browser metadata
- Accurate dwell³ time measurements

[Huang, 2011]

³German: to dwell: verweilen - Source <http://www.dict.cc>

Parallel Browsing Huang [Huang and White, 2010] found out that 57.4% of desktop surfing tab sessions involve parallel browsing, but users are multitasking by splitting their browsing activity into different tabs rather than viewing more pages overall.

4.2 Eye-Tracking

Eye Tracking is the field of study where most information about the participants attention can be recorded [Posner, 1980]. Different levels of cognitive processes, including oculo- motor and semantic processes [John M. Henderson, Phillip A. Weeks Jr, 1999] are involved when the eyes are moved.

Eye-Tracking systems typically use multiple cameras (there are also efforts to build systems with only one camera [Hennessey et al., 2006]) that are positioned around the display participants look on. These systems are called remote eye tracking systems as they do not have physical contact with the participant:

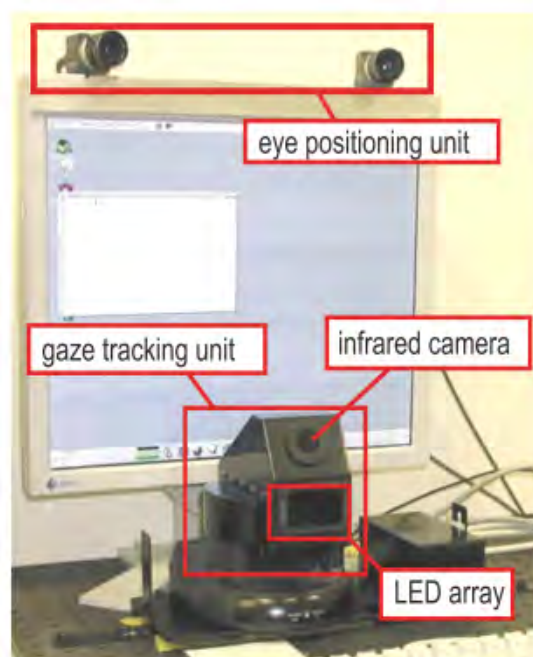


Figure 4.5: Prototype of a fixed Eye Tracking System [Ohno et al., 2003]



Figure 4.6: Commercial fixed Eye Tracking System - SMI Red 500 [SMI - Senso Motoric Instruments, 2013b]

Beside remote eye tracking systems, there are also systems that are mounted on the head of the participant. These systems allow free movement of the participant:



Figure 4.7: Prototype of a head-mounted low-cost Eye Tracking System [Li et al., 2006]



Figure 4.8: Commercial head mounted Eye Tracking System - SMI IVIEW X HED [SMI - Senso Motoric Instruments, 2013a]

4.2.1 Important terms

“People make rapid eye movements called *saccades*” [Irwin, 1998] about three or four times per second while reading or examining the world around them. Successive saccades are separated by *fixations*, during which the eyes are relatively still [Irwin, 1998]. *Fixations* are defined as a relatively motionless gaze⁴ lasting about 200-300 milliseconds (ms) [Rayner, 1998]. In this time, visual attention is aimed at a specific area of a display [Rayner, 1998]. “*Saccades* are rapid movements of the eyes with velocities as high as 500 degrees per second [Rayner, 1998]”. Saccades and Fixation times are dependent upon the task that should be done - e.g. “aspects of the text currently fixated” [Rayner, 1998]. Typical durations can be seen in table 4.1. Figure 4.9 shows the fixations and the *scanpath* that is the line that connects a “sequence of fixations and saccades” [Pan et al., 2004] “as a movement of attention” [Pan et al., 2004].

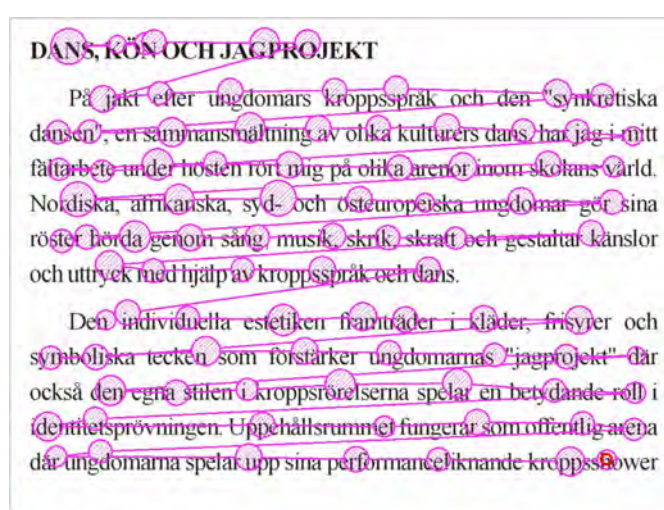


Figure 4.9: Fixations and Saccades taken from a study of speed reading made by Humanistlaboratoriet, Lund University, in 2005 using a SMI iView X 240 Hz [Humanistlaboratoriet, Lund University, 2005]

4.2.2 Perception

During eye movements, the sensitivity to visual input is reduced. This phenomenon is called *saccadic suppression* [Rayner, 1998]. During this time, we do not perceive information [Rayner, 1998]. *Masking* caused by the information available before and after the saccade makes it possible that we do not perceive any type of blurring effect [Brooks et al., 1981] [Campbell and Wurtz, 1978]. Due to the quick eye movements, it would be impossible to obtain new information during a saccade because the eyes are “moving so quickly across the stable visual stimulus that only a blur would be perceived” [Uttal and Smith, 1968]. One problem that limits eye-tracking is that:

⁴German: to gaze: starren - Source: <http://www.dict.cc/>

Although it is often necessary to move our eyes to identify objects in our environment, we can move attention without moving eyes [Posner, 1980].

This clearly limits eye-tracking as it should only be seen as a prediction of the participants attention - some inaccuracy that is impossible to avoid with this approach will stay. Even during saccades, the processing of information is not suppressed - “processing continues to take place while the eyes are in motion” [Irwin, 1998].

4.2.3 Metrics and Characteristics

Table 4.1: Approximate Mean Fixation Duration and Saccade Length [Rayner, 1998]

Task	Mean fixation duration	Mean saccade size (degree)
Silent reading	225 ms	2 (about 8 letters)
Oral reading	275 ms	1.5 (about 6 letters)
Visual search	275 ms	3
Scene perception	330 ms	4
Music reading	375 ms	1
Typing	400 ms	1 (about 4 letters)

The duration of a saccade is influenced by the distance covered. A two degree saccade which is typical for reading takes around 30 ms, whereas a five degree saccade, which is typical of scene perception, takes about 40-50 ms [Meyer and Kornblum, 1989] [Abrams and Jonides, 1988]. As it can be seen in table 4.1, the saccades and fixation times are dependent upon the task. The mean fixation duration is used frequently in eye tracking studies as an indication of information complexity and task difficulty [Rayner, 1998].

The *fixation duration* as well as the *saccade rate* can be seen as common metrics for eye tracking [Pan et al., 2004]. Other typical metrics are *fixations per 100 words* and the *frequency of regressions* [Rayner, 1998, Table 4]. All these metrics are dependent upon the abilities of the participant.

Table 4.2: Characteristics of eye movements dependent on school grade [Buswell, 1922]

Metric	1 grade	2 gr.	3 gr.	4 gr.	5 gr.	6 gr.	Adult
<i>Fixation duration (ms)</i>	432	364	316	268	252	236	252
<i>Fixations per 100 words</i>	182	126	113	92	87	87	75
<i>Frequency of regressions (percent)</i>	26	21	20	19	20	21	8

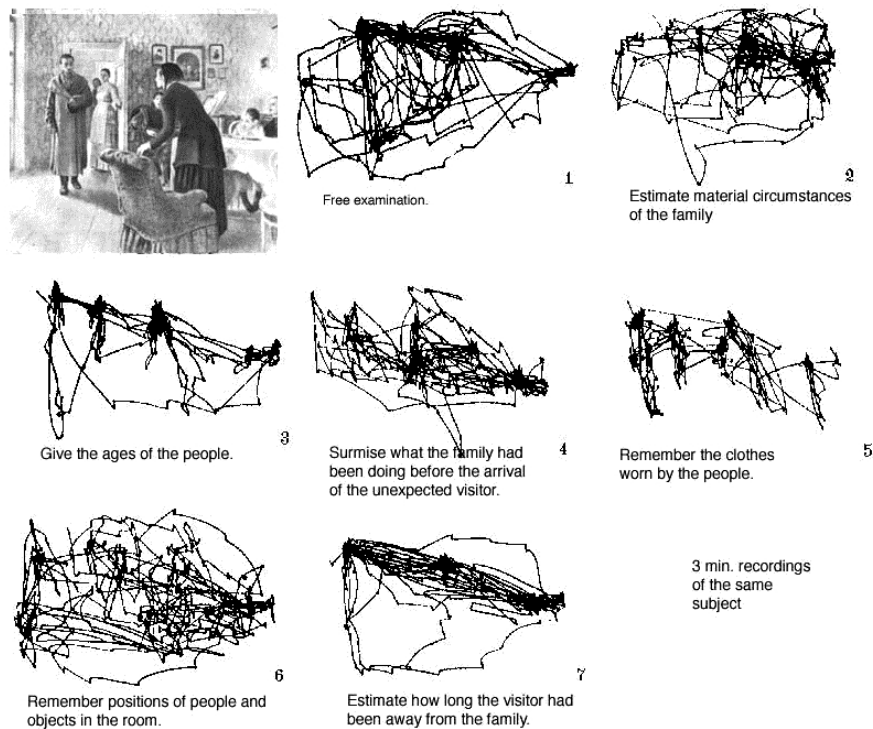


Figure 4.10: Yarus the visitor: Influence of the given task on eye movements [Yarus and Riggs, 1967] [Lucs-kho and Yarus, 1967]

The scanpath is also an important characteristic that can be measured via eye-tracking. Figure 4.10 illustrates how different given tasks lead to different scanpaths.

4.2.4 Problems

Most eye-tracking systems are expensive - the costs varied from 5.000 to 40.000 USD [Kumar, 2006] for commercial systems in the year 2006. Nowadays, first commercial systems are sold starting at about 2.500 USD⁵. A lot of research is done to build cheap solutions [Li et al., 2006] [Hennessey et al., 2006]. Homegrown solutions start at hardware costs of about 500 USD⁶ and will need a lot of time to build the system. There are also some solutions available that just use normal webcams for eye tracking. Typical scan rates of webcams are about 30 frames per second [Chau and Betke, 2005], leading to an detection interval of 33.3ms (1000 / 30). As stated in 4.2.3, typical saccade times start at about 30ms. As long as time resolution of these ultra-low-cost solutions is not significantly higher, these solutions will not be of peculiar interest for usability studies. Sony's Playstation Eye could be a solution to this problem as it delivers a data rate of 60fps (640x480) [Sony, 2013] leading to a time resolution of 15ms. A recently funded

⁵<http://www.grinbath.com/>

⁶<http://thirtysixthspan.com/openEyes/hardware.html>

kickstarter project (NUIA eyeCharm: Kinect to eye tracking ⁷) is targeting Microsofts Kinect to be used for eye-tracking. A major drawback is the data rate of only 30fps (640x480) [Janes, 2011] targeting “mainly interactions on Gaze-coordinates and fixations” ⁸.

Commercial systems have a typical data rate of 60Hz (16.6ms) to 120Hz (8.3ms) [Tobii, 2013]. Eye tracking systems typically need some calibration before using [Beauvillain and Beauvillain, 1995]. Even experienced researchers sometimes have problems with the calibration of eye tracking systems [Navalpakkam and Churchill, 2012].

4.2.5 Eye-Tracking Web Analytics

[Pan et al., 2004] did a study on Web Analytics using an Eye-Tracking system as it “is still not clear how different web users view web pages differently” [Pan et al., 2004]. Their key findings were that “Web page viewing behavior is driven by the gender of subjects, the order of web pages being viewed, and the interaction between site types and the order of the pages being viewed” [Pan et al., 2004]. These paths tend to drift over time for a given subject [Josephson and Holmes, 2002].

[Josephson and Holmes, 2002] and [Pan et al., 2004] state that important information is processed during the first 15 seconds. Only data from this time period was used for their analysis as it was found that 15 seconds are “sufficient for participants to completely examine the Web pages” [Josephson and Holmes, 2002].

Note: As MVT uses smartphones with limited screen size, this limit was made a soft limit devaluing the weight-increase on heatmaps as participants need some more time to orientate with this limitation after 20 seconds.

In eye-tracking web analytics, there are also surprising findings like the impact of the order of items of a navigation menu showed that:

There was no significant relationship between the distance of a menu item to the centre of the menu and the time elapsed before users first fixated on that menu item. There was a slightly increased performance when locating items in the first half of the menu as opposed to the second half [DeWitt, 2010].

or that help items are best placed on the right upper corner of the websites because people that lost orientation tend to look in this corner [Rakoczi,].

4.3 Mouse-Tracking

In this section, the term Mouse Tracking is not used in a strict manner. This section will deal with the tracking of mouse movements as well as mouse clicks, scroll tracking and form/input tracking as these tracking methods are often combined in the solutions available. In fact, most

⁷<http://www.kickstarter.com/projects/4tiitoo/nuia-eyecharm-kinect-to-eye-tracking>

⁸Email-Response of Tore Meyer (4tiitoo AG - owner of the project) to my question on the low framerate of Microsoft Kinect being not a good base for the solution created

of the solutions available have their focus on mouse clicks as this data is more easy to interpret and usually more descriptive than the movements itself. Currently some research is done in the field of interpretation of the mouse movements [Huang et al., 2011] itself.

Mouse tracking can be very useful on highly interactive sites like the check-out pages of web shops. On pages that are not very interactive like news pages, articles etc., mouse tracking is not the tool of choice to analyze user behaviour as shown in section 4.3.1.

4.3.1 Mouse- / Eye-correlation

The correlation between Eye and Mouse is not researched very well up to the present. There are only a few papers dealing with this field of research. According to [Mueller and Lockerd, 2001], 35% of people move their mouse cursor while reading a web page. This implies that the mouse cursor is an indicative to the users focus of attention [Arroyo et al., 2006]. When visiting websites of commercial mouse tracking providers, one will for sure read the value of 88% correlation multiple times - as well as the name 'Carnegie Mellon University'. This number is often just called correlation without any further information⁹. This paper describes a "study on the relationship between gaze position and cursor position on a computer screen during web browsing" [Chen et al., 2001]. In this work, a mouse-tracking system and an eye-tracking system were used on different web page types [Chen et al., 2001] to identify and evaluate Mouse- /Eye-correlation using five participants. The average distance between gaze and cursor range from 0.1 pixels to 1137.4 pixels, with 290.5 pixels in average [Chen et al., 2001]. "Of the regions that a mouse cursor visited, 84% of them were also visited by an eye gaze. Furthermore, among the regions that the eye gaze didn't visit, 88% of them were not visited by the mouse cursor, either" [Chen et al., 2001]. An example segmentation of a heatmap created with MVT can be seen in figure 7.12. The average dwell time on a page in the study was 8.9 seconds [Chen et al., 2001]. There are some points that could be criticized: In the year 2001 when this study was done, 95% of users had a display resolution lower or equal to 1024 x 768 pixels and 66% lower than or equal to 800 x 600 pixels [W3Schools, 2013]. An average distance of gaze and cursor of 290.5 pixels is a very high number as it is nearly one third / the half of the display in vertical direction. The study does not mention the monitor size and resolution used.

Another point to criticize is the average dwell time of 8.9 seconds. Our own studies had an dwell time of about 65.0 seconds (see section 7.1.2). One can argue, that due to the trend to single-page-websites, the website changed more frequently in 2001 than in 2013, but an average dwell time of 8.9 seconds seems very low, especially when taking into account that you would expect users of smartphones to have an lower dwell time than desktop users as it can be assumed that only important information is consumed when using a mobile device.

With a short dwell time of average 8.9 seconds per page it seems that the pages used in this study were highly interactive or the participants just clicked around, because they did not have a task to accomplish. As illustrated in section 4.3, highly interactive sites can be tracked much better with mouse tracking than other pages.

As most of the current studies do [Guo and Agichtein, 2010] [Huang et al., 2011] [Cox and Silva, 2006] [Rodden et al., 2008], also Googles studies [Rodden and Fu, 2007] target web

⁹<http://blog.clicktale.com/2010/10/14/eye-tracking-vs-mouse-tracking/>

search result pages (search engine result pages - SERPSs). This study found an median of 191 pixels and mean of 257 pixels (s.d. 237) of distance between eye gaze and mouse position, a slightly better result than [Chen et al., 2001] could achieve.

As a result, mouse tracking can be seen as having a weak relation [Navalpakkam and Churchill, 2012] between mouse and eye gaze as long as the web pages are highly interactive.

Websites that are not highly interactive - e.g. news websites, articles or blogs - cannot be tracked in a appropriate manner. This can be seen as one of the major limitations of mouse-tracking. For such sites, simple click-tracking still can give some very basic information but the movements of the mouse-cursor should not be taken into account.

4.3.2 Identified Movement patterns

There were three mouse movement patterns that where identified by current research [Arroyo et al., 2006] [Rodden and Fu, 2007].

While some people use their mouse pointer as a reading aid, other people move their cursor far away from what they want to read [Arroyo et al., 2006]. According to [Arroyo et al., 2006], this can help distinguish these groups of people [Arroyo et al., 2006]. Identified movement patters are:

- "Following the eye horizontally" [Rodden et al., 2008] - Using the Mouse as a Reading Aid [Rodden and Fu, 2007] behaviour was generally quite rare [Rodden et al., 2008].
- "Marking a particular result" was determined sometimes by most of the participants in the studies of [Rodden et al., 2008].
- Following the eye vertically in vertical direction [Navalpakkam and Churchill, 2012] behaviour was the most common of the active behaviours [Rodden et al., 2008]
- "Keeping the Mouse Still While Reading" [Rodden and Fu, 2007] - the user holds the mouse far away from the place where they are currently reading, keeping it still until they found a search result they want to click [Rodden and Fu, 2007].

These patterns could give useful information on usage behaviour, "but the automated analysis of this data is complex" [Rodden and Fu, 2007] and it was hard to classify users [Rodden and Fu, 2007] and identify metrics for classification. According to [Rodden and Fu, 2007], such metrics would be a prerequisite to conducting any larger-scale studies (e.g. with remote users) where only mouse data can be collected" [Rodden et al., 2008]. According to [Huang, 2011], an high amount of independent data could beat a sophisticated algorithm - this could be an approach solving this problem.

Only through such metrics and analysis, the mouse movements can give useful information. As it will be mentioned in section 4.3.4, most of the commercial providers simply draw movement maps without any reflection on the meaning and expressiveness of the data.

4.3.3 Distraction

According to Navalpakkam et al. [Navalpakkam and Churchill, 2012], movements of a mouse could be seen as a weak proxy for eye gaze mainly in the vertical direction but not in the horizontal direction [Rodden and Fu, 2007]. Also, high distraction leads to uselessness of mouse tracking [Navalpakkam and Churchill, 2012].

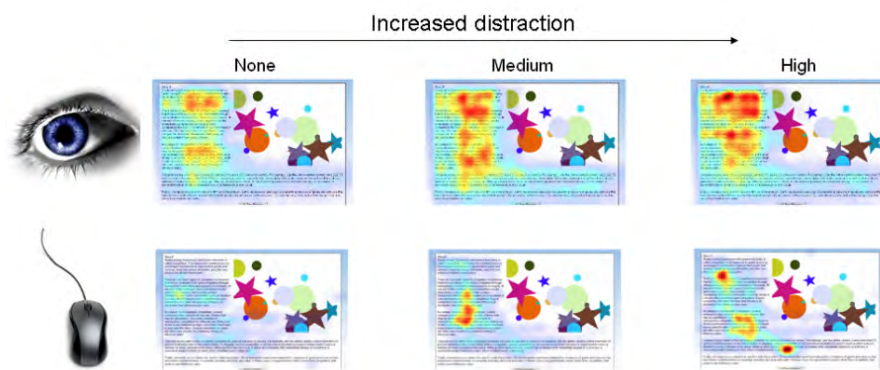


Figure 4.11: Increased distraction leads to uselessness of mouse tracking [Navalpakkam and Churchill, 2012]

Even when the correlation between eye and mouse is not always very high, mouse tracking can lead to useful information on pages that are highly interactive when the users are not distracted [Navalpakkam and Churchill, 2012]. Mouse tracking can also give very important information for web usability design on all pages:

It can show points where users clicked - even when it was not a link. Typical cases are Logos or Images that are not hyperlinked while a small text next to the image contains the link. This is a very simple information but it can be useful to reduce dissatisfaction of the user. One could make a hyperlink out of such elements to improve user experience.

This form of tracking also allows in-place click-tracking - Google Analytics uses your page-level data to track click maps [Crazy Egg, 2013], so with Google Analytics, it is only possible to say what pages the users visited but not on which link they clicked. If there are multiple links to a certain page, it is not possible to say which one was clicked. All mouse trackers used for this thesis support click tracking solving this limitation of Google Analytics.

4.3.4 Existing / Commercial Products

Mouse tracking is a field of study where the academic part - in comparison to the commercial part - is relatively small. As mouse tracking has a focus which is very close to the focus of the *Mobile Viewport Tracking* project, this section is handled in more detail. If *Mobile Viewport Tracking* would be a commercial product, this section could be seen as an overview of its competitors. Due to the overwhelming number of commercial providers, some of them are just mentioned in a light-minded manner.

USAProxy

One existing academic approach to mouse-tracking (including scrolling events) from the University of Munich is the Usability Proxy¹⁰. It is a very simple tool that is easy to set up but it lacks support for SSL, this leads to many problems even on not encrypted connections as referenced JavaScript from CDNs¹¹ and IFrames like the Facebook Likebox use secured connections, causing even simple web pages to fail. Also, the data is only recorded to log files, this makes it a little bit harder to evaluate. The source of the transmitted web page is stored to files - these files could be used to create screenshots of the web pages. Similar to *Mobile Viewport Tracking*, it uses a proxy approach to inject JavaScript into the web pages. This differentiates it from typical commercial solutions that use beacon scripts included by the website owner. All other solutions discussed here use the beacon script approach - they are included by the website owner.

Clickheat

Clickheat¹² is an OpenSource project which is just recording clicks [Labsmedia, 2013]. The interface is very simple but straight forward. Heatmaps are used for visualisation. As with the most other systems, the website is not archived, so the heatmap is always an overlay of the current version of the page.

