FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

Web Interaction Archiving and Replay using Advanced User Behaviour Analysis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

Sebastian Morgenbesser
Matrikelnummer 0826197

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.Prof. Mag.rer.nat. Dr.techn. Reinhard Pichler
Mitwirkung: Projektass. Dipl.Ing. Christoph Herzog

Wien, 29.07.2013

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Web Interaction Archiving and Replay 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

Sebastian Morgenbesser
Registration Number 0826197

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ.Prof. Mag.rer.nat. Dr.techn. Reinhard Pichler
Assistance: Projektass. Dipl.Ing. Christoph Herzog

Vienna, 29.07.2013

(Signature of Author) (Signature of Advisor)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

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

At first I want to thank my colleagues and project team members Maximilian Aster and David Neukam for their commitment to the project and the interesting work together with them.

I also want to thank Robert Baumgartner, Reinhard Pichler and their whole team at the DBAI institute of the Vienna University of Technology, namely Christoph Herzog, Bernhard Kruepl-Sypien and Ruslan Fayzrakhmanov who did support us during the whole duration of the project and the writing of this thesis.

Last but not least, I want to thank my lovely girlfriend Christina Schenk who did support me in any possible kind and helped me to succeed with this thesis.

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. Due to the increasing domination of smartphones in the World Wide Web these days, it is very important to analyse and optimise web pages according to those devices. This thesis introduces a new technology called *Mobile Viewport Tracking*, that is capable of tracking all kinds of actions users can perform on mobile devices and also offers functions for analysis and graphical presentation of the results. The data that is gathered by *Mobile Viewport Tracking* is, on the one hand used for optimizing web pages, but on the other hand for designing and introducing a completely new approach for web archiving based on web interactions, which is the main focus of this thesis. This new type of web archive does not focus on the archiving of pure website content, as most common web archive solutions do, instead it targets at the user's experience by archiving and replaying user interactions on web pages. A conceptual design and architecture is developed, that describes in detail how such a user-centric web archive can be built and which technical components are needed. Apart from the web archive itself, also an architecture and concept for possible contributors to the archive is introduced, as the approach relies on data donations from external parties. Another part of this thesis shows how the idea of archiving and replaying user interactions could be used in order to improve the effectiveness of web sites, and help to adapt web pages to different user groups needs.

Kurzfassung

Aktuelle Verfahren um Nutzerverhalten aufzuzeichnen, mit dem Ziel Webseiten zu optimieren konzentrieren sich hauptsächlich auf Desktop Benutzer und beachten dabei die spezielle Gruppe der mobilen Benutzer fast überhaupt nicht. Wegen der steigenden Zahl an Smartphones im World Wide Web ist es von großer Wichtigkeit, Webseiten und deren Benutzer zu analysieren und anhand der Ergebnisse diese auf die Bedürfnisse der Benutzer hin zu optimieren. Diese Arbeit stellt eine neue Technologie mit dem Namen *Mobile Viewport Tracking* vor, die es ermöglicht, diverse Aktionen die von Benutzern von mobilen Endgeräten getätigt werden, aufzuzeichnen, diese zu analisieren und in einer grafischen Form wiederzugeben. Die Daten die durch *Mobile Viewport Tracking* verfügbar sind, können einerseits verwendet werden um Webseiten zu optimieren, auf der anderen Seite wird in dieser Arbeit basierend auf diesen Daten ein komplett neuer Ansatz für Web Archiving präsentiert. Dieser Ansatz verzichtet auf das reine Archivieren von Webseiteninhalt, stattdessen steht der Benutzer und seine Interaktionen auf der Webseite im Vordergrund. Das Design eines Konzeptes und einer Architektur für ein solches benutzerorientiertes Web Archiv ist wesentlicher Bestandteil dieser Arbeit. Außerdem wird auch ein Konzept für mögliche Datenlieferanten vorgestellt da dieser Ansatz abhängig von externen Institutionen ist, die das Archiv mit aktuellen Daten versorgen. Ein weiterer Teil dieser Arbeit beschäftigt sich mit der Optimierung von Webseiten , ebenfalls auf Basis einer benutzerorientierten Analyse und der Wiedergabe von aufgezeichneten Benutzerinteraktionen im Internet.

Contents

CHAPTER 1

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

The main problem this project and master thesis is trying to solve, is the lack of archiving user experience in the World Wide Web. The Web is growing and changing very fast and as it is a very important part of everyday life nowadays, it should be historicised in a proper way because it belongs to our history. As already stated, common approaches just try to archive content of web pages, but the actual interactions users perform on the web are lost. A basic use case would be to look up what pages did people use e.g. 10 years ago and what they did do on these pages. What inputs did they use? How long did they stay on a web page? Which parts of different pages did they prefer?

The second goal is to prevent bad website layout and uninteresting content. This thesis will provide data for optimisation of web pages by possible analyses of those archived user experiences. It will be a good opportunity for website operators to improve websites according to the results of such an analysis. During the master thesis, the following research questions will be answered:

- How can the web be archived in a meaningful way, so that people in the future would be able to get a feeling of how the web looked like, and more specifically how it was used in the past?
- How could web pages be improved to reach a level of maximum usability and satisfy all needs of different user groups?
- How can meaningful data be retrieved that is needed to answer the previous two questions?
- What architecture and which technical requirements are needed to fulfil these tasks?

1.3 Aim of the work

The main focus of this thesis is to archive user interactions on web pages in a way that will be used to historicise the web. A common approach to keep a history of the web is to store entire web pages and archive the content at regular intervals. In the future one would be able to take a look at those old web pages and get a feeling of how the web actually looked like in the past. One problem with this approach is that it is not easy to store runnable copies of web pages. A common web page does not only consist of static content anymore, more likely it is highly dynamic and relies on various databases and logic components that render the actual content based on context. Another 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 memorise how people actually perceived web pages at different times, rather than just storing the plain content itself. Another point is the archiving of trends. It would be nice to historicise how people acted or behaved using the web. In these days it would be strongly affected by the usage of smartphones, which is also the main target group we want to address with our project.

The idea is now to create a mechanism that enables the archiving of 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. The different actions on specific websites are combined in a sequence and interpreted as interactions, and so it is possible to reconstruct whole surf sessions of users. There are several ways to visualise those sessions. The simplest form will be 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 rendering a video that shows the user session as it actually took place in the past in a visual way. Such a video will not only describe the surf session quite precise, it will also give the viewer a good impression of how the web was perceived at that time. In order to limit the amount of space and computing power needed to realise such a system, the user sessions that will be rendered as videos could be selected by different attributes. Therefore, it has to be possible to compare and categorise those user sessions.

The archived user sessions could either be used to create a solution for historicising the web, as already discussed, or for analysis and optimisation of websites. The visualisations of archived user sessions would be a great possibility for website operators to study how their web page is used and how to improve it accordingly. Of course it would be be hard to automatically analyse those sessions, but if someone regularly watched selected videos out of the recorded pool, this person would get a good impression of how users perceive the web page and how it could be optimised.

A main problem in the whole project will be to get meaningful data. For a serious approach of archiving the web, or at least a part of the web, a lot of recorded user sessions will be needed. Existing approaches that archive entire web pages use crawlers to search the web, very similar to search engines. A possibility to get roughly as much data as is needed in order to have a good representation of the history of the web would be to use the crowd as a "human crawler". An example would be a Wireless Internet Hotspot that is placed in a public space and enables the tracking of all users. A similar scenario was implemented at the Vienna University of Technology during an exercise. More details can be found in chapter 7.

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.

In chapter 7 the small introduction section 7.3.1 was written by David Neukam and the description of the task in section 7.3.2 was written in collaboration. This can also be found in the table of authors on the next page.

Table 1.1: Author catalog

Chapter/Section	**Author**
1 Introduction	
1.1 Motivation	Collaborative
1.2 Problem statement	Sebastian Morgenbesser
1.3 Aim of the work	Sebastian Morgenbesser
1.4 Authorship	Sebastian Morgenbesser
1.5 Methodology	Maximilian Aster
1.6 Related project work	Collaborative
1.7 Relevance for Business Informatics	Sebastian Morgenbesser
1.8 Structure of the work	Sebastian Morgenbesser
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	Sebastian Morgenbesser
5 Web Interaction Archiving	Sebastian Morgenbesser
6 Website Optimisation	Sebastian Morgenbesser
7 Evaluation	
7.1 Introduction	Sebastian Morgenbesser
7.2 Evaluation	Sebastian Morgenbesser
7.3 Studies	
7.3.1 Introduction	David Neukam
7.3.2 Task description	Collaborative
7.3.3 Results	Sebastian Morgenbesser
8 Summary and future work	Sebastian Morgenbesser
9 Conclusion	Sebastian Morgenbesser

1.5 Methodology

The methodology describes how a certain type of problem is approached. As pointed out by [16] 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. [16]

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. [16]

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 cycles, i.e. the requirements and the existing theoretical and practical knowledge for the design

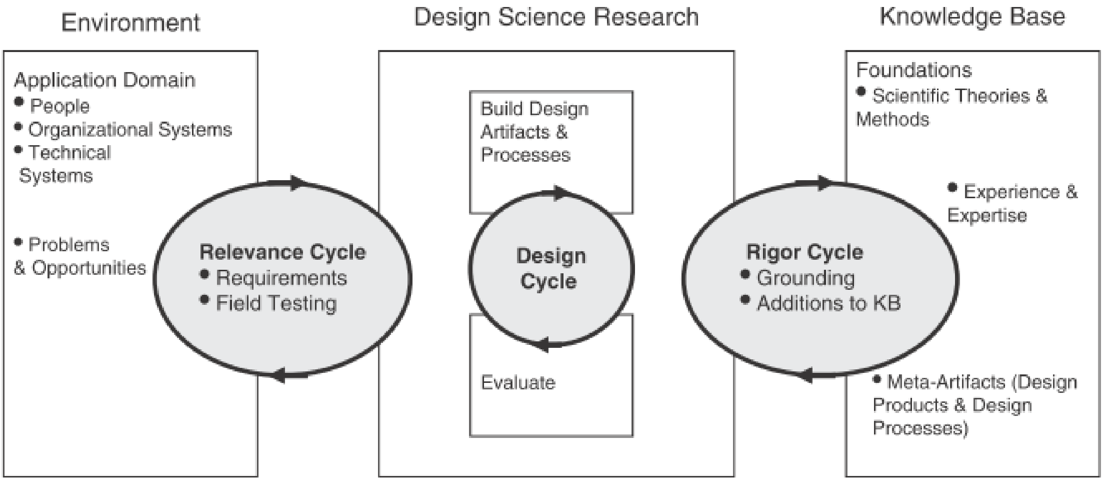


Figure 1.1: Design science research cycles [16]

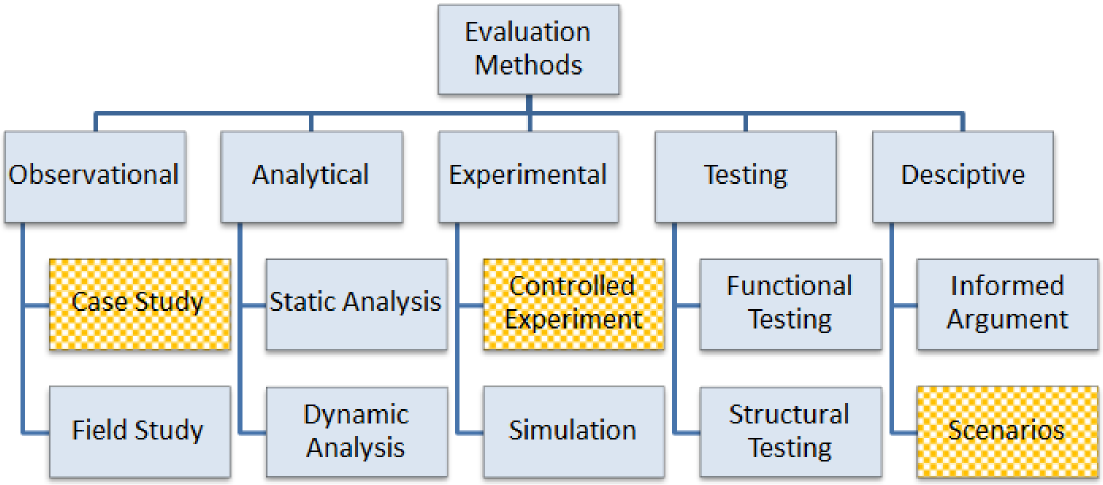


Figure 1.2: Design evaluation methods [16]

and the evaluation. The solution part of the thesis and the evaluation are documented in separate chapters.

In [16] 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

The World Wide Web is dominated by Smartphones these days. Therefore, it is very important to analyse and optimise web pages according to those devices to be capable of competing. This is necessary for all types of businesses that have an online representation, especially for E-Commerce Businesses, because Smartphones are representing a very valuable and rapidly growing market segment, which cannot be ignored from a business perspective.

As Business Informatics is intended “to build an interface between humans, organizations, and information technology” [2] , also the Archiving aspect has a strong context to it. The thesis intends to show an approach of historicising the web in a way that it can be easily used by private users, as well as by businesses in order to learn from the past and build up-to-date web applications that are fitting the needs of their potential customers.

1.8 Structure of the work

Chapter 2 and 3 introduce the common part. Chapter 2 explains *Advanced User Behaviour Analysis* which is the basic idea of analysing user behaviour using sophisticated approaches and technical enhancements. In Chapter 3 *Mobile Viewport Tracking* is introduced, which is the technical solution for tracking and analysing users, that was built up during this project. Chapter 4 is dealing with research regarding the topics Digital Preservation, Web Archiving and Website Optimisation. The basics are clarified there. After that, chapter 5 explains a new approach for Web Archiving using principles from the *Mobile Viewport Tracking* system but extending it by additional components in order to implement a user-centric solution. Chapter 6 focuses on how the approaches mentioned in chapter 5 could be used to optimise web pages regarding

[1] `http://www.dbai.tuwien.ac.at/proj/tamcrow/`

[2] `http://www.informatik.tuwien.ac.at/lehre/studien/master/business-informatics/`

different user groups needs. Chapter 7 describes evaluations that have been made during this thesis. Chapter 8 contains a summary and the final conclusion.

CHAPTER 2

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 [1] of a user on web pages. Many studies using eye tracking technologies were done testing theories like the scanpath theory [20] on websites, identifying determinants and metrics of web page viewing behaviour [31], general user behaviour pattern identification [11] and specific optimisation of web pages [34] targeting the dynamic user behaviour. Static information like server logs is used by [21] to analyse web page viewing behaviour.

As opposed to common web analytics solutions such as Google Analytics[1] 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 [**?**]. 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 DOM[2] 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 Archive [3]. 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.

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 [31]" which is heavily influenced by the stimulus [20]. These fixation points - defined as "relatively motionless gaze" [31] 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 [7]. 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" [7]. According to [7], the average correlation is about 58% in sections.

Mouse Tracking is offered by multiple vendors. Some solutions are free[6], others [7] [8] cost from a few Euros a month [9] up to multiple thousand Euro per Website Optimisation project[10].

[2] Document Object Model `http://www.w3.org/DOM`

[3] `http://www.archive.org`

[4] `http://www.tobii.com/`

[5] `http://www.smivision.com/`

[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/`

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” [6] 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*.

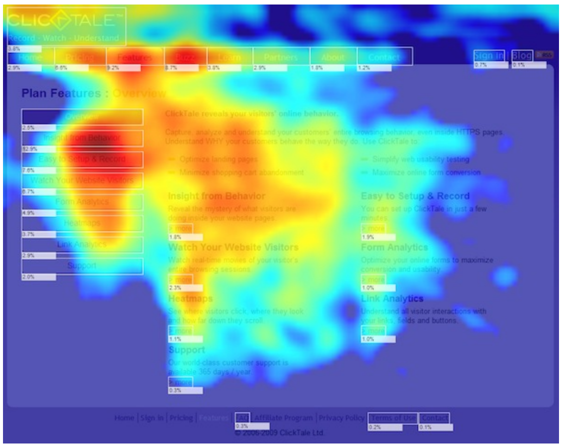


Figure 2.1: Heatmap: www.clicktale.com, Made by Clicktales Mouse Tracking Solution [8]

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 [8]. 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* where the whole screen of a mobile device is tracked, but the correlation of attention and mouse-movements is not very high [6].

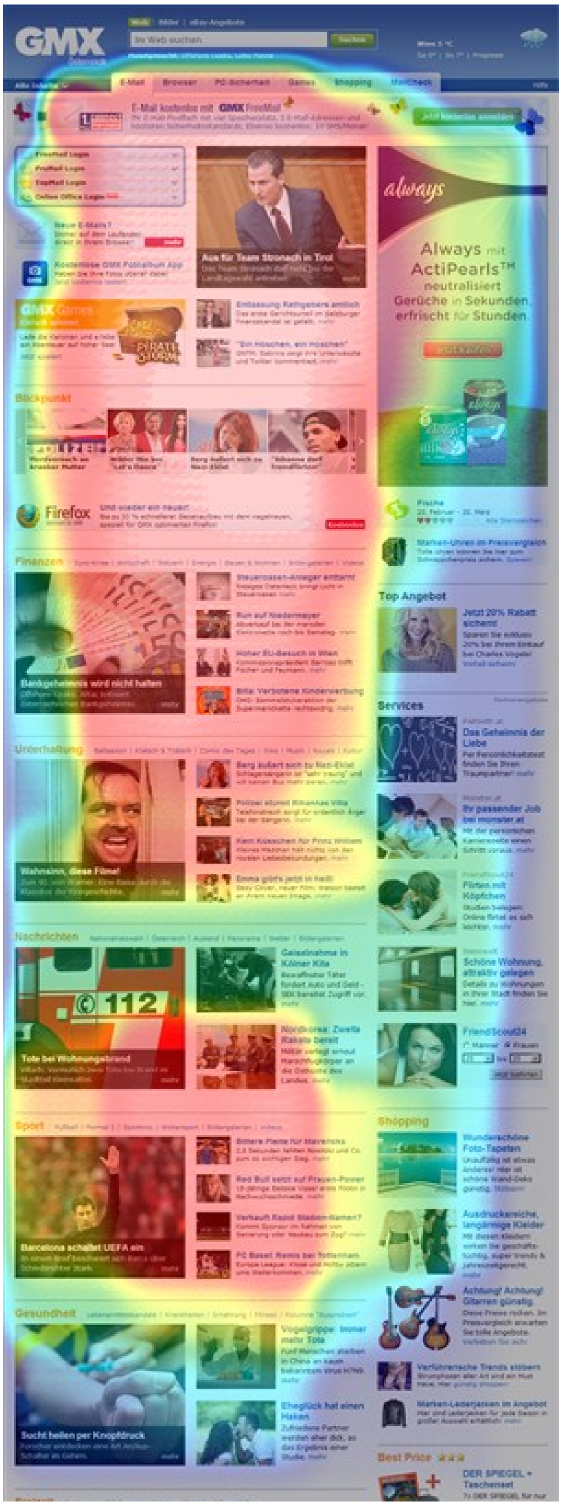


Figure 2.2: Heatmap: www.gmx.at, Aggregated data from 11 users created with *Mobile Viewport Tracking*

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.

CHAPTER 3

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 [13]. 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 HTTP[1] response of each request returning HTML[2] 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

[1] Hypertext Transfer Protocol - `http://www.w3.org/Protocols/`

[2] Hyper Text Markup Language - `http://www.w3.org/standards/`

The following image illustrates the viewport on a mobile device:

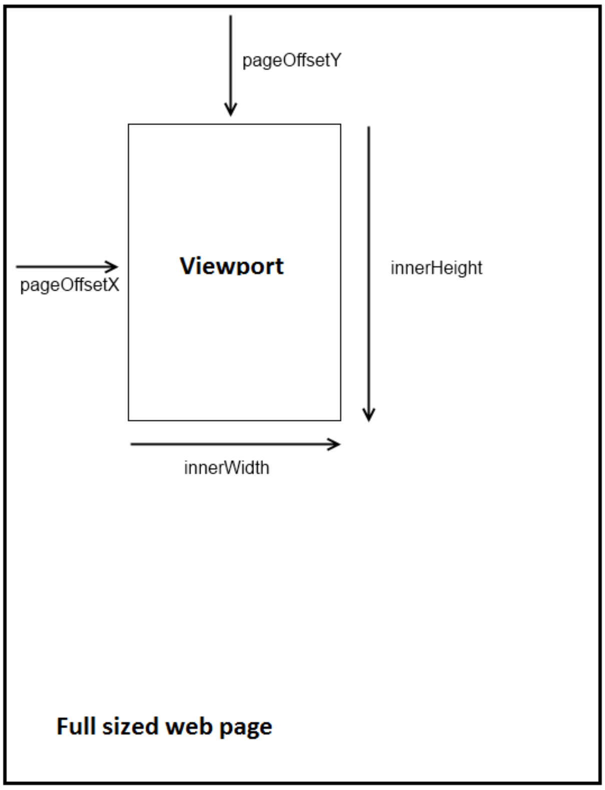


Figure 3.1: Illustration of mobile devices viewport in relation to the full size web page

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 URI[3] which provides the access to the service. The communication is very likely performed over the HTTP protocol and the content is transferred in XML[4] 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 Protocol [5] 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

[3] Unified Resource Identifier - `http://www.w3.org/Addressing`

[4] Extensible Markup Language - `http://www.w3.org/XML`

[5] `http://www.w3.org/TR/soap`

called WSDL [6] 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” [29]. 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" [50].