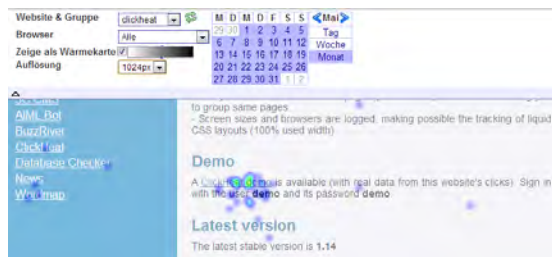


Figure 4.12: The Clickheat Interface [Labsmedia, 2013]

Crazy Egg

Crazy Egg is one of the cheapest commercial providers for mousetracking solutions, starting at 9 USD per Month billed yearly. Only clicks are tracked. This information is reshaped in a neat user interface. The heatmap is an overlay over the current version of the page¹³. Crazy Egg has an Confetti View where clicks of users with specific information like the referrer is differently coloured.

¹⁰<http://fnuked.de/usaproxy/>

¹¹Content Delivery Network

¹²<http://www.labsmedia.com/clickheat>

¹³www.crazyegg.com/

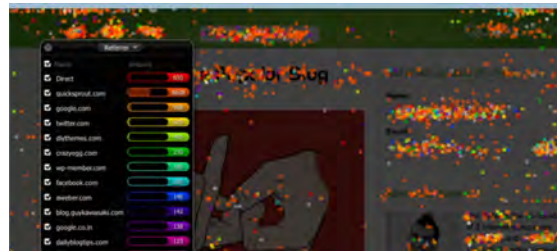


Figure 4.13: The Crazy Egg Confetti View [Crazy Egg, 2013]

Mouseflow

Mouseflow¹⁴ offers heatmaps, visibility maps and form analysis as well as recording and replay on user sessions (on one domain). Mouse moves, clicks and form inputs are recorded and can be replayed in realtime or fast forwarded. Their recording and playback service is very simple to use.

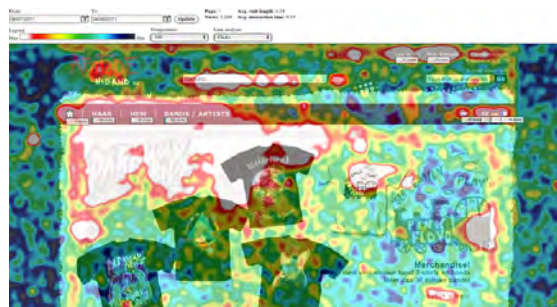


Figure 4.14: The Mouseflow heatmap View [Mouseflow, 2013]

m-pathy

m-pathy¹⁵ is targeting professional customers. They offer individual projects with consulting and optimisation studies. They offer the most complete approach in comparison to all other competitors. They are recording the delivered HTML document, so it is possible to precisely replay user sessions. Their product UX Insight mobile¹⁶ is in fact very similar to the approach used by MVT. It offers tracking of the current viewport. From their homepage, it is not clear if UX Insight mobile is ready to use or if it is in an early development stage as also diploma thesis topics with this focus are offered by m-pathy¹⁷. One important difference between m-pathy and

¹⁴<http://www.mouseflow.com/>

¹⁵<http://www.m-pathy.com/>

¹⁶<http://www.m-pathy.com/cms/m-pathy-ux-insight-mobile>

¹⁷http://www.m-pathy.com/cms/tl_files/m-pathy/Unternehmen/Themen_Abschlussarbeiten_052012.pdf

Mobile Viewport Tracking is that m-pathy just records the data but does not further process it ¹⁸.



Figure 4.15: The m-pathy heatmap view [M-pathy, 2013]

Clickdensity

Clickdensity¹⁹ is a simple tool that records clicks and mouse-overs. The feature-set is very limited and the pricing is above average. No replay, check-out recording or other more valuable functionality is present.

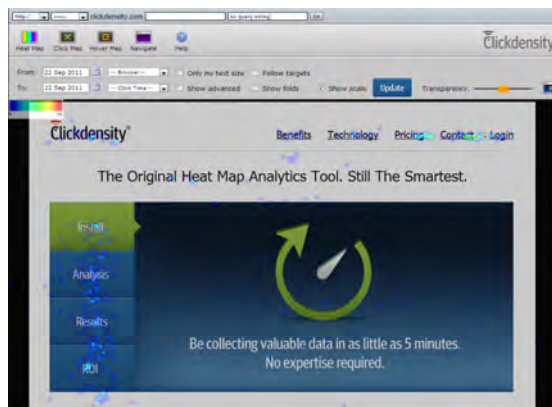


Figure 4.16: The Clickdensity heatmap view [Clickdensity, 2013]

ClickTale

ClickTale²⁰ is a very complete solution with heatmaps, filter-options, realtime monitoring, and web analytics. The core feature is a dashboard and multiple KPIs give a fast overview. Also integration with Google Analytics is possible.

¹⁸Contact to the CEO of m-pathy Tobias Kroha was established to get a access to the analysis platform. After some mails explaining the target of *Mobile Viewport Tracking*, no further response was given.

¹⁹<http://www.clickdensity.com>

²⁰<http://www.clicktale.com>



Figure 4.17: The Clicktale In-Page Web Analytics View [Clicktale, 2013c]

etracker

etracker²¹ is a web analytics suite that can include their mouse tracking solution 'Visitor Motion' - which creates heatmaps and videos of user sessions. Very interesting is the inclusion of a one-dimensional heatmap that records how deep user have scrolled down the page.



Figure 4.18: The etracker Visitor Motion Map [Etracker, 2013]

²¹<http://www.etracker.com>

4.4 Scrolltracking

[Atterer and Lorenzi, 2008] proposes a one-dimensional heatmap as a navigation aid. Only the vertical direction is tracked recording the time spent at certain scroll offsets.

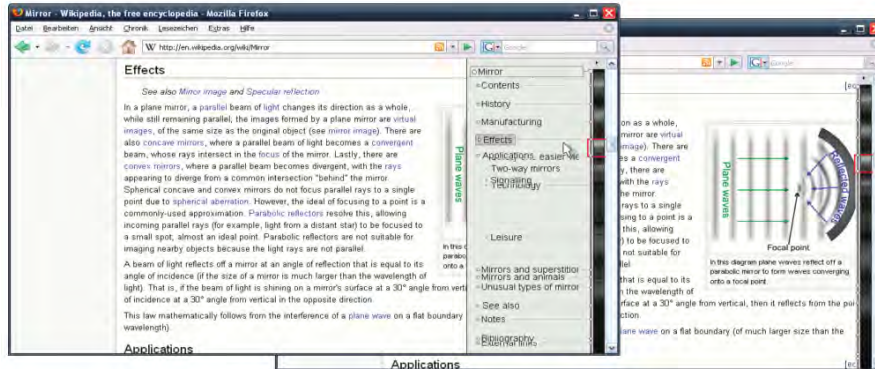


Figure 4.19: The scroll-heatmap navigation aid [Atterer and Lorenzi, 2008]

This approach is not only for usability studies but aims to help the user navigating within one web page to find "sections that have been visited earlier" [Atterer and Lorenzi, 2008], visualizing it next to the browser scrollbar [Buscher et al., 2010].

By looking at scrolling behaviour of documents, it was observed that documents are sometimes read linearly, sometimes parts are skipped, and sometimes parts are read multiple times [Iga and Shinnishi, 2006].

This information could be useful to be evaluated by website-owners - even if it is simple, it can deliver useful information. Since this information could be processed automatically, it would be feasible to create simple indicators of the websites average scroll depth, how many sections were read multiple times - they could be difficult to understand and many more.

4.5 Viewport Tracking on the Desktop

Langun [Lagun and Agichtein, 2011] proposed:

ViewSer, a novel methodology for performing web search examination studies remotely, at scale, and without requiring eye-tracking equipment [Lagun and Agichtein, 2011].



Figure 4.20: The ViewSer interface showing “a blurred search engine result page (SERP) for the query “toilet”, with the viewport revealing the first result” [Lagun and Agichtein, 2011]

Though the ViewSer’s restriction of the viewport, it is obvious that only the information shown in the available viewport is read actively by users/participants. Participants can select the entry to unblur with arrow-down / arrow-up key presses. This leads to data that is easy to interpret. As the viewport is an area as wide as the whole page, the heatmap generated from this data is one-dimensional.

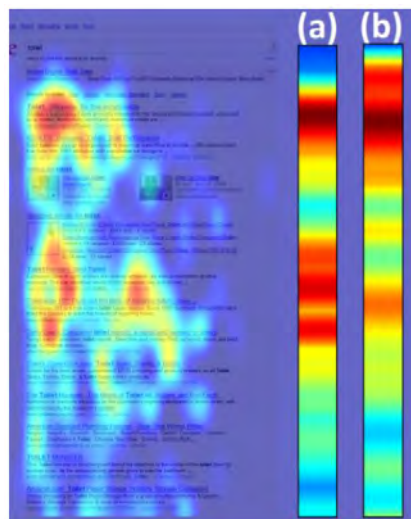


Figure 4.21: A heatmap showing the eye-tracking heatmap in background and its one-dimensional projection (a) compared to the heatmap generated by ViewSer (b) [Lagun and Agichtein, 2011]

It is conceivable that the ViewSer approach could be adapted to be used with arbitrary web pages. The commercial project Spotlight Viewer²² uses the mouse position to unblur the current position. The unblurred area is about as big as a 2 Euro coin [Dr. Berger Consult KG, 2013].

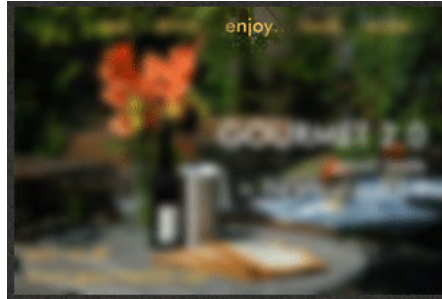


Figure 4.22: The Spotlight Viewer Interface [Dr. Berger Consult KG, 2013]

The Spotlight viewer started as a Ph.D.-thesis on the University of Vienna. It was founded by the “Hochschuljubiläumsstiftung der Stadt Wien” [Dr. Berger Consult KG, 2013] and won the “Triconsult Preis für innovative Marktforschung TIM 2011” [Marketing Department, 2013].

One major drawback of these approaches is that the user experience is affected - this limits the usage of ViewSer and Spotlight Viewer to usability studies with a limited number of participants as nobody would use a browser with such a limitation. Also the normal reading behaviour is impaired by the use of these systems so results can be swayed.

The idea behind ViewSer and Spotlight Viewer is very similar to MVT - the limitation of the viewport is just created in an artificial way while MVT uses a natural limitation (screen size). One major benefit of ViewSer and Spotlight Viewer compared to MVT is that the viewport is significantly smaller than the viewport of current smartphones, so data is more precise and more easy to interpret.

²²<http://www.mindspotter.at/services.html>

Setup of Mobile Viewport Tracking

5.1 Client Side Configuration

The configuration of mobile devices for usage with *Mobile Viewport Tracking* is as simple as the configuration of a proxy server on the device. If *Mobile Viewport Tracking* is used as a beacon script included in a web page or the network used by the device uses the proxy server as transparent proxy, no configuration is needed on the mobile device.

Figures 5.1 and 5.2 show the configuration tutorial as given to the participants of the case study.

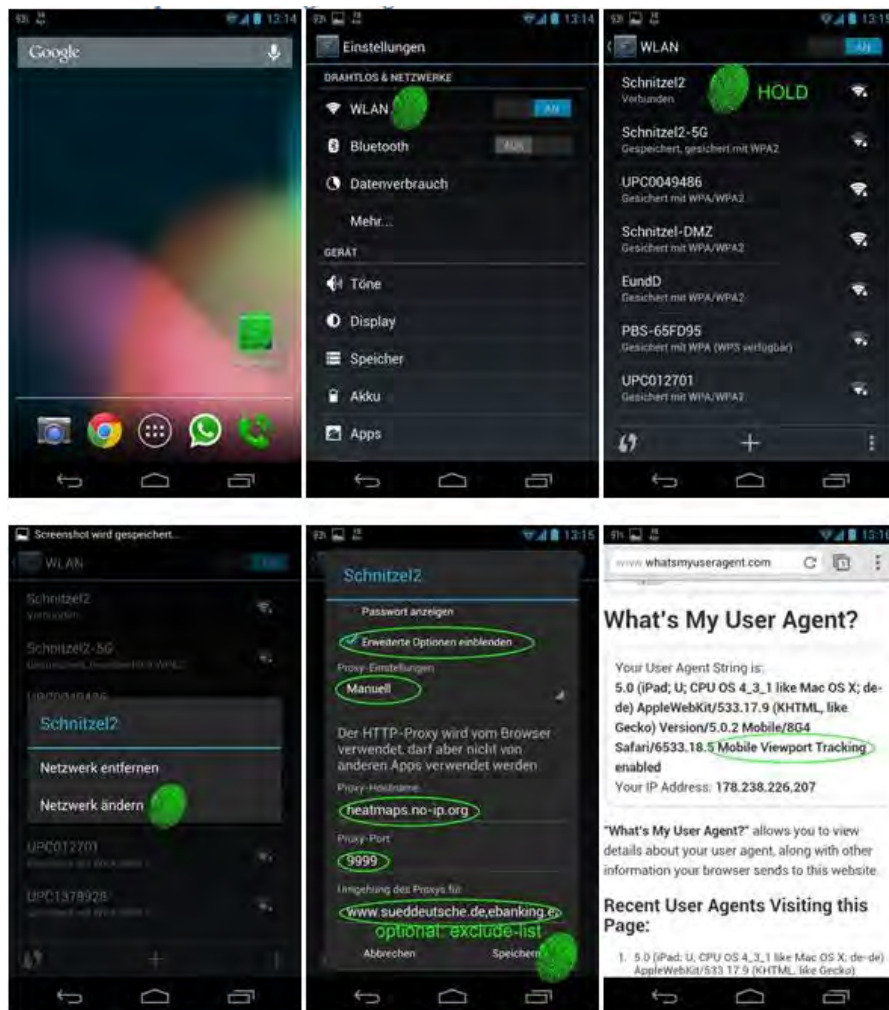


Figure 5.1: Configuration Tutorial for Android 4.X



Figure 5.2: Configuration Tutorial for iOS

5.2 Server Side Configuration

The server side of *Mobile Viewport Tracking* consists of five Java projects:

- *MVTPProxy*: The MVTPProxy is a fork of the PAW Proxy server ¹. The project had to be forked because of crucial bugfixes (these were later on returned to the original project) and the removal of unneeded end user configuration tools. The proxy server is a simple Java application that can be started as follows:

Listing 5.1: Proxy start command

```
1 java -jar publish/MVTPProxy.jar -p 8080
```

where 8080 has to be substituted by the desired port.

¹<http://paw-project.sourceforge.net/>

The file `conf\reg-filter.xml` has to be adjusted to point to the IP address or host-name of the MVTService (in this example: `heatmaps.no-ip.org`) at the three occurrences in the document. If your servlet container (Tomcat) is running on port other than 80, be sure to add the port to the URL (e.g. `tomcats_base_url:8000`)

Listing 5.2: Configuration file `reg_filter.xml`

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <reg-filters>
3   <reg-filter type="replace" status="active">
4     <name>MVTInjection</name>
5     <description>Injects the Javascript for Mobile Viewport Tracking</description>
6     <regexp>
7       <match>&lt;\[/\[*head[ ]*&gt;</match>
8       <subst>&lt;script type="text/javascript" src="http://heatmaps.no-ip.org/
          MVTService/viewport.js"&gt;&lt;/script&gt;&lt;script type="text/
          javascript"&gt; MVT.init("heatmaps.no-ip.org"); &lt;/script&gt;&lt;/
          head&gt;</subst>
9     </regexp>
10  </reg-filter>
11  <reg-filter type="replace" status="active">
12    <name>MVTUserInjection</name>
13    <description>Injects the IFrame for Global Cookies</description>
14    <regexp>
15      <match>&lt;\[/\[*body[ ]*&gt;</match>
16      <subst>&lt;iframe src="http://heatmaps.no-ip.org/MVTService/mvt_user.html"
          height="1" width="1" style="visibility:hidden; top:-10000; left
          :-10000;" id="mvt_iframe"/&gt;&lt;/body&gt;</subst>
17    </regexp>
18  </reg-filter>
19 </reg-filters>

```

- **MVTPersistence:** The MVTPersistence project is a decoupled library for persistence tasks using the OR mapper hibernate and business-logic like heatmap drawing and action type determining. Its configuration is based on the `persistence.xml` file for hibernate. The configuration is similar to other hibernate based projects ². Main configuration properties are the `hibernate.dialect`, the `hibernate.connection.url`, the `hibernate.connection.username` and the `hibernate.connection.password`.

Listing 5.3: Hibernate configuration file `persistence.xml`

```

1 <?xml version='1.0' encoding='utf-8'?>
2 <persistence xmlns="http://java.sun.com/xml/ns/persistence"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://
      java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
5   version="2.0">
6   <persistence-unit name="mvtManager" transaction-type="RESOURCE_LOCAL">
7     <provider>org.hibernate.ejb.HibernatePersistence</provider>
8     <class>at.tuwien.viewport.persistence.entities.ViewportEntry</class>
9     <class>at.tuwien.viewport.persistence.entities.Heatmap</class>

```

²http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/ch01s02s01.html

```

10 <class>at.tuwien.viewport.persistence.entities.ImportanceMatrix</class>
11 <class>at.tuwien.viewport.persistence.entities.Image</class>
12 <properties>
13   <property name="hibernate.cache.use_query_cache" value="no"/>
14     <property name="hibernate.cache.use_second_level_cache" value="no
15       "/>
16       <property name="cache.provider_class" value="org.hibernate.cache.
17         NoCacheProvider"/>
18     <property name="hibernate.dialect" value="org.hibernate.dialect.
19       MySQLDialect"/>
20   <property name="hibernate.hbm2ddl.auto" value="update"/>
21     <property name="hibernate.connection.driver_class" value="com.
22       mysql.jdbc.Driver"/>
23     <property name="hibernate.connection.url" value="jdbc:mysql://
24       localhost:3306/mvt?autoReconnect=true"/>
25     <property name="hibernate.connection.username" value="mvtuser"/>
26     <property name="hibernate.connection.password" value="
27       some_secret_password"/>
28     <property name="hibernate.connection.pool_size" value="10"/>
29 </properties>
30 </persistence-unit>
31 </persistence>

```

The screenshot component needs some additional configuration when it should be used (e.g. from MVTService) - mainly the paths to the browsers that should be used. The current implementation supports Firefox and Google Chrome. The `browser` property is used to select the browser (*FF* for Mozilla Firefox, *Chrome* for Google Chrome). When using Google Chrome the Selenium Chrome Driver³ has to be in the same directory as the browsers executable. The `screenfolder` is the path where the screenshots are stored. `screenshotOnVisit` should be set to *true* if a screenshot should be taken for every page visit. If it is set to *false*, screenshots are created only when a new heatmap is created. These screenshots are stored in the database.

Listing 5.4: The `config.properties` configuration file

```

1 # Use forward slashes
2 screenfolder = C:/MVT/Images/
3 firefoxExe = C:/MVT/Browser/Firefox15/firefox.exe
4 chromePath = C:/MVT/Browser/chrome/
5 # "FF" or "Chrome"
6 browser = Chrome
7 heatmapFolder = C:/MVT/Heatmaps/
8 heatmapUserSessionFolder = C:/MVT/UserSessions/
9 screenshotOnVisit = true

```

- **MVTService:** The MVTService is based on the MVTPersistence and exposes some Web Service endpoints like the one the injected JavaScript is sending the user actions to. It has to be deployed as a WAR⁴-file on a servlet container like Apache Tomcat. The configuration is done via the contained MVTPersistence.

³<https://code.google.com/p/selenium/wiki/ChromeDriver>

⁴Web Module - <http://docs.oracle.com/javaee/5/tutorial/doc/bnadx.html>

- *MVTRegistration*: The MVT Registration page is a decoupled part of the MVTAalysis application. Its purpose is to allow users the registration of their username and the creation of the global tracking cookie for iOS devices. No configuration is needed as the configuration of the included MVTPersistence is used. This component is not needed in all cases as the same functionality is provided via the MVTAalysis application.
- *MVTAalysis*: The MVT Analysis application which was developed as a simple analysis tool for case studies which is further described in chapter 6 does not need its own configuration as the configuration of the included MVTPersistence is used. It is dependent on the user roles of the servlet container (e.g Apache Tomcat). For normal access, the user should have the *webapp* role, for admin rights, the *webappadmin* role should be given.

Listing 5.5: Tomcat configuration file `tomcat_users.xml`

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tomcat-users>
3     <role rolename="webapp" />
4     <role rolename="webappadmin" />
5     <user username="exin" password="some_secret_password" roles="webapp" />
6     <user username="admin" password="some_secret_password" roles="webapp,
7         webappadmin" />
8 </tomcat-users>

```

5.3 Build and Run Mobile Viewport Tracking

To build and run the project from the source code, a small guide is given:

1. *Check-out from source code management system*: The source code management system used is Mercurial. Pull and update from the repository. Each of the projects uses its own repository. For each project needed, run:

Listing 5.6: Check out using Mercurial Console

```

1 hg init
2 hg pull pathToRepository
3 hg update

```

to get a working copy of the project.

2. *Import into Eclipse*: For development, the Eclipse IDE was used. Import the projects into an Eclipse workspace by right clicking the Project Explorer and selecting *Import -> Import -> General -> Existing Projects into Workspace*. Select browse and browse to the the root directory of the projects. Select all projects that should be imported. Optionally, select *copy projects into workspace*, then select *Finish*.
3. *Select the targeted runtimes*: If your target runtime is not called Apache Tomcat v7.0, then you have to set the target runtime to the servlet container you are using by right clicking

the project and selecting *Properties* -> *Targeted Runtimes* and selecting the runtime of your choice. It is recommended to use Apache Tomcat 7.0.

4. *Fix project references*: If the project folders were not renamed after checkout - the dependencies between the projects are set correctly. If project folders were renamed, please be aware that the MVTSservice and MVTAnalysis projects need the MVTPersistence project on their build path. Set it via right click on the project -> *Properties* -> *Java Build Path* -> *Projects* -> *Add* and selecting MVTPersistence and -> *Properties* -> *Deployment Assembly* *Add* and again selecting MVTPersistence. Store via *OK*.
5. *Download the Vaadin Plugin*: If not already installed, download the Vaadin Plugin via *Eclipse Marketplace* or *Install New Software* and entering the repository url `http://vaadin.com/eclipse`. This step is optional but the plugin helps updating the Vaadin / GWT version. The current versions of libraries are included in the repository. This only has an impact on Vaadin based projects (MVTAnalysis and MVTRegistration).
6. *Configure Projects*: Follow the instructions in section 5.2 to configure the applications. Be sure to set the database credentials correctly. The database schema is automatically created if it does not exist.
7. *Build using Eclipse*: All projects can be build via Eclipse. The JAR-file of the MVTProxy can also be build with Apache Ant as a build script is provided. For the Web applications, this is optional as the projects will be built automatically when exporting.
8. *Export*: Export MVTAnalysis, MVTRegistration and MVTSservice as WAR files via right-click on the project -> *Export* -> *War file*.
9. *Deploy*: Deploy the WAR files via Tomcats Admin Web Page or via Tomcats file system watcher. Deploy the MVTProxy by copying its JAR-file on the server and starting it via a command shell by running

Listing 5.7: Proxy start command

```
1 java -jar publish/MVTProxy.jar -p 8080
```

10. *Use*: Configure the smartphones as described in 5.1. The MVTAnalysis application can be found at `http://tomcats_base_url:tomcat_port/MVTAnalysis`, the registration application can be found at `http://tomcats_base_url:tomcat_port/MVTRegistration`.

The server firewall has to forward communication on the ports of the servlet container - this is typically port 80 - and the port of the proxy server - 8080 in the default configuration. Additionally, outgoing requests have to be permitted for the screenshot component - Mozilla Firefox or Google Chrome.

Implementation of an Analysis Application

6.1 The Need of such an Application

The shared part of the *Mobile Viewport Tracking* Project ended where the data was analyzed, categorized and stored into a database. In order to visualize the data recorded, also heatmap drawing was implemented.

In order to increase the user base of this project to gain data for analysis, a simple web application was created which gives access to the data and drawing mechanisms.

In the course *Applied Web Data Extraction* in Summer Term 2013, it was one of the available homeworks to use MVT for at least 30 minutes and do some simple tasks (see section 7.1.1). To convince students to select this task, they got a small demonstration of the Analysis Application and access to the system - as soon, as they completed their tasks.

Heatmaps cannot only be drawn from the predefined heatmaps from MVT (which are aggregations of all data for a specific URL from a start date to an end date) but also on-the-fly from selected data. Also some simple charting should be implemented to show the usability of the data created. It was not target of this master thesis to develop a fully fledged application but to show some points how such an application should look like today.

6.2 Technical Background / Implementation

The persistence layer used in MVT (based on hibernate OR-mapper) was reused for this web-application (see 3.2.7). The Vaadin framework¹ which is based on the Google Widget Toolkit GWT² was used to create the application.

¹<https://www.vaadin.com/home>

²<https://developers.google.com/web-toolkit/>

6.3 Features

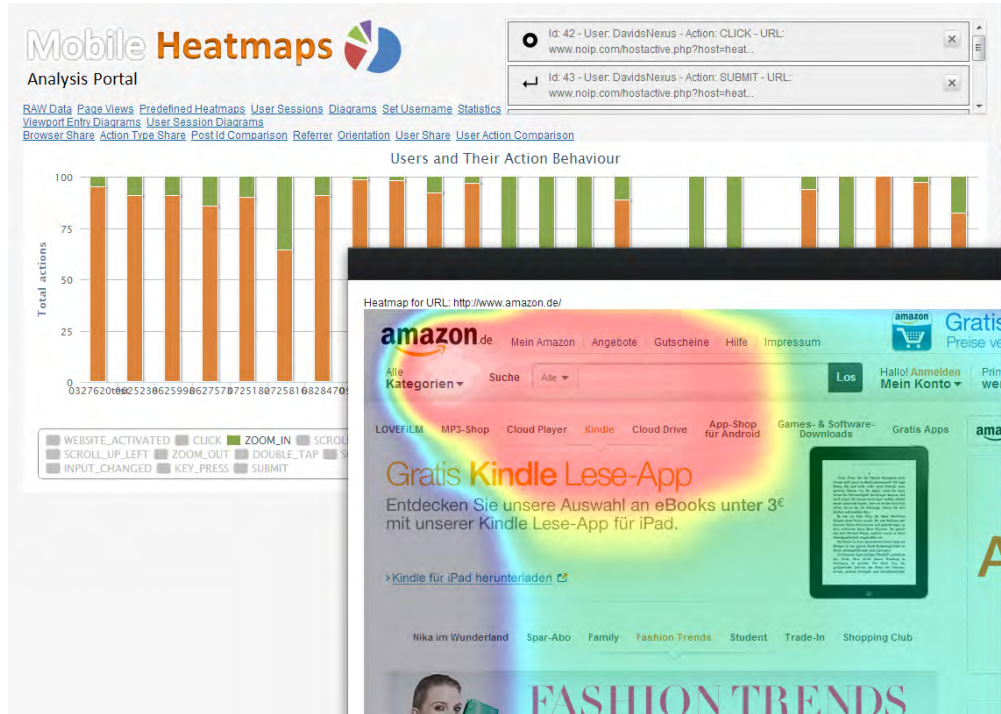


Figure 6.1: The Mobile Heatmaps Analysis Portal in an overview

Fig. 6.1 shows the Mobile Heatmaps Analysis Portal. The Update Panel is positioned on the upper right.



Figure 6.2: The Update Panel where new Information is pushed to the users to see what is currently happening

Figure 6.2 shows the Update Panel where new entries are shown. The type of action, the user, the URL and the PostId is shown. The main target of the update panel was to visualize what is happening in live demonstrations. It takes about two seconds to see the action done on the smartphone popping up in the Update Panel.

VIEWPORTENTRYCOUNT	URL	VIEWPORTENTRYCOUNT	RECORDTIME	MOBILEACTION	VIEWPORTENTRYCOUNT
319	http://www.sch.com/de/leistungen/operateuraktivitaet.aspx?group=...	0029428	2013-04-10 20:41:00.0	CLICK	1
293	http://www.vogel-ohne.de/flag.htm	0029428	2013-04-10 23:26:54.0	CLICK	1
256	http://mobi1.denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-04-02 00:39:08.0	CLICK	1
191	http://www.ski.at/ski-angebote/2015/04/06/...	0029428	2013-04-08 08:10:48.0	CLICK	1
167	http://www.smeil.gu.at/index.php?module=1002	0029428	2013-04-08 06:31:58.0	CLICK	1
165	http://appreak.com/	0029428	2013-04-10 09:50:36.0	CLICK	1
163	http://mobi1.denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-04-02 00:39:08.0	CLICK	1
161	http://www.ski.at/ski-angebote/2015/04/06/...	0029428	2013-03-31 21:27:33.0	CLICK	1
160	http://mobi1.denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-04-02 00:39:08.0	CLICK	1
156	http://www.ski.at/ski-angebote/2015/04/06/...	0029428	2013-04-08 08:10:48.0	CLICK	1
154	http://mobi1.denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-03-29 13:33:37.0	CLICK	1
150	http://denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-04-10 14:44:34.0	CLICK	1
150	http://denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-04-04 22:19:02.0	CLICK	1
148	http://mobi1.denstandard.at/13379134031/Content/Widget-Cen-Link	0029428	2013-04-04 22:14:06.0	CLICK	1
144	http://mobi1.denstandard.at/	0029428	2013-04-10 09:57:31.0	CLICK	1

Figure 6.3: Page views shown in a list

Viewport Entries are shown as given in the database, Page views (one user on one URL) and aggregated predefined heatmaps are shown in filterable lists (see fig. 6.3) where it is possible to search or filter according to the specific needs. Columns that are not important for most research are hidden and can be added to the grid. Heatmaps and Rastermaps of the specific data set can be shown via the actions in the grid - these are generated on-the-fly.

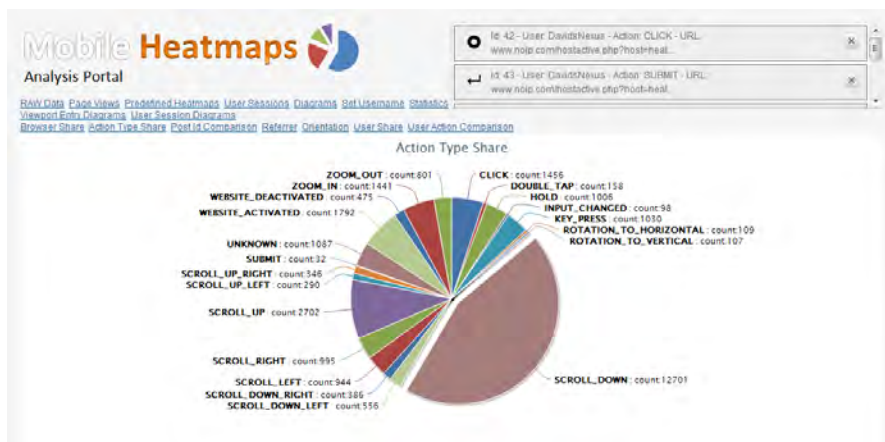


Figure 6.4: A pie chart showing the share of Action Types

Charts help to visualize the data (see fig. 6.4). More complex charts also allow to interact with the chart and to add or remove data sets like action types. (See fig. 6.5)

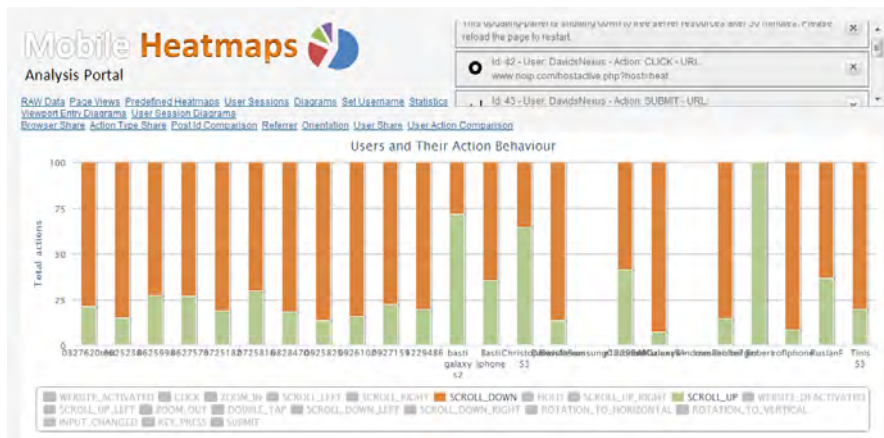


Figure 6.5: A stacked bar chart showing the share of Action Types for each user, only Scroll Up and Scroll Down are shown by default, others can be added/removed by selecting them in the chart

AVERAGE PAGE VIEW DURATION	ACTION COUNT	DOMAINS VISITED	URLS VISITED	DURATION IN SECONDS
19	10	1	1	19
136	31	1	1	136
171	183	3	10	513
22	5	1	2	22
4	7	1	1	4
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
80	3	1	1	80
0	1	1	1	0
153	161	1	3	153
22	8	1	1	22
7	2	1	1	7
0	1	1	1	0
0	1	1	1	0
55	44	4	4	55

Figure 6.6: The average page view duration for each user session - Min, Max and Mean are also calculated and shown at the end of the table

The Analysis Portal makes use of the user and role management of the servlet container (Apache Tomcat³). Different roles were needed as every participant got access to the analysis portal after he ended his studies but only had access to viewing functions and no access to administrative functions like the finalisation of heatmaps (A predefined heatmap can be finalized for multiple reasons - e.g. a new study beginning or a layout change on the website which would lead to inconsistent data).

³<http://tomcat.apache.org>

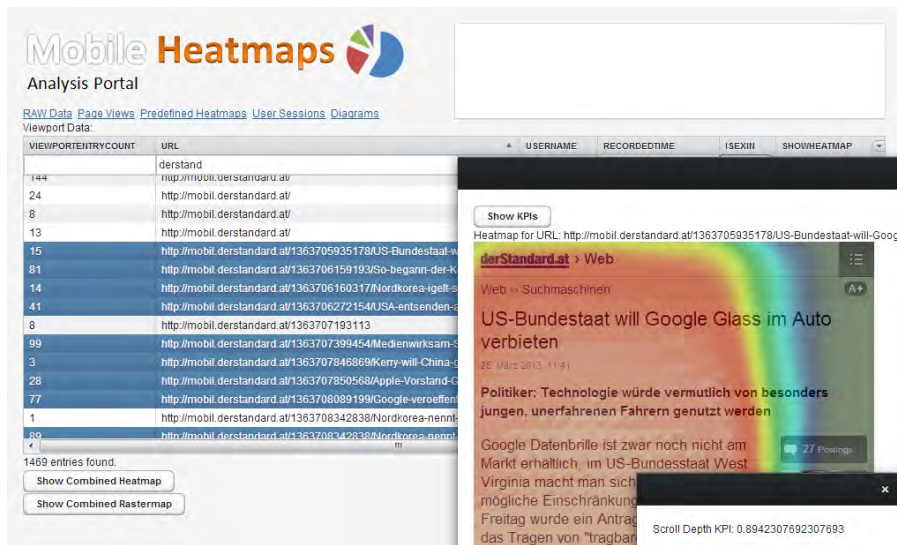


Figure 6.7: Selection of page views (one visit to a specific URL) can be selected and a combined heatmap can be generated

Figure 6.7 shows the very important functionality of combination of web site visits to a heatmap. With this functionality it is possible to filter for specific data sets (like views generated from AEXIN users or just iPhones or specific recording date) and generate a heatmap and corresponding KPIs just for this selection. As many page visits also include multiple screenshots of the web pages visited, the underlying screenshot can be selected from the list of contributing page views and changed via a paging functionality to compare the underlying visits and ensure consistency over the pages.

Evaluation

The evaluation of the given approach is done by showing the functionality of the solution created. Furthermore, possible performance indicators based on the solution implemented are proposed.

7.1 Data source

During the summer term 2013 of *Applied Web Data Extraction and Integration*, an optional task was given as homework. This task was selected by 14 students of which 11 returned the questionnaires. Other data was gathered by eight namely known participants and one and a half hour of 'anonymous' traffic over two months. There was no given task to these users and no time constraints - their smartphones just where set to use the proxy-server of the project so people contributed by their normal surfing behaviour.

7.1.1 AEXIN Task description

The task description as given to the students. It was one of four given tasks of which two had to be done.

In this task you have the possibility to support and participate in the evaluations conducted for a thesis about *mobile viewport tracking* and generating mobile heatmaps, performed by students at DBAI. For more details about the setting, please refer to the lecture slides. Moreover, please look at the online guide provided by the project members on the project's welcome page at <http://heatmaps.no-ip.org>. There you find a step-by-step guide with all the information about how to configure your cellphone and about the requirements.

1. Follow the instructions at <http://heatmaps.no-ip.org> [(see appendix 5.1)] step-by-step to configure the MVT proxy server with a WiFi of your choice (e.g. eduroam or your private WiFi). Please verify that the proxy is

really used with sites such as `whatsmyuseragent.com`, `whatismyip.com`.

2. On your mobile browser, access the MVT registration page ¹
3. Please enter a unique alias for your cookie. With the alias we can identify your browsing sessions. It is required to set some alias to verify that you chose Task D. Furthermore, mention your used alias in the exercise deliverable (e.g. use your university registration number, but any other alias is fine as well).
4. Part A: Browse with your smartphone with activated proxy server for at least 20 minutes and access at least 5 different Web sites where you perform some interactions. Please do not use sites such as Facebook or Google as they are SSL encrypted and cannot be tracked this way. Using Google as search is fine, but you should spend most of the time of different (non-SSL) Web sites.
5. Part B: Furthermore, please browse to 3 newspapers of your choice and search for articles about one particular topic of your choice - either about latest developments in *North Korea*, or about *Google Glass* or *Google Ingress*, or about *Eurobonds*. Please rate how fast and intuitively you found the information on the news portals (finding, readability on the mobile browser, navigation on the mobile browser).

After you finished your contributions, you can ask for access to the recorded interaction steps (clicking, zooming, scrolling...), and the generated mobile heatmaps. Please send a message to `mobile.viewport.tracking@gmail.com` (Sebastian/David) to get access, furthermore, in case of questions, ideas, comments please feel free to get in touch with them as well. Deliverables: As deliverable, please hand in the filled out answer sheet which is provided as part of the lecture resources. The answer sheet must contain your used alias, a short summary of part A (some sites you visited, how long you did surf...) and a short summary of part B (topic you did chose, the 3 newspaper sites you did use, comparison between these 3 sites regarding usability, how fast you did find the information and some other criteria. Finally please provide some general feedback in the answer sheet as described. Summary of Requirements:

- Smartphone (Android at least 4.x, iPhone - all models)
- WLAN with Internet access

What is tracked: We only track actions that run in the browser via the HTTP protocol, excluding the following: no password fields on non-SSL pages, no SSL-encrypted pages and no App traffic as well. The goal is to track mobile surf behaviour and nothing else.

¹<http://heatmaps.no-ip.org/MVTRegistration>

7.1.2 Data Amount and Quality

14 students participated in the exercise of which 11 students returned their questionnaires. An overall of 15251 Actions were recorded in 9 hours, 5 minutes and 4 seconds of overall surfing time. That is an average of 1087 actions and 38 minutes and 56 seconds per participant. About 100 different domains were “surfed”.

The *data collected in this study* is compared to the *data that was not collected in this study* (Data contributed by ourselves and some of our friends over two months without any given task) and to the *overall data* (Sum of both).

This comparison should help get an overview of the quality of data.

Table 7.1: User Statistics (AEXIN)

Measure	Actions (count)	Time (hours:min:sec)
Sum	15251	09:05:04
Mean	1089,36	00:38:56
Std. Dev.	793,10	00:12:57
Median	875	00:37:14
Maximum	3254	01:11:18
Minimum	362	00:20:40

Table 7.2: User Statistics (other)

Measure	Actions (count)	Time (hours:min:sec)
Sum	12562	06:33:00
Mean	1570,25	00:56:09
Std. Dev.	3365,60	01:04:28
Median	185	00:44:46
Maximum	9827	03:00:50
Minimum	33	00:00:21

Table 7.3: User Statistics (all)

Measure	Actions (count)	Time (hours:min:sec)
Sum	27813	15:38:04
Mean	1264,23	00:44:40
Std. Dev.	2054,56	00:37:45
Median	764	00:37:23
Maximum	9827	03:00:50
Minimum	33	00:00:21

Tables 7.1, 7.2 and 7.3 show the basic statistical data from the list of users. E.g. Minimum time 00:20:40 means that every user from the AEXIN group contributed at least 20 minutes and 40 seconds to the project.

Table 7.4: User Session Statistics (AEXIN)

Measure	AvgPageDur. (sec)	Actions	Domains visited	URLs visited	Duration (sec)
Sum	—	15386,00	238,00	742,00	34392,00
Median	81,00	40,00	1,00	3,00	125,00
Std. Dev.	117,16	292,62	1,60	9,61	441,97
Mean	116,43	129,29	2,00	6,24	289,01
Maximum	527,00	2773,00	10,00	78,00	3699,00
Minimum	0,00	1,00	1,00	1,00	0,00

Table 7.5: User Session Statistics (other)

Measure	AvgPageDur. (sec)	Actions	Domains visited	URLs visited	Duration (sec)
Sum	—	13826,00	594,00	803,00	33528,00
Median	24,00	11,00	1,00	1,00	31,00
Std. Dev.	76,37	56,93	0,59	1,56	110,94
Mean	51,95	29,73	1,28	1,73	72,10
Maximum	517,00	728,00	4,00	16,00	874,00
Minimum	0,00	1,00	1,00	1,00	0,00

Table 7.6: User Session Statistics (all)

Measure	AvgPageDur. (sec)	Actions	Domains visited	URLs visited	Duration (sec)
Sum	—	29212,00	832,00	1545,00	67920,00
Median	31,50	13,00	1,00	1,00	37,00
Std. Dev.	89,98	146,70	0,94	4,89	238,70
Mean	65,09	50,02	1,42	2,65	116,30
Maximum	527,00	2773,00	10,00	78,00	3699,00
Minimum	0,00	1,00	1,00	1,00	0,00

As the given task was to surf at least 30 minutes, the surf sessions tend to be longer than normal. This is a known issue in quality of the data but was accepted as this limitation was also accepted by academic researchers [Chen et al., 2001] and could not be avoided. This can be seen as the median of the duration of a surf session is 125 seconds in AEXIN data and 31 seconds in other data. One can say that the surf duration is in this regard biased. Also the 78 different URLs (= web pages) visited without putting the smartphone aside shows that the length of the surf sessions cannot be taken into account (Maximum URLs visited from table 7.4. As the data was collected in the course *Applied Web Data Extraction and Integration* which is a free elective or a specialization in different Informatics Bachelor / Master studies on a University of Technology, it is obvious that data does not represent a wide range of society as the participants are mostly male, typically between 20 and 30 years old and have very good knowledge of the

World Wide Web. People used their own smartphones so they should be familiar with their device and should not have any problems when using the device. So no learning effects or technical problems were assumed. As the user-agent is altered by the proxy server, in most cases, the users got the regular page and not mobile sites (see section 3.4.3) which could lead to some irritation if the users know that a mobile version would typically appear on a certain web page.

7.2 Studies

7.2.1 Intention of user



Figure 7.1: derstandard.at (aggregated: LEFT: 3 pageviews, AEXIN data, given task, RIGHT: 10 pageviews, NON-EXIN data, no given task) cropped vertically after 40 percent

Figure 7.1 shows the landing page of the news website derStandard ². The image is cropped vertically after about 40% as there was no data recorded in the last 60% of the page. **The surfing behaviour is significantly different between the users with a given task (AEXIN) and the users that surfed regularly.** On the one hand, this figure shows the impact of the task on the viewing behaviour of the page as it is not the typical usage of a news website to use the search field.

On the other hand it shows that the intention of the user on a page significantly affects the viewing behaviour. This could be used to determine the originating intent of the user from the recorded data.



Figure 7.2: derstandard.at/... (various article urls) (aggregated: LEFT: 5 pageviews, AEXIN data, given task, RIGHT: 52 pageviews, non-AEXIN data, no given task) comparing reading behaviour with given task to reading behavior without a given task. Please note that the length of the pages differs between different articles.