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 JavaScript[7] , 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 [14].

[7] `http://www.ecma-international.org/publications/standards/Ecma-262.htm`

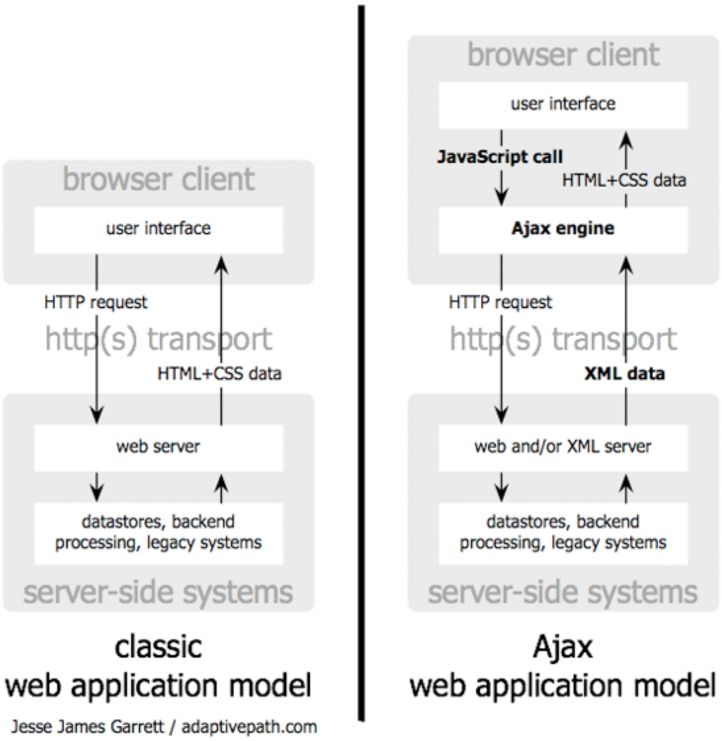


Figure 3.2: The traditional model for web applications (left) compared to the AJAX model (right). [14]

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 than 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

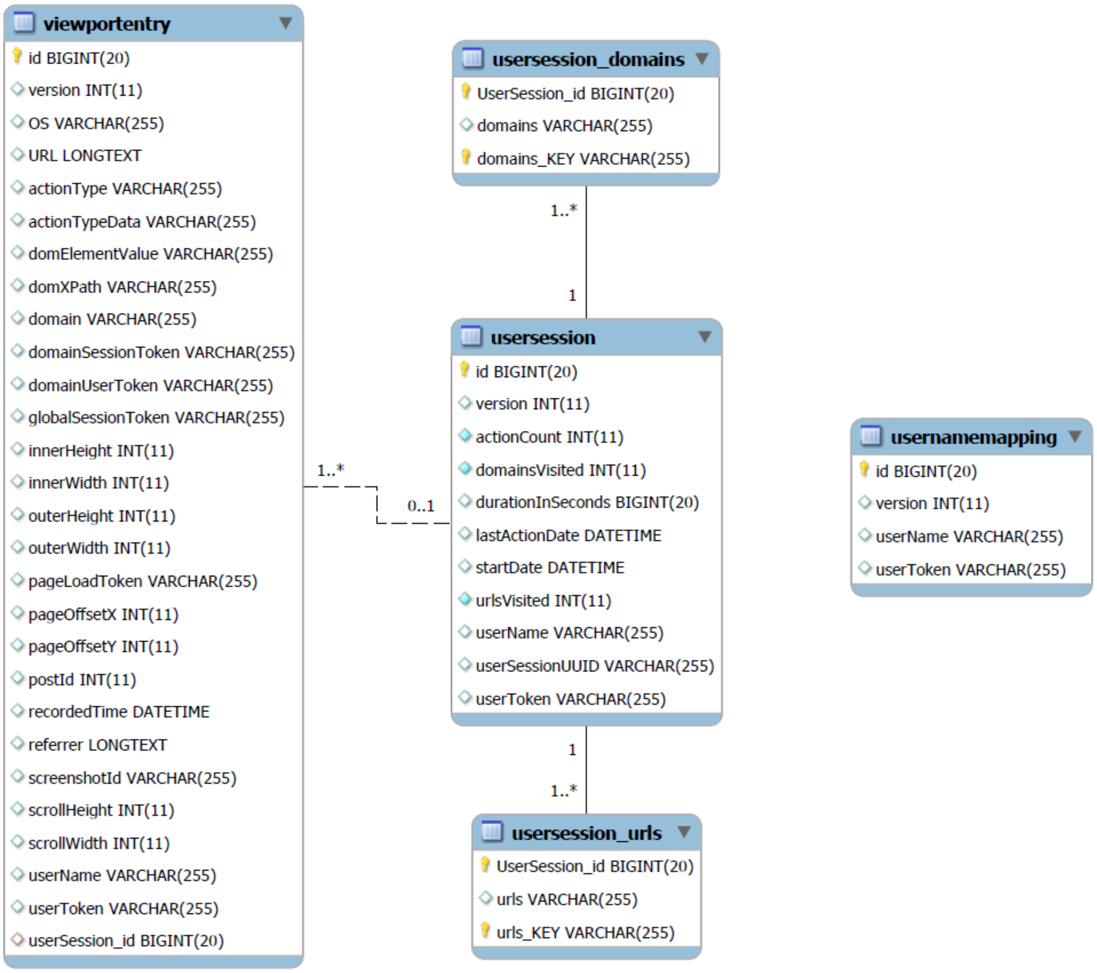


Figure 3.3: Data Model - part 1

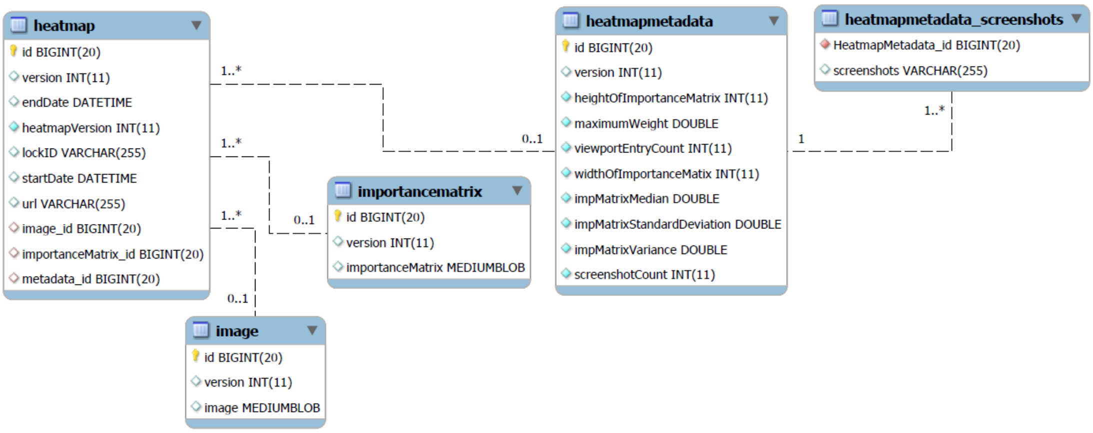


Figure 3.4: Data Model - part 2

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 screenhot 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

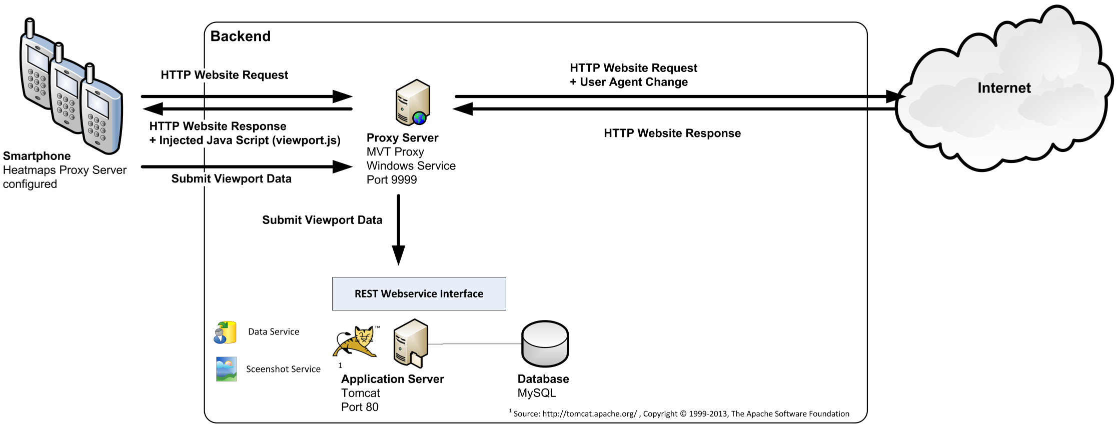


Figure 3.5: Architecture Overview

3.4.2 Client-side script

As can be seen in the architecture diagram the client side is 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 Proxy [9]. The configuration of the proxy on the supported mobile platforms can be found in the appendix section A.2.1.

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 Java[10] and published as web application on a Tomcat[11] 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 Driver[12]. 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 hibernate[13] 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.js[14], 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 JSONP[15] 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

[15] JSON 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 [20] and [31], 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 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 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 where created that both have their benefits and drawbacks. While the Raster Drawer is closer to the implementation and more technical, the Colourful Drawer has it's focus on a pleasant graphical representation. The sources of the concrete implementation of both drawers are available in the Appendix A.1.1.

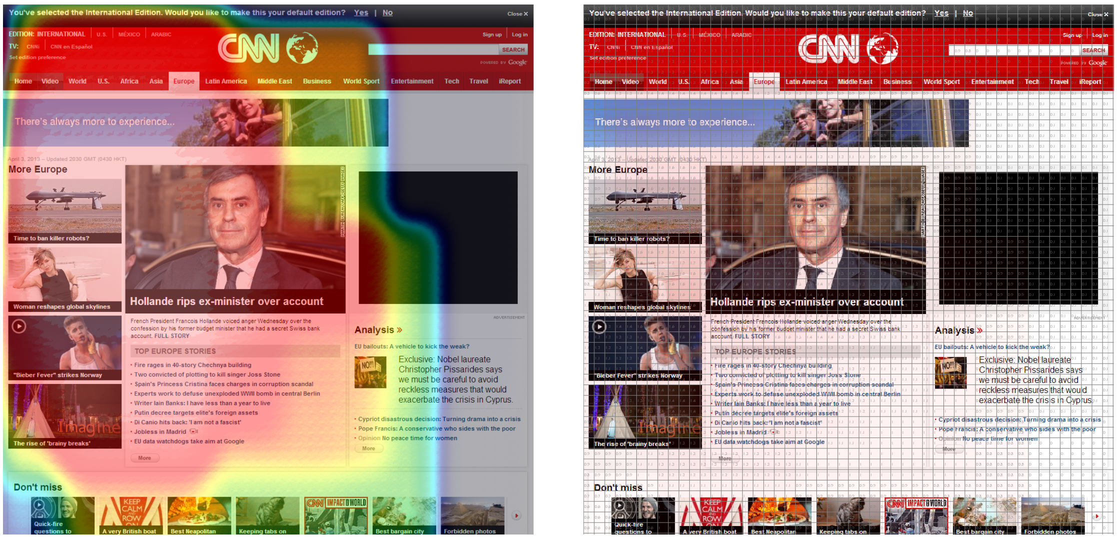

CNN
There's always more to experience...
More Europe
Hollande rips ex-minister over account
Analysis
Don't miss

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

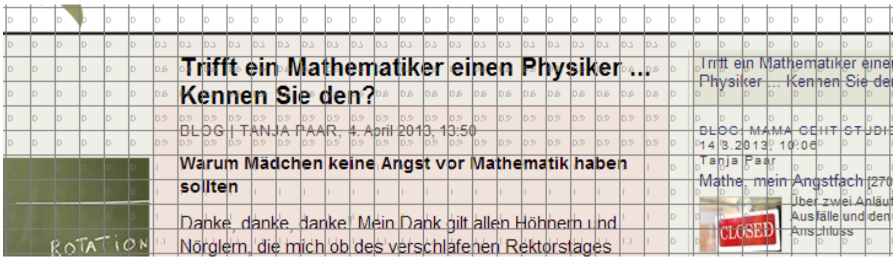
Raster Drawer

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

$$weight = 10^{cellValue} \tag{3.1}$$

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.



Trifft ein Mathematiker einen Physiker ... Kennen Sie den?

BLOG | TANJA PAAR, 4. April 2013, 13:50

Warum Mädchen keine Angst vor Mathematik haben sollten

Danke, danke, danke! Mein Dank gilt allen Höhnern und Nörglern, die mich ob des verschlafenen Rektorstages

Trifft ein Mathematiker einer Physiker ... Kennen Sie de

BLOG: MAMA GEHT STUDI

14.3.2013, 10:06

Tanja Paar

Mathe, mein Angstfach [270

Über zwei Anläu Ausfälle und den Anschluss

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 it's focus on a pleasant graphical representation. It maps the value of the cells to a colour spectrum seen in figure 3.8.

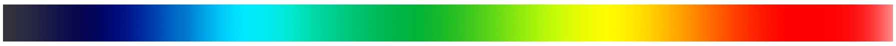


Figure 3.8: The colour spectrum used to map the value to the resulting colour

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

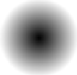


Figure 3.9: The circle brush used to smooth data

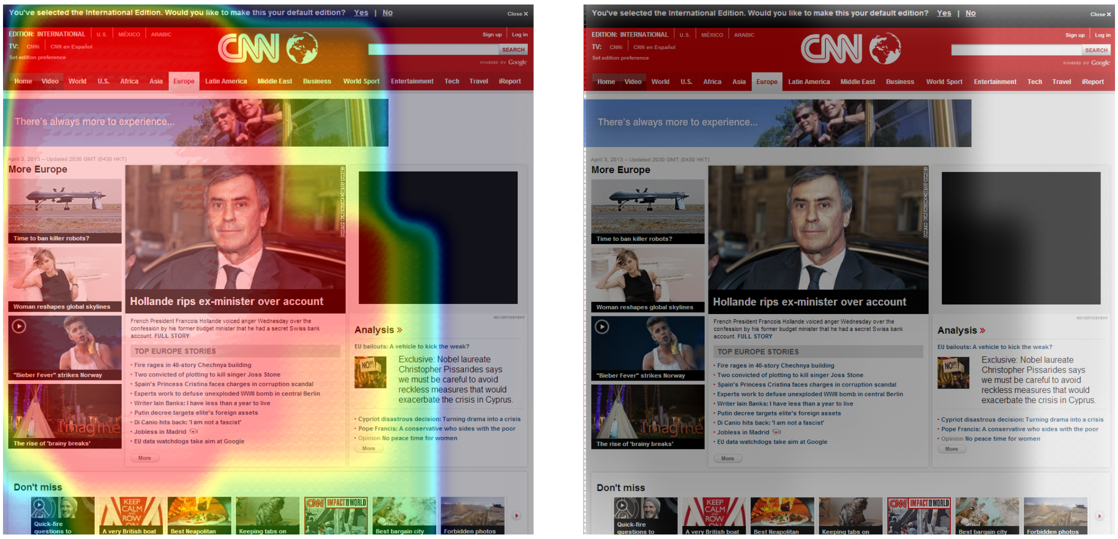
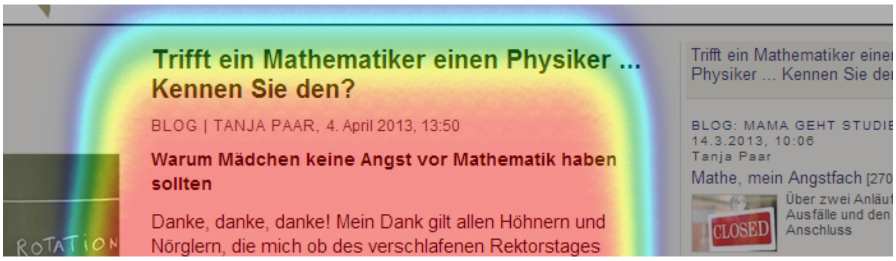


Figure 3.10: The colour scale is mapped from a transparent black and white image with different alpha-values (right side) to the colours of the colour-spectrum (left side)

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.



Trifft ein Mathematiker einen Physiker ... Kennen Sie den?

BLOG | TANJA PAAR, 4. April 2013, 13:50

Warum Mädchen keine Angst vor Mathematik haben sollten

Danke, danke, danke! Mein Dank gilt allen Höhnern und Nörglern, die mich ob des verschlafenen Rektorstages

Trifft ein Mathematiker eine
Physiker ... Kennen Sie de

BLOG: MAMA GEHT STUDIE
14.3.2013, 10:06
Tanja Paar

Mathe, mein Angstfach [270

Über zwei Anläu
Ausfälle und den
Anschluss

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

```
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.

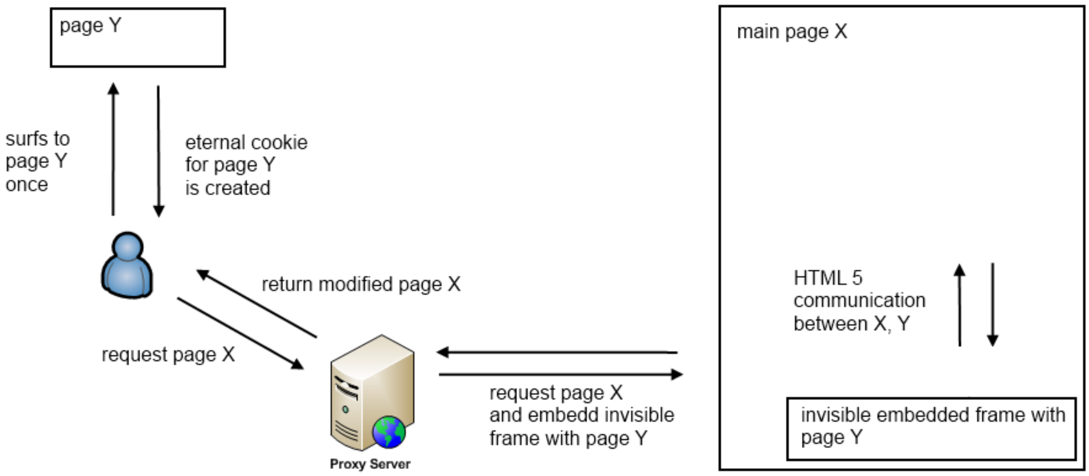

page Y
surfs to page Y once
eternal cookie for page Y is created
return modified page X
request page X
Proxy Server
request page X and embedd invisible frame with page Y
main page X
HTML 5 communication between X, Y
invisible embedded frame with page Y

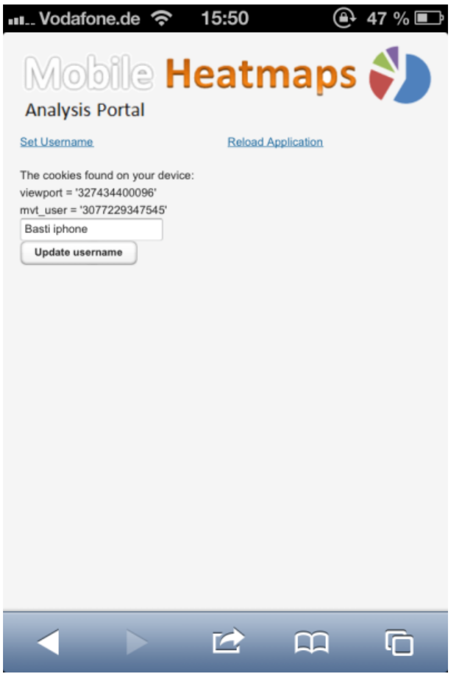
Figure 3.12: IFrame embedding trick using HTML5

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.



The cookies found on your device:
viewport = '327434400096'
mvt_user = '3077229347545'

Basti iphone

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 RDF[16] and OWL[17] 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 SPARQL[18] .

The analysis web application offers the possibility to export all data to a simple CSV[19] 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).

[16] Resource Description Framework - http://www.w3.org/RDF

[17] Web Ontology Language - http://www.w3.org/OWL

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

[19] Comma Separated Values - http://www.ietf.org/rfc/rfc4180.txt

CHAPTER 4

State of the art

4.1 Introduction

The previous two chapters were handling the common part of this thesis and were written in collaboration between the members of the project team. This chapter now focuses on the specific topic of this thesis by introducing a definition of terms and technologies that will be used in the proceeding chapters, research and analysis of literature. A comparison of existing approaches in the field of Digital Preservation, Web Archiving and Website Optimisation is presented, as well.

4.2 Digital Preservation

4.2.1 What is Digital Preservation?

The term “Digital Preservation” is widely used, also in many different ways and meanings. Therefore it is not easy to give a universally valid definition. However Jones and Beagrie from the Digital Preserveration Coalition [1] came up with a very general description which seems to be suited for an introduction of the term:

> Digital Preservation refers to the series of managed activities necessary to ensure continued access to digital materials for as long as necessary. [19]

Jones and Beagrie also distinguish between three types of Digital Preservation based on the time frame the information has to be accessible:

- Long-term preservation - Continued access to digital materials, or at least to the information contained in them, indefinitely. [19]

- Medium-term preservation - Continued access to digital materials beyond changes in technology for a defined period of time but not indefinitely. [19]

[1] `http://www.dpconline.org/`

- Short-term preservation - Access to digital materials either for a defined period of time while use is predicted but which does not extend beyond the foreseeable future and/or until it becomes inaccessible because of changes in technology. [19]

Another very short but precise definition is given by Strodl in "How to Choose a Digital Preservation Strategy: Evaluating a Preservation Planning Procedure" [44]:

> Digital Preservation - the process of keeping electronic material accessible and usable for a certain period of time. [44]

What is not yet fully covered by these very basic definitions is the process behind the scenes to reach this goal of availability including all the necessary planning and resources, problems such as privacy issues, legal aspects and various other difficulties that can make it hard to fulfil the task within a specific case with well defined requirements.

These aspects will be covered in the following sections.

4.2.2 History and Goals of Digital Preservation

As Digital Preservation was now introduced, this section shall take a look at the history and also the general goals of this discipline. To explain the possible reasons why such a process or discipline is needed, we have to take a look at the history of mankind first. Long before the first computer existed, people tried to preserve information for the future generations in order to preserve knowledge. This was a significantly important process for mankind. Without such efforts a lot more knowledge would have disappeared over time. In the early times of information preservation tools such as stones, parchments, paintings and later on also books were used. These artifacts of knowledge could be preserved over a very long period of time if well treated and stored.

The problem we are facing today in the computer or Internet era, is that the amount of data that is available and produced every day is exploding. In contrary to ancient times where a small amount of people such as monks, educated people or philosophers controlled the information, today nearly everyone who has access to the Internet is able to contribute and publish information using blogs, forums, wikis, social networks, file servers, photo galleries and many other resources. The problem that is arising is: it is not possible to preserve all the data over time. A distinction has to be made in order to decide what data can be defined as qualitative information and which information can even be specified as knowledge that is worth to be preserved in whatever scope of time. The general goals that are derived from this conclusion are:

- Identification of qualified information and knowledge,
- decision of how long it should be preserved,
- storing and preserving the information/knowledge over time using different strategies which are discussed later,
- and providing an interface to make it easily accessible for users.

4.2.3 Examples / fields of operation

The term Digital Preservation is a general one and therefore it has many fields of operations , which can have completely different requirements. However the main goal is always the same: preserve digital information over a certain period of time. Because of the wide usage it is not possible to cover all fields within this thesis, instead a selected number of interesting applications is outlined in this section.

Audio Preservation

A very important field of operation of Digital Preservation is Audio Preservation. This discipline focuses on the long-term preservation of all kinds of audio data.

> The world's stock of audio recordings is estimated to be more than 50 Mh (million hours) of materials. The greater part of these recordings is analog, and many are unique documents of cultural, scientific, or artistical value. [43]

The major challenges in this field are technical, legal and organizational issues. As already stated by Schuller a large amount of audio recordings are still existing on analog media. Sooner or later this analog media has to be converted somehow to a digital format because analog audio media such as vinyl or compact cassettes have a limited time of survival. In contrary to digital media, analog media also loses quality with each replication, which limits the number of times it can be copied. Nowadays nearly every new audio recording is made digitally.

Figure 4.1: Process of digitalizing audio data which is stored on magnetic film at the National Archives of Australia , http://www.naa.gov.au/

Once the data is available digitally, the questions is how to store it? Which format is best suited for long term archiving? What if new formats come up?

> With the development of the World Wide Web have come new digital sound formats and delivery systems that offer archivists, as well as home consumers, a wider variety of recorded sound, instantaneously, than in any time in history. [5]

Formats such as MP3 are very popular today because of the high compression rate. This enables to store audio recordings or music titles with a very low amount of data needed. In respect of storage costs, such formats could also be interesting for audio archivists.

> Whether these sound recordings are going to be maintained for posterity or only for the next 10 years, if they are to persist, it will be as digital recordings of some type. [5]

How to face the challenge of finding the right format? How to handle format changes over time? Concerning these questions the section 4.2.4 will go into more detail.

Moving Image Preservation initiatives which are focused on preserving analog film and digital video are facing very similar problems and challenges as in the field of audio preservation. Again digitalization plays a major role because of already discussed problems in preservation of analog media.

E-book Preservation

E-book Preservation is focusing on archiving items of electronic publishing.

> The concept of electronic publishing was first articulated by Vannevar Bush of the Massachusetts Institute of Technology (MIT) in the seminal 1945 article “As We May Think”. [40]

The challenges in this field are competing technical standards, digital rights management, definitional issues, restructuring within traditional publishing, selection of the best suited format and also user acceptance. [40]

Another critical issue is the screen size. The size and resolution capabilities of e-books vary. [40] On top of this, nearly every e-book provider comes with proprietary software and different formats claiming that his format is the best for reading and publishing. This leads to major problems concerning long-term archiving of e-books.

Although at the moment the user acceptance concerning e-books and e-book readers in the future is very hard to predict, the resulting publications will have to be preserved somehow.

> Librarians and others involved in digital asset management will have to preserve at least some of this material for future reference, since it is expected that original works will be created and many of these may exist only in electronic form. [40]

Personal data preservation

Among all the various fields of applications for generally persevering content either produced or provided for the public, also private users have needs to preserve all kinds of personal data. A good example would be private photographs or videos. Prior to the introduction of digital photography, people archived their photographs in hardcover photo albums. Nowadays the major part of private photographs is made digitally, most likely in the JPEG [2] format, and stored somewhere on a hard or flash- drive. How can private persons ensure that their albums are still viewable for their descendants? A possibility would be of course to print out all the photographs and archive them in hardcover books as usual, but it should be also possible to archive the data somehow digitally. Of course the data itself can be archived, but will the JPEG format still be viewable in 30 years? Such questions that cannot be answered with certainty at the moment demonstrate that Digital Preservation also plays a role on the private sector.

4.2.4 Strategies

In order to preserve content in terms of the definition of Digital Preservation as stated in [19], several different strategies seem feasible. The major problem of finding the right strategy for Digital Preservation, is the need of long-term preservation. For short-term preservation many strategies and solutions are available. For long-term preservation however a common recipe is not yet found. Various different strategies may have to be combined in order to achieve this goal. The most relevant strategies are migration and emulation, but also other strategies will be shortly summarized at the end of this section.

Migration

The most often used strategy of content preservation is migration. Basically objects that need to be preserved are simply converted into new and accessible formats, or newer versions of its own format repeatedly. [44] This process may take place whenever new formats arise or differently stated if old formats tend to be outdated and accessibility is decreasing or support is no longer guaranteed.

> The purpose of migration is to preserve the integrity of digital objects and to retain the ability for clients to retrieve, display, and otherwise use them in the face of constantly changing technology. [52]

A possible drawback of continuous migration of objects over time is the risk of losing authenticity. A very primitive example would be the conversion of a Microsoft Word [3] document from an older version to a newer one. The conversion would most likely cause a minor change in the appearance and layout of the original document. Of course this could be acceptable in specific scenarios. For document archiving PDF/A [4] has been adopted as the current most used format. It is implementing a subset of the PDF Format and has proven to be well-suited for mid or long-term preservation. [44]

[2] Most common image format using lossy compression created by the Joint Photographic Experts Group

[3] `http://office.microsoft.com/en-us/word/`

[4] `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38920`

Emulation

The second most relevant strategy is called Emulation. The difference compared to a migration approach is made clear by van der Hoeven in [48]

> Emulation does not focus on the digital object, but on the hard- and software environment in which the object is rendered. It aims at (re)creating the environment in which the digital object was originally created. [48]

In other words the (old) system, with all its requirements, which is capable of presenting the content in the way it was intended to, is emulated on a modern system. Therefore the content itself has not to be changed, because it can be used by an emulated version of the originating system environment.

The technique is very popular in the area of computer games. Various old gaming consoles can be emulated nowadays using a standard pc. Also virtual computers provide the technical prerequisites to emulate old operating systems that are no longer supported or available. Unfortunately the problem is only shifted with such an approach, because it has to be ensured that the technology or virtualization software is still available in order to use the virtual environment in the future.

Other strategies

Migration and Emulation are the two most popular strategies, although there are many others which are also focusing not only on the readability in terms of formats and environments, but also on preventing loss or decay of the actual data. Some other strategies are:

- *Encapsulation:* The details of how to interpret the content are packed within the encapsulated information so that the content is self describing. The creation of the software that created the content in the first place on the future platform is also part of this strategy. For establishing relations between information components within the content object so called containers or wrappers are used. For example a relation can be made from the digital object itself and its describing meta data. [23]

- *Refreshing:* The transfer of digital information of one storage medium to another of the same kind without changing the bit-stream. For example refreshing data that is stored on a five year old compact disc, by reading the content and re-writing it on a new CD. The reason of doing this is the problem of the possible decay of the storage media on which the content is stored.

- *Replication:* This strategy is used for preventing eventual loss of data due to various reasons such as computer failure, natural disasters or decay of storage media by replicating the information to several locations. In other words distributed redundancy is used to ensure that the content is not damaged or lost completely over time.

4.2.5 Problems / Difficulties

At this point it should be clear that the discipline of Digital Preservation is needed in many fields and that it represents an important task for mankind in order to preserve information in this highly dynamic digital era. Despite the fact that each field of application, each project and each use case of Digital Preservation has its own specific problems and difficulties to cope with, there are some general ones that affect the majority of applications. These general problems are discussed within this section to give an overview of the complexity of the task of Digital Preservation.

- *Physical Objects vs. Digital Objects:*

  > The process of preserving digital objects is fundamentally different from that of preserving physical objects such as traditional books or documents on paper. [46]

  As described by Thibodeau in [46] it is not possible to preserve a digital object in the same way as a physical object. He shows that for preservation an extraction of logical units of the digital object has to be done and further relations between these units have to be identified and established in order to be able to reproduce the original object. This process is similar to the strategy of Encapsulation which was already discussed in the previous section.

- *Technological Change:* The main problem that turns the preservation of digital objects into a hard task is the danger of technological change. As shown in the previous point preserving digital objects is much different from preserving physical objects. A book for example is not dependent on any technology. As long as the language the book is written in is still understood, the original content of the book can be retrieved easily in the future if it was protected from decay. A document written on a computer, which is very similar to a hardcover book, is much harder to preserve. The format used for reading documents not only might but will change over time. How can one guarantee that the original document is still accessible on future computer systems? It is a major problem to deal with this kind of technological change in the digital world in order to ensure that digital objects survive over time. The last section did show some strategies to overcome those problems, but each solution leads to another bunch of problems such as storage costs for emulation solutions or effort needed for migration.

- *Storage:* Apart from the problem of changing formats and accessibility in the future, the fundamental task is to store the actual digital object somehow safely for long periods. This of course requires a thoroughly planned storage system that provides functions for data replication and long term reliability ensuring that no data gets lost or damaged over time. Data centers storing huge amounts of business data are facing similar problems and already built solutions for that matter although the periods they have to store the data are most likely clearly defined and shorter than needed for Digital Preservation projects. The long periods of storage may require a change of storage media at some point. This would than need effort for copying the data onto the new media.

- *Legal and organizational issues:* Digital Preservation projects and initiatives are often facing legal and organizational issues on the way of building up databases to preserve the digital parts of our history. First of all, it is not easy to establish international initiatives or alliances where multiple nations take part in and provide resources, funds and also information in order to create a meaningful archive of certain topic. An example of legal issues would be the already introduced discipline of Audio Preservation. An attempt of building an international library of digital music with the goal to preserve every piece of music that was published on earth would definitely face another problem, apart from accessing and storing the huge amount of data. The music industry would not allow the storage of protected music titles in a central database for free even if the accessibility of the archive would be limited to certain people. Another general issue is data protection. How can be ensured that no private information is preserved and human rights are violated? The process of preservation has to be well planned and documented in order to fulfil the legal requirements.