Figure 7.2 shows an aggregation of article pages of the news website derStandard. As this is

²<http://www.derstandard.at>

an aggregation over multiple different articles, no conclusions on the content can be made but on the layout of the page and typical reading behaviour. It can be seen that the reading length differs more on people who do not have a given task (right side) as typical article length is about as long as the example articles shown in the background. This could be due to the user comments that are at the end of the page - people with a specific task more probably stop at the end of an article or just very shortly after starting to read user comments. The users that did not have a given task are more likely to digress and more probably go on reading user comments.

Another interpretation can be done on the right heatmap in figure 7.2. The **zoom level gets adjusted in small steps when reading on**. Despite the fact that the white ('hottest') area has a clear boundary on both sides which is pretty close to the vertical limits of the text of the article, the red ('pretty hot') areas width thins down when scrolling further down the page. This can be interpreted as in some sessions, people are instantly getting the right zoom level and focus while in other sessions (or other people) are struggling to get the right level of zoom and the right focus and adjust this when scrolling down. This is one of the major problems of websites that are not optimized for mobile phones. When comparing this to the mobile version of this website (see figure 7.3) this effect cannot be seen as the heatmap is very simple on a news article.



Figure 7.3: Example non-aggregated heatmap of the mobile version of derStandard. Please note that this mobile version is not corresponding well with the background image as the layout of the page is adaptive to the screen width and the screenshot is taken with a screen width that is (in this example) slightly higher than the screen width of the mobile device.

7.2.2 News Websites

The AEXIN survey asked to rate *Fast Info Finding*, *Usability* and *Readability* with grades from 1 = very good to 5 = poor. As the pages could be selected by the participants, many different news websites were chosen. Only the pages with the most users choosing them are shown here, other websites with just one user are omitted here (kleinezeitung.at, sueddeutsche.de, bostonherald.com, net.hr, dailynews.com, news.at. . .). As news websites typically have a high percentage of reading content and cannot be evaluated very well using mouse tracking, this type was chosen to find out how well the implemented solution can be used for evaluation of such websites.

Table 7.7: derstandard.at Statistics

<i>derstandard.at</i>	Fast Info Finding	Usability	Readability
Sample-Size	6		
Median	2.00	1.50	1.00
Std. Dev.	0.75	0.55	1.22
Mean	1.83	1.50	1.50
Maximum	3.00	2.00	4.00
Minimum	1.00	1.00	1.00

Table 7.8: diepresse.com Statistics

<i>diepresse.com</i>	Fast Info Finding	Usability	Readability
Sample-Size	5		
Median	1.00	3.00	3.00
Std. Dev.	0.89	0.84	1.14
Mean	1.60	2.80	2.60
Maximum	3.00	4.00	4.00
Minimum	1.00	2.00	1.00

Table 7.9: orf.at Statistics

<i>orf.at</i>	Fast Info Finding	Usability	Readability
Sample-Size	3		
Median	1.00	2.00	2.00
Std. Dev.	0.00	0.58	1.15
Mean	1.00	2.33	2.67
Maximum	1.00	3.00	4.00
Minimum	1.00	2.00	2.00



Figure 7.4: Aggregated heatmaps of different news articles of derStandard (left), diePresse (center) and orf.at (right) from AEXIN

Readability of the website on a mobile device

Figure 7.4 shows the reading behaviour of three different news websites.

The readability of the web pages was rated with a mean of 1.50 for derStandard, 2.60 for diePresse and 2.67 for orf.at. orf.at delivered the mobile version of the web page even though the proxy pretended that the website is delivered to an iPad. Visitors of derStandard read typically about three times as far as the visitors of diePresse.com. This is also represented by the grading of readability (see table 7.7 and 7.8). But also the mobile “optimized” website of orf.at gets a

bad grading ³. Although the layout of these two pages is a very similar three column layout with illustrating images in the left column, the main content in the middle column and links and abstracts of other articles in the right column, there are significant differences in viewing behaviour. A possible interpretation of these differences is the layout of the heading and abstract of these two pages. **derStandard** has a layout where the heading and abstract are as wide as the main content - so a mobile **user will be able to zoom in instantly and begin to read whereas diePresse has the articles heading and abstract spanning over the left column and center column**. Especially on smaller devices, it can be **difficult to read** them. Then after reading these few lines, the text size significantly decreases so the users have to zoom in. The text size of the abstract of articles on derStandard.at are the same as the text size of the main content.



Figure 7.5: mytimes.com (left), thomsonreuthers.com (center left), heute.at (center), bbc.co.uk (center right), kurier.at (right)

The news pages shown in figure 7.5 that were rated for readability all had a mean between 1.0 and 2.0. All these were read very linearly from top to bottom except kurier.at where the region with related articles was also read. This region is near the end of the article and was read by most users. This site also got a good rating of 1.5 for readability.

³please note the layout issue of this heatmap as already mentioned at figure 7.3

7.2.3 Mobile Version vs. Desktop Version



Figure 7.6: Desktop version (left) and mobile version (right) of the news website kurier.at - aggregated over 2-3 page views

Figure 7.6 compares the mobile and the standard version of article pages on the news website Kurier. On the desktop version, it can be seen that the scrolling behaviour is not linearly down but some more to the left and right of the page. Also, it can be noticed, that the links to **further articles on the right were typically perceived on the desktop website** but the links to further articles at the end of the page were not. One reason could be that the advertisements that are at the end of the main article disrupted the page viewing. The mobile version of the page has a very linear viewing behaviour. **Further articles at the end of the mobile page were not perceived.** The keyword-list and creation time were the last parts of the website that were seen (this view is aggregated but the underlying page views show the same behaviour which can be found in the appendix A.2.1).

It should be tested if the position of these two lines is a show stopper and removing these lines would lead to further reading. As MVT is using a proxy approach, the website could be altered on the proxy server to enable such tests.



Figure 7.7: Search on news websites - desktop vs. mobile websites - derStandard.at (left) and kurier.at (right) aggregated over 3 - 7 page views

The search results pages of derStandard.at and kurier.at show pretty similar behaviour when comparing the desktop version of the page with the mobile version. **On both mobile version, more content is consumed as on the desktop versions.** One possible reason could be the optimized layout leading to an improved user experience and people reading on. When looking at the raw data, it can be seen that more Window-Deactivated events were recorded on **mobile pages** and people **followed links more often compared to the desktop version.**

7.2.4 Amazon Article Pages

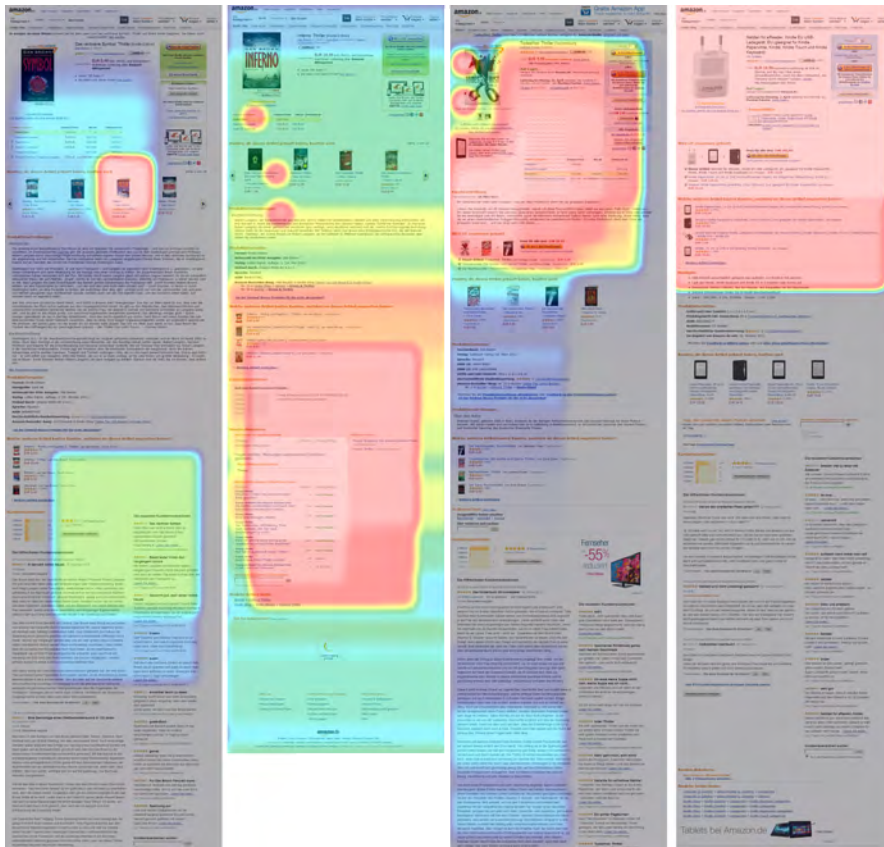


Figure 7.8: Visits on four different <http://www.amazon.de> product pages (four examples shown out of about 25 recorded)

Figure 7.8 shows product pages of the online shop Amazon.de. The most significant difference to news pages as seen in figure 7.2 and figure 7.5 is the **discontinuity of viewing behaviour**. One possible explanation are the user recessions at the end of article pages which are read frequently. The **product description is sometimes skipped** and shows a similar relevance as the user recessions. It should be tested if a higher placement of user reviews as prominent as the article description would lead to a better user experience.

The placement of **cross selling products** seems to be good as they are displayed in most of the product pages and **earned high attention**. In the example pages shown here, this is visualized through a high zoom level and a click in the left two heatmaps.

7.2.5 Weather Pages

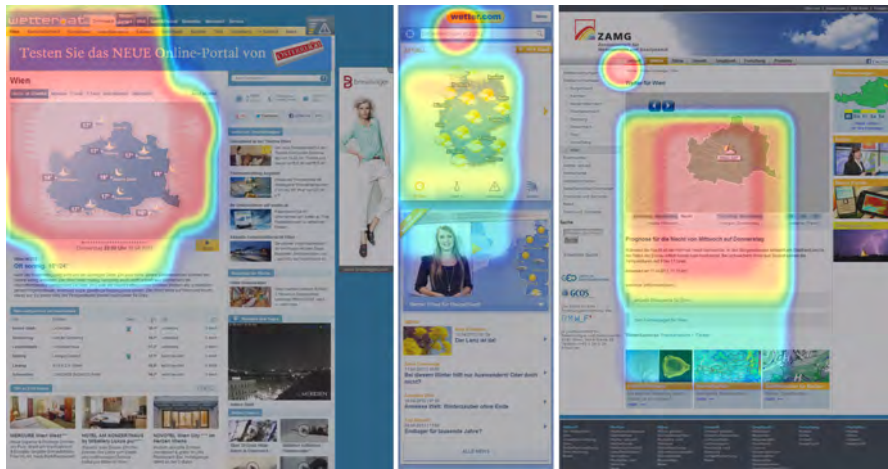


Figure 7.9: Visits of the current weather on <http://www.wetter.at>, <http://www.wetter.com> and <http://www.zamg.ac.at>

The weather pages shown in 7.9 all have a very similar viewing behaviour. **Just the key information is viewed, further articles or advertisements are not displayed and recognised.**

7.2.6 Orientation

In nearly 71% of all viewports recorded, the device was held vertically and just 29% of the actions were done when the device had a horizontal orientation.

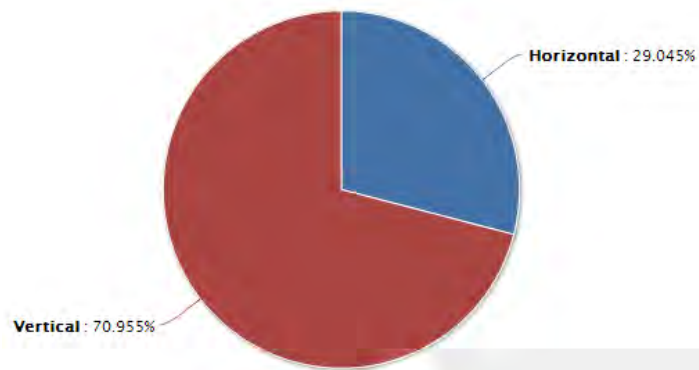


Figure 7.10: Device orientation (overall)

7.3 Key Performance Indicators

7.3.1 Scroll Depth



Figure 7.11: Heatmaps of four different news website start pages compared (left to right: derStandard.at, sueddeutsche.de, kleinezeitung.at, edition.cnn.com)

Figure 7.11 compares the start page of four different news websites. The heatmaps are all aggregated data from multiple users. It can be seen that **the overall depth of scrolling was always under 50% of the total depth of the website**. When looking at the overall data recorded, it can be seen that the first three pages range in the upper 30% in the metadata field 'heightOfImportanceMatrix' representing the overall count of raster cells in vertical direction. It can be assumed that this viewing behaviour is typical for mobile users as these three websites show a very long layout with a maximum of up to 14500 pixels which is even 13 times the height of a current [W3Schools, 2013] full hd computer monitor. A **one-dimensional scroll map** as mentioned in 4.4 could be created that aggregates all cell values of one row to a single aggregated value. The percentage-value of the **95%-quantile** of this one-dimensional array is used as an **KPI showing the typical maximum of scroll depth of the website**. As viewing behaviour is certainly different for different types of devices, this factor should be recorded separately for different device-types like desktops, tablets or smartphones. Website owners could adapt their

Content Management System to use this value to **automatically deliver more content if the value of this KPI is over a certain limit or reduce content if the value is below a certain threshold.**

Table 7.10: Example Values for Scroll Depth KPI

Landing pages shown in figure 7.11	
derStandard	0.2669
sueddeutsche	0.3031
kleinezeitung	0.0909
edition.cnn.com	0.2717
mobil.derStandard.at	0.7089
Other pages examples of landing pages	
Amazon.com	0.4898
Web.de	0.6351
Heise.de	0.5536
Articles (with user comments at the end):	
derstandard.at/1363711043869/Kaffeekraenzchen-mit-Apple...	0.6121
derstandard.at/1363711036424/Kommt-ein-weisses-Nexus-4...	0.6300
derstandard.at/1363710892408/Tweets-verraten-Alter...	0.5376
Other pages:	
sixx.at/stars-style/beauty	0.4824
nationalparks.org	0.6415
heimfest.at/aktuelles	0.6783
orf.at	0.6694

A value between 0.5 to 0.9 is proposed as a good value, but further studies would be needed to get well-founded values for a range definition. It should be mentioned that many homepages have a footer at the end that should not be taken into account for this KPI - this is only possible with usage of website element classification or by manual adaption. Table 7.10 shows example values for this indicator.

7.3.2 Horizontal Scroll Depth

The Scroll Depth 7.3.1 represents the visit duration and interaction count of the website projected on one vertical axis. Especially on websites that are very wide, a similar KPI for the horizontal axis can be defined. As such websites are more uncommon and the algorithm is very similar to 7.3.1, an implementation of this KPI is omitted.

7.3.3 Visit Duration on Different Areas



Figure 7.12: Web.de landing page segmented

With the usage of an segmentation algorithm, the website can be divided into sections like advertisements, main content, header, footer, navigation... These areas can be mapped to raster cells. With this the **importance of specific areas can be determined - e. g. by the average cell weight of cells in a specific area in comparison to the average overall weight**. This approach is more similar to the approach used in studies of [Chen et al., 2001] which were described in 4.3.1. In this study, one of the main findings was that “regions that the eye gaze didn’t visit, 88% of them were not visited by the mouse cursor, either” [Chen et al., 2001]. Figure 7.12 illustrates that this high percentage is often misused as it only gives information of unvisited areas (of the mouse cursor).

An example for interesting indicators based on the segmentation approach is shown in figure 7.13 where one visitor has viewed the image area on the left side whereas the another visitor ignored this area. This indicator could be calculated for each area of a web site.

As this approach of segmentation is used in the master thesis “Repackaging Web Pages using *Advanced User Behaviour Analysis*” done using the same data base and common part, this work will not go into further detail.

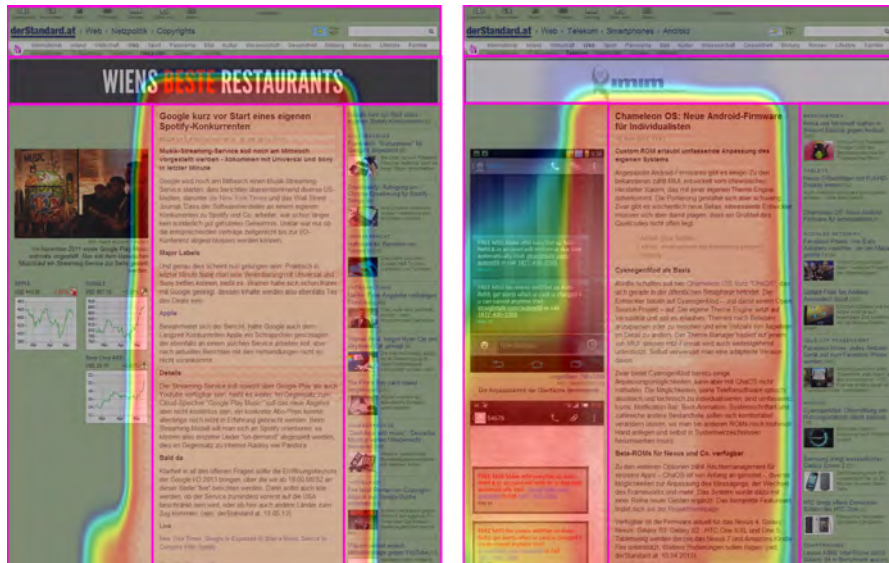


Figure 7.13: Page views on derStandard.at - one were the image area was ignored (left) and one were the image area was viewed (right)

7.3.4 Basic Statistical Data of Importance Matrix

Standardized metrics of the importance matrix (the raster cell concept of MVT) like the maximum weight of a cell, the standard deviation, the mean and median and many others can be used as KPIs. In the current solution, these values are stored as metadata to each heatmap to help identify important heatmaps with filtering and sorting after these values.

7.3.5 Interaction Sequence of Zooms/Scrolls

The sequence of interactions could be an interesting indicator showing how the user experience was. In theory, **a sequence of scrolls in different directions could be an indicator for a bad user experience whereas a targeted zoom to a certain position followed by reading and finally zooming out to get an overview over the page could be an indicator of a good user experience.** To prove or falsify this theory, a larger data pool with classification of the user experience would be needed to appropriately make use of machine-learning algorithms to identify such patterns.

7.3.6 The Sequence of Interactions

More general, all sequences of interactions can be used to classify viewing behaviour on a certain page and the page type that is viewed. For this, not only zoom and scroll actions should be taken into account but also other action types like textual input.

7.3.7 Homogeneity of the Distribution of Weights

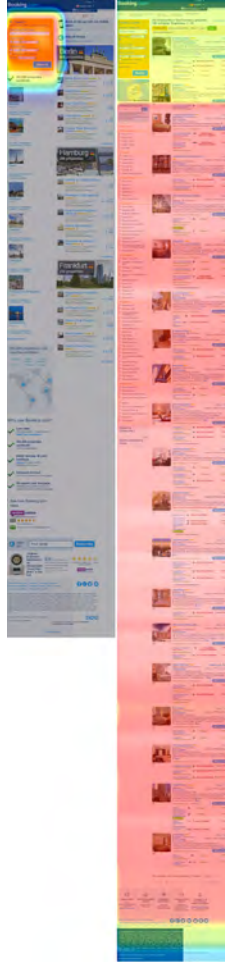


Figure 7.14: Booking.com landing page (left) and corresponding result page (right)

The distribution of weights over cells could be an interesting metric showing the **homogeneity of the distribution of significance over a certain website**. Figure 7.14 shows the landing page of the booking portal `booking.com` where only a small area of the website was important whereas the search result page was read very uniformly. The count of values being larger than 30% of the maximum weight divided by the overall count leads to a value between 0 and 1. For the landing page shown, the calculated value is 0.0218 whereas the value for the shown result page is 0.5609. This factor **can be seen as the “percentage of the page that was typically read”**. To gain a more stable value, a percentile value could be used to calculate the limit between importance and unimportance as the maximum could also be a unstable value when getting a lot of clicks on a certain position.

7.3.8 Interactivity of a Web Page

One of the main differences of the two pages seen in figure 7.14 is the level of interactivity. While the landing page is used as a **highly interactive site**, (even when there is much reading content on the page that is ignored) **the ratio of interaction events (like key presses, taps, holds and website deactivated events) is much higher than on typical reading content**. So the percentage of interaction events on a site can be used as indicator how interactive the site was used. As seen in figure 7.14, this can be different to the design of the website. This value can be calculated for each page visit and they can be compared to each other by simple statistical methods. This would show how differently pages are used that can be used as interactive or as reading pages.

In the example given in figure 7.14, the value for the first page is 0.5897 (46 / 78) whereas the value for the only slightly interactive result page is 0.1553 (16 / 103).

7.3.9 Problems finding the right Zoom Level



Figure 7.15: Heatmap of the desktop version of derStandard showing a v-form zooming pattern

On news website article pages, the reading behaviour that is expected is very simple: One would expect the reader to zoom in till the text is readable and then read from the beginning of the text till the end of the article, page or interest. As it can be seen in figure 7.15, **sometimes people struggle to find the right zoom level** and read further on as they can read the text but the left side of the viewport is just showing empty space or some parts of images or advertisements shown in that place.

When reading on the text, the zoom level gets adjusted step by step by people zooming into the right zoom level. When viewing this behaviour on a heatmap, a **V-form can be identified**.

This pattern could be used to create an indicator for zoom level problems for certain pages. For indicators like this one, it is important to drill the analysis down to the page view level to ensure that the pattern is not an artefact of the aggregation of multiple page views with different behaviour.

7.3.10 Average Viewport Entry Count

Per page load, an average of 19.45 actions was recorded. Of course, this value has a high range of variation due to the different types of web pages, e.g. articles, landing pages or checkout pages of online shops. For each type of website, an average can be calculated and each web page of that type should be compared to this average. When the **page significantly differs from the average, a possible reason could be that the page has a too high / low complexity.**

7.3.11 Percentage of Horizontal Scrolls

As figure 7.16 shows, the average percentage of scrolls in horizontal directions (scroll left, scroll right) is below 8%. **If a web page has a significantly higher value than 8%, the page layout could be too wide for mobile devices.**

7.3.12 Action Type

Figure 7.16 shows the total percentages of action types, e.g. over 45% of the overall actions recorded were scroll down actions (scroll down straight or diagonal). If a website significantly differs from these values and the reason is not an high amount of text inputs, an analysis should be done finding the reason for the deviation.