- *Costs:* Another problem is the financing of such preservation projects. Who will pay for an archive that “only” serves the need of preserving parts of our history? There is no direct economic value in a digital preservation system. Of course one could argue that the access to the archive in the future doesn’t have to be free. The owner of the archive could request a fee for the usage of the system but how many people would use it and pay for it? It probably would not finance the whole project which would be very expensive depending on the scope of archiving. The more likely scenario would be that the projects are funded by governments and governmental organizations. An example would be the state of Australia which has been examining Digital Preservation since 1994. [23] But also many other countries and organizations are working on Digital Preservation projects and initiatives.

4.3 Web Archiving

The previous section gave a general introduction to Digital Preservation and showed different approaches. This section now introduces Web Archiving which can be seen as a branch of Digital Preservation that focuses on preserving the contents of the World Wide Web.

4.3.1 What is Web Archiving?

The term Web Archiving describes a discipline that pursues the goal of preserving the history of the World Wide Web. In contrary to other Digital Preservation disciplines where the main problem is to keep information available despite of the changes of formats and storage media, Web Archiving is facing another problem: rapid change and highly dynamic content.

4.3.2 Why Web Archiving?

A suitable explanation is found in [36] by Rauber:

> With the amount of information being made available in digital form, the need to archive and preserve access to these documents becomes an essential asset, which is only starting to be appreciated to its full extent. [36]

As already mentioned in the previous chapter, preserving cultural artifacts, information and even knowledge has always been a crucial task for mankind. Since the Internet era started in the 90s, the amount of information that is published digitally on a daily basis increased dramatically. One could argue that the World Wide Web has already become a very important part of our history, as many artifacts characterizing our lives are contained within it. This is confirmed by Masans who stated in [18] that

> Web Preservation is a cultural and historical necessity. [18]

Although it looks like the goal of preserving the web is broadly adopted and accepted, there are also people disagreeing and questioning the discipline of Web Archiving [18]. In [18] Masans identifies three different types of arguments against Web Archiving:

1. *Question of Quality:* A major problem of Web Archiving is the huge amount of data that is available. The question is how to archive the important parts in order to ensure a minimum of information quality that is stored within the archive. How is it possible to do such a selection? How is it possible to distinguish important content from definitely unimportant content? The classification of information published on the web has become a general problem. The massive amount of individual publishers generates the need of strategies to evaluate and merge information of various different sources. An example would be the wikipedia[5] platform, where the content that is published by individuals is double checked and evaluated by the platform operators using some kind of reviewing process.

2. *Lack of necessity:* This refers to people who believe that archiving the web is not needed because the web is self preserving. The idea behind is that the web itself is changing and adapting over time but meaningful content is preserved “automatically” by individuals, companies or organizations. To take again the example of wikipedia into consideration, the contributors are updating the content that is available on the platform constantly. Also old articles are not deleted except they are outdated or replaced by new ones handling the same topic.

3. *Impossibility of the task:* The third group sees the preservation of the web as an almost impossible task. The complexity and the amount of resources needed to achieve this goal are estimated too high to be managed by single organizations or countries. Only a global alliance with many members and large investments would be a possible solution. However according to the very low economic value of such an institution, it is not very likely that all of the stakeholders would invest that much money and resources into such a project.

[5] `http://www.wikipedia.org/`

4.3.3 Benefits of Web Archiving

The concept of a classic Web Archive describes the idea of making Internet content from the past available in some form in the future. The general idea is always the same, however there are various interest groups that would benefit in different ways from such an Web Archive. These are shortly described in the following listing:

- Historians: People investigating the history of mankind in the future obviously would benefit from the possibility of making use of the web content from the past. As the Internet is a crucial part of our history, an archive would give an opportunity to understand the past in a deeper sense.

- Countries / States: The benefit for single countries or states is also strongly related to historic reasons. One could say that a state has the obligation to preserve its own history to preserve its identity. This also happened in the very beginnings of humankind, where different peoples preserved their cultural artifacts and techniques for their descendants. Of course this does not mean that each state should build up its own archive, single states would also benefit from a global Web Archiving alliance that would save money and resources and would also help to overcome the gap between different states.

- Enterprises: Companies could also benefit from a Web Archive in terms of learning and improving. They could perform an analysis on past web appearances from competitors or even from their own, draw conclusions from it and increase usability, efficiency and quality of the companies' web portal and also its products and services.

- Private persons: People from the private sector would mostly benefit from the possibility of looking up information or content that is no longer available on the current web. Imagine the fictitious scenario that the youtube[6] portal would go offline by tomorrow and no following project that migrates the old data is planned. If the content of the youtube portal had been archived, users would still be able to access the content although it has been deleted from the web.

4.3.4 Challenges of Web Archiving

As introduced by Rauber in [36], a common process for building up an archive of the Web basically consists of four main phases: *acquisition, storage, preservation, and access provision*. For a better understanding of the complexity and challenges of Web Archiving these four phases are now illustrated shortly using the conclusions from [36]:

1. Acquisition: The very first challenge that comes up when an approach for building up a Web Archive is evaluated, is data acquisition. This task might be a lot more complicated than it looks like at first sight. Data acquisition can be further divided into the sub-tasks of selecting the relevant content and defining the method of how to retrieve it. Before a decision about selection can be made, the scope of the archive has to be defined. [36] In

[6] `http://www.youtube.com`

most archiving projects the scope will be limited somehow e.g. to a nation or a specific topic that is addressed in order to be able to deal with the amount of data and resources needed for it. When the scope is clear, the method of how to retrieve the relevant content has to be defined. Most projects use a crawler[7] based approach in order to cover a majority of the web pages that refers to the defined scope automatically. This is done similar to search engine crawlers. The crawler simply follows all outgoing links from a given input page, based on the predefined criteria. The main problem evolves when specific content has to be chosen to be in the scope of the archive, that cannot easily be selected by an automated program. In this case it has to be done either manually or by a complex software that is able to derive facts that indicate if a certain website is within scope or not. These programs are obviously not available for each desired scope and apart from that they are hard to develop and often need advanced computing power in order to find solutions for a given problem. Therefore this content has to be selected manually in most of the cases, which leads to personal costs.

2. Storage: Once the appropriate content is retrieved, it has to be stored somewhere. This again sounds like an easy task, but in fact it is not. The data has to be available for a very long time, otherwise the effort of building up the Web Archive does not makes sense. Two major problems arise when it comes down to long-term storage: [36]

   - Storage media decay: When digital information is stored on whatever media that is known until now, one has to cope with decay. Each used storage media has a limited time of survival, which in fact means that the data that is stored upon the media is no longer guaranteed to be safe anymore at some point and loss of data is inevitable. To prevent this from happening and therefore guarantee the authenticity of the archive, the stored data has to be copied to new media from time to time. The required interval of doing this is determined by the average time of survival of the used storage media.
   - Technological obsolesce: Another problem that can come up with storage is technical obsolesce. Who can guarantee that the storage media that is used is still accessible in the future despite the fact that the stored data is still intact? The answer is: nobody can. Therefore the data that is stored on a specific type of storage media, has to be shifted to new types of media at some point when the old technology is about to become obsolete in order to guarantee accessibility in the future.

3. Preservation: This challenge refers to the general problem of preserving digital objects, which was already discussed in section 4.2. The major difference between preserving physical objects and new media objects is also pointed out precisely in [36]:

   > While ancient stone plates, papyrus scrolls, and codices can be accessed as long as their physical being is preserved, additional tools are necessary for gaining access to electronic media. [36]

[7] an automatic program that retrieves information of the web using one or multiple starting pages as input and following each link that is found on these and the following websites

For preserving digital information for long term accessibility, therefore special methods have to be implemented as already shown in section 4.2.

4. Access provision: The last step of building a Web Archive is the provision of access to the archived content in some form. Based on the available resources, similar navigation as in regular search engines could be provided using a keyword search on the indexed website content, direct search on domains or simple navigation based on related topics of the website. Another problem could arise when granting access to the archive. Some people, organizations or companies could object to the public availability of such an archive on whatever reasons. This has also to be taken into consideration when starting such an archiving project. [36]

The life-cycle that is described above, is also known as *Web Curation*. *Digital Curation* in general describes the gathering, preservation, maintenance, collection and archiving of digital assets [10]. Concerning *Web Curation* also the indexing, sorting and presentation of the archived content plays a role.

4.3.5 Strategies

Web Crawling

This is the most promising approach for Web Archiving projects at the moment. The procedure of crawling the web using one or multiple starting pages and following all the links was already pointed out, but the process can be further divided into different approaches regarding methods, scope and quality of the resulting web archive: [26]

- *Method*
  - Automatic: The approaches that are fully automatic use the same procedure as search engines. The target very often is the whole Internet or at least a specified part of it and the crawlers are grabbing all sites they come along upon a predefined maximum deep, in order to avoid that they get trapped. This is the only approach that would lead to a fully usable archive of a certain part of the web where accessing users would be able to navigate through the archived websites like in the real Internet.
  - On-demand: On the contrary, on-demand approaches do not archive web pages automatically but offer the archiving functionality on request to the public. Services like peeep.us[8] use the crowd for building up a web archive. Every user can enter an URL of a web page he wants to have archived. The services archives the requested page and hands a link back to the user, which redirects to the archived version of the page. Crowd sourcing[9] is an interesting strategy for Web Archiving. The resources needed would be much lower than for archiving the whole web, and only websites that users want to have in the archive would actually be in there. The prerequisite for a successful project using this approach of course would be participation of the

[8] http://www.peeep.us/

[9] approach to use the power of many people (the crowd), to tackle with a difficult or time consuming task.

user base, otherwise the archive would not contain any serious parts of the web and therefore it would be uninteresting.

- *Scope*

  - Site vs. topic vs. domain-centric: Site-centric archiving is focused on a number of preselected websites only. This is very often used by corporate bodies or institutions. The capabilities of such archives are limited [26]. Topic-centric archiving is gaining more and more popularity. Such approaches clearly focus on a specific topic or a specific type of content that has to be archived. An example would be the French elections web archive fulfilled by the Bibliotheque nationale de France (BNF) or the Miverva project from the Library of Congress which is explained in [42]. Domain-centric archiving is not focused on the content type but on the content location [26]. A specific domain is selected as archiving target such as e.g.

    .gov. It could also be a national web archive which only targets at archiving web pages from the national domain.

  - Selective vs. comprehensive: A similar distinction is made by selective and comprehensive web archives. A comprehensive approach tries to archive the whole web or at least everything that is possible to retrieve with common techniques. In contrary, a selective approach has a clearly defined scope of which part of the web should be archived.

  - Content types: Another possibility to set the focus of the web archive is by content type. The web does not only consist of web pages. Many other files such as documents in various formats, audio or video files are available. A possible scope would be e.g. to archive all pdf files that can be retrieved from the World Wide Web.

- *Quality / Completeness*

  - Extensive vs. Intensive:

    > Graphically, completeness can be measured horizontally by the number of relevant entry points found within the designated perimeter and vertically by the number of relevant linked nodes found from this entry point. [26]

    Differently stated, it can be distinguished between the number of pages a crawler archives, and the level of deepness one site in particular is archived.

    > Ideally any archive should be complete vertically as well as horizontally. [26]

    As the “ideal” archive is not existing, a choice has to be made. Extensive approaches accept that not each page is gathered down to the deepest level. In figure 4.2 is shown that an extensive approach misses some pages at the bottom level. The advantage is, that more pages can be processed in less time. The horizontal completeness increases. In contrary, intensive approaches aim to collect fewer websites but collect deeper content. In this case, the vertical completeness is increased [26]. Figures

4.2 and 4.3 illustrate the difference between extensive and intensive Web Archiving approaches.

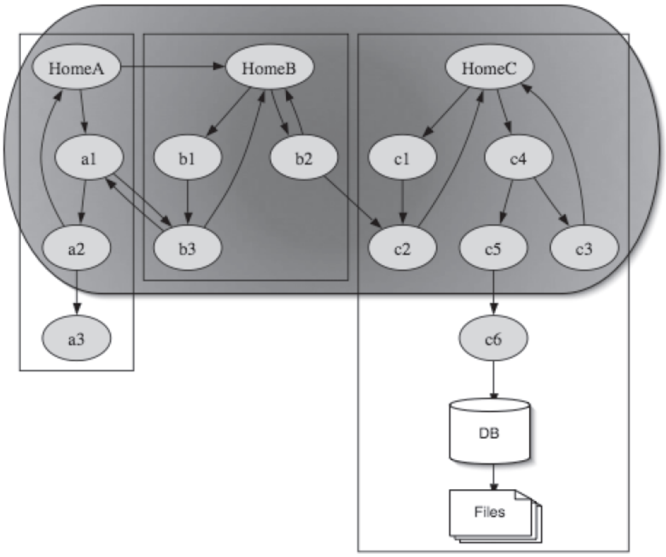


Figure 4.2: Extensive Archiving (Shaded Area). Some pages are missed (a3, c6) as well as the “hidden“ part of sites [26]

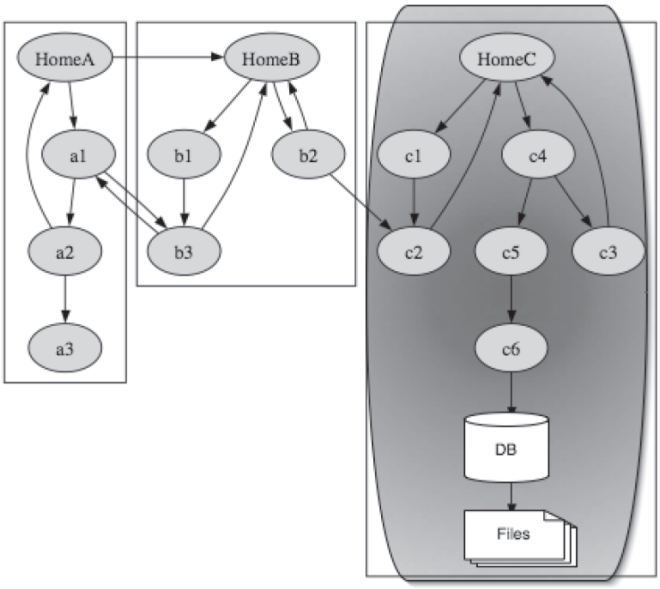


Figure 4.3: Intensive archiving (shaded area). Aims to collect fewer sites but collects deeper content, including potentially “hidden“ web [26]

Alternative approaches

- Database Archiving: This strategy targets at the archiving of web page databases, omitting the appearance of the website. The focus is on extracting the pure information that most likely is buried in some kind of database. The retrieved data is then translated and stored in a common XML format that allows querying. Many projects are focusing on archiving relational databases using the XML format. In [51] an evaluation on possible approaches is made for "preserving, publishing, and querying efficiently the history of a relational database". The main limitation towards Web Archiving is obviously that access to the websites databases needs to be granted, which most likely will only happen for a selected number of specific websites.

- Transaction Archiving: Another technique is Transactional Archiving which tries to archive all incoming HTTP requests to a web server, as well as all outgoing responses from the web server. This is usually done by intercepting the traffic between the web server and the user. The result is some kind of audit trail that represents all the requested HTML sources for each transaction that is made on the website. The usage for building large Web Archives is limited, because firstly the amount of data would be overwhelming as numerous requests occur on each website constantly, secondly a direct interception of the content would be necessary, which means that website owners would have to participate in the archiving initiative. A real life application for this matter is dealing with legal issues. Some companies use this approach on intranet or even Internet pages, in order to have a detailed audit trail of all accesses, in case it comes to legal action for some reason. [47]

The WARC format

Another thing that has to be introduced in terms of Web Archiving strategies is the WARC format. WARC stands for Web ARChive file format and specifies a method for combining multiple digital resources into an aggregate archival file together with related information. [30] It is an extended version of the ARC format which has been introduced by The Internet Archive [10] [30].

> The WARC format generalises the older format to better support the harvesting, access, and exchange needs of archiving organizations. [30].

Using common formats is very important in Web Archiving strategies as the task of archiving the whole web or at least a major part of it, is to complex for single organisations. In order to create synergies or distribute the task of archiving the web to multiple instances, common formats and also common procedures can help to achieve this goal.

4.3.6 Problems / Difficulties