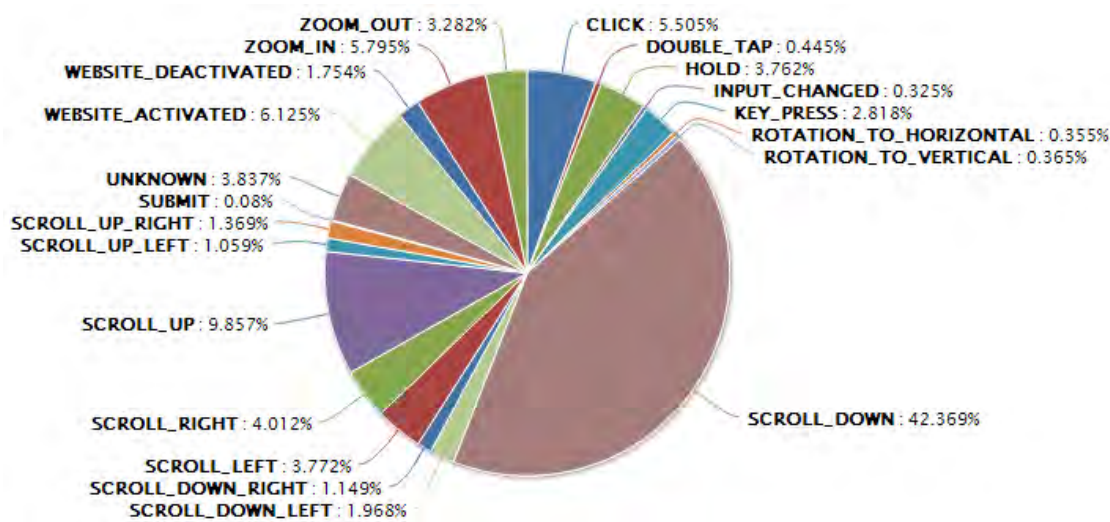


Figure 7.16: Action types (overall)

The 3.837% of 'unknown' actions exhibits some limitation of the classification algorithm and shows some potential for improvements. For further information see 8.1.5.

7.3.13 Other possible KPIs

There are 385 metrics that were mentioned and classified in the Web Quality Model (see 4.1.2) that could be used to create KPIs. As these metrics are not specific to this approach on usability studies, they are not explicitly mentioned here but could help to implement a more holistic analysis portal.

Conclusion

The key concept of this project is the recording of the limited visible area of a website on a mobile device which is used as a surrogate for the attention of a user. By tracking this small area, it can be reconstructed what a user could have read. Also, interactions like taps, clicks or inputs are recorded. The data is classified (see 3.5.2) and stored in flat tables to enable easy mining of the data. Also, multiple keys are generated (see 3.5.5) which connect entries to page views, user sessions and user data which help filter and aggregate data. A state-of-the-art analysis (see chapter 4) with focus on mouse tracking, eye tracking and similar technologies that enable page level interactions to be recorded and evaluated was done to create and position the solution that was implemented. A study (see 7.1.1) was done with 14 students from the course *Applied Web Data Extraction* with given tasks and a minimum amount of time (30 minutes) the participants had to surf. A second study (see 7.1) was done with eight namely known participants and one and a half hour of 'anonymous' traffic over two months. In this study, no constraints were given - just the tracking was enabled on their mobile devices and they could do whatever they want. The data from these studies was used to evaluate the system created and also influences of the given task on user behaviour were discussed. The data gained was used to do qualitative studies and to define some key performance indicators that were theoretically discussed and some that were implemented and their results were shown. To conduct the studies and to gain more voluntaries for the studies, an analysis application was created (see chapter 6) which can be used to gain an overview over the data and combine selected data sets and do evaluations on the selected data. This application acts as a simple interface to the large amount of data recorded by *Mobile Viewport Tracking* (see 3). A key functionality is the selection of data sets and on-the-fly creation of the corresponding heatmaps and KPIs. This application was used to perform data evaluation and calculation of KPIs and acted as a simple user interface for generation of heatmaps (see 2.4).

The proxy approach proved to be very reasonable for usability studies as studies with different websites are possible without much effort. Other usability studies often access downloaded and prepared websites [Lagun and Agichtein, 2011] which is a limitation that can be bypassed. Other academic usability projects like the Usability Proxy (see 4.3.4) record very complete data but lack a simple way to do first evaluations and give an overview over data. The web application created (see 6) showed to be very useful for getting an overview over the recorded data. Studies showed that a given task has a very high impact on user behaviour for specific web site types like news paper landing pages (see 7.2.1) whereas no differences were found in the reading behaviour of news articles, that small layout issues (see 7.2.3) could stop the visitors from reading on the page and that the mobile version is not always better (see 7.2.3). When smartphone users do not immediately find the right zoom level while reading text, this can be seen (see 7.2.1) in the heatmaps created by *Mobile Viewport Tracking* (see 3). The heatmaps also helped getting an overview over the recorded data and helped identify some interesting KPIs - that is the target of a visualization. Studies showed that the overall scroll depth of news websites is far too long for mobile users on the most popular news websites among the participants of the study - even the mobile layouts showed this phenomenon. A KPI was proposed and implemented that is a measurement of the average scroll depth. With the usage of a segmentation algorithm, comparable data to the studies of Chen [Chen et al., 2001] can be created and with the usage of an eye tracking system, statistical evidence of the quality of data of MVT can be given. Also, a KPI is proposed and implemented that shows the homogeneity of reading over the whole page - a simple indicator of the percentage of the page that was read. Another KPI proposed gives an overview over the interactivity of web pages (see 7.3.8) that can give useful information for the categorization of web pages.

Some findings and their implications for improving web page design using the implemented solution were:

- About 71% of the overall actions were done when the smartphone was held vertically showing that web pages for mobile devices should be optimised for vertical usage.
- It was found that news websites that typically have extensively long landing / start pages are not read till the end of the page (see 7.3.1). Only about the first 30% of the page read by most visitors using mobile devices. On the mobile device optimized version of the landing page of derStandard, about the first 70% of the page were viewed.
- While news web pages are read very linearly, web shop article pages have a higher discontinuity of viewing behaviour. Also the main part of an article page - the article description was regularly skipped while user recessions earned high attention (see 7.2.2 and 7.2.4).
- The percentage of the page that was typically read can be identified using the homogeneity of the distribution of weights as proposed in section 7.3.7. This indicator can be used to help classify web pages or to quickly gain an overview over user behaviour on a certain page. If the value is very low, this is an indicator that the pages needs further analysis and a possible redesign.

- Users with a given task behave differently than users without a given task on certain page types like news website landing pages (see 7.2.1).
- A KPI which could be very useful for web page classification tasks is the “Interactivity of a web page” (see 7.3.8) which shows significantly different values for interactive and non-interactive pages.
- Studies (see 7.2.3) also showed that the simplified layout of web pages that are optimised for mobile devices also sometimes stop the user reading further on. On desktop websites, further articles or advertisements are recognised while reading the main article as they are often positioned on the left or right side of the main article so some users will come back to these areas and follow the links provided while the user behaviour on mobile versions shows to be very limited to reading an article and stop reading at the end of the article. In the qualitative study done, links to further articles on mobile pages shown at the end of a page were not recognised by the users whereas similar ‘related articles’ were visited when the smartphone users viewed the desktop version of a web page. This shows that the layout that is used by nearly all mobile optimised websites needs further improvements.

The approach of using the limited viewport of smartphones and the whole solution implemented showed its potential to be used to record user behaviour and to visualize data in a simple manner and help identify KPIs. Also the KPIs proposed showed to be useful for website evaluation.

The main limitations of eye tracking - the high price of devices and the laboratory situation which limits user count and influences the participants are not applicable to *Mobile Viewport Tracking*. The recorded data does not have such a high resolution than the fixation points of the human eye as the viewport is significantly larger but showed to be useful for web usability studies. The main limitation of mouse tracking - the low correlation between the mouse cursor and the visual attention of the participant is not applicable to *Mobile Viewport Tracking* because it is not possible to view something on the page not displayed in the viewport of the device. A limitation of *Mobile Viewport Tracking* is that the viewport is just a proxy for visual attention that also could be totally different as people could theoretically scroll on their device while not watching it - similar to mouse tracking. *Mobile Viewport Tracking* and mouse tracking have the benefits of being very cheap while it is also possible to record people without their knowledge in blind studies or even with an very high number of users. When the proxy server is used as a transparent proxy in a public institution or on a wireless hotspot, this could lead to a very high amount of data from different websites - which is not possible with typical solutions provided - with identification of a single user over all pages he visits - due to the global cookie trick implemented - over a long time - as studies are not limited to a laboratory - without the influence of given tasks.

As for all studies in web user behaviour, appropriate tools for the job have to be selected and typically triangulated with other research methods like interviews or surveys [Khoo et al., 2008]. *Mobile Viewport Tracking* could be one of them delivering not as precise data as eye tracking but an large amount possibly as high as the amount of actions done on the observed page with correlation being significantly higher than with mouse tracking solutions.

8.1 Future Work

Due to the limited time and overwhelming possibilities of the project, there are many things that remain for the future of this project.

8.1.1 Other platforms

The current implementation only targets Android and iOS devices as these two devices had a market share of 92,6 % in Q1 2013 [Gartner, 2013] but there are also some other mobile platforms like Windows Phone 7.5 and 8 ¹ or BlackBerry ² available and many others like Sailfish OS ³, Ubuntu for Phones ⁴ and Firefox OS ⁵ announced.

Also data from desktop browsers could be used, e.g. some mouse tracking that leads to additional data but without the usage of the viewport as the viewport on desktop systems is far too large to be used as a surrogate for user attention.

8.1.2 Use recorded data from Proxy

The current approach needs websites to always return the same content as the screenshot service of the application re-accesses the URL and expects the website to deliver the same or very similar content. As the proxy delivers the altered websites to the users, it has full access to the sources of the website and could store them. Then, the original data that was transmitted to the user could be used to create screenshots or directly use this data and draw the heatmaps in the browser of the person viewing the recorded data. This would also be the first step when making the solution fully AJAX aware.

8.1.3 DOM-events / AJAX-aware

The current solution is not capable of handling AJAX-events - till now, they are just recorded. When using data recorded by the proxy, the full information of the DOM is available. The recorded actions could be used to replicate the actions on the server part and update the DOM-tree accordingly. Also, the concept of the importance-matrix would need an extension as the website layout changes due to the actions triggered. So, importance information could be stored to the corresponding elements of the website.

8.1.4 Include in websites as beacon script

The inclusion of the script in web pages by website administrators would solve the problem of the current solution not being able to track SSL-pages. When included in the websites itself, it should also be possible to track user behaviour on SSL-protected pages like the checkout pages of a web shop.

¹<http://www.windowsphone.com>

²<http://www.blackberry.com>

³<http://www.sailfishos.org>

⁴<http://www.ubuntu.com/phone>

⁵<http://www.mozilla.org/de/firefox/partners/>

8.1.5 Improve action classification

An overall of 3.837% (see 7.3.12) of the actions recorded could not be classified with the current implementation of Action Type Classification (see 3.5.2). With some effort spend on the analysis of this specific actions, this number can probably be reduced drastically.

8.1.6 Transparent Proxy

To gain more data, it would be possible to create free WiFi hotspots which use a transparent proxy server. In this way, people would benefit from free Internet access and researchers would gain a lot of data. Obviously, a start page mentioning the tracking should be included. This start page could also be used to create the global tracking cookie (see section 3.5.5).

8.1.7 Reconsider the concept of predefined heatmaps

The predefined heatmaps showed not to be very useful for usability studies. It was found that these heatmaps can be used for usage like repackaging were importance information of a large user group needs to be evaluated and processed nearly immediately because loading times of websites are influenced but are too simple for usability studies. The current aggregation over all traffic for one URL to be added to the existing heatmap for that URL just with timestamps for start and end will not be enough to be really usable. So rules on all data should be available to define which entries to aggregate - eg. by the users operating system or user agent or by a specific repackaged version of the website. CrazyEggs solution (see 4.3.4) for aggregation could be an example.

8.1.8 Improve features of the analysis web application

The web application (see chapter 6) that was created to help perform studies intentionally limits itself to page level analysis of web pages as this is the level that lacks support on most web analysis tools available. Apparently also data recorded on the inter-page level like absolute visits and visit paths between pages are important information that can be used in usability studies. This information should be taken into account when planning future extensions of the analysis web application.

8.1.9 Web page variant comparison

As already mentioned in 7.2.3, it would be feasible to select a specific website and do tests on this page by delivering different variants of the original page. The website could be repackaged (some parts moved or removed) by the proxy server and differences in the viewing behaviour of the page could be found.

8.1.10 Evaluation of derived KPIs

In section 7.3, some KPIs were proposed that were not evaluated adequately due to the fact that the data base was too small and somewhat biased due to given tasks so an evaluation would

not have been convincing. Also, it was only target of this work to identify possible KPIs as an evaluation would have needed an iteration in the studies that was not possible within the time limits. With a larger study focusing on evaluation of the proposed KPIs, it will be possible to select the KPIs that give best information for website usability studies. A very simple example would be the variation of the length of a search result page by limiting the results returned.

8.1.11 Evaluation using eye-tracking system

An evaluation on the comparability of eye-tracking data and data generated from *Mobile Viewport Tracking* would be very helpful in understanding the limitations of the systems. Initially, such a study was planned for this thesis but could not be realised as the eye tracking equipment of the Technical University of Vienna was broken during the time this thesis was written.

8.1.12 Commercial usage of the project

With some time and venture capital, this solution could have the potential of getting a great alternative to typical mouse tracking systems. A possible risk would be competitors now already offering fully fledged mouse tracking solutions like m-pathy (see section 4.3.4) that have some of the features proposed here already implemented and just would need to borrow the key idea of viewport tracking and do some changes in their solutions to offer *Mobile Viewport Tracking*.

Appendix

This appendix contains additional information like source code and is related to the common basis of the theses, i.e. the chapters about *Advanced User Behaviour Analysis 2* and *Mobile Viewport Tracking 3*.

A.1 Mobile Viewport Tracking Sources

A.1.1 Heatmap Drawers

Listing A.1: The Interface for Heatmap Drawers

```
1 package at.tuwien.viewport.utils.heatmapdrawer;
2
3 import java.awt.image.BufferedImage;
4
5 import at.tuwien.viewport.persistence.entities.Heatmap;
6 import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
7
8 public interface IHeatmapDrawer {
9
10     public BufferedImage drawHeatmap(Heatmap heatmap);
11
12     public boolean drawAndSaveHeatmap(Heatmap heatmap, String outputDirectoryPath,
13         String filename)
14         throws MVTScreenshotReadException;
15 }
```

Listing A.2: The Abstract Base Class for Heatmap Drawers

```
1 package at.tuwien.viewport.utils.heatmapdrawer;
2
3 import java.awt.image.BufferedImage;
4 import java.io.File;
```

```

5  import java.io.FileNotFoundException;
6  import java.io.IOException;
7  import java.util.logging.Logger;
8
9  import javax.imageio.ImageIO;
10
11 import at.tuwien.viewport.persistence.entities.Heatmap;
12 import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
13 import at.tuwien.viewport.screenshot.MVTPropertyNotFoundException;
14 import at.tuwien.viewport.screenshot.ScreenshotLoader;
15
16 public abstract class AbstractMapDrawer implements IHeatmapDrawer {
17
18     protected final static Logger log = Logger.getLogger(AbstractMapDrawer.class.
19         getName());
20
21     protected String fileExtension = ".png";
22
23     protected String filePostfix = "";
24
25     public AbstractMapDrawer(String filePostfix)
26     {
27         this.filePostfix = filePostfix;
28     }
29
30     @Override
31     public boolean drawAndSaveHeatmap(Heatmap heatmap, String outputDirectoryPath,
32         String filename)
33         throws MVTScreenshotReadException {
34         File file = new File(outputDirectoryPath + filename + "-" + heatmap.
35             getHeatmapVersion() + filePostfix
36             + fileExtension);
37         if (heatmap.getImage() == null || heatmap.getImage().getImage() == null)
38         {
39             boolean alternativeScreenshotTaken = false;
40
41             if (heatmap.getImage() != null && heatmap.getMetadata() != null && heatmap.
42                 getMetadata().getScreenshots() != null)
43             {
44                 try {
45                     for (String s : heatmap.getMetadata().getScreenshots())
46                     {
47                         if (ScreenshotLoader.exists(s))
48                         {
49                             heatmap.getImage().setImage(ScreenshotLoader.loadScreenshot
50                                 (s));
51                             alternativeScreenshotTaken = true;
52                             break;
53                         }
54                     }
55                 }
56                 if (!alternativeScreenshotTaken)
57                 {
58                     for (String s : heatmap.getMetadata().getScreenshots())
59                     {
60                         if (ScreenshotLoader.exists("RECREATED_" + s))
61                         {
62                             heatmap.getImage().setImage(ScreenshotLoader.
63                                 loadScreenshot("RECREATED_" + s));
64                             alternativeScreenshotTaken = true;
65                             break;

```

```

61         }
62     }
63 }
64     } catch (MVTPropertyNotFoundException e) {
65         e.printStackTrace();
66     } catch (FileNotFoundException e) {
67         e.printStackTrace();
68     } catch (IOException e) {
69         e.printStackTrace();
70     }
71 }
72
73
74 }
75 BufferedImage img = this.drawHeatmap(heatmap);
76 if (img == null || file == null)
77 {
78     return false;
79 }
80 try {
81     ImageIO.write(img, "png", file);
82 } catch (IOException e) {
83     throw new MVTScreenshotReadException("Could not write Screenshot");
84 } catch (NullPointerException e) {
85     throw new MVTScreenshotReadException(e);
86 }
87 return true;
88 }
89
90
91 }

```

Listing A.3: The Colorful Heatmap Drawer

```

1  package at.tuwien.viewport.utils.heatmapdrawer;
2
3  import java.awt.AlphaComposite;
4  import java.awt.Color;
5  import java.awt.Graphics2D;
6  import java.awt.image.BufferedImage;
7  import java.io.IOException;
8  import java.net.URL;
9  import java.util.Arrays;
10 import java.util.logging.Level;
11
12 import javax.imageio.ImageIO;
13
14 import at.tuwien.viewport.persistence.entities.Heatmap;
15 import at.tuwien.viewport.persistence.exceptions.MVTPersistenceException;
16 import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
17 import at.tuwien.viewport.utils.HeatmapManager;
18
19 public class HeatMapDrawer extends AbstractMapDrawer {
20
21     /** half of the size of the circle picture. */
22     private static int HALFCIRCLEPICSIZE = 64;
23
24     /** path to picture of circle which gets more transparent to the outside. */
25     private static String CIRCLEPIC;
26     private static String SPECTRUMPIC;

```

```

27 private static String EMPTYIMAGE = "/empty.png";
28
29 private float multiplier = 1.0f;
30
31 private int[] circles = new int[] { 256, 128, 96, 64, 48, 32, 24, 16, 8 };
32
33 public HeatMapDrawer() {
34     super("heat");
35     SPECTRUMPIC = "/colors.png";
36     CIRCLEPIC = "/circle_128.png";
37 }
38
39 /**
40  * @param multiplier
41  *         calculated opacity of every point will be
42  *         multiplied by this value. This leads to a HeatMap that is easier to
43  *         read,
44  *         especially when there are not too many points or the points are too
45  *         spread out. Pass 1.0f for original.
46 */
47 public HeatMapDrawer(float multiplier, int circleSize) {
48     super("heat-" + circleSize);
49
50     if (!Arrays.asList(circles).contains(circleSize))
51     {
52         log.log(Level.SEVERE, "Could not create drawer because circle size not
53             available.");
54         return;
55     }
56     this.multiplier = multiplier;
57     SPECTRUMPIC = "/colors.png";
58     CIRCLEPIC = "/circle_" + circleSize + ".png";
59     HALFCIRCLEPICSIZE = circleSize / 2;
60 }
61
62 @Override
63 public BufferedImage drawHeatmap(Heatmap heatmap) {
64
65     BufferedImage mapPic = heatmap.getImage().getImage();
66
67     if (mapPic == null) {
68         log.log(Level.INFO, "Drawing Heatmap with empty image for URL: " + heatmap.
69             getUrl());
70         // return null;
71         mapPic = loadImage(EMPTYIMAGE);
72     }
73
74     int maxXValue = mapPic.getWidth();
75     int maxYValue = mapPic.getHeight();
76
77     BufferedImage circle = loadImage(CIRCLEPIC);
78     BufferedImage heatMap = new BufferedImage(maxXValue, maxYValue, 6);
79     paintInColor(heatMap, Color.white);
80
81     double maxWeight = heatmap.getMaximumWeightOfImportanceMatrix();
82     double lowerLimit = 0.0;
83
84     for (int i = 0; i < heatmap.getImportanceMatrix().length; i++) {
85
86         int positionY = HeatmapManager.TILE_SIZE * i;
87
88         for (int j = 0; j < heatmap.getImportanceMatrix()[i].length; j++)

```

```

86     {
87         int positionX = HeatmapManager.TILE_SIZE * j;
88         // now we are on the pos i j
89         double valueAtCurrentPos = heatmap.getImportanceMatrix()[i][j];
90
91         if (valueAtCurrentPos > lowerLimit)
92         {
93             float weight = (float) (valueAtCurrentPos / maxWeight) * multiplier
94             ;
95             addImage(heatmap, circle, weight,
96                     (positionX - HeatmapManager.TILE_SIZE / 2 -
97                      HALFCIRCLEPICSIZE),
98                     (positionY - HeatmapManager.TILE_SIZE / 2 -
99                      HALFCIRCLEPICSIZE));
100         }
101     }
102     heatmap = colorImage(heatmap);
103     addImage(mapPic, heatmap, 0.4f);
104     return mapPic;
105 }
106 /**
107  * First it negates the image.
108  * Then remaps black and white picture with colors.
109  * It uses the colors from SPECTRUMPIC. The whiter a pixel is, the more it
110  * will get a color from the bottom of it. Black will stay black.
111  *
112  * @param heatmapBW
113  *         black and white heat map
114  */
115 private BufferedImage colorImage(BufferedImage img) {
116     BufferedImage colorGradient = loadImage(SPECTRUMPIC);
117     int gradientHeight = colorGradient.getHeight() - 1;
118     int width = img.getWidth();
119     int height = img.getHeight();
120     for (int x = 0; x < width; x++) {
121         for (int y = 0; y < height; y++) {
122
123             // (1) Negate
124             int rGB = img.getRGB(x, y);
125
126             // Swaps values
127             // i.e. 255, 255, 255 (white)
128             // becomes 0, 0, 0 (black)
129             int r = Math.abs(((rGB >>> 16) & 0xff) - 255); // red inverted
130             int g = Math.abs(((rGB >>> 8) & 0xff) - 255); // green inverted
131             int b = Math.abs((rGB & 0xff) - 255); // blue inverted
132
133             // get heatmapBW color values:
134             rGB = (r << 16) | (g << 8) | b;
135
136             // (2) Remap
137
138             // calculate multiplier to be applied to height of gradient.
139             float multiplier = rGB & 0xff; // blue
140             multiplier *= ((rGB >>> 8) & 0xff); // green
141             multiplier *= (rGB >>> 16) & 0xff; // red
142             multiplier /= 16581375; // 255f * 255f * 255f
143
144             // apply multiplier

```

```

145         int colorY = (int) (multiplier * gradientHeight);
146
147         // remap values
148         // calculate new value based on whiteness of heatMap
149         // (the whiter, the more a color from the top of colorGradient
150         // will be chosen.
151         int mappedRGB = colorGradient.getRGB(0, colorY);
152         // set new value
153         img.setRGB(x, y, mappedRGB);
154
155     }
156 }
157 return img;
158 }
159
160 /**
161  * changes all pixel in the buffer to the provided color.
162  *
163  * @param buff
164  *         buffer
165  * @param color
166  *         color
167  */
168 private void paintInColor(BufferedImage buff, Color color) {
169     Graphics2D g2 = buff.createGraphics();
170     g2.setColor(color);
171     g2.fillRect(0, 0, buff.getWidth(), buff.getHeight());
172     g2.dispose();
173 }
174
175 /**
176  * prints the contents of buff2 on buff1 with the given opaque value
177  * starting at position 0, 0.
178  *
179  * @param buff1
180  *         buffer
181  * @param buff2
182  *         buffer to add to buff1
183  * @param opaque
184  *         opacity
185  */
186 private void addImage(
187     BufferedImage buff1, BufferedImage buff2, float opaque) {
188     addImage(buff1, buff2, opaque, 0, 0);
189 }
190
191 /**
192  * prints the contents of buff2 on buff1 with the given opaque value.
193  *
194  * @param buff1
195  *         buffer
196  * @param buff2
197  *         buffer
198  * @param opaque
199  *         how opaque the second buffer should be drawn
200  * @param x
201  *         x position where the second buffer should be drawn
202  * @param y
203  *         y position where the second buffer should be drawn
204  */
205 private void addImage(BufferedImage buff1, BufferedImage buff2,
206     float opaque, int x, int y) {

```



```

207     Graphics2D g2d = buff1.createGraphics();
208     g2d.setComposite(
209         AlphaComposite.getInstance(AlphaComposite.SRC_OVER, opaque));
210     g2d.drawImage(buff2, x, y, null);
211     g2d.dispose();
212 }
213
214 private BufferedImage loadImage(String ref) {
215     BufferedImage b1 = null;
216     try {
217         URL url = getClass().getResource(ref);
218         b1 = ImageIO.read(url);
219     } catch (IOException e) {
220         log.log(Level.SEVERE, "error loading the image.", e);
221     }
222     return b1;
223 }
224
225 public static void main(String[] args) throws MVTScreenshotReadException,
226     MVTPersistenceException
227 {
228     HeatmapManager.getInstance().drawAllHeatmapsFromDatabase(new HeatMapDrawer());
229 }

```

Listing A.4: The Rastermap Drawer

```

1  package at.tuwien.viewport.utils.heatmapdrawer;
2
3  import java.awt.AlphaComposite;
4  import java.awt.BasicStroke;
5  import java.awt.Color;
6  import java.awt.Font;
7  import java.awt.Graphics2D;
8  import java.awt.image.BufferedImage;
9  import java.text.NumberFormat;
10
11 import at.tuwien.viewport.persistence.entities.Heatmap;
12 import at.tuwien.viewport.persistence.exceptions.MVTPersistenceException;
13 import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
14 import at.tuwien.viewport.utils.HeatmapManager;
15
16 public class RasterMapDrawer extends AbstractMapDrawer {
17
18     public RasterMapDrawer() {
19         super("raster");
20     }
21
22     @Override
23     public BufferedImage drawHeatmap(Heatmap heatmap) {
24         BufferedImage img = heatmap.getImage().getImage();
25         if (img == null)
26         {
27             return null;
28         }
29
30         Graphics2D g2d = img.createGraphics();
31
32         BasicStroke bs = new BasicStroke(1);
33         g2d.setStroke(bs);

```

```

34     Font font = new Font("Serif", Font.PLAIN, 8);
35     g2d.setFont(font);
36
37     // needed for scaling, making it bigger than the maximum to have it always a
38     // bit transparent
39     double maxWeight = Math.log(heatmap.getMaximumWeightOfImportanceMatrix()) *
40     1.2;
41
42     boolean drawRaster = true;
43     for (int i = 0; i < heatmap.getImportanceMatrix().length; i++) {
44         int positionY = HeatmapManager.TILE_SIZE * i;
45         if (positionY < img.getHeight())
46         {
47             // Drawing horizontal line
48             if (drawRaster)
49             {
50                 g2d.setColor(Color.GRAY);
51                 g2d.drawLine(0, positionY, img.getWidth(), positionY);
52             }
53
54             for (int j = 0; j < heatmap.getImportanceMatrix()[i].length; j++)
55             {
56                 int positionX = HeatmapManager.TILE_SIZE * j;
57                 if (positionX < img.getWidth())
58                 {
59                     // Drawing vertical line
60                     // Because this is a nested loop has to be checked that it is only
61                     // done once
62                     if (drawRaster && i == 0)
63                     {
64                         g2d.setColor(Color.GRAY);
65                         g2d.drawLine(positionX, 0, positionX, img.getHeight());
66                     }
67                     // now we are on the pos i j
68                     double valueAtCurrentPos = heatmap.getImportanceMatrix()[i][j];
69
70                     double loggedValue = Math.log10(valueAtCurrentPos);
71                     if (loggedValue < 0)
72                     {
73                         loggedValue = 0;
74                     }
75                     setColorForCurrentPos(loggedValue, maxWeight, g2d);
76                     drawRectToTile(i, j, loggedValue, g2d, drawRaster);
77                 }
78             }
79             g2d.drawImage(img, null, 0, 0);
80             g2d.dispose();
81             return img;
82         }
83     }
84
85     private void drawRectToTile(int y, int x, double valueAtCurrentPos, Graphics2D g2d,
86     boolean drawRaster) {
87         int tileSize = HeatmapManager.TILE_SIZE;
88         int positionX = tileSize * x;
89         int positionY = tileSize * y;
90         g2d.fillRect(positionX + 1, positionY + 1, tileSize - 1, tileSize - 1);
91
92         // Lighter text color
93         g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 0.5f));

```

```

92
93     if (drawRaster)
94     {
95         g2d.setColor(Color.GRAY);
96         NumberFormat nf = NumberFormat.getInstance();
97         nf.setMaximumFractionDigits(1);
98         g2d.drawString(nf.format(valueAtCurrentPos), positionX + 2, positionY + 13)
99         ;
100    }
101
102    // Reset composite
103    g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 1));
104
105
106    private void setColorForCurrentPos(double valueAtCurrentPos, double maxWeight,
107        Graphics2D g2d) {
108        float r = 1.0f, g = 0f, b = 0f;
109        float opacity = (float) (valueAtCurrentPos / maxWeight); // between 0 and 1
110
111        g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, opacity));
112        g2d.setColor(new Color(r, g, b, opacity));
113    }
114
115    public static void main(String[] args) throws MVTScreenshotReadException,
116        MVTPersistenceException
117    {
118        HeatmapManager.getInstance().drawAllHeatmapsFromDatabase(new RasterMapDrawer())
119        ;
120    }

```

A.1.2 Action Type Determiner

Listing A.5: The Action Type Determiner

```

1  package at.tuwien.viewport.utils;
2
3  import java.lang.reflect.Method;
4  import java.util.logging.Logger;
5
6  import at.tuwien.viewport.persistence.entities.ActionType;
7  import at.tuwien.viewport.persistence.entities.ViewportEntry;
8  import at.tuwien.viewport.persistence.entities.ViewportEntry.Fields;
9
10 public class ActionTypeDeterminer {
11
12     private ViewportEntry oldViewport;
13     private ViewportEntry newViewport;
14     private static final int tolerance = 5; //Tolerance in pixels
15
16     private enum Change {
17         BIGGER,
18         EQUAL,
19         SMALLER;
20     }
21
22     private final static Logger log = Logger

```

```

23         .getLogger(ActionTypeDeterminer.class.getName());
24
25     public ActionTypeDeterminer(ViewportEntry oldViewport,
26         ViewportEntry newViewport) {
27         this.oldViewport = oldViewport;
28         this.newViewport = newViewport;
29     }
30
31
32
33     public ActionType getActionType() {
34
35         // ZOOM-IN
36         if (is(ViewportEntry.Fields.innerHeight, Change.SMALLER)
37             && is(ViewportEntry.Fields.innerWidth, Change.SMALLER)) {
38             return ActionType.ZOOM_IN;
39         }
40         // ZOOM-OUT
41         if (is(ViewportEntry.Fields.innerHeight, Change.BIGGER)
42             && is(ViewportEntry.Fields.innerWidth, Change.BIGGER)) {
43             return ActionType.ZOOM_OUT;
44         }
45         // SCROLL_UP_LEFT
46         if (is(ViewportEntry.Fields.pageOffsetX, Change.SMALLER)
47             && is(ViewportEntry.Fields.pageOffsetY, Change.SMALLER)
48             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
49             return ActionType.SCROLL_UP_LEFT;
50         }
51         // SCROLL_UP_RIGHT
52         if (is(ViewportEntry.Fields.pageOffsetX, Change.BIGGER)
53             && is(ViewportEntry.Fields.pageOffsetY, Change.SMALLER)
54             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
55             return ActionType.SCROLL_UP_RIGHT;
56         }
57         // SCROLL_DOWN_LEFT
58         if (is(ViewportEntry.Fields.pageOffsetX, Change.SMALLER)
59             && is(ViewportEntry.Fields.pageOffsetY, Change.BIGGER)
60             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
61             return ActionType.SCROLL_DOWN_LEFT;
62         }
63         // SCROLL_DOWN_RIGHT
64         if (is(ViewportEntry.Fields.pageOffsetX, Change.BIGGER)
65             && is(ViewportEntry.Fields.pageOffsetY, Change.BIGGER)
66             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)
67             && is(ViewportEntry.Fields.innerHeight, Change.EQUAL)) {
68             return ActionType.SCROLL_DOWN_RIGHT;
69         }
70
71         // SCROLL_LEFT
72         if (is(ViewportEntry.Fields.pageOffsetX, Change.SMALLER)
73             && is(ViewportEntry.Fields.pageOffsetY, Change.EQUAL)
74             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
75             return ActionType.SCROLL_LEFT;
76         }
77         // SCROLL_RIGHT
78         if (is(ViewportEntry.Fields.pageOffsetX, Change.BIGGER)
79             && is(ViewportEntry.Fields.pageOffsetY, Change.EQUAL)
80             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
81             return ActionType.SCROLL_RIGHT;
82         }
83         // SCROLL_UP
84         if (is(ViewportEntry.Fields.pageOffsetX, Change.EQUAL)

```

```

85         && is(ViewportEntry.Fields.pageOffsetY, Change.SMALLER)
86         && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
87             return ActionType.SCROLL_UP;
88         }
89         // SCROLL_DOWN
90         if (is(ViewportEntry.Fields.pageOffsetX, Change.EQUAL)
91             && is(ViewportEntry.Fields.pageOffsetY, Change.BIGGER)
92             && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
93             return ActionType.SCROLL_DOWN;
94         }
95
96
97         return null;
98     }
99
100
101     private Boolean is(Fields field, Change change) {
102         String methodName = "get"
103             + field.toString().substring(0, 1).toUpperCase()
104             + field.toString().substring(1);
105         try {
106             Method getter = oldViewport.getClass().getMethod(methodName);
107             int oldValue = (Integer) getter.invoke(oldViewport);
108             int newValue = (Integer) getter.invoke(newViewport);
109
110             switch (change) {
111                 case BIGGER:
112                     if (newValue > oldValue + tolerance) {
113                         return true;
114                     }
115                     break;
116                 case SMALLER:
117                     if (newValue < oldValue - tolerance) {
118                         return true;
119                     }
120                     break;
121                 case EQUAL:
122                     if (newValue == oldValue) {
123                         return true;
124                     } else if (newValue > oldValue) {
125                         if (newValue <= oldValue + tolerance) {
126                             return true;
127                         }
128                     } else if (newValue < oldValue) {
129                         if (newValue + tolerance >= oldValue) {
130                             return true;
131                         }
132                     }
133                     break;
134             }
135             return false;
136         } catch (Exception e) {
137             log.severe("Exception: " + e.getMessage());
138         }
139     }
140
141     return null;
142 }
143
144 }

```

A.1.3 Client Side - Viewport.js

Listing A.6: The JavaScript that is injected into the websites

```
1 //Version 2.1
2
3 // MVT namespace to avoid conflicts
4 var MVT = {};
5 // Basic configuration
6 MVT.Config =
7 {
8     SERVER_URL : null // Set within "init"
9 };
10
11 MVT.DOMEvents =
12 {
13     CLICK : "CLICK",
14     SUBMIT : "SUBMIT",
15     INPUT_CHANGED : "INPUT_CHANGED",
16     KEY_PRESS : "KEY_PRESS",
17     WEBSITE_ACTIVATED : "WEBSITE_ACTIVATED",
18     WEBSITE_DEACTIVATED : "WEBSITE_DEACTIVATED",
19     ROTATION_TO_HORIZONTAL : "ROTATION_TO_HORIZONTAL",
20     ROTATION_TO_VERTICAL : "ROTATION_TO_VERTICAL",
21     DOUBLE_TAP : "DOUBLE_TAP",
22     HOLD : "HOLD"
23 };
24
25 //Variables
26 MVT.currentX = -1;
27 MVT.currentY = -1;
28 MVT.innerWidth = -1;
29 MVT.innerHeight = -1;
30 MVT.outerWidth = -1;
31 MVT.outerHeight = -1;
32 MVT.scrollWidth = -1;
33 MVT.scrollHeight = -1;
34
35 MVT.queuedData = [];
36 //check for race conditions
37 MVT.post_id;
38 MVT.OSName;
39 // Cookie within this page
40 MVT.pageLoadToken;
41 // Cookie within same domain
42 MVT.domainUserToken;
43 MVT.userToken;
44
45 //filter Keywords - Do not report Viewport of Ads
46 MVT.AD_KEYWORDS = new Array("/uim.html", "partner=", "banner.html",
47 "/AdServer/", "/MetaAdServer/", "a.ligatus.com", "doubleclick.net",
48 "plugins/like.php", "plugins/activity.php", "likebox.php", "Banner.php", "ads.", "
49     heatmaps.no-ip.org"
50 );
51 MVT.lastEventTimeStamp = 0;
52 MVT.scrolling = false;
53 MVT.pauseDetection = false;
54 MVT.scrollTimer = false;
55 MVT.blockPasswordTransmission = true;
56
```

```

57 MVT.init = function(serverIP)
58 {
59     if(!serverIP)
60     {
61         console.error("Server IP of MVTService needed.");
62         return;
63     }
64     MVT.Config.SERVER_URL = "http://" + serverIP + "/MVTService/rs/dataCollector/
        storeData";
65
66     //Setting, getting cookie
67     var found = false;
68     var cookies = document.cookie.split(";");
69     for (var i=0; i<cookies.length; i++) {
70         var keyValue = cookies[i].split("=");
71         if (keyValue[0].match("viewport")) {
72             MVT.domainUserToken = keyValue[1];
73             found = true;
74             break;
75         }
76     }
77     if (!found) {
78         MVT.domainUserToken = MVT.getRandom();
79         //expiration is 365 days
80         document.cookie = "viewport=" + MVT.domainUserToken + "; expires=" +
            MVT.getExpireDate(365, 0) + "; path=/";
81     }
82
83     //Setting One-Time-Variables
84     MVT.post_id = 1;
85     MVT.pageLoadToken = MVT.getRandom();
86     MVT.OSName = navigator.appVersion;
87     if(!MVT.filterDocuments(document.URL))
88     {
89         var iDevice = navigator.userAgent.match(/iPad/i) != null || navigator.
            userAgent.match(/iPhone/i) != null;
90         var androidDevice = navigator.userAgent.match(/Android/i) != null;
91
92         window.addEventListener('message', function (e)
93         {
94             if (e.origin == ("http://" + serverIP))
95             {
96                 var data = e.data;
97                 if(data)
98                 {
99                     MVT.userToken = data.substring(data.indexOf("=")+1);
100                 }
101             }
102         }, false);
103
104         window.addEventListener('load', function()
105         {
106             MVT.registerDOMEvents();
107
108             //alert("w:"+document.body.scrollWidth + ",h:" + document.body.
                scrollHeight);
109
110             window.addEventListener('pageshow', function()
111             {
112                 MVT.queueData(MVT.DOMEvents.WEBSITE_ACTIVATED, null, "
                    window-pageshow");
113             }, false);

```

```

114 window.addEventListener('focus', function()
115 {
116     MVT.queueData(MVT.DOMEvents.WEBSITE_ACTIVATED, null, "
117         window-focus");
118 }, false);
119
120 // Android fires "blur" but iPhone the "pagehide"
121 if(androidDevice)
122 {
123     window.addEventListener('blur', function(){ MVT.
124         queueData(MVT.DOMEvents.WEBSITE_DEACTIVATED, null,
125             "window-blur"); }, false);
126
127     }
128     else if (iDevice)
129     {
130         window.addEventListener('pagehide', function(){ MVT.
131             queueData(MVT.DOMEvents.WEBSITE_DEACTIVATED, null,
132                 "window-pagehide"); MVT.postData(); }, false);
133     }
134
135     window.addEventListener('scroll', function( event )
136     {
137         if (!MVT.scrolling)
138         {
139             MVT.scrolling = true;
140         }
141
142         clearTimeout( MVT.scrollTimer );
143         MVT.scrollTimer = setTimeout(function()
144         {
145             MVT.scrolling = false;
146             if(MVT.pauseDetection)
147             {
148                 MVT.pauseDetection = false;
149             }
150             }, 100 );
151     }, false);
152
153     window.addEventListener('orientationchange', function()
154     {
155         if ( window.orientation == 0 || window.orientation == 180 )
156         {
157             MVT.queueData(MVT.DOMEvents.ROTATION_TO_VERTICAL);
158         }
159         else if ( window.orientation == 90 || window.orientation == -90 )
160         {
161             MVT.queueData(MVT.DOMEvents.ROTATION_TO_HORIZONTAL);
162         }
163     }, false);
164
165     var hammer = new Hammer(document.body, {
166         swipe: false
167     });
168
169     hammer.ondragstart = function(ev) {
170         MVT.pauseDetection = true;
171     };
172     //hammer.ondrag = function(ev) { };
173     hammer.ondragend = function(ev) {
174         if(!MVT.scrolling)

```



```

171         {
172             // No "lazy" scrolling, directly capture data
173             MVT.pauseDetection = false;
174         }
175     };
176
177     //hammer.ontap = function(ev) { };
178     hammer.ondoubletap = function(ev, data) {
179         MVT.queueData(MVT.DOMEvents.DOUBLE_TAP, null, null, ev.
            position[0].x + ";" + ev.position[0].y);
180     };
181     hammer.onhold = function(ev) {
182         MVT.queueData(MVT.DOMEvents.HOLD, null, null, ev.
            position[0].x + ";" + ev.position[0].y);
183     };
184
185     hammer.ontransformstart = function(ev) {
186         MVT.pauseDetection = true;
187     };
188     //hammer.ontransform = function(ev) { };
189     hammer.ontransformend = function(ev) {
190         MVT.pauseDetection = false;
191         //MVT.queueData();
192     };
193
194     //hammer.onrelease = function(ev) { };
195
196     setInterval("MVT.detectChanges()", 200);
197     setInterval("MVT.postData()", 1500);
198     }, false);
199     }
200 };
201
202 MVT.registerDOMEvents = function()
203 {
204     var fnRegisterListener = function(aElement, sListenerName, aDOMEventEnum,
        bStoreValue)
205     {
206         aElement.addEventListener(sListenerName, function (event)
207         {
208             var aValue = bStoreValue ? aElement.value : null;
209             // Click is performed for an element and all underlying
                elements
210             // therefore the timestamp is compared to just check the first
                element
211             if(MVT.lastEventTimeStamp == event.timeStamp)
212             {
213                 return;
214             }
215             var actionTypesData = null;
216             if(aDOMEventEnum == MVT.DOMEvents.CLICK)
217             {
218                 actionTypesData = event.x + ";" + event.y;
219             }
220             if(aDOMEventEnum == MVT.DOMEvents.INPUT_CHANGED ||
                aDOMEventEnum == MVT.DOMEvents.KEY_PRESS)
221             {
222                 if(event.target && event.target.type && event.target.
                    type == "password" && MVT.
                    blockPasswordTransmission)
223                 {
224                     var aLength = aValue.length;