The problems and difficulties that occur when designing a strategy for building a Web Archive are countless as it is not yet fully understood of how to achieve this goal until now. The complexity depends on the selected scope of the archive, as already mentioned before. Some even

[10] `http://archive.org/`

think that focusing on the whole web and trying to create a global archive is a goal that cannot be reached with today's capabilities. Despite this fact, some organizations and initiatives evolved, which accepted the challenge and try to do exactly that, with some exceptions towards the scope. Whether the scope is chosen to be globally or limited as e.g. preserving only websites that refer to a specific country, a number of problems would occur to most of the projects. These problems are discussed in the following list:

- Amount of data: The size of of the public indexable web was estimated at 11.5 billion pages in 2005 by Gulli in [15]. It can be assumed that this number has increased by a certain growth factor until now. The amount of storage space that would be needed is enormous, which in fact will raise the question of costs and funding.

- Ephemeral web pages: Content that is stored in the World Wide Web generally has a high volatility [36], which means that the documents on the web are changing very fast. Using a crawler based approach for example, it is very likely that a document changes during the extraction process of the crawler. [36] The time that crawlers need to find new websites is called "delay". When a crawler comes to an ephemeral website e.g. a website about an event, the delay could be to long in order to gather the related content [26]. This leads to the fact that it is not possible to create a holistic snapshot of the web at a given time, it is only possible to create an archive that contains web page snapshots taken within a specific time period - the time the crawler needs for one extraction run.

- Identifying websites within the scope: As already said, most of the Web Archiving projects have a predefined scope. A possible scope could be all sites that are located or refer to one country. The problem that arises after the selection of the scope is how can the websites that are within the scope be identified? The example of [36] is taken for illustration. There would be three groups of websites that should be considered in a national archive: (1) websites using the national domain (e.g. .at), (2) websites having a different domain but are physically located in the respective country and (3) websites that are stored abroad, but contain interesting content that is related to the national archives' country. The first can be found easily, the second could be extracted from domain registration services if they cooperated, but the third would be very hard to identify. It would most likely need a manual selection in order to include these sites. [36]

- Privacy and legal issues: Especially when using a crawler based approach, the problem of legal issues could come up very fast.

  > On the Internet there are sites publicly available, that are prohibited by law, such as Nazi propaganda or child pornography. Since automatic crawlers are not able to discern these illegal sites, they will include them in the archive along with all the other documents. Thereby the archive may contain offensive material without being aware of it. [36]

  Another problem that could come up, is that different companies, organizations or private persons would object to specific sites being archived. [36] A possible solution for this could be an opt-out option for website owners to prevent their website from being archived. Of course this in fact would lead to a limitation of the quality of the archive if many took this option.

- Storage and Preservation: These topics have already been illustrated in section 4.2.5 and 4.3.4.

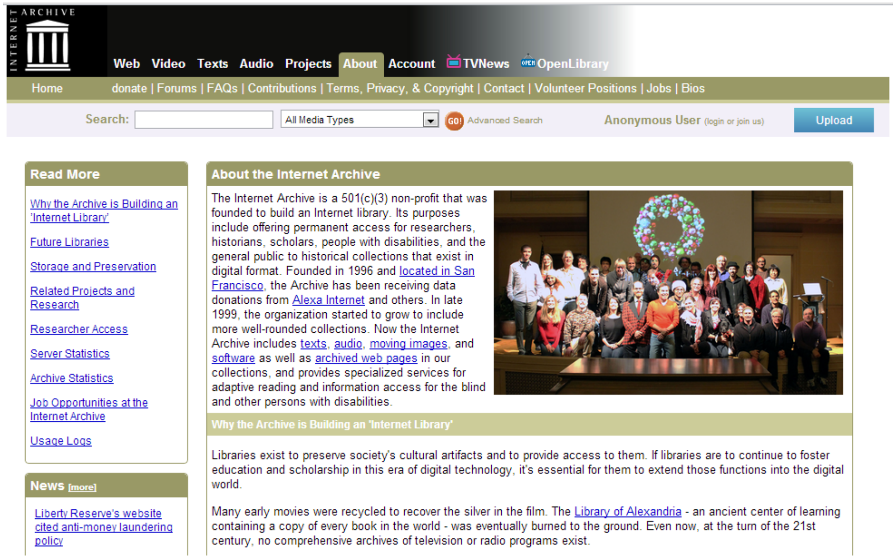
4.3.7 Existing projects / initiatives

The following section shall give an overview on existing projects and initiatives that have been or are currently focusing on the challenge of Web Archiving. Since the number of projects that deal with Web Archiving is too high to be fully discussed, a selection of the most interesting ones regarding current importance has been made.

The Internet Archive

The most prominent project is the Internet Archive [11] also known as archive.org. It was founded by Kahle Brewster in 1996 as a non-profit initiative, with the goal to archive public available information such as websites, audio data, video data or texts.

> "Its purposes include offering permanent access for researchers, historians, scholars, people with disabilities, and the general public to historical collections that exist in digital format". [2]


INTERNET ARCHIVE
Web Video Texts Audio Projects About Account TVNews OpenLibrary
Home donate | Forums | FAQs | Contributions | Terms, Privacy, & Copyright | Contact | Volunteer Positions | Jobs | Bios
Search:
All Media Types
GO!
Advanced Search
Anonymous User (login or join us)
Upload
Read More
Why the Archive is Building an 'Internet Library'
Future Libraries
Storage and Preservation
Related Projects and Research
Researcher Access
Server Statistics
Archive Statistics
Job Opportunities at the Internet Archive
Usage Logs
News [more]
Liberty Reserve's website cited anti-money laundering policy

About the Internet Archive

The Internet Archive is a 501(c)(3) non-profit that was founded to build an Internet library. Its purposes include offering permanent access for researchers, historians, scholars, people with disabilities, and the general public to historical collections that exist in digital format. Founded in 1996 and located in San Francisco, the Archive has been receiving data donations from Alexa Internet and others. In late 1999, the organization started to grow to include more well-rounded collections. Now the Internet Archive includes texts, audio, moving images, and software as well as archived web pages in our collections, and provides specialized services for adaptive reading and information access for the blind and other persons with disabilities.

Why the Archive is Building an 'Internet Library'

Libraries exist to preserve society's cultural artifacts and to provide access to them. If libraries are to continue to foster education and scholarship in this era of digital technology, it's essential for them to extend those functions into the digital world.

Many early movies were recycled to recover the silver in the film. The Library of Alexandria - an ancient center of learning containing a copy of every book in the world - was eventually burned to the ground. Even now, at the turn of the 21st century, no comprehensive archives of television or radio programs exist.

Figure 4.4: Screenshot taken from archive.org portal at 06/2013 - http://archive.org/about/

The archive is located in San Francisco and has gathered an impressive amount of data by now. In June 2013 the Internet Archive contained 342 billion archived web pages, about 1.6 million

[11] http://archive.org

audio recordings, 1.2 million video recordings and 4.5 million texts. The data is mostly provided by donations from the Alexa Service[12]. Alexa is using a web crawling robot that constantly gathers data from all over the world. A portion of the content is at some point copied to the Internet Archive. [36] The scope of the Internet Archive is global. Every public available document, including websites, audio, video and texts is included in the archive, if detected by the crawling service or manually contributed. Therefore the project is also receiving criticism from various people or organizations, because also data material that is prohibited or violating laws, human rights or privacy could be stored within the archive as it is gathered by the crawler.

> For this reason the Internet Archive states explicitly, that the collections may contain information that might be deemed offensive, disturbing, pornographic, racist, or otherwise objectionable, not to mention the accuracy and completeness of the information. [36]

The heart of the project is the so called “Wayback Mashine”. The service offers the historians, researchers, scholars and other interested people or organizations to go back in time and browse the Internet of the past using over 340 billion web pages which are stored within the archive. To access previous data one just enters a web address and selects a date from the available archived copies of the web page. Links that are embedded within the website are pointing to the closest archived version if available.

The Internet archive does not violate copyrights, because all information that is gathered is publicly available. There is also an opt-out possibility for website owners using the robot.txt file. With this file a website operator can prevent web crawlers from gathering information from the respective site.

Heritrix Heritrix[13] is an open-source web-crawler project started by the Internet Archive in 2003. As mentioned before, the Internet Archive is mainly supplied by the Alexa Service founded by Brewster Kahle and Bruce Gilliat in 1996. However, in the end of 2002 the Internet Archive decided that they would need their own crawler as well in order to address special use cases and perform crawls internally. Therefore the project called “Heritrix” was developed. [28]

> The Internet Archive believed it was essential the software be open source to promote collaboration between institutions interested in archiving the web. [28]

Heritrix is written in Java and is using a recursive procedure for crawling, as most web crawlers do, containing the following steps: [28]

1. Choose a URI from among all those scheduled

2. Fetch that URI

3. Analyze or archive the results

[12] `http://www.alexa.com/`

[13] `http://crawler.archive.org`

4. Select discovered URIs of interest, and add to those scheduled 5. Note that the URI is done and repeat

The next figure shows the architecture of the Heritrix crawler, illustrating the various components and the interlinkage between them:

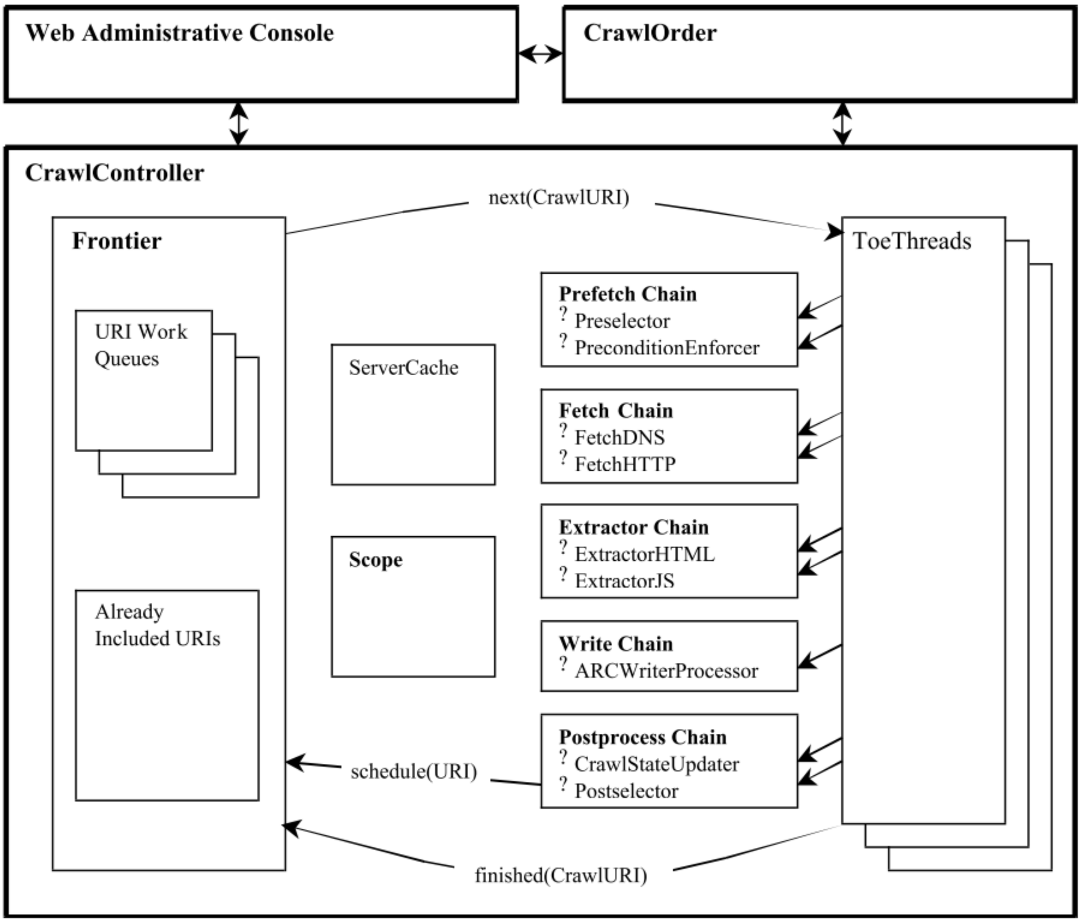


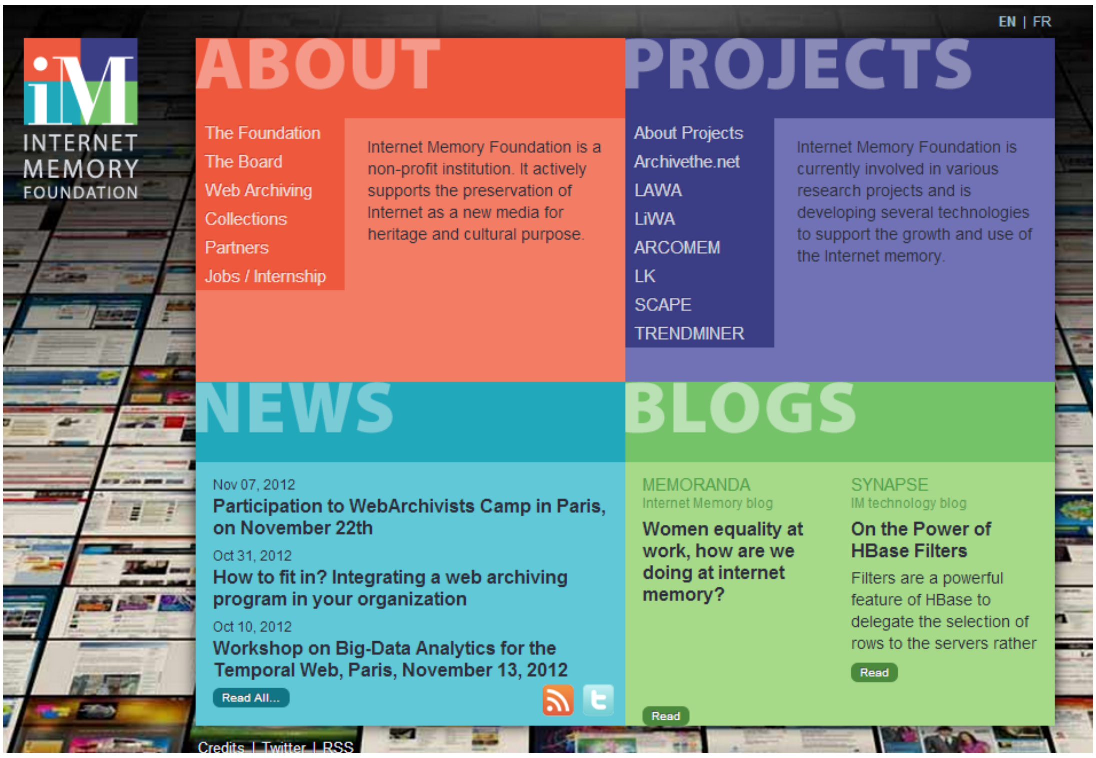
Figure 4.5: Major Components of Heritrix [28]

The three most important components of the crawler are the *Scope*, the *Frontier*, and the *Processor Chains*. The *Scope* has the control over the URIs that are processed within a crawl. It decides which URIs are used as a seed for the algorithm, as well as which URIs are selected within Step 4 on visited websites. The *Frontier* is responsible for scheduling the URIs that are to be collected, and also keeps track of those already selected. The component decides which is the next URI to be collected and also prevents redundant cycles for already scheduled URIs. The *Processor Chains* are modular components processing each URI that is selected. [28]

These include fetching the URI (as in step 2 above), analyzing the returned results (as in step 3 above), and passing discovered URIs back to the Frontier (as in step 4 above). [28]

Internet Memory Foundation

The Internet Memory Foundation[14] which was formerly known as the European Archive, is a non-profit institution which is originating in Amsterdam and Paris. Since 2004 the preservation of the Internet is within the focus. The foundation is archiving terabytes of data per month and is currently working on developing several Web Archiving technologies. The institution serves as a platform for various projects focusing on issues such as Digital Preservation and Web Archiving. One example for such a project is the Living Web Archive which is introduced in section 4.3.7. [27]



Internet Memory Foundation is a non-profit institution. It actively supports the preservation of Internet as a new media for heritage and cultural purpose.

Internet Memory Foundation is currently involved in various research projects and is developing several technologies to support the growth and use of the Internet memory.

Nov 07, 2012
Participation to WebArchivists Camp in Paris, on November 22th

Oct 31, 2012
How to fit in? Integrating a web archiving program in your organization

Oct 10, 2012
Workshop on Big-Data Analytics for the Temporal Web, Paris, November 13, 2012

Women equality at work, how are we doing at internet memory?

On the Power of HBase Filters

Filters are a powerful feature of HBase to delegate the selection of rows to the servers rather

Figure 4.6: Screenshot taken from Internet Memory web page at 06/2013 - http://internetmemory.org/

LiWA - Living Web Archive

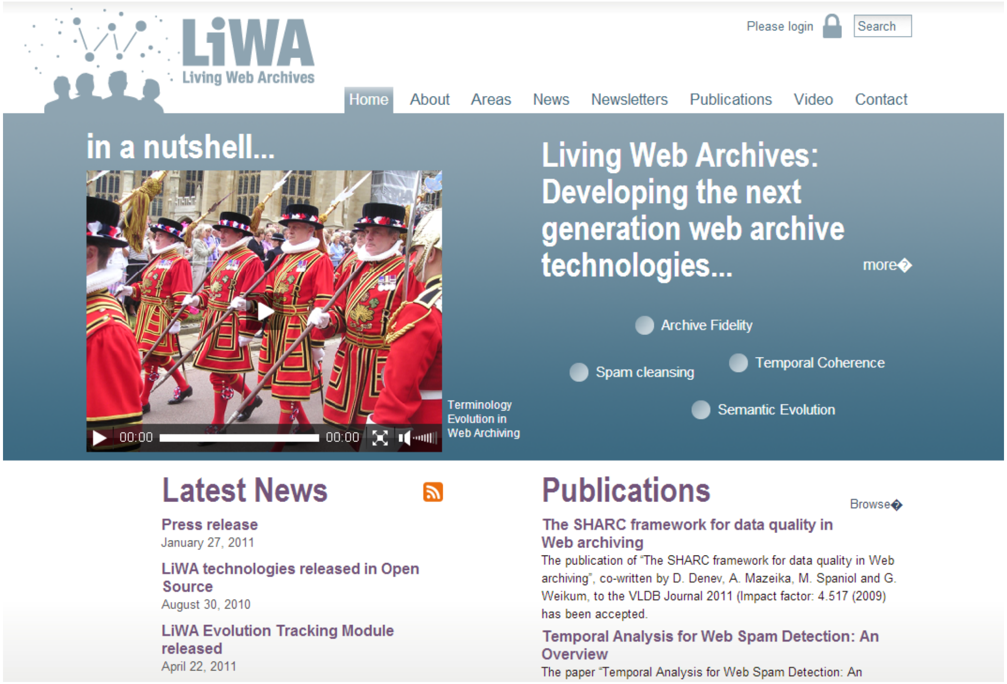
LiWA stands for Living Web Archive [15] and is a project which was originally founded within the Internet Memory Foundation in February 2008 as a three year project funded by the European

[14] http://internetmemory.org

[15] http://internetmemory.org/en/index.php/projects/liwa

Commission. The aim of the project was to improve techniques and methods for Web Archiving in respect of the new dynamics and interrelations of the Web content. Major topics that were addressed within the project were the problem of the deep or hidden web[16], and extracting all types of content that is very hard to extract with the current abilities, as well as filtering specific types of content that may be unwanted within an archive. Also social media information that gains more and more popularity was taken into consideration. [27] [3]

The LiWA consortium has not built up an own Web Archive, instead it provides new techniques and procedures in order to cope with known Web Archiving problems. A cooperation is also existing between the LiWA consortium and the Internet Preservation Consortium (IIPC) which is a global network of experts in the field of Web Archiving and is mentioned in the next section. [27] [3]



Living Web Archives: Developing the next generation web archive technologies...

Latest News

Press release
January 27, 2011

LiWA technologies released in Open Source
August 30, 2010

LiWA Evolution Tracking Module released
April 22, 2011

Publications

The SHARC framework for data quality in Web archiving
The publication of "The SHARC framework for data quality in Web archiving", co-written by D. Denev, A. Mazeika, M. Spaniol and G. Weikum, to the VLDB Journal 2011 (Impact factor: 4.517 (2009) has been accepted.

Temporal Analysis for Web Spam Detection: An Overview
The paper "Temporal Analysis for Web Spam Detection: An

Figure 4.7: Screenshot taken from LIWA project page at 06/2013 - http://www.liwa-project.eu/

The major areas that LIWA explores, concerning the improval of Web Archiving technologies are: [3]

- Archive Fidelity: capturing a complete and authentic version of web content.
- Spam Cleansing: Detecting and filtering out web spam and traps.
- Temporal Coherence: Provide coherent capture and navigation.

[16] Web content that is hidden behind web forms, dynamic web pages or locked areas of web pages that require a login

- Semantic Evolution: Keep track of evolving terminology in order to enable future interpretability of web archives.

- Social Web: Archiving rich and complex social web material.

- Rich Media: dealing with rich media websites.

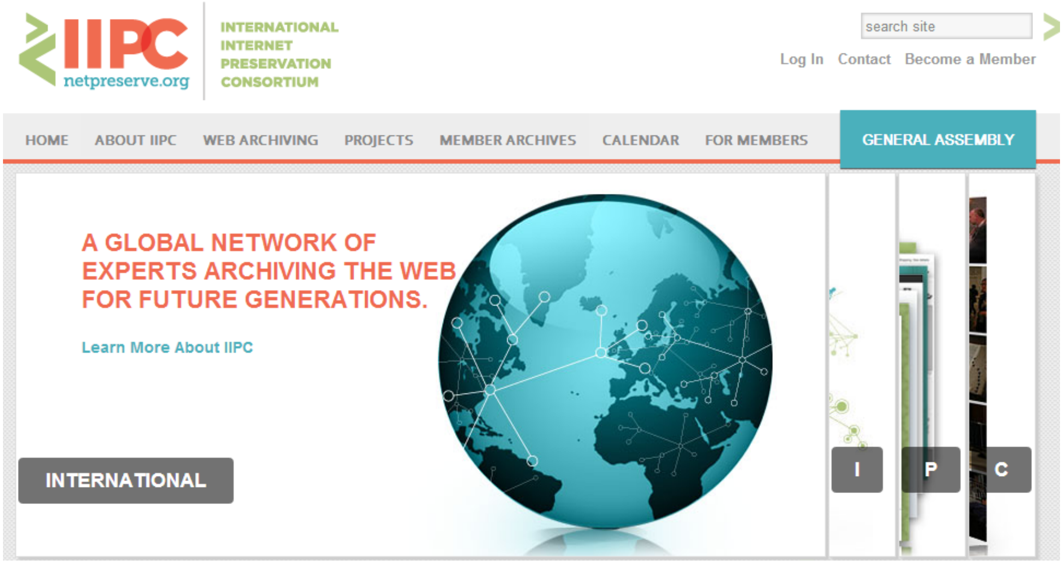
IIPC - The International Internet Preservation Consortium

IIPC stands for International Internet Preservation Consortium[17] and is a membership organization, with the target to create standards and best practices as well as improving tools or procedures of Web Archiving. Another goal of IIPC is to promote international collaboration and the broad access of Web Archiving and to assemble experts in the field in order to develop the discipline. [17]

The IIPC was founded in 2003 at the National library of France with 12 participating institutions. The members are now from over 25 countries, including national, academical and regional libraries and archives. [17]

Projects The IIPC did many projects in the past and also has a few ongoing projects at the moment as e.g. *The Live Archiving HTTP Proxy* project which was approved in Ausgust 2012 and introduces a new approach to minimise duplicate efforts in crawling and addressing the problem of website rendering. The project aims at building an HTTP Proxy that can be placed in the middle between a crawler and the web and capsules the task of capturing the content in a centralised component. The proxy can receive requests from multiple crawlers in parallel and put the communication into the archive. This helps to prevent code duplication and double efforts.

[17] urlhttp://netpreserve.org/



A GLOBAL NETWORK OF EXPERTS ARCHIVING THE WEB FOR FUTURE GENERATIONS.

Figure 4.8: Screenshot taken from IIPC web page at 06/2013 - http://netpreserve.org

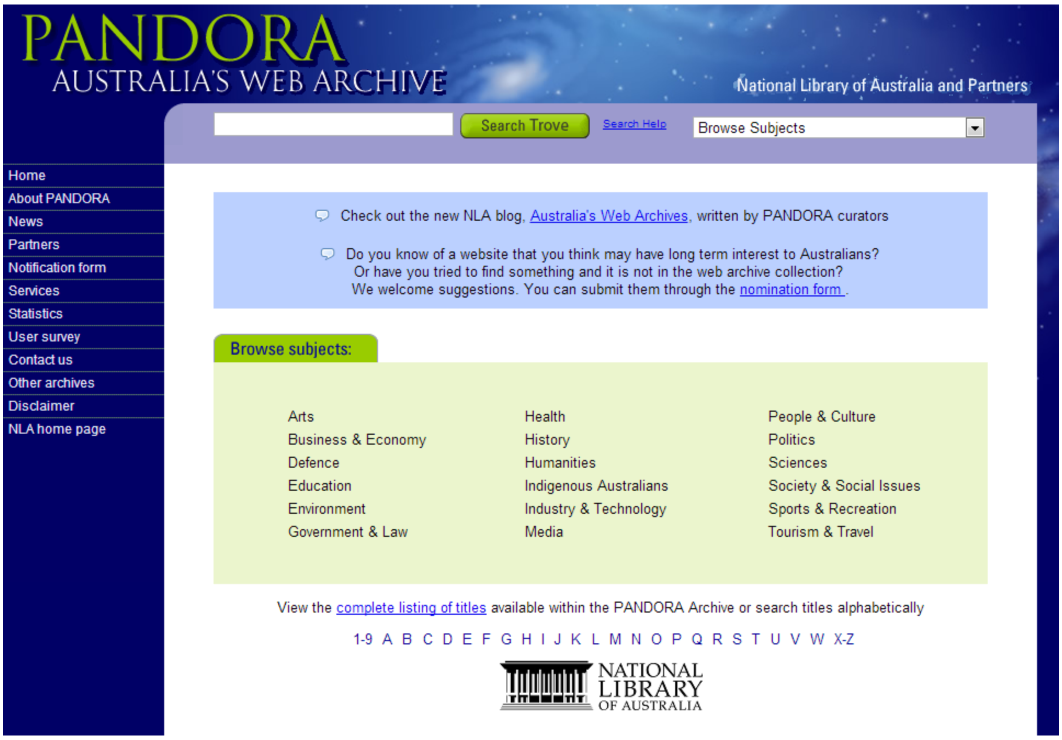
PANDORA - Australia's Web Archive

PANDORA stands for Preserving and Accessing Networked Documentary Resources of Australia[18] and is a "growing collection of Australian online publications, established initially by the National Library of Australia in 1996, and now built in collaboration with nine other Australian libraries and cultural collecting organisations". [32] The full-text of archived content is stored within the National Bibliographic Database and access is granted over the Library's national single search discover service. Also alphabetical listings are available on the PANDORA Home Page[19]. [32]

Selective approach The scope of the archive is very specific as it only targets online publications relating to Australia or Australians. The publications and websites that are chosen to be in the archive are selected manually by the National Library of Australia. The PANDORA project is therefore an example for a selective archiving approach. The advantage of such an approach is the limited amount of data that is needed and also the quality of the content stored within the archive. On the other hand, one could say that a selective archive is not serving the need to preserve the history of the web because many websites and other resources are kept out and are lost over time. [32]

[18] `http://pandora.nla.gov.au/`

[19] `http://pandora.nla.gov.au/index.html`



Check out the new NLA blog, Australia's Web Archives, written by PANDORA curators

Do you know of a website that you think may have long term interest to Australians? Or have you tried to find something and it is not in the web archive collection? We welcome suggestions. You can submit them through the nomination form .

Arts	Health	People & Culture
Business & Economy	History	Politics
Defence	Humanities	Sciences
Education	Indigenous Australians	Society & Social Issues
Environment	Industry & Technology	Sports & Recreation
Government & Law	Media	Tourism & Travel

View the complete listing of titles available within the PANDORA Archive or search titles alphabetically

Figure 4.9: Screenshot taken from PANDORA project page at 06/2013

4.4 Website Optimisation

4.4.1 Introduction

The previous sections within the state of the art chapter focused on introducing the disciplines of Digital Preservation and more specifically the archiving of the world wide web. The existing approaches within these fields are of major importance for this thesis, as a new approach for archiving the web is presented in the next chapter. This section will introduce Website Optimisation, especially targeting at new platforms such as smartphones and tablets. Similar to the previous sections, an introduction to the topic and the resulting problems will be made and currently existing tools and approaches will be shown. Also a small part of this thesis (see chapter 6) is introducing an approach to use the data from *Advanced User Behaviour Analysis* in a user-centric way in order to optimise web pages according to different user groups (such as smartphone users) needs.