```

```

225         var aFirstPWChars = "";
226         if(aLength > 0)
227         {
228             aFirstPWChars = aValue.substring(0,1);
229         }
230         aValue = "Password: ";
231         if(aLength > 0)
232         {
233             aValue += aFirstPWChars;
234         }
235
236         for(var i=1; i<aLength; i++)
237         {
238             aValue += "*";
239         }
240     }
241 }
242
243     MVT.lastEventTimeStamp = event.timeStamp;
244     MVT.queueData(aDOMEventEnum, MVT.getElementXPath(this), aValue,
        actionTypesData);
245     MVT.postData();
246     }, false);
247 };
248
249 MVT.forAllTags("a", MVT.bind(fnRegisterListener, ['click', MVT.DOMEVENTS.CLICK,
        false], true));
250 MVT.forAllTags("area", MVT.bind(fnRegisterListener, ['click', MVT.DOMEVENTS.
        CLICK, false], true));
251 MVT.forAllTags("span", MVT.bind(fnRegisterListener, ['click', MVT.DOMEVENTS.
        CLICK, false], true));
252 MVT.forAllTags("div", MVT.bind(fnRegisterListener, ['click', MVT.DOMEVENTS.
        CLICK, false], true));
253 MVT.forAllTags("p", MVT.bind(fnRegisterListener, ['click', MVT.DOMEVENTS.CLICK,
        false], true));
254
255 // button, checkbox, radio
256 MVT.forAllTags("input", MVT.bind(fnRegisterListener, ['change', MVT.DOMEVENTS.
        INPUT_CHANGED, true], true));
257 MVT.forAllTags("textarea", MVT.bind(fnRegisterListener, ['change', MVT.
        DOMEVENTS.INPUT_CHANGED, true], true));
258 MVT.forAllTags("input", MVT.bind(fnRegisterListener, ['keydown', MVT.DOMEVENTS.
        KEY_PRESS, true], true));
259 MVT.forAllTags("textarea", MVT.bind(fnRegisterListener, ['keydown', MVT.
        DOMEVENTS.KEY_PRESS, true], true));
260
261 MVT.forAllTags("select", MVT.bind(fnRegisterListener, ['change', MVT.DOMEVENTS.
        INPUT_CHANGED, true], true));
262 MVT.forAllTags("form", MVT.bind(fnRegisterListener, ['submit', MVT.DOMEVENTS.
        SUBMIT, false], true));
263 };
264
265 MVT.forAllTags = function(sTagName, fnCallback)
266 {
267     if(!fnCallback)
268     {
269         return;
270     }
271     var elements = document.getElementsByTagName(sTagName);
272     for(var i = 0; i < elements.length; i++)
273     {
274         fnCallback(elements[i]);

```

```

275     }
276 };
277
278 /**
279  * Possibility to specify arguments which are always appended when a function is
        called
280  */
281 MVT.bind = function(fn, args, appendArgs)
282 {
283     var method = fn,
284         slice = Array.prototype.slice;
285
286     return function() {
287         var callArgs = args || arguments;
288
289         if (appendArgs === true) {
290             callArgs = slice.call(arguments, 0);
291             callArgs = callArgs.concat(args);
292         }
293         else if (typeof appendArgs == 'number') {
294             callArgs = slice.call(arguments, 0); // copy arguments first
295             Ext.Array.insert(callArgs, appendArgs, args);
296         }
297
298         return method.apply(window, callArgs);
299     };
300 };
301
302 MVT.getRandom = function()
303 {
304     return Math.floor(Math.random()*1000000000000001);
305 };
306
307 MVT.getElementXPath = function(element) {
308     // TODO: Also handle className and name attributes to optimize xpath
309     if (element && element.id)
310     {
311         return '//*[@id="' + element.id + '"']';
312     }
313     else
314     {
315         return MVT.getElementTreeXPath(element);
316     }
317 };
318
319 MVT.getElementTreeXPath = function(element) {
320     var paths = [];
321
322     // Use nodeName (instead of localName) so namespace prefix is included (if any).
323     for (; element && element.nodeType == 1; element = element.parentNode) {
324         var index = 0;
325         // EXTRA TEST FOR ELEMENT.ID
326         if (element && element.id) {
327             paths.splice(0, 0, '//*[@id="' + element.id + '"']');
328             break;
329         }
330
331         for (var sibling = element.previousSibling; sibling; sibling = sibling.
            previousSibling) {
332             // Ignore document type declaration.
333             if (sibling.nodeType == Node.DOCUMENT_TYPE_NODE)
334                 continue;

```

```

335         if (sibling.nodeName == element.nodeName)
336             ++index;
337     }
338 }
339
340     var tagName = element.nodeName.toLowerCase();
341     var pathIndex = (index ? "[" + (index+1) + "]" : "");
342     paths.splice(0, 0, tagName + pathIndex);
343 }
344
345     return paths.length ? "/" + paths.join("/") : null;
346 };
347
348 MVT.getExpireDate = function(tage, stunden)
349 {
350     var jetzt = new Date();
351     var zeit = jetzt.getTime();
352     var zukunft = zeit + (((tage * 24) + stunden) * 3600 * 1000);
353     jetzt.setTime(zukunft);
354     var haltbarkeit = jetzt.toUTCString();
355     return haltbarkeit;
356 };
357
358 MVT.filterDocuments = function(url) {
359     ow = window.outerWidth;
360     oh = window.outerHeight;
361
362     // Filter for iframes
363     if (window.top != window.self)
364     {
365         return true;
366     }
367     if(ow < 80 && oh < 80){
368         //filter small add-frames or faceook-like-frames or something like that
369         return true;
370     }
371     for (var i=0; i< MVT.AD_KEYWORDS.length; i++)
372     {
373         if(url.match(MVT.AD_KEYWORDS[i])){
374             return true;
375         }
376     }
377     return false;
378 };
379
380 MVT.detectChanges = function() {
381     if (!MVT.pauseDetection && (MVT.changed(window.pageXOffset, MVT.currentX)
382         || MVT.changed(window.pageYOffset, MVT.currentY)
383         || MVT.changed(window.innerWidth, MVT.innerWidth)
384         || MVT.changed(window.innerHeight, MVT.innerHeight)
385         || MVT.changed(window.outerWidth, MVT.outerWidth)
386         || MVT.changed(window.outerHeight, MVT.outerHeight))
387         )
388     {
389         MVT.queueData();
390     }
391 };
392
393 MVT.changed = function(val1, val2)
394 {
395     //TOLERANCE for equals
396     var tol = 5;

```

```

397     if (val1 > val2) {
398         if(val1 <= val2 + tol)
399             {
400                 return false;
401             }
402     } else if (val1 < val2) {
403         if(val1 + tol >= val2)
404             {
405                 return false;
406             }
407     } else {
408         return false;
409     }
410     return true;
411 };
412
413 MVT.queueData = function(aEventType, sXPath, sElementValue, actionTypeData)
414 {
415     MVT.currentX = window.pageXOffset;
416     MVT.currentY = window.pageYOffset;
417     MVT.innerWidth = window.innerWidth;
418     MVT.innerHeight = window.innerHeight;
419     MVT.outerWidth = window.outerWidth;
420     MVT.outerHeight = window.outerHeight;
421     MVT.scrollWidth = document.body.scrollWidth;
422     MVT.scrollHeight = document.body.scrollHeight;
423
424     var recorded_millis = new Date().getTime();
425
426     var aData = {
427         pageLoadToken : MVT.pageLoadToken,
428         URL : document.URL,
429         OS : MVT.OSName,
430         pageOffsetX : MVT.currentX,
431         pageOffsetY : MVT.currentY,
432         innerWidth : MVT.innerWidth,
433         innerHeight : MVT.innerHeight,
434         outerWidth : MVT.outerWidth,
435         outerHeight : MVT.outerHeight,
436         scrollWidth : MVT.scrollWidth,
437         scrollHeight : MVT.scrollHeight,
438         domainUserToken : MVT.domainUserToken,
439         recordedTime : recorded_millis,
440         postId : MVT.post_id++,
441         actionType : aEventType ? aEventType : null, //
442             explicit set null to avoid "undefined"
443         actionTypeData : actionTypeData ? actionTypeData :
444             null,
445         domXPath : sXPath ? sXPath : null,
446         domElementValue : sElementValue ? sElementValue :
447             null,
448         referrer : document.referrer != "" ? document.
449             referrer : null,
450         domain : window.location.host
451     };
452
453     // Message from iframe not yet received
454     MVT.waitForUserToken(aData);
455 };

```

```

455 MVT.waitForUserToken = function(aData)
456 {
457     if(!MVT.userToken)
458     {
459         setTimeout(function()
460             {
461                 MVT.waitForUserToken(aData);
462             }, 300);
463     }
464     else
465     {
466         aData.userToken = MVT.userToken;
467         MVT.queuedData.push(aData);
468     }
469 };
470 };
471
472 MVT.postData = function()
473 {
474     if(MVT.queuedData.length == 0)
475     {
476         return;
477     }
478     var sData = JSON.stringify(MVT.queuedData);
479
480     MVT.queuedData = [];
481
482
483     var xmlHttp = new XMLHttpRequest();
484     xmlHttp.open("POST", MVT.Config.SERVER_URL, true);
485     xmlHttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
486     xmlHttp.send("jsonData=" + MVT.Base64.encode(sData));
487 };
488
489 MVT.postJSONP = function()
490 {
491     var script = document.createElement("script");
492     script.type = "text/javascript";
493     script.src = MVT.Config.SERVER_URL + "?data=" + encodeURIComponent(MVT.Base64.
494         encode(sData));
495     var head = document.getElementsByTagName("head")[0];
496     (head || document.body).appendChild(script);
497 };
498
499 MVT.callback = function()
500 {
501     console.info("stored data");
502 };
503
504 //-----
505 //-----      Base 64 Encoding
506 //-----      http://www.webtoolkit.info/
507 //-----
508 MVT.Base64 =
509 {
510     /* The sources for Base 64 encoding where included here but are not
511     * included in this listing. They can be found on the project's
512     * homepage:
513     * http://www.webtoolkit.info/
514     */
515 };

```

```
515 |
516 |
517 |
518 |
519 | /*
520 |  * Hammer.JS
521 |  * version 0.6.4
522 |  * author: Eight Media
523 |  * https://github.com/EightMedia/hammer.js
524 |  * Licensed under the MIT license.
525 |  */
526 | function Hammer(element, options, undefined)
527 | {
528 |   /* The sources of Hammer.js were included here (Version 0.6.4) but are not
529 |      * included in this listing. They can be found on the project's github:
530 |      * https://github.com/EightMedia/hammer.js
531 |      */
532 | }
```

A.2 Heatmaps

A.2.1 Visits of mobile version on kurier.at

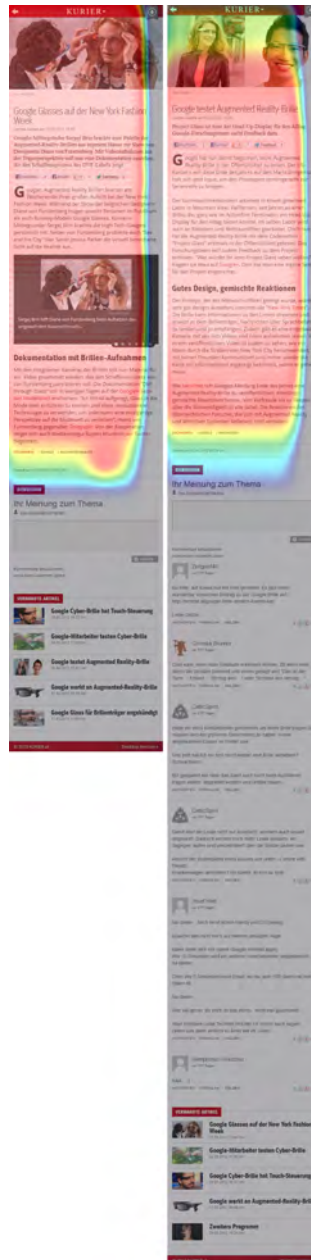


Figure A.1: The two page visits of the mobile version of kurier.at

A.2.2 Unused heatmaps worthwhile showing

As there are innumerable aggregations with different criteria and a large number of page views each with its own heatmap, there are some heatmaps that did not find its way into the thesis but are worth mentioning. It is up to the reader to interpret these heatmaps and draw own conclusions.



Figure A.2: Homepage of the Vienna University of Technology

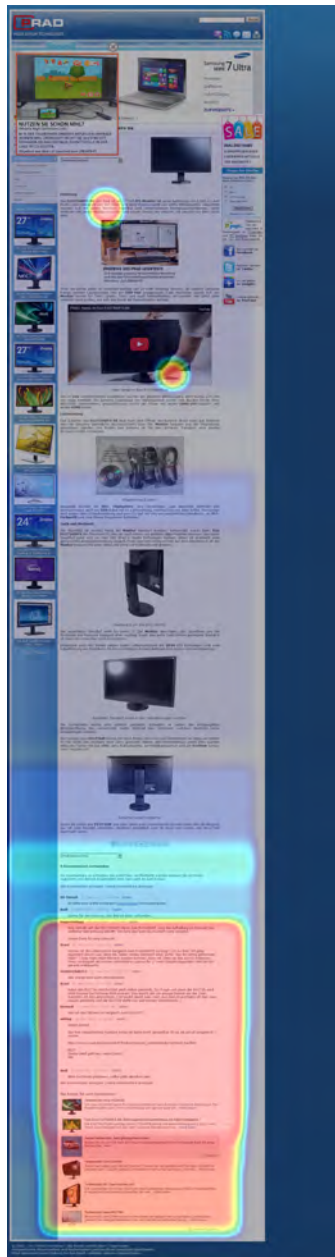


Figure A.3: PRAD Computer Monitor Tests - user comments more interesting than the main article - single visit

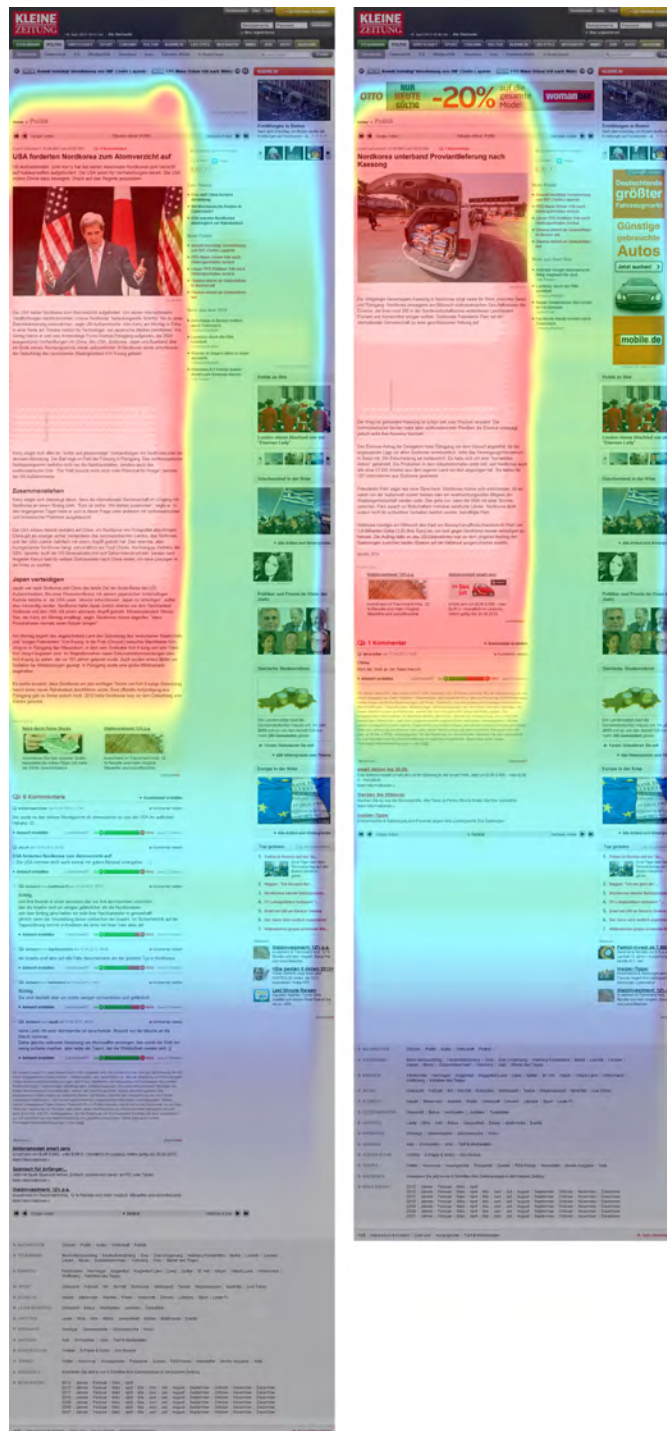


Figure A.4: Articles of the newspaper <http://www.kleinezeitung.at>

golem.de
NEWS FÜR PROFS

HOME THEMEN TICKER SUCHE

PC-MARKT
Windows 8 nützt nicht nur nichts, sondern schadet sogar

Tablets und Smartphones boomen, der PC verliert an Bedeutung. Windows 8 habe dem nicht nur nichts entgegengesetzt, sondern diese Entwicklung noch verstärkt, urteilt das Marktforschungsinstitut IDC.

Der weltweite PC-Markt ist im ersten Quartal 2013 um 13,9 Prozent geschrumpft. Das gab der IT-Marktforscher IDC bekannt. Die Zahl der weltweit ausgelieferten PCs sank auf 76,3 Millionen. IDC hatte nur einen Rückgang um 7,7 Prozent erwartet.

"An diesem Punkt dürfte leider klar sein, dass der Windows-8-Launch nicht nur keinen positiven Impuls für den PC-Markt schaffen konnte, der Start scheint den Markt sogar verkümmert zu haben", sagte Bob O'Donnell, IDC Program Vice President für Clients und Displays. Der Rückgang ist der stärkste, seit IDC im Jahr 1994 begonnen hat, die Marktzahlen zu ermitteln.

Einige Nutzer schätzen zwar die Touch-Funktionen und andere Neuerungen in Microsofts neuem Betriebssystem, sagte O'Donnell - dennoch: "Die radikalen Änderungen an der Benutzerschnittstelle, das Entfernen des populären Startknopfs und die Kosten für Touch-Funktionen in der Hardware machen den PC zu einer weniger attraktiven Alternative zum Tablet und anderen konkurrierenden Geräten. Microsoft wird um einige sehr schwierige Entscheidungen nicht herumkommen, wenn es den PC-Markt beleben will."



Video: Windows 8 - Test (6:55)

Der Markt für PCs in den USA gab um 12,7 Prozent nach. Apple entwickelte sich laut IDC aber besser als der US-Markt. Dennoch gingen die PC-Absatzzahlen auch für Apple zurück, durch die Konkurrenz des eigenen iPads, so IDC.

IDC zählt Tablets nicht in seiner PC-Statistik mit. Der starke Rückgang der Netbooks und der Wechsel zu Tablets und Smartphones hätten die Entwicklung weiter beschleunigt.

Laut Gartner st sanken die weltweiten PC-Verkaufszahlen im ersten Quartal 2013 auf den niedrigsten Stand seit dem zweiten Quartal 2009. Die PC-Verkäufe waren in allen Weltregionen rückläufig. In der EMEA-Region gab es den stärksten Rückgang seit Beginn der Aufzeichnungen durch Gartner. EMEA ist die Abkürzung für einen Wirtschaftsraum, die für Europe (West- und Osteuropa), Middle East (Naher Osten) und Afrika steht.

Laut vorläufigen Ergebnissen Gartner wurden im ersten Quartal 2013 weltweit 78,2 Millionen PCs verkauft. Das entspricht einem Rückgang von 11,2 Prozent im Vergleich zum entsprechenden Vorjahreszeitraum. Die weltweiten PC-Verkäufe sanken erstmals seit dem zweiten Quartal 2009 wieder unter die 80-Millionen-Marke. Während die Verkäufe in allen Regionen rückläufig waren, wurde in der EMEA-Region der stärkste Rückgang festgestellt.

Datum: 11.4.2013, 11:40
Autor: Achim Cayall
Themen: PC, Studie, Tablet, Gartner, Mobil

188 64 77
Facebook Twitter LinkedIn

310 Kommentare

© 1997 - 2013 Golem.de. Alle Rechte vorbehalten.
IMPRESSUM DATENSCHUTZ ANSOFT

Figure A.5: Aggregated article view of <http://www.golem.de>



Figure A.6: Article view of <http://www.reddit.com>

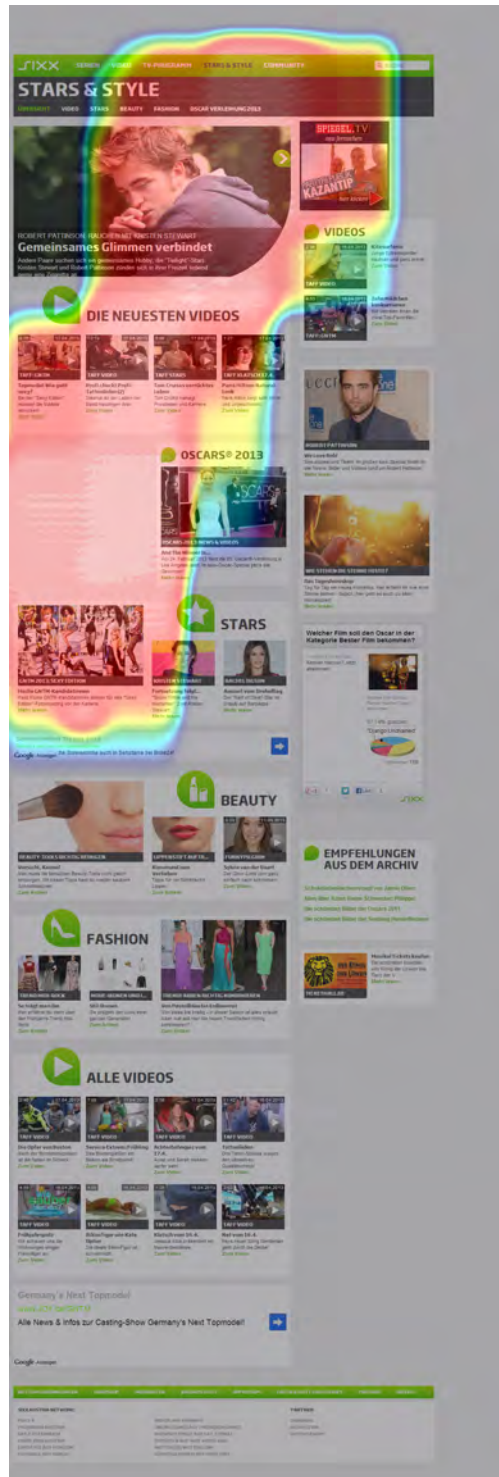


Figure A.7: Page view of a navigation page of <http://www.sixx.at>



Figure A.8: primetime television program <http://www.tele.at>



Figure A.9: Different page views of <http://www.alpbach.org>

Bibliography

- [Abrams and Jonides, 1988] Abrams, R. and Jonides, J. (1988). Programming saccadic eye movements. *Journal of Experimental Psychology: Human Perception and Performance*, 14(3):428.
- [Akkus et al., 2012] Akkus, I. E., Chen, R., Hardt, M., Francis, P., and Gehrke, J. (2012). Non-tracking web analytics. In *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12*, page 687, New York, New York, USA. ACM Press.
- [Albanese et al., 2004] Albanese, M., Picariello, A., Sansone, C., and Sansone, L. (2004). Web personalization based on static information and dynamic user behavior. In *Proceedings of the 6th annual ACM international workshop on Web information and data management - WIDM '04*, page 80, New York, New York, USA. ACM Press.
- [Arroyo et al., 2006] Arroyo, E., Selker, T., and Wei, W. (2006). Usability tool for analysis of web designs using mouse tracks. *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06*, page 484.
- [Atterer and Lorenzi, 2008] Atterer, R. and Lorenzi, P. (2008). A heatmap-based visualization for navigation within large web pages. *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08*, page 407.
- [Atterer et al., 2006] Atterer, R., Wnuk, M., and Schmidt, A. (2006). Knowing the User 's Every Move – User Activity Tracking for Website Usability Evaluation and Implicit Interaction.
- [Badre, 2002] Badre, A. N. (2002). Shaping web usability: interaction design in context. *Ubiquity*, 2002(February).
- [Beauvillain and Beauvillain, 1995] Beauvillain, C. and Beauvillain, P. (1995). Calibration of an eye-movement system for use in reading. *Behavior Research Methods, Instruments, & . . .*, 27(3):331–337.
- [Bigham et al., 2009] Bigham, J. P., Kaminsky, R. S., and Nichols, J. (2009). Mining web interactions to automatically create mash-ups. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*, page 203, New York, New York, USA. ACM Press.

- [Brooks et al., 1981] Brooks, B. a., Impelman, D. M., and Lum, J. T. (1981). Backward and forward masking associated with saccadic eye movement. *Perception & psychophysics*, 30(1):62–70.
- [Burklen et al., 2005] Burklen, S., Marron, P., Fritsch, S., and Rothermel, K. (2005). User Centric Walk: An Integrated Approach for Modeling the Browsing Behavior of Users on the Web. In *38th Annual Simulation Symposium*, pages 149–159. IEEE.
- [Buscher et al., 2010] Buscher, G., Biedert, R., Heinesch, D., and Dengel, A. (2010). Eye tracking analysis of preferred reading regions on the screen. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems - CHI EA '10*, page 3307.
- [Buswell, 1922] Buswell, G. (1922). Fundamental reading habits: a study of their development. *Supplementary educational monographs*, (21).
- [Calero et al., 2005] Calero, C., Ruiz, J., and Piattini, M. (2005). Classifying web metrics using the web quality model. *Online Information Review*, 29(3):227–248.
- [Campbell and Wurtz, 1978] Campbell, F. and Wurtz, R. (1978). Saccadic omission: why we do not see a grey-out during a saccadic eye movement. *Vision research*.
- [Chau and Betke, 2005] Chau, M. and Betke, M. (2005). Real time eye tracking and blink detection with USB cameras. Technical report, Boston University Computer Science Department.
- [Chen et al., 2001] Chen, M., Anderson, J., and Sohn, M. (2001). What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. *CHI'01 extended abstracts on . . .*, pages 281–282.
- [Clickdensity, 2013] Clickdensity (2013). Homepage. <http://www.clickdensity.com/>.
- [Clicktale, 2013a] Clicktale (2013a). Google Analytics integration. http://wiki.clicktale.com/Article/Google_Analytics_integration.
- [Clicktale, 2013b] Clicktale (2013b). Homepage. <http://www.clicktale.com/>.
- [Clicktale, 2013c] Clicktale (2013c). Homepage. <http://www.clicktale.com/>.
- [Cloyd, 2001] Cloyd, M. H. (2001). Designing user-centered Web applications in Web time. *Software, IEEE*, 18(February):62—69.
- [Cox and Silva, 2006] Cox, A. and Silva, M. (2006). The role of mouse movements in interactive search. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, pages 1156—1161.
- [Crazy Egg, 2013] Crazy Egg (2013). Homepage. <http://www.crazyegg.com/>.

- [DeWitt, 2010] DeWitt, A. J. (2010). Examining the Order Effect of Website Navigation Menus With Eye Tracking. *Journal of Usability Studies*, 6(1):39–47.
- [Dr. Berger Consult KG, 2013] Dr. Berger Consult KG (2013). Online Eye Tracking and Brain Scanning. <http://www.mindspotter.at/services.html>.
- [Etracker, 2013] Etracker (2013). Homepage. <http://www.etracker.com>.
- [Fang, 2007] Fang, W. (2007). Using google analytics for improving library website content and design: a case study. 2007:1—17.
- [Ganapathi and Zhang, 2011a] Ganapathi, A. and Zhang, S. (2011a). Web analytics and the art of data summarization. In *Managing Large-scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques on - SLAML '11*, pages 1–9, New York, New York, USA. ACM Press.
- [Ganapathi and Zhang, 2011b] Ganapathi, A. and Zhang, S. (2011b). Web analytics and the art of data summarization. In *Managing Large-scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques on - SLAML '11*, pages 1–9, New York, New York, USA. ACM Press.
- [Garret, 2005] Garret, J. J. (2005). Ajax: A New Approach to Web Applications. <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>.
- [Gartner, 2013] Gartner (2013). Smartphone Marketshares. <http://www.gartner.com/newsroom/id/2482816>.
- [Guo and Agichtein, 2010] Guo, Q. and Agichtein, E. (2010). Towards predicting web searcher gaze position from mouse movements. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems - CHI EA '10*, page 3601.
- [Hammoud, 2008] Hammoud, R. (2008). *Passive eye monitoring: Algorithms, applications and experiments*. Springer.
- [Hennessey et al., 2006] Hennessey, C., Nouredin, B., and Lawrence, P. (2006). A single camera eye-gaze tracking system with free head motion. In *Proceedings of the 2006 symposium on Eye tracking research & applications - ETRA '06*, page 87, New York, New York, USA. ACM Press.
- [Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.
- [Huang, 2011] Huang, J. (2011). On the value of page-level interactions in web search. *Proc. HCIR*.
- [Huang et al., 2011] Huang, J., White, R., and Dumais, S. (2011). No clicks, no problem: using cursor movements to understand and improve search. ... of the 2011 annual conference on
....

- [Huang and White, 2010] Huang, J. and White, R. W. (2010). Parallel browsing behavior on the web. *Proceedings of the 21st ACM conference on Hypertext and hypermedia - HT '10*, page 13.
- [Humanistlaboratoriet, Lund University, 2005] Humanistlaboratoriet, Lund University (2005). Reading Fixations Saccades. http://commons.wikimedia.org/wiki/File:Reading_Fixations_Saccades.jpg.
- [Iga and Shinnishi, 2006] Iga, S. and Shinnishi, M. (2006). SnapShoot: integrating semantic analysis and visualization techniques for web-based note taking system. . . . *Asia-Pacific Symposium on Information Visualisation*
- [Irwin, 1998] Irwin, D. E. (1998). Lexical processing during saccadic eye movements. *Cognitive psychology*, 36(1):1–27.
- [Jacob, 1991] Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems*, 9(2):152–169.
- [Janes, 2011] Janes, C. (2011). Using the Microsoft Kinect API. <http://answers.oreilly.com/topic/2994-using-the-microsoft-kinect-api-part-1-video-stream/>.
- [John M. Henderson, Phillip A. Weeks Jr, 1999] John M. Henderson, Phillip A. Weeks Jr, A. H. (1999). The effects of semantic consistency on eye movements during complex scene viewing. *Journal of Experimental Psychology Human Perception & Performance*, 01/1999:210–228.
- [Joint Industry Committee for Web Standards, 2013] Joint Industry Committee for Web Standards (2013). JICWEBS Reporting Standards. <http://www.abc.org.uk/PageFiles/1489/JICWEBSWebTrafficReportingStandards%20version%202%20March%202013.pdf>.
- [Josephson and Holmes, 2002] Josephson, S. and Holmes, M. (2002). Visual attention to repeated internet images: testing the scanpath theory on the world wide web. *Proceedings of the 2002 symposium on Eye . . .*, 0535(March).
- [Khoo et al., 2008] Khoo, M., Pagano, J., and Washington, A. (2008). Using web metrics to analyze digital libraries. *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 375–384.
- [Krüpl-Sypien et al., 2011a] Krüpl-Sypien, B., Fayzrakhmanov, R. R., Holzinger, W., Panzenböck, M., and Baumgartner, R. (2011a). A versatile model for web page representation, information extraction and content re-packaging. In *Proceedings of the 11th ACM symposium on Document engineering - DocEng '11*, page 129, New York, New York, USA. ACM Press.
- [Krüpl-Sypien et al., 2011b] Krüpl-Sypien, B., Fayzrakhmanov, R. R., Holzinger, W., Panzenböck, M., and Baumgartner, R. (2011b). A versatile model for web page representation,

- information extraction and content re-packaging. In *Proceedings of the 11th ACM symposium on Document engineering - DocEng '11*, page 129, New York, New York, USA. ACM Press.
- [Kumar et al., 2012] Kumar, L., Singh, H., and Kaur, R. (2012). Web analytics and metrics: a survey. . . . *of the International Conference on Advances . . .*, page 966.
- [Kumar, 2006] Kumar, M. (2006). Reducing the cost of eye tracking systems. Technical report, Stanford University, Stanford.
- [Labsmedia, 2013] Labsmedia (2013). Clickheat Homepage. <http://www.labsmedia.com/clickheat>.
- [Lagun and Agichtein, 2011] Lagun, D. and Agichtein, E. (2011). ViewSer : Enabling Large-Scale Remote User Studies of Web Search Examination and Interaction Categories and Subject Descriptors. Number Section 3, pages 365–374. ACM.
- [Li et al., 2006] Li, D., Babcock, J., and Parkhurst, D. (2006). openEyes: a low-cost head-mounted eye-tracking solution. . . . *of the 2006 symposium on Eye tracking . . .*, 1(March):27–29.
- [Lucs-kho and Yarus, 1967] Lucs-kho and Yarus, A. (1967). Yarus The Visitor. http://commons.wikimedia.org/wiki/File:Yarus_The_Visitor.jpg.
- [M-pathy, 2013] M-pathy (2013). Homepage. <http://www.m-pathy.com>.
- [Marketing Department, 2013] Marketing Department, U. o. V. (2013). Triconsult Preis für innovative Marktforschung – TIM. <http://marketing.univie.ac.at/home/tim-2011-geht-an-dr-berger-fuer-mindspoterr/>.
- [Meyer and Kornblum, 1989] Meyer, D. and Kornblum, S. (1989). Speed and accuracy of saccadic eye movements: Characteristics of impulse variability in the oculomotor system. *Journal of Experimental . . .*, 15(3):529–43.
- [Mouseflow, 2013] Mouseflow (2013). Homepage. <http://www.mouseflow.com>.
- [Mueller and Lockerd, 2001] Mueller, F. and Lockerd, A. (2001). Cheese: tracking mouse movement activity on websites, a tool for user modeling. *CHI'01 extended abstracts on Human factors in . . .*, pages 279–280.
- [Navalpakkam and Churchill, 2012] Navalpakkam, V. and Churchill, E. (2012). Mouse tracking: measuring and predicting users' experience of web-based content. pages 2963–2963–2972–2972.
- [Navalpakkam et al., 2013] Navalpakkam, V., Jentzsch, L., and Sayres, R. (2013). Measurement and Modeling of Eye-mouse Behavior in the Presence of Nonlinear Page Layouts. *sravi.org*.

- [Network Working Group and Crockford, 2006] Network Working Group and Crockford, D. (2006). JavaScript Object Notation (JSON). <http://www.ietf.org/rfc/rfc4627.txt>.
- [Nielsen, 2006] Nielsen, J. (2006). *Ensuring Web Usability: Understanding What Users Want*. New Riders Publishing.
- [Nielsen and Loranger, 2006] Nielsen, J. and Loranger, H. (2006). *Prioritizing Web Usability*. Pearson Education.
- [Ohno et al., 2003] Ohno, T., Mukawa, N., and Kawato, S. (2003). Just blink your eyes: A head-free gaze tracking system. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 950—957. ACM.
- [Pan et al., 2004] Pan, B., Hembrooke, H. A., Gay, G. K., Granka, L. A., Feusner, M. K., and Newman, J. K. (2004). The determinants of web page viewing behavior. In *Proceedings of the Eye tracking research & applications symposium on Eye tracking research & applications - ETRA'2004*, pages 147–154, New York, New York, USA. ACM Press.
- [Plaza, 2011] Plaza, B. (2011). Google Analytics for measuring website performance. *Tourism Management*, 32(3):477–481.
- [Posner, 1980] Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32(1):3–25.
- [Punera and Merugu, 2010] Punera, K. and Merugu, S. (2010). The anatomy of a click. In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, page 989, New York, New York, USA. ACM Press.
- [Rakoczi,] Rakoczi, G. Design Optimierung der Benutzeroberfläche einer zentral eingesetzten Moodle Instanz (28 . 000 + User) anhand von Eye Tracking. pages 1–11.
- [Rakoczi, 2010] Rakoczi, G. (2010). Cast your eyes on Moodle: An eye tracking study investigating learning with Moodle. . . . *of the 4th International Conference Moodle. si*, (May):203–213.
- [Rayner, 1998] Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372–422.
- [Rodden and Fu, 2007] Rodden, K. and Fu, X. (2007). Exploring how mouse movements relate to eye movements on web search results pages. . . . *2007 Workshop on Web Information Seeking and . . .*, (July):29–32.
- [Rodden et al., 2008] Rodden, K., Fu, X., Aula, A., and Spiro, I. (2008). Eye-mouse coordination patterns on web search results pages. *CHI'08 Extended Abstracts on Human . . .*, pages 2997–3002.
- [SMI - Senso Motoric Instruments, 2013a] SMI - Senso Motoric Instruments (2013a). IVIEW X HED Product Page. <http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/iview-x-hed.html>.

- [SMI - Senso Motoric Instruments, 2013b] SMI - Senso Motoric Instruments (2013b). Red 500 Product Page. <http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/red-red250-red-500.html>.
- [Sony, 2013] Sony (2013). Playstation Eye Brings Next-Generation Communication to Playstation 3. <http://us.playstation.com/corporate/about/press-release/396.html>.
- [Tobii, 2013] Tobii (2013). Tobii T/X series Eye Trackers Product Description. http://www.tobii.com/Global/Analysis/Downloads/Product_Descriptions/Tobii_TX_Product_description.pdf.
- [Uttal and Smith, 1968] Uttal, W. and Smith, P. (1968). Recognition of alphabetic characters during voluntary eye movements. *Perception & Psychophysics*, pages 257–264.
- [W3C, 2005] W3C (2005). Document Object Model (DOM). <http://www.w3.org/DOM/>.
- [W3Schools, 2013] W3Schools (2013). Historic display resolutions. http://www.w3schools.com/browsers/browsers_display.asp.
- [W3Techs - Web Technology Surveys, 2013] W3Techs - Web Technology Surveys (2013). Usage statistics and market share of Google Analytics for websites. <http://w3techs.com/technologies/details/ta-googleanalytics/all/all>.
- [Weischedel and Huizingh, 2006] Weischedel, B. and Huizingh, E. (2006). Website optimization with web metrics: a case study. ... *of the 8th international conference on ...*, pages 463–470.
- [Yan and Guo, 2010] Yan, P. and Guo, J. (2010). The research of Web usability design. *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, 4:480–483.
- [Yarbus and Riggs, 1967] Yarbus, A. and Riggs, L. (1967). *Eye movements and vision*. Plenum Press, New York.

List of Figures

1.1	Design science research cycles [Hevner et al., 2004]	6
1.2	Design evaluation methods [Hevner et al., 2004]	6
2.1	Heatmap: www.clicktale.com, Made by Clicktales Mouse Tracking Solution [Clicktale, 2013b]	11
2.2	Heatmap: www.gmx.at, Aggregated data from 11 users created with <i>Mobile Viewport Tracking</i>	12
3.1	Illustration of mobile devices viewport in relation to the full size web page	17
3.2	The traditional model for web applications (left) compared to the AJAX model (right). [Garret, 2005]	20
3.3	Data Model - part 1	21
3.4	Data Model - part 2	22
3.5	Architecture Overview	23
3.6	Comparing the Colourful Drawer (left side) with the Raster Drawer (right side)	31
3.7	Raster Drawer in detail: the logarithmized weight and the opacity	31
3.8	The colour spectrum used to map the value to the resulting colour	32
3.9	The circle brush used to smooth data	32
3.10	The colour scale is mapped from a transparent black and white image with different alpha-values (right side) to the colours of the colour-spectrum (left side)	32
3.11	Colourful Drawer in detail	33
3.12	IFrame embedding trick using HTML5	34
3.13	Registration page	35
4.1	Unobserved User Actions [Punera and Merugu, 2010]	37
4.2	The Web Quality Model Cube [Calero et al., 2005]	40
4.3	The Web Quality Model Metrics Chart [Calero et al., 2005]	41
4.4	Google Analytics Dashboard	42
4.5	Prototype of a fixed Eye Tracking System [Ohno et al., 2003]	43
4.6	Commercial fixed Eye Tracking System - SMI Red 500 [SMI - Senso Motoric Instruments, 2013b]	43
4.7	Prototype of a head-mounted low-cost Eye Tracking System [Li et al., 2006]	44
4.8	Commercial head mounted Eye Tracking System - SMI IVIEW X HED [SMI - Senso Motoric Instruments, 2013a]	44

4.9	Fixations and Saccades taken from a study of speed reading made by Humanistlaboratoriet, Lund University, in 2005 using a SMI iView X 240 Hz [Humanistlaboratoriet, Lund University, 2005]	45
4.10	Yarbus the visitor: Influence of the given task on eye movements [Yarbus and Riggs, 1967] [Lucs-kho and Yarbus, 1967]	47
4.11	Increased distraction leads to uselessness of mouse tracking [Navalpakkam and Churchill, 2012]	51
4.12	The Clickheat Interface [Labsmedia, 2013]	52
4.13	The Crazy Egg Confetti View [Crazy Egg, 2013]	53
4.14	The Mouseflow heatmap View [Mouseflow, 2013]	53
4.15	The m-pathy heatmap view [M-pathy, 2013]	54
4.16	The Clickdensity heatmap view [Clickdensity, 2013]	54
4.17	The Clicktale In-Page Web Analytics View [Clicktale, 2013c]	55
4.18	The etracker Visitor Motion Map [Etracker, 2013]	55
4.19	The scroll-heatmap navigation aid [Atterer and Lorenzi, 2008]	56
4.20	The ViewSer interface showing “a blurred search engine result page (SERP) for the query “toilet”, with the viewport revealing the first result” [Lagun and Agichtein, 2011]	57
4.21	A heatmap showing the eye-tracking heatmap in background and it’s one-dimensional projection (a) compared to the heatmap generated by ViewSer (b) [Lagun and Agichtein, 2011]	57
4.22	The Spotlight Viewer Interface [Dr. Berger Consult KG, 2013]	58
5.1	Configuration Tutorial for Android 4.X	60
5.2	Configuration Tutorial for iOS	61
6.1	The Mobile Heatmaps Analysis Portal in an overview	68
6.2	The Update Panel where new Information is pushed to the users to see what is currently happening	68
6.3	Page views shown in a list	69
6.4	A pie chart showing the share of Action Types	69
6.5	A stacked bar chart showing the share of Action Types for each user, only Scroll Up and Scroll Down are shown by default, others can be added/removed by selecting them in the chart	70
6.6	The average page view duration for each user session - Min, Max and Mean are also calculated and shown at the end of the table	70
6.7	Selection of page views (one visit to a specific URL) can be selected and a combined heatmap can be generated	71
7.1	derstandard.at (aggregated: LEFT: 3 pageviews, AEXIN data, given task, RIGHT: 10 pageviews, NON-EXIN data, no given task) cropped vertically after 40 percent	77

7.2	derstandard.at/... (various article urls) (aggregated: LEFT: 5 pageviews, AEXIN data, given task, RIGHT: 52 pageviews, non-AEXIN data, no given task) comparing reading behaviour with given task to reading behavior without a given task. Please note that the length of the pages differs between different articles.	78
7.3	Example non-aggregated heatmap of the mobile version of derStandard. Please note that this mobile version is not corresponding well with the background image as the layout of the page is adaptive to the screen width and the screenshot is taken with a screen width that is (in this example) slightly higher than the screen width of the mobile device.	79
7.4	Aggregated heatmaps of different news articles of derStandard (left), diePresse (center) and orf.at (right) from AEXIN	81
7.5	mytimes.com (left), thomsonreuters.com (center left), heute.at (center), bbc.co.uk (center right), kurier.at (right)	82
7.6	Desktop version (left) and mobile version (right) of the news website kurier.at - aggregated over 2-3 page views	83
7.7	Search on news websites - desktop vs. mobile websites - derStandard.at (left) and kurier.at (right) aggregated over 3 - 7 page views	84
7.8	Visits on four different http://www.amazon.de product pages (four examples shown out of about 25 recorded)	85
7.9	Visits of the current weather on http://www.wetter.at , http://www.wetter.com and http://www.zamg.ac.at	86
7.10	Device orientation (overall)	86
7.11	Heatmaps of four different news website start pages compared (left to right: derStandard.at, sueddeutsche.de, kleinezeitung.at, edition.cnn.com)	87
7.12	Web.de landing page segmented	89
7.13	Page views on derStandard.at - one were the image area was ignored (left) and one were the image area was viewed (right)	90
7.14	Booking.com landing page (left) and corresponding result page (right)	91
7.15	Heatmap of the desktop version of derStandard showing a v-form zooming pattern	92
7.16	Action types (overall)	93
A.1	The two page visits of the mobile version of kurier.at	122
A.2	Homepage of the Vienna University of Technology	123
A.3	PRAD Computer Monitor Tests - user comments more interesting than the main article - single visit	124
A.4	Articles of the newspaper http://www.kleinezeitung.at	125
A.5	Aggregated article view of http://www.golem.de	126
A.6	Article view of http://www.reddit.com	127
A.7	Page view of a navigation page of http://www.sixx.at	128
A.8	primetime television program http://www.tele.at	129
A.9	Different page views of http://www.alpbach.org	130

Abbreviations

KPI Key Performance Indicator
MVT Mobile Viewport Tracking