The term "Website Optimisation" implies that many web pages that can be found on the world wide web are not designed in a way that fits with the users needs. In the past, web pages were designed with one single appearance, presuming that every user has equal requirements concerning look and feel, usability and the amount of information that is displayed on the web page. Of course the problem of different resolutions is a very old one, but the majority of website creators dealt with the problem by designing the web page for a minimum resolution on which the information would still be readable or they used basic HTML functionality to resize the content for non-standard resolutions. The development of mobile devices such as smartphones and tablets created a real need for Website Optimisation. For such devices it is not enough to just downsize the pages according to the different solutions. The boom of mobile devices also lead to a different style of browsing. Modern touchscreens enable very fast and intuitive navigation if the interface (e.g. an app) that is used is designed accordingly. Smartphone or tablet users are used to having a different type of interface containing less information, bigger fonts and navigation elements and are optimised for touch screens. This also applies to surfing the World Wide Web.. At the moment the majority of websites is not really optimised for mobile devices in a meaningful way. Many websites have special mobile appearances, but they often miss important information and functions, which leads to the fact that the user adoption is very low and people switch to the desktop versions again.

Apart from users utilizing mobile devices, Website Optimisation also applies to other user groups such as handicapped or elderly people. The Web Accessibility Initiative (WAI) [20] founded by W3C[21] is focusing on guidelines and specifications for making the web accessible for all people regardless the device they are using, their age or their disabilities. The interrelationship between developers, the content and users that consume the produced content somehow is illustrated in the following figure:

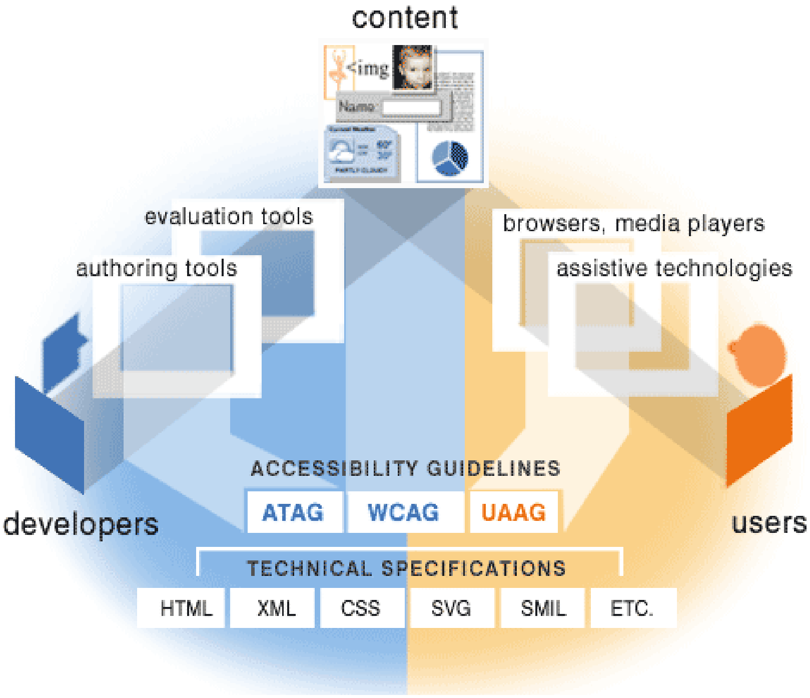


Figure 4.10: Interrelationship between developers, content and users, source: http://www.w3.org/WAI/guid-tech.html

[20] http://www.w3.org/WAI/

[21] http://www.w3.org/

The WAC is specifying guidelines for accessibility and is also driving forward technical specifications such as HTML, XML and CSS and the conformance of websites and browsers to it. WAC is also focusing on mobile accessibility and states the term as the following:

> Mobile accessibility generally refers to making websites and applications more accessible to people with disabilities when they are using mobile phones. [9]

WAI's work in this area includes people using a broad range of devices to interact with the web: phones, tablets, TVs, and more. [9]

The following guidelines address mobile accessibility: [9]

- WCAG (Web Content Accessibility Guidelines) covers web pages and web applications, including content used on mobile devices.
- UAAG (User Agent Accessibility Guidelines) covers web browsers and other 'user agents', including mobile browsers.
- ATAG (Authoring Tool Accessibility Guidelines) covers web browsers and other 'user agents', including mobile browsers.

WAI is also working on new technologies and enhancements of existing technologies in order to improve mobile accessibility: [9]

- IndieUI (Independent User Interface) is a way for user actions to be communicated to web applications, including mobile applications.
- WAI-ARIA (Accessible Rich Internet Applications) defines a way to make web content more accessible, especially dynamic content and advanced user interface controls.

The following sections will focus on the general goals of Website Optimisation, target user groups and existing tools for tracking and analyzing users in order to improve websites accordingly.

4.4.2 Goals of Website Optimisation

These are main goals or targets of Website Optimisation towards a specific user group:

- *Accessibility:* The most important concern is to make the information of the website accessible to the person that is trying to retrieve it in a meaningful way. This topic is mostly targeting at handicapped or elderly people that need an alternative way to access information on websites. For example for blind people, a screen reader that transforms the written content into an acoustic output can be used. A prerequisite that this works correctly is that the web page conforms to certain guidelines and is structured in a way that can be interpreted by the screen reader. Otherwise it would read anything on the page and it would be hard for the consuming person to retrieve the desired content.

- *Usability:* Another point is usability. This is very important for the new generation of mobile devices. Users utilizing smartphones or tablets have different needs than desktop users. The main limitation is the smaller screen size, which especially is a problem on smartphones, because a lot of scrolling and zooming has to be done in order to get to the information on standard desktop designed web pages. A different way of presenting the information has to be found for this special group of users.
- *Efficiency:* The ultimate goal of designing websites is efficiency for all user groups. Accessibility and usability are preconditions for a successful information retrieval, but the time that is needed in order to do that is also an important factor. The ideal website would provide each user group with the information they want to retrieve in the best possible way and in the shortest possible amount of time. When focusing on mobile users for example, a selection of the important content has to be made. Mobile websites try to do it, but often fail at selecting the right content and functions that are needed from the majority of the users. Chapter 6 will introduce an approach in order to provide a possibility to determine which information is important especially for mobile users.

4.4.3 Target user groups

A target user group could be any group of people having special needs in order to access information on websites. A possible selection is made in the following list:

- By technology:
  - Desktop users
  - Smartphone users
  - Tablet users
- By different needs of people:
  - Handicapped people
  - Young people
  - Elderly people
  - People not capable of the website's language

In the following this thesis is focusing on mobile users as does the approach of *Advanced User Behaviour Analysis* which is targeting at this user group specifically. The next section will introduce tools for tracking and analyzing users.

4.4.4 Tracking and analyzing users

Many tools and applications are available focusing on the analysis of web pages, and also on the tracking and analysis of the users surfing on this web pages. The key of understanding the users needs is to analyse their surfing behaviour and to draw conclusions about possible improvements from it. In the following two prominent tools will be introduced that use very different approaches for addressing the tracking and analysis of users surfing the web.

Google analytics

The most prominent tool for analyzing website users is Google Analytics[22]. Google Analytics is using the classic approach which tracks users regardless of the platforms or devices they are using. It is integrated via a script that is embedded in the target website , tracks all users surfing on the website and then transmits the data to the Google servers. The results can be viewed in the Google Analytics portal based on reports showing all kinds of quantitative statistics and KPIs[23] that can be used to analyse and improve the website. An example of the Google Analytics dashboard is shown in the following figure:

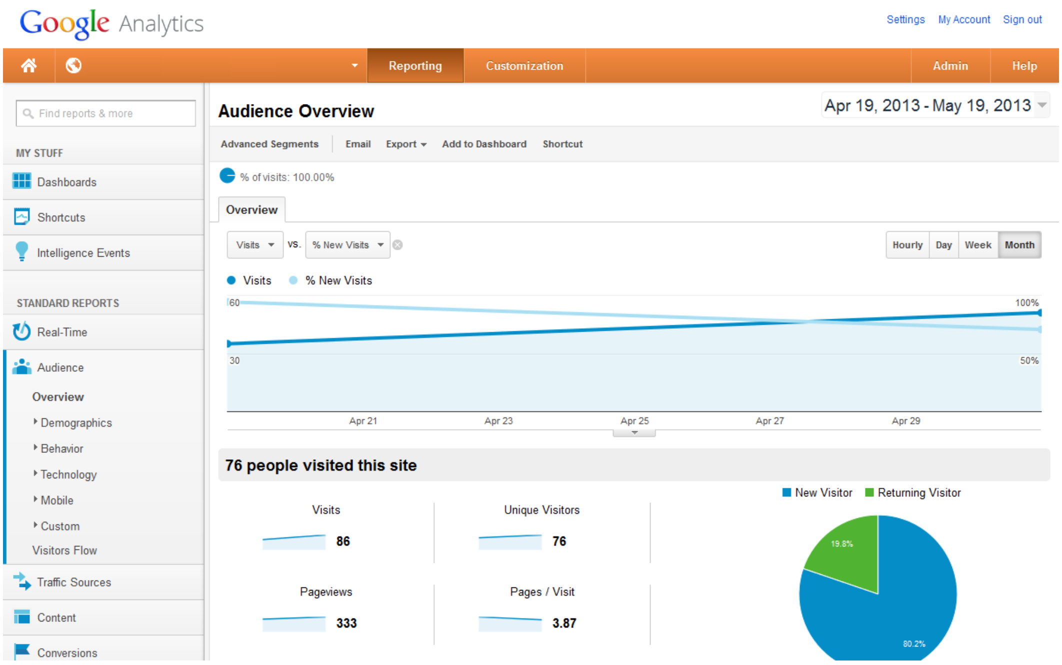


Figure 4.11: Google Analytics Dashboard, source: Screenshot taken from Google Analytics Account (http://www.google.com/analytics/)

The reports can also be received in PDF format by email on a configurable schedule and they contain quantitative data such as:

- Number of visiting users per day, week, month or overall
- Average duration that users stay on the website
- Location of surfing users
- Platform of surfing users

[22] `http://www.google.com/analytics/`

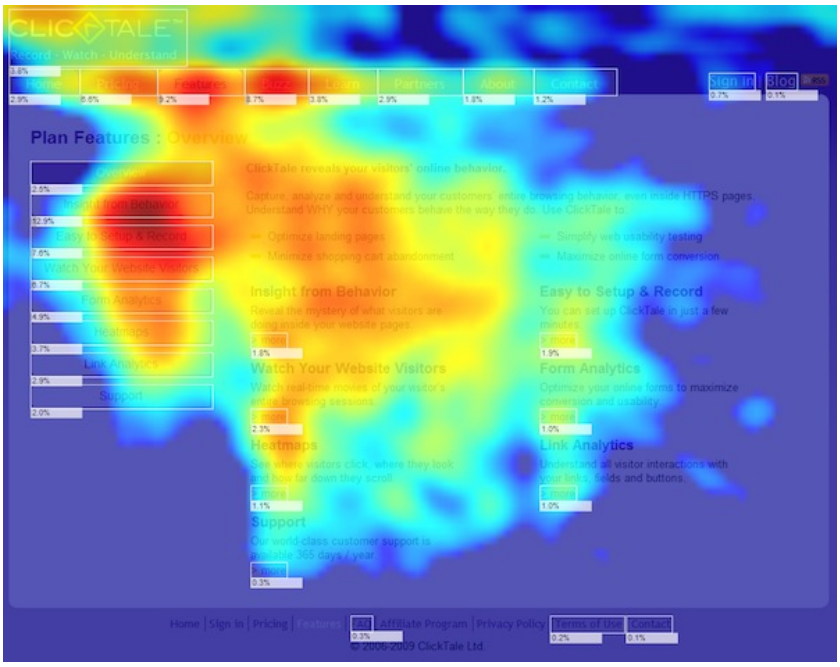
[23] Key Performance Indicator

- Screen resolution of surfing users
- and many more..

At the moment there is no real focus on mobile users concerning their surfing behaviour. A mobile user is handled like any other user within Google Analytics although Google has started to take mobile traffic into account. They analyse for example, when users decide to buy apps instead of using a mobile website, or the locations of mobile users in order to determine in which areas mobile traffic is increasing etc.

Clicktale

Tools that use heatmap based data for analysis are not that widely spread at the moment. The biggest competitor would be Clicktale.[24] They are mainly targeting at desktop users and use mouse tracking in order to generate heatmaps. As outcome they can produce click heatmaps showing the various mouse clicks of users, move heatmaps visualising mouse movement, and they also produce visitor recordings that show information about the surfing behaviour such as scrolling and keystrokes. The main drawback is again that they do not focus on mobile users at all. An example of a mouse movement heatmap is shown in the following figure:



ClickTale reveals your visitors' online behavior.

Capture, analyze and understand your customers' entire browsing behavior, even inside HTTPS pages. Understand WHY your customers behave the way they do. Use ClickTale to:

- Optimize landing pages
- Minimize shopping cart abandonment
- Simplify web usability testing
- Maximize online form conversion

Insight from Behavior
Reveal the mystery of what visitors are doing inside your website pages.

Easy to Setup & Record
You can set up ClickTale in just a few minutes.

Watch Your Website Visitors
Watch real-time movies of your visitor's entire browsing sessions.

Form Analytics
Optimize your online forms to maximize conversion and usability.

Heatmaps
See where visitors click, where they look and how far down they scroll.

Link Analytics
Understand all visitor interactions with your links, fields and buttons.

Support
Our world-class customer support is available 365 days / year.

Figure 4.12: Mouse movement heatmap produced by clicktale, source: http://insightr.com/blog/2009/11/25/user-experience-analysis-with-clicktale-to-save-your-busines.html

[24] `http://www.clicktale.com/`

The darker spots indicate that more mouse movement occurred within this sector. The biggest problem of mouse tracking systems is that the correlation between mouse and eye movement is hard to prove and does only apply in certain circumstances. This raises the question of the actual value of such an system concerning user behaviour analysis.

CHAPTER 5

Web Interaction Archiving

5.1 Introduction

In the last section the term “Web Archiving” has already been mentioned and thoroughly described. In this section a connection is made from classic Web Archiving, as mentioned before to a new type of Web Archiving using the data that is collected by *Advanced User Behaviour Analysis*, more specifically from the Mobile Viewport Tracking system.

Classic Web Archiving approaches are content centric. The main goal or target of such systems is to get a as complete as possible and runnable copy of a website and store it somehow for later usage. This process has to be repeated periodically, e.g, monthly / quarterly or yearly to historicise the changes of this web page. The approach is very similar to search engines. They also use so called *crawlers* to grab the content of web pages periodically over time. The resulting web archive that develops over time, gives end users the possibility to take a look at how a specific website did look like at a specific time in the past. As already discussed in the previous chapter, this leads to various problems such as storage costs, inability to copy websites entirely, especially very dynamic websites and many more. Another very important drawback is the lack of user experience[1]. The user perspective is completely ignored by those classic archiving systems. The following sections will now introduce a completely new method of archiving the web, based on a user centric approach which is called “Web Interaction Archiving”.

5.2 The idea of Web Interaction Archiving

The main idea of Web Interaction Archiving is not to preserve the Internet itself, but to preserve the user experience of people surfing in the web. The assumption behind that is that the history of the world wide web does not only consist of the content of websites, but also of the users' interaction with the content. The way of how the content is actually perceived by people using different devices, inputs or navigation techniques, is also a very interesting and therefore an

[1] described in detail in section 5.3

important part of our history. Historians could argue that such an archive would not cover the whole history of the web, but at least it could be seen as an extension to classic web archives, that enables people to discover the user experience of the past. The idea, that is also the topic of this thesis, is to use the crowd represented by users utilizing mobile devices, such as smartphones in combination with the technology of *Mobile Viewport Tracking*, in order to create a user-centric Internet archive. The approach refers to a known method to cope with complex or time-consuming problems which is called "crowd sourcing". The term is further described in the next section 5.3. The following sections start with a clarification of important terms. Subsequently the data that is available from *Advanced User Behaviour Analysis* is discussed once again in order to link the idea of Web Interaction Archiving to the project of *Mobile Viewport Tracking* that was performed during this thesis. In the end a description of how such a user-centric web archive could be implemented in detail is presented.

5.3 Clarification of important terms

This section is introducing some important terms that are used later on in this chapter. The exact meaning is defined in order to create a common base of understanding.

5.3.1 User experience

User experience is a term that is not easy to define precisely. In general this could be anything a person sees, feels, hears, smells or tastes while using some kind of technology. In this definition the focus is on users surfing the web using any kind of device that is capable of that. In this case the term user experience describes the perception of the content that the user is accessing while surfing in the Internet. The user experience basically consists of the following two components:

- *Surf actions:* the actual actions a user performs on a web page during surfing such opening a website, clicking links on it, entering text in a search box, viewing the results and so on. This is also referred to as "Web Interactions" or "surf sessions".

- *Reaction of the user:* apart from the consumption of the content, the reaction of the user (e.g. facial expression), or the impact of the content on the user is part of the user experience, as well.

An example of how these two aspects could be captured would be two cameras, one facing the screen of the user, recording the web interactions (it could also be replaced by a screen recording software), and the other one targeting at the face of the user, capturing his facial expressions and reactions in relation to the content that is being perceived. The outcome would be a split-screen video showing on the one side a video of the surf session of the user performing all kinds of actions in the web and on the other side the reactions of the user. The following image shows how a single frame [2] of such a video could look like:

[2] a single image of a video

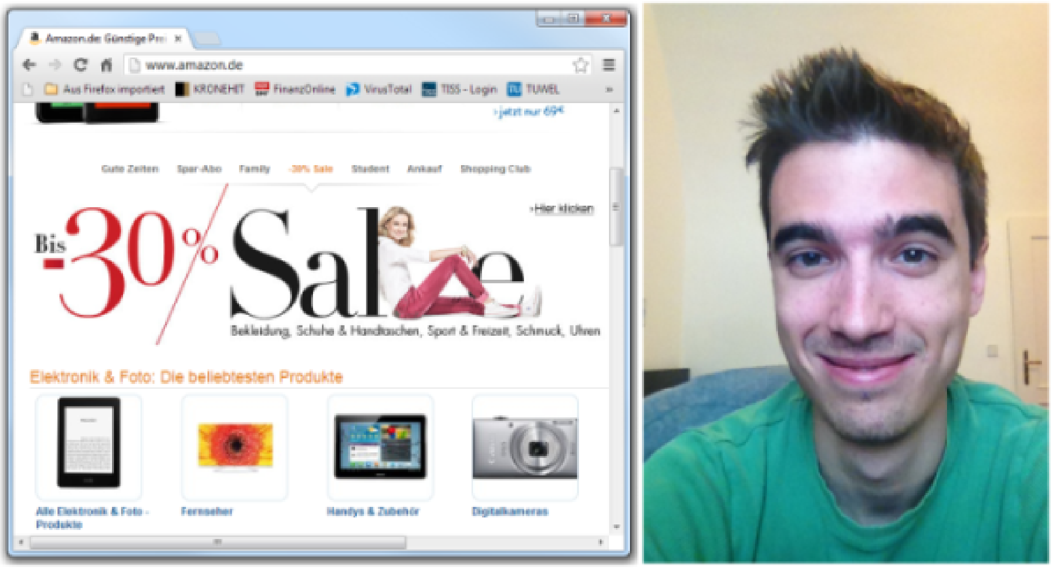


Figure 5.1: Illustration of a single frame of a camera based user experience video

The image shows a person's facial expression (on the right side) while surfing on the amazon[3] website (on the left side). A relation from the facial expression to the content currently viewed can be made.

Since it would be very hard to find a relevant amount of users willing to install cameras in their homes in order to take part in a study for web archiving, the second part covering the reaction of the user is omitted in the following. Instead, the approach that is shown will work without cameras and will only focus on capturing the surf actions of mobile users.

5.3.2 User session

Another important term that is used in the following sections is a *user session*. In general the term user session is often used in the Internet when talking about sessions of websites or locked parts of websites that require a login. Usually such websites have a predefined timeout after that the user session is closed when the respective user is not active any more. This means the user has then to log in again, if he was inactive (not interacting with the website) for this period of time. Within the project supporting this thesis, the term is used a bit differently. In the focus of tracking users while surfing in the world wide web, a user session is basically an amount of time the user is continuously surfing, having a defined start and also a defined end of the session, and every surf action that happened in between. The starting point and the end are not always easy to define. In the case of smartphone users, an ideal capture of a user session would be to start when the user is unlocking his phone and opening a browser and to end when he closes the browser or locks the phone again.

[3] `http://www.amazon.com`

5.3.3 Crowdsourcing

The term crowdsourcing describes a strategy that copes with complex or time-consuming problems or tasks with the power of the mass. Differently stated, many different people are working on the same problem in order to solve it in a reasonable amount of time. In [12] a crowd sourcing (CS) system is defined in the following way by Doan:

> A CS system enlists a crowd of users to explicitly collaborate to build a longlasting artifact that is beneficial to the whole community.

There are many known examples for projects based on crowd sourcing such as the wikipedia platform or the operating system linux. As also stated in [12], crowdsourcing systems mainly face four key challenges: (1) How to recruit contributors, (2) what they are able to do, (3) how to combine their contributions and (4) how to manage abuse? [12]

5.4 Available data from Advanced User Behaviour Analysis

This section will cover the relevant data that can be used for a web interaction archiving approach and that is available from *Advanced User Behaviour Analysis* more specifically from the *Mobile Viewport Tracking* system. For further details about *Advanced User Behaviour Analysis* and Mobile Viewport Tracking take a look at chapter 2 and 3.

Basically what is recorded with *Mobile Viewport Tracking* is every action that a smartphone user can perform during a user session, such as zooming, scrolling, rotating the phone, clicking links and so on. An action is called a “ViewportEntry”. Each single change of the viewport, clicks or key events trigger a new ViewportEntry. A ViewportEntry basically consists of the following information that is relevant for web interaction archiving:

- The exact date and time when the action occurred.
- The scrolling location of the viewport.
- The zoom level of the viewport.
- The URL of the website on which the action occurred.
- The type of the action such as e.g. zoom or scroll.
- The DOM path to the related element if its a click or key event.
- Multiple session tokens in order to connect the action to a specific user.

In principle the ViewportEntries cover all the information that is required in order to replicate the surf actions of the user. However, what is missing is a clear cut between different user sessions as defined in the previous section. Therefore, the concept of user sessions is used in the Mobile Viewport Tracking system, as well.

5.4.1 User Sessions

The term User Session describes all the actions of one specific user that he performed during a certain period of time. A User Session can be understood as one complete set of actions within one Internet surfing session. In technical terms a user session groups a specific number of ViewportEntries together, adding additional meta data such as:

- The exact start time of the session
- The exact end time of the user session
- The duration of the user session
- The count of actions performed during the session
- The URLs and domains visited during the session

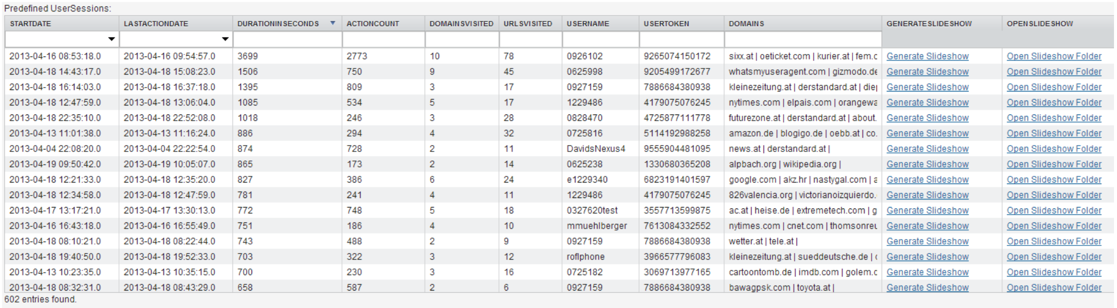
The user session implemented in the *Mobile Viewport Tracking* system tries to approximate the real user session as described in the previous section with the following mechanism:

- If the closing event of the browser on the mobile device can be captured, the user session ends, and the next time the same user triggers an action a new user session is created.
- If no closing event is received for five minutes, the user session also ends.

The assumption behind this mechanism is that typical smartphone users more likely have rather short user sessions. They look something up, perform some actions on the web and after that they close the browser again or lock the phone. Therefore the timeout is set to five minutes which is the maximum duration of a user session. The timeout is needed, because the browser closing or phone lock events cannot be captured on each phone. This depends on the operating system of the mobile device. Another problem is that the transmittal of the closing event is not reliable and can be delayed. It could happen that the event is only sent when the user performs the next action which could be hours after the last action occurred. This has technical reasons because the event cannot be transmitted during the closing process of the browser.

Implementation in the Mobile Viewport Tracking system

In the *Mobile Viewport Tracking* System an implementation was made, in order to be able to browse existing recorded user sessions, see the meta data and also have a prototypical graphical representation of the user session using the heatmap drawer. The heatmap drawer was modified in order to draw multiple images for one user session. The heatmap is split into chronological parts, which enables the possibility to watch the heatmap develop over the duration of the user session. The following screenshots illustrate the implementation of user sessions in the Mobile Viewport Tracking system:



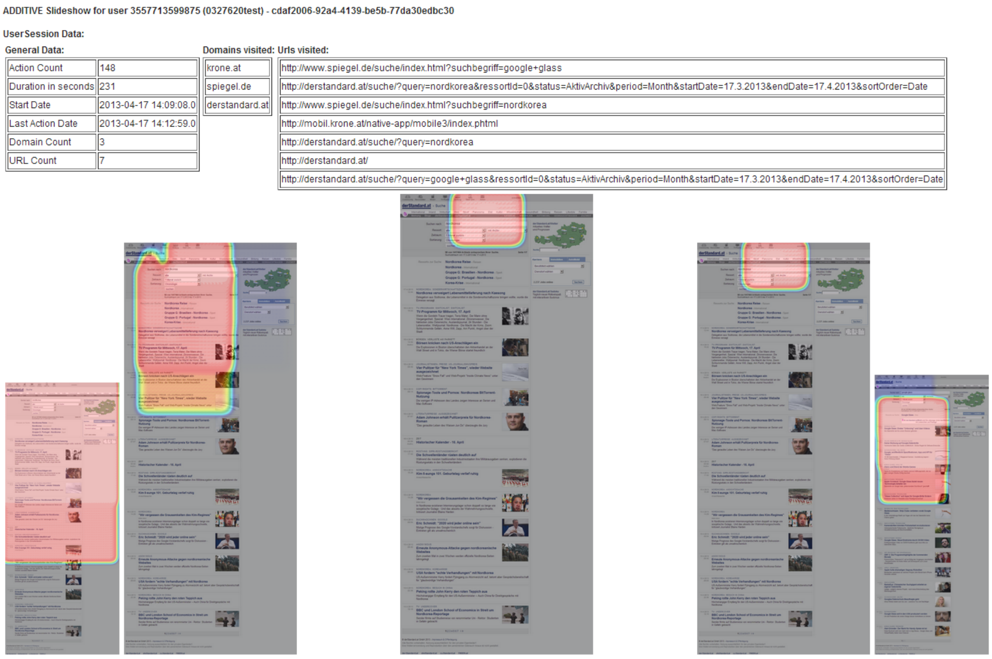
Predefined UserSessions:

STARTDATE	LASTACTIONDATE	DURATIONINSECONDS	ACTIONCOUNT	DOMAINSVISITED	URLSVISITED	USERNAME	USERTOKEN	DOMAINS	GENERATESLIDESHOW	OPENSLIDESHOW
2013-04-16 08:53:18.0	2013-04-16 09:54:57.0	3699	2773	10	78	0926102	9265074150172	sixx.at | oeticket.com | kurier.at | fem.c	Generate Slideshow	Open Slideshow Folder
2013-04-18 14:43:17.0	2013-04-18 15:08:23.0	1506	750	9	45	0625998	9205499172677	whatsmyuseragent.com | gizmodo.de	Generate Slideshow	Open Slideshow Folder
2013-04-18 16:14:03.0	2013-04-18 16:37:18.0	1395	809	3	17	0927159	7886684380938	kleinezeitung.at | derstandard.at | die	Generate Slideshow	Open Slideshow Folder
2013-04-18 12:47:59.0	2013-04-18 13:06:04.0	1085	534	5	17	1229486	4179075076245	nytimes.com | elpais.com | orangewa	Generate Slideshow	Open Slideshow Folder
2013-04-18 22:35:10.0	2013-04-18 22:52:08.0	1018	246	3	28	0828470	4725877111778	futurezone.at | derstandard.at | about.	Generate Slideshow	Open Slideshow Folder
2013-04-13 11:01:38.0	2013-04-13 11:16:24.0	886	294	4	32	0725816	5114192988258	amazon.de | blogigo.de | oebb.at | co.	Generate Slideshow	Open Slideshow Folder
2013-04-04 22:08:20.0	2013-04-04 22:22:54.0	874	728	2	11	DavidsNexus4	9555904481095	news.at | derstandard.at |	Generate Slideshow	Open Slideshow Folder
2013-04-19 09:50:42.0	2013-04-19 10:05:07.0	865	173	2	14	0625238	1330680365208	alpbach.org | wikipedia.org |	Generate Slideshow	Open Slideshow Folder
2013-04-18 12:21:33.0	2013-04-18 12:35:20.0	827	386	6	24	e1229340	6823191401597	google.com | akz.hr | nastygal.com | a	Generate Slideshow	Open Slideshow Folder
2013-04-18 12:34:58.0	2013-04-18 12:47:59.0	781	241	4	11	1229486	4179075076245	826valencia.org | victorianoizquierdo.	Generate Slideshow	Open Slideshow Folder
2013-04-17 13:17:21.0	2013-04-17 13:30:13.0	772	748	5	18	0327620test	3557713599875	ac.at | heise.de | extremetech.com | g	Generate Slideshow	Open Slideshow Folder
2013-04-16 16:43:18.0	2013-04-16 16:55:49.0	751	186	4	10	mmuehlberger	7613084332552	nytimes.com | cnet.com | thomsonreu	Generate Slideshow	Open Slideshow Folder
2013-04-18 08:10:21.0	2013-04-18 08:22:44.0	743	488	2	9	0927159	7886684380938	wetter.at | tele.at |	Generate Slideshow	Open Slideshow Folder
2013-04-18 19:40:50.0	2013-04-18 19:52:33.0	703	322	3	12	roflphone	3966577796083	kleinezeitung.at | sueddeutsche.de |	Generate Slideshow	Open Slideshow Folder
2013-04-13 10:23:35.0	2013-04-13 10:35:15.0	700	230	3	16	0725182	3069713977165	cartoontomb.de | imdb.com | golem.c	Generate Slideshow	Open Slideshow Folder
2013-04-18 08:32:31.0	2013-04-18 08:43:29.0	658	587	2	6	0927159	7886684380938	bawagpsk.com | toyota.at |	Generate Slideshow	Open Slideshow Folder

602 entries found.

Figure 5.2: Prototypical implementation of user session browser in the *Mobile Viewport Tracking* system

As far as figure 5.2 is concerned, the user sessions are displayed in a sortable table, offering basic meta data for fast identification in test scenarios. On the right side a user session can be generated and after that also opened by clicking on the "Open Slideshow Folder" link. The generation triggers the heatmap drawer that draws up to 30 images which represent the status of the heatmap at a specific time during the user session. As figure 5.3 demonstrates, when opening the user session, additional meta data such as all URLs and domains that were visited are displayed and also a slideshow of the heatmap pictures is offered. By clicking on the images, one can navigate through the user session going forwards and backwards in time. Figure 5.3 shows the open view of a user session:



ADDITIVE Slideshow for user 3557713599875 (0327620test) - cdaf2006-92a4-4139-be5b-77da30edbc30

UserSession Data:

General Data:

Action Count	148
Duration in seconds	231
Start Date	2013-04-17 14:09:08.0
Last Action Date	2013-04-17 14:12:59.0
Domain Count	3
URL Count	7

Domains visited:

krone.at
spiegel.de
derstandard.at

Urls visited:

http://www.spiegel.de/suche/index.html?suchbegriff=google+glass
http://derstandard.at/suche/?query=nordkorea&ressortId=0&status=AktivArchiv&period=Month&startDate=17.3.2013&endDate=17.4.2013&sortOrder=Date
http://www.spiegel.de/suche/index.html?suchbegriff=nordkorea
http://mobil.krone.at/native-app/mobile3/index.phtml
http://derstandard.at/suche/?query=nordkorea
http://derstandard.at/
http://derstandard.at/suche/?query=google+glass&ressortId=0&status=AktivArchiv&period=Month&startDate=17.3.2013&endDate=17.4.2013&sortOrder=Date

Figure 5.3: Prototypical visualisation of a user session in the *Mobile Viewport Tracking* system

5.5 The Web Interaction Archive

As the background of Digital Preservation and also Web Archiving was now explained thoroughly in chapter 4 and also an introduction to the idea of Web Interaction Archiving has been made at the beginning of chapter 5, this section now contains a detailed description about how such an user-centric Internet archive, based on web interactions could be designed and also implemented. The system that is described is called “The Web Interaction Archive”. The basic idea behind it is to have a centralised archive very similar to The Internet Archive, and to use a crowd sourcing approach in order to obtain data for it. The archive therefore is capable of receiving user centric data from numerous locations by providing a common and public available interface for contributions. The following graphic shows a possible architecture of the archive, which is described in more detail on the following page by explaining the components and the linkages between them.

5.5.1 Architecture

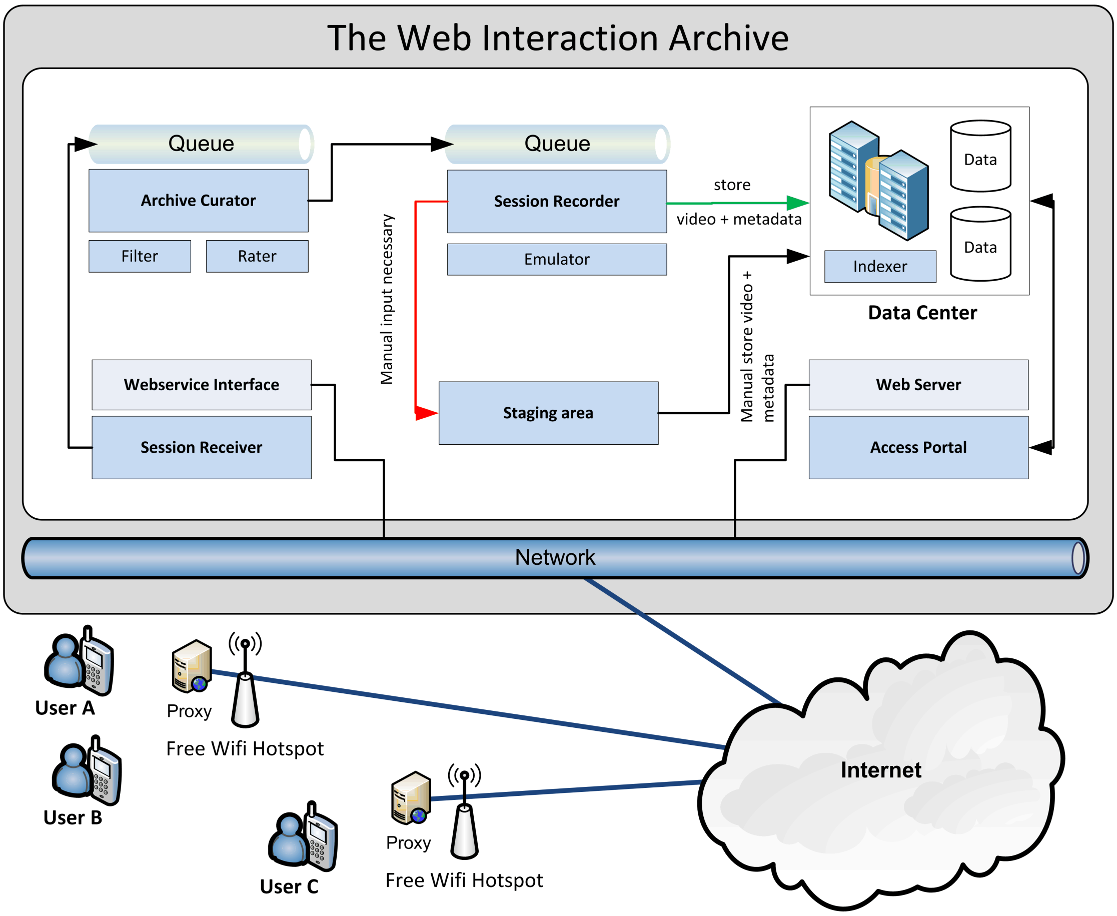


Figure 5.4: Components and architecture of the Web Interaction Archive

Figure 5.4 shows that the Web Interaction Archive basically consists of two main parts:

1. The centralised archive itself (grey rectangle), containing all the archived data, providing access via a portal and enabling contributions by a common available Web Service interface.
2. The contributors consisting of a tracking proxy server, the network attached to the proxy server and the users surfing within this network (shown at the bottom of the graphic).

In the following the components of the archive itself are described at first, and afterwards the design of the contribution system is shown.

5.5.2 Components of the archive

As shown in figure 5.4 the main components of the archive are:

- *Session Receiver:* The Session Receiver is mainly responsible for receiving data from contributors. The component consists of a common Web Service interface that allows to submit one or more user sessions in a predefined format very similar to the data model used in the *Mobile Viewport Tracking* system. Each submission has to contain at least one recorded user session as defined in section 5.4.1. In order to prevent abuse or flooding of the archive, credentials are provided for interested contributors. Before each submission an authentication has to be made, using a login method of the Web Service interface. This enables the providers of the archive to exclude contributors from the archive if they would violate the terms and conditions. If the Session Receiver receives a submission from an authenticated source, it hands the data over to the Archive Curator, where it is queued.

- *Archive Curator:* The Archive Curator is responsible for analyzing incoming submissions to the archive. It is the guard of the archive regarding content. There are many different reasons why a specific user session should not be stored within the archive. For example the user session could contain login procedures to private areas, or unwanted website content such as child pornography or Nazi related topics. The Archive Curator is the component that decides if a submitted user session will be recorded and stored within the archive or not. The decision is mainly based on a blacklisting approach combined with a semantic text analysis. Another criteria is the rating and also the coverage of the user session in order to decide if the submission is of interest for the archive. Assuming a high amount of contributors, it will not be possible to store all submitted user sessions. Therefore a selection has to be made. For some prominent websites a lot of submissions will occur. That is why a limit must be defined in order to restrict the maximum number of user sessions that are archived for one specific website for a specific period of time. The Archive Curator checks if the maximum number of user sessions is already exceeded for similar submissions and if that is the case, it is rated. A user session is considered similar if the coverage contains less than 25 percent new content in total over all web pages visited. The coverage defines which parts of a web page were visited or even read by the user. The concept is based on the heatmap data. If a certain area of a web page is weighted higher than a critical value, than this area is considered to be covered by this user session. If a newly contributed user session covers more than 25 percent new content that is not yet covered by existing user sessions, it is not ruled as similar and therefore added to the archive. The goal is to have a very high coverage of each website within the archive, by providing enough user sessions for this particular web page. If a similar user session already exists the user session is rated based on the duration of the session, the number of actions that were recorded and the date the session was recorded. If the rating is lower than the lowest rating of already archived similar user sessions, the submission is rejected. When the Archive Curator decides to accept the submission, the user session is forwarded to the Session Recorder where each single user session is queued again.

- *Session Recorder:* The Session Recorder is the most important component. Its purpose is to create a visual representation of the user session in the form of a archivable video. The video shall represent the original user session as accurately as possible as it took place. The Session Recorder relies on a web automation software in order to reproduce the user session in a browser. An example would be the Selenium WebDriver[4], which allows to automate different browsers using a simple programming interface that can be embedded into various programming languages, as browser tests. With the data that is available from the submissions, such as zoom levels, scrolling positions, actions such as clicks or key presses, the whole user session can be reproduced in a browser test. The video can be made with a screencasting software or component, that captures the screen while the automated browser test is running. Since each user session has to be replayed in real-time, the task of session recording could be very time consuming. In order to cope with a high amount of submissions, the component has to run on a powerful server that is capable of recording multiple sessions in parallel. Due to the fact that it would be possible that the user session which was submitted is missing data, is containing errors or the recording process failed for some reason, the session is forwarded into the staging area, from where it could be fixed manually. If the session was recorded successfully, the video is stored in the data center together with the available meta data where it gets indexed for later access.

- *Staging area:* The Staging area is used to collect user sessions that could not be processed correctly for some reason. The archive operators have the possibility to check submissions that were put into the Staging area and try to fix them or restart the recording process. They could also decide to delete single user sessions. In order to avoid flooding of the Staging area, the user sessions stored there are deleted automatically after a configurable period of time, if not processed manually.

- *Data center:* The Data Center is the heart of the Web Interaction Archive. It basically provides safe storage for all data that is put into the archive. Common technologies are used in order to cope with the challenges of Digital Preservation as already thoroughly described in chapter 4. The Indexer is responsible for indexing all the data and the available meta data in order to provide fast access.

[4] `http://docs.seleniumhq.org/projects/webdriver/`

- *Access portal:* The last component is the Access portal. This is a simple web application that can be used as user interface for searches and inquires on the archive. The search interface provides standard parameters such as time, website, topic or the originating country of a user session. The result page lists all the user sessions that match the inputs and gives the user the opportunity to look at the meta data of these user sessions and also to watch the recorded videos.

5.5.3 Tagging of user sessions

In order to provide easy and fast access to the archived user sessions, they have to be tagged or categorized somehow. A basic classification of user sessions can be done by using derived meta data such as:

- Domains and Websites visited during the session
- Originating country of user session / contributor
- Starting date of user session
- Duration of user session / average duration on one domain
- Mobile device used for surfing / operating system

Accessing users of the Web Interaction Archive could use a filter in order to search for recorded user sessions matching these arguments. However, this may not be enough. For inquiries where the desired scenario or website is already known this is well suited, but if someone would like to consume the Web Interaction Archive in terms of browsing without having predefined criteria in mind, a more detailed categorization e.g. based on the website topics would be needed. The idea is now to use existing web taxonomies in order to categorize user sessions and offer a topic browsing tree. In [33] such taxonomies and web page classification strategies are evaluated and described in detail. Those strategies mostly divide into subject classification, which tries to obtain the topic of a website, and functional classification, which tries to determine the type of a web page such as “corporate website”, “forum” etc. [33] For the tagging of user sessions in the Web Interaction Archive, both classification types seem interesting.

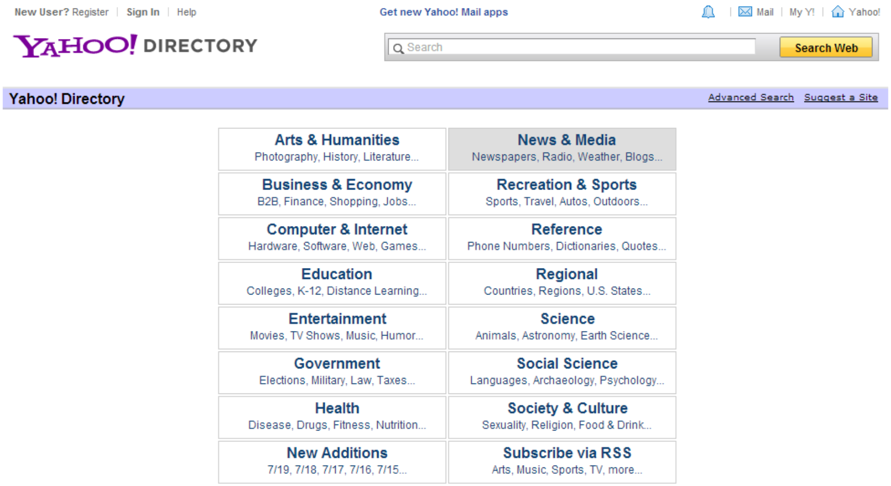
For the functional classification the following simple taxonomy is used:

- Published information (e.g. news page, weather, corporate web page etc.)
- Web shop
- Search engine/comparison portal
- Wiki
- Forum
- Blog

- Social Network/Communication
- Crowd contributed content (e.g. youtube)
- Games and other entertainment
- Course/Learning portal
- Other

The assignment of the functional category has to be done manually by the users or the archive operators, if no classification is already existing for a given website.

For subject classification existing web taxonomies can be used. Famous web directories that categorize web pages based on their topic are Yahoo Directories[5] or the Open Directory Project[6] which is also known as DMOZ. The topological and hierarchical structuring of web sites used in this directories could also be applied to the recorded user sessions of the Web Interaction Archive based on the visited web pages. The following figure 5.5 shows the top level structuring of Yahoo Directories.



Arts & Humanities Photography, History, Literature...	News & Media Newspapers, Radio, Weather, Blogs...
Business & Economy B2B, Finance, Shopping, Jobs...	Recreation & Sports Sports, Travel, Autos, Outdoors...
Computer & Internet Hardware, Software, Web, Games...	Reference Phone Numbers, Dictionaries, Quotes...
Education Colleges, K-12, Distance Learning...	Regional Countries, Regions, U.S. States...
Entertainment Movies, TV Shows, Music, Humor...	Science Animals, Astronomy, Earth Science...
Government Elections, Military, Law, Taxes...	Social Science Languages, Archaeology, Psychology...
Health Disease, Drugs, Fitness, Nutrition...	Society & Culture Sexuality, Religion, Food & Drink...
New Additions 7/19, 7/18, 7/17, 7/16, 7/15...	Subscribe via RSS Arts, Music, Sports, TV, more...

Figure 5.5: Yahoo Directories - `http://dir.yahoo.com/`, accessed at July 2013

In order to classify user sessions, the visited web pages have to be checked against an existing web taxonomy database, and could be classified automatically. A cooperation with one of the providers would be needed, in order to have access to the database e.g. via a web service interface.

Using this, the basic meta data model that was described at the beginning of this section has to be extended by two attributes:

[5] http://dir.yahoo.com/

[6] http://www.dmoz.org/

- Topics: for each visited web page of a user session a topic based on an existing web taxonomy is assigned.
- Functional types: for each visited web page of a user session one of the functional types outlined above is assigned.

The extended meta data model allows to build a hierarchical navigation structure on the access portal of the Web Interaction Archive, based on functional and subject classifications very similar to the web directories that were mentioned within this section.

5.5.4 The contribution system

This section now describes the architecture of an archive contributor. The following graphic illustrates the components needed for the contribution system:

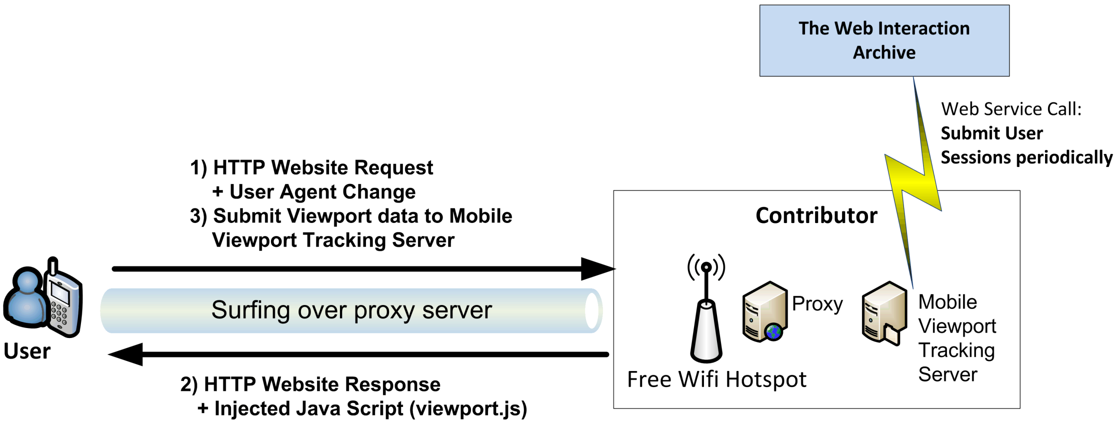

The Web Interaction Archive
Web Service Call:
Submit User Sessions periodically
1) HTTP Website Request
+ User Agent Change
3) Submit Viewport data to Mobile Viewport Tracking Server
Contributor
Surfing over proxy server
User
Proxy
Mobile Viewport Tracking Server
Free Wifi Hotspot
2) HTTP Website Response
+ Injected Java Script (viewport.js)

Figure 5.6: Components and architecture of the contribution system

The contribution system basically consist of the following components:

- *Proxy server:* responsible for the injection of the *Mobile Viewport Tracking* script. At first a HTTP request is sent by a connected device. Then the proxy server changes the user agent in order to avoid mobile web pages. The response is returned with the *Mobile Viewport Tracking* script already injected. After that the device is reporting each viewport change to the *Mobile Viewport Tracking* server.
- *Mobile Viewport Tracking Server:* stores the tracking data that is sent by the mobile devices and constantly submits finished user sessions to the Web Interaction Archive.
- *Hotspot:* a wireless hotspot, that provides Internet access over the proxy server and is placed within the same network as the *Mobile Viewport Tracking* server.
- *Users:* the users that are connected to the hotspot with their devices and therefore tracked by the *Mobile Viewport Tracking* Server.

The whole contribution system could be designed as one software package, containing all the necessary configuration for easy setup and immediate usage. Each provider of a public wireless hotspot would be a possible contributor to the Web Interaction Archive. He would just need one server and the contribution software, which could be downloaded for free from the Web Interaction Archive Portal. Another possibility would be to cooperate with wireless hotspot hardware vendors and sell the contribution system together with the hotspot as a complete ready-to-use package.

5.5.5 Problems and Challenges

The last section showed how an Internet archive based on user-centric data could be implemented and what infrastructure would be needed. Although the idea is realisable, such a project is facing some technical and organisational challenges that have not been mentioned yet. These problems and possible solutions will be discussed in this section.

Organisational challenges

- *Funding:* The main problem that arises even before the project can be started, is the question of funding. As the architecture of the Web Interaction Archive, as shown in figure 5.4, implies, and a lot of technical resources such as servers, storage or Internet connection is needed the question of financing has to be clarified first. Another huge cost factor would be the development of the different components needed for the archive. So how can a project like this be financed? The simple answer is: an investor is needed. Either a private investor, some governmental department or Web Archiving initiatives would have to invest in order to develop the archive. The operating costs of the Web Interaction Archive could be covered e.g. with advertisements on the archive's portal web page, once the archive is running, and established within the Web Archiving community. Another possibility would be to start it as an open source project in order to prove the concept but this would also be a huge organisational challenge.

- *Motivation of contributors:* The second challenge that arises is to get enough contributors for the archive. The Web Interaction Archive relies on contributors in order to have a reasonable amount of data within the archive. A possibility to motivate potential contributors would be e.g. the selling of complete ready-to-use wireless hotpots for attractive prices as already mentioned shortly in section 5.5.4. Another possibility would be to offer to advertise for the locations and businesses (e.g. restaurants) that contribute to the archive. However, this would require a high degree of popularity of the archive in order to attract possible contributors.

Technical challenges

- *Session Recording:* The main technical challenge is the implementation of the *Session Recorder* component. The recorded sessions should be as accurate as possible, which means that the resulting video should represent the session as it originally took place on the mobile device. The variety of smartphone models, versions and operating systems makes it hard to reproduce a user session precisely. Current web browser automation tools such as *Selenium*, which has already been explained, have limited capabilities regarding mobile devices, more specifically mobile browsers. At the moment, a *Selenium* component is available for Android which is called *AndroidDriver* [7] that is able to run browser tests on mobile devices or mobile device emulators for Android. However, the current version is not capable of replaying zooms, which is a major drawback because the zooms are an important part of the recorded user sessions. For iPhones the recording of user sessions would be even harder, because no similar component is available for free. A possible solution would be to use the standard *Selenium WebDriver* and resize the browser window to the phone's resolution. If such an approach is producing results accurately enough, that has to be evaluated.

- *Detection of similar user sessions:* Another problem is the comparison between user sessions that are added to the Web Interaction Archive, and those already existing in the archive. Not all user sessions that are added can be archived basically because of the amount of resources needed and also because there is no need to have many thousand user sessions of one specific web page. The concept of coverage is introduced in section 5.5.2 where the Archive Curator component of the Web Interaction Archive is described. The coverage is a possibility to distinguish between user sessions that took place on the same website. If new parts of a web page are covered in a newly published user session, it makes sense to store this session in the archive. But what about user sessions that cover the same parts of a web page? How can they be compared? In this case a possible way would be to analyze the sequence of actions that were performed in that user session. The following quantitative indicators can be derived from the interactions:

  - Percentage of zooming-in and double-tap actions: strong indicators that content was read
  - Number of clicks: interaction with the web page e.g. displaying/hiding additional content
  - Time resided on specific parts: strong indicator that content was read
  - Viewport rotations performed on the website: different user behaviour

  Another possible way would be to make use of the heatmap data. In order to compare a user session on a specific web page, the weighting matrix that is needed to generate a heatmap can be used. A distance value between the weighting matrices of the two user sessions that need to be compared, is calculated. A decision about the similarity can then be made upon this value.

- *Consolidation of user sessions:* Very similar to the challenge of the distinction between user sessions, is the idea of consolidating user sessions. This breaks with the original idea of the Web Interaction Archive, that user sessions are archived and replayed exactly in the way they happened on the users mobile device. Nevertheless the concept can be

[7] `http://code.google.com/p/selenium/wiki/AndroidDriver`

interesting for the archive. The idea behind this is to merge various user sessions that took place on the very same web page together to one user session which is representing the summarized surfing behaviour of all users. This is a powerful and fast way to look up user behaviour of specific web pages. In order to decide which user sessions are grouped together and how, again the coverage can be used. The goal is to have a as high as possible coverage in the resulting user session. Possible problems and challenges that occur when trying to merge user sessions are:

  - The content of web pages changes, so only user sessions that occurred within a short time frame can be merged together.
  - The interactions have to be re-arranged into a logical order that allows fluent replaying of the resulting user session.
  - Similar interactions have to be skipped in order to avoid redundancy in the user session.
  - Only actions on one specific web page can be merged. All actions that trigger a change of the URL have to be skipped.

- *Tagging of user sessions*: In order to provide an easy way to find user sessions stored within the archive, the user sessions have to be categorized or classified in a meaningful way. This challenge was already thoroughly described in section 5.5.3.

- *Rising number of mobile web pages:* The rising number of mobile web pages creates a problem for *Mobile Viewport Tracking*. The approach of tracking mobile users' surfing behaviour relies on standard desktop pages, since mobile pages would limit the amount of actions that could be tracked. Most likely no zooming will be required, which reduces the possibility of analysis. In a proxy based tracking system, as used with the Web Interaction Archive, this problem can be solved by changing the user agent of each request in order to avoid mobile pages.

CHAPTER 6

Website Optimisation

6.1 Introduction

In the state of the art chapter (4) the term *Website Optimisation* was introduced and explained. This chapter shall provide a new way to optimise web pages according to mobile users' needs based on the techniques used by the Web Interaction Archive, which was introduced in chapter 5.

Existing approaches are mainly focused on desktop users, and many websites are only optimised for big screen resolutions. Mobile versions of websites often lack important functions or content and do not offer the user the same experience as desktop versions. The number of smartphone users is rising and mobile devices get more and more important, especially in terms of surfing the web on the go. The following sections show on the one hand, how important content can be identified on web pages using *Mobile Viewport Tracking* and on the other hand, how the user-centric view, introduced with the Web Interaction Archive, can provide useful information for improving websites in terms of usability and efficiency. At the end of this chapter a comparison with existing tools is made and the economic potential of such an approach will be outlined.

6.2 Important Content

The identification of important content is a major task when thinking about Website Optimisation. Nearly every website in the World Wide Web has parts that are not often visited at by users. This can have a variety of reasons, such as:

- *Complex navigation path:* Most of the users simply don't find the information, because the navigation path to the respective part of the website is too complex.

- *Uninteresting content:* The content on this part is not of interest to the majority of the users surfing on the website.

- *Not optimised for user groups:* The content is not optimised for the major user groups of the website e.g. mobile users. If the user group is not able to view the content correctly, it is not interested any more.

There could be many other reasons for why specific content is not accepted and utilised by users. Many businesses strongly rely on the usability and attractiveness of their website. If they do not constantly adapt their web appearances to their customers' needs, they will most likely lose money due to less closed deals. Therefore, the identification of important content as well as unimportant content, that can be removed or has to be re-arranged, is the key to satisfy users.

6.2.1 Detection

The question is how to detect this important and unimportant content? A first possibility was introduced in chapter 3 in section 3.5.3. The concept of visualising the data that is retrieved in *Mobile Viewport Tracking* by the usage of heatmaps is already a first step towards identifying important content. Figure 6.1 illustrates the capabilities of the *Mobile Viewport Tracking* approach using heatmaps. As already explained in chapter 3, the darker spots (red) indicate that more users visited specifically that area and most likely also read the content within this part.

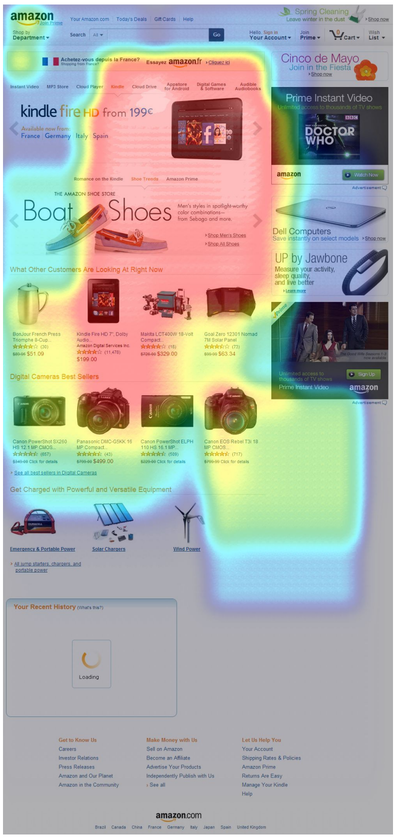


Figure 6.1: Heatmap created from the amazon.com page using *Mobile Viewport Tracking* data

Although *Mobile Viewport Tracking* mainly focuses on mobile devices, it could be assumed that due to the rising number of smartphone users the conclusions made from the analysis of those mobile users could also be applied to other user groups, such as standard desktop or laptop users. This makes *Mobile Viewport Tracking* a good choice for Website Optimisation. The standard functionality of *Mobile Viewport Tracking* is focused on the aggregation of multiple user sessions that occurred on one page in order to identify the important parts. The question is why not to look at it from a single user's perspective? The next section explains an extension to *Mobile Viewport Tracking* in order to visualise single user sessions using heatmaps.

User session visualisation using heatmaps

The idea is very similar to the Web Interaction Archive which also focuses on the user experience. In order to take the user's view into account, the *Mobile Viewport Tracking* system was extended by a new functionality that enables the analysis of specific user sessions. The implementation has already been introduced in the last chapter in section 5.4.1. It offers the possibility to look at the data from the user's perspective and analyse specific user sessions, making also use of the heatmap drawer for visualisation. For each user session a series of heatmap images can be generated in order to illustrate what the user actually did. The system provides two ways of generating the heatmap images:

1. *Additive:* The *additive* mode allows to create multiple heatmap images per user session that constantly add up to the final heatmap representing the whole user session containing all actions. This enables to analyse how the heatmap develops during the user session.

2. *Current actions only:* The *current actions only* mode also creates multiple heatmap images per user session, but for each new image only the actions that occurred within the current timeslot are displayed.

Both functions can be useful for analysing the user session. The *additive* mode for example would be like a heatmap with a timeline slider. Going forwards and backwards within the slideshow of heatmap images illustrates how the heatmap looks like at a specific point of time during the user session. This can support the interpretation of the final heatmap. The *current actions only* mode can be used for visualising single actions of a user during the user session. Going forwards and backwards within the slideshow in this case illustrates what the user actually did at a specific point of time. This visualisation is very close to a video that is replaying the user session. Each heatmap image could be seen as a frame of the video. The next two figures show an example of a short user session on the GMX [1] website, once visualised with the *additive* mode (figure 6.2) and another time using the *current actions only* mode (figure 6.3):

[1] http://www.gmx.at

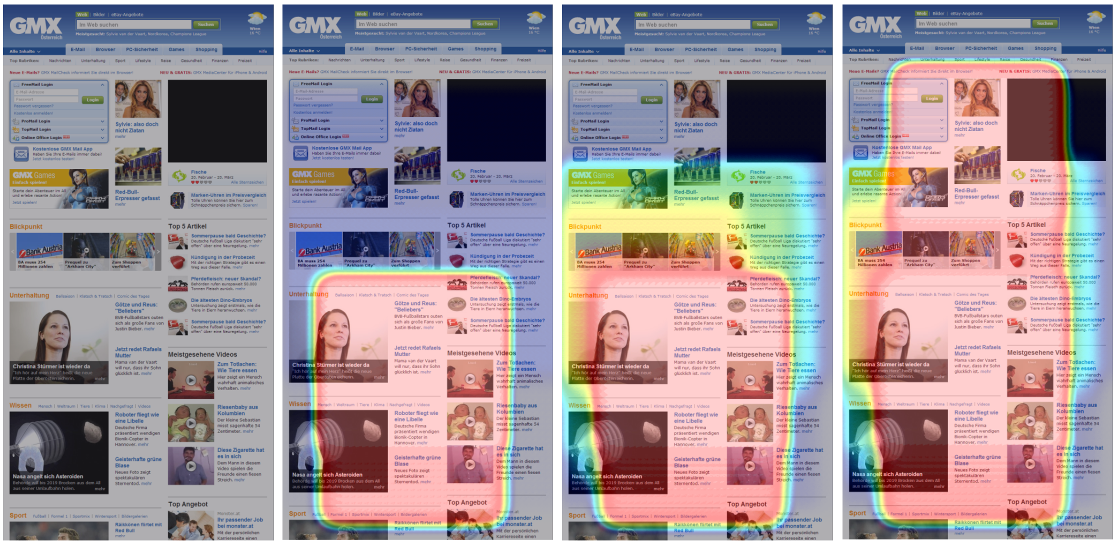


Figure 6.2: Heatmaps generated for user session with *additive* mode

Figure 6.2 shows the *additive* mode. With each picture that is taken during the user session the newest actions are added up until the final heatmap has developed (last picture).

Figure 6.3 shows the *current actions only* mode. In this case, each picture basically visualises the current actions that happened at this point of time. This helps to understand the sequence of actions and in which order they occurred.

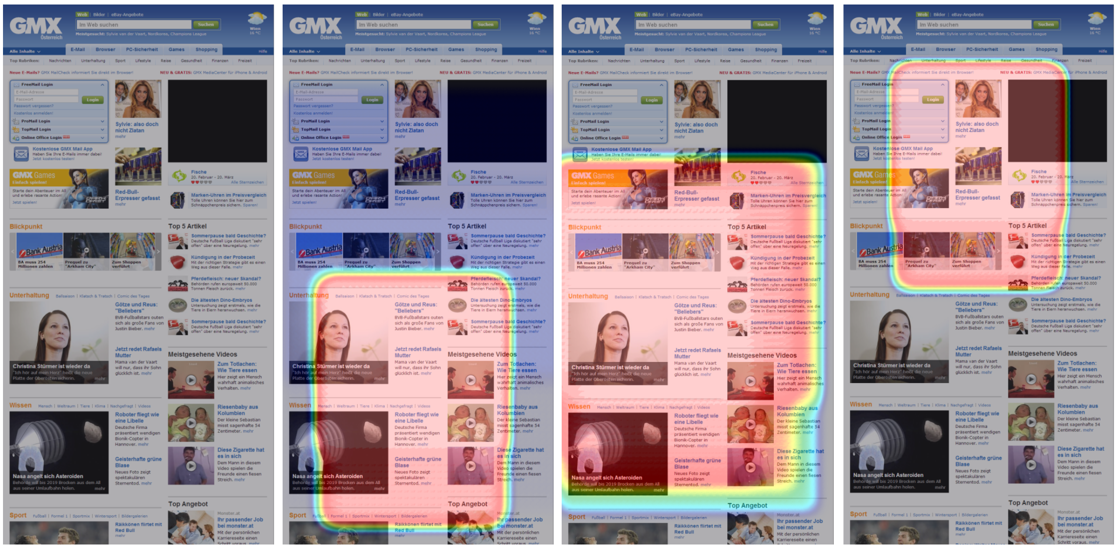


Figure 6.3: Heatmaps generated for user session with *current actions only* mode

User session videos

The visualisation of user sessions using heatmaps, as shown in the last section, is a good starting point for understanding what single user's did on specific websites. An even better way to reproduce the users' actions would be a video showing the user session in a similar way as it happened on the users' mobile device. The idea behind this is the same as already introduced with the Web Interaction Archive. From the components of the archive, which were introduced in section 5.5.2, only the *Session Recorder* would be needed in combination with the classic *Mobile Viewport Tracking* system. The generated videos could be embedded in the user session browser (as introduced in figure 5.2) in the form of links in order to provide easy access. Website owners could view this videos on a regular basis in order to analyse the user behaviour and draw conclusions from it about possible optimisations of the web appearance. The videos could also help with the decision of what information and which functions should be available in a mobile version of the website. The following image illustrates how such a video could look like by showing a series of single screenshots that were taken during a user session which took place on the amazon website:

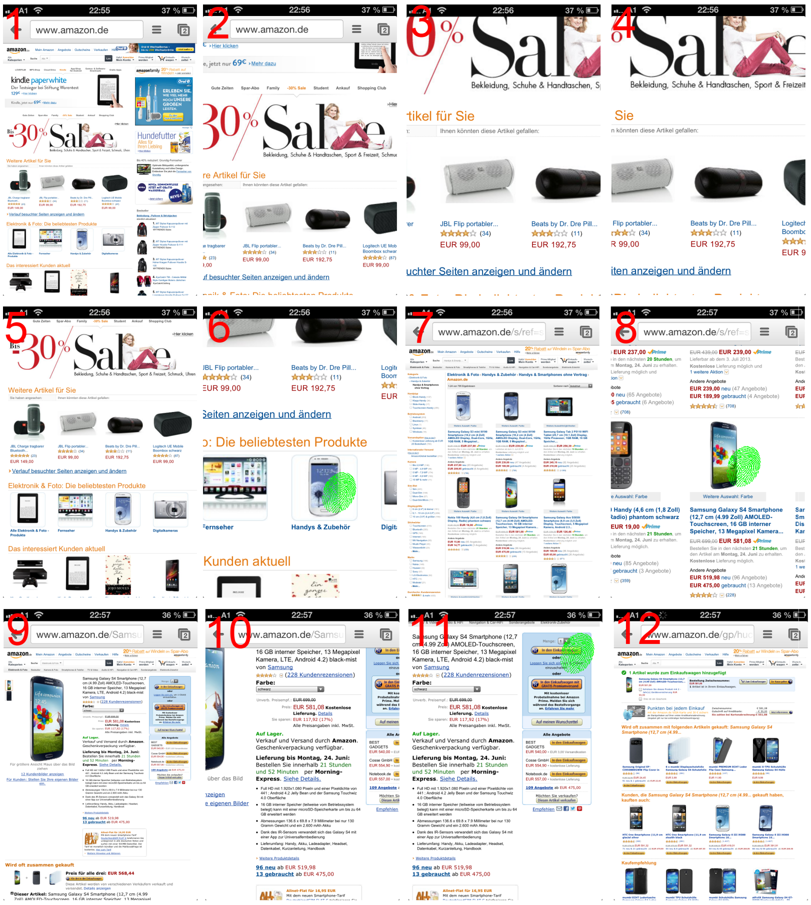


Figure 6.4: Single screenshots taken on a iPhone during a surf session on the amazon web page

The single screenshots within the image are numbered beginning at the top left corner from one to twelve in order to understand the sequence of the user session. The green fingerprint indicates a touch event on the respective area. Figure 6.4 shows a user session that is starting on the root page of amazon.de. The user then zooms-in a little and looks at the two portable speakers which can be seen on screenshot number 3. After that a minor scroll to the right is made (screenshot number 4) and then a zoom-out (screenshot number 5). Then the user zooms-in on the mobile phone section and touches the image (screenshot number 6). A new page is opened showing a list of smartphones (screenshot number 7). The user zooms-in on the product "Samsung Galaxy S4" and touches the image (screenshot number 8). Again a new page is opened showing the product details (screenshot number 9). After that the user zooms-in a bit, most likely reading the product information (screenshot number 10). Then he scrolls to the top right corner and touches the "Add to shopping cart" button (screenshot number 11). The last screenshot shows that the product was added to the shopping cart.

If videos like this one were available for all users visiting a certain web page, website owners would have a good chance to understand the users' needs and also the problems and difficulties during surfing on the website. A possible drawback of this solution is that the system only provides the recorded user sessions, but the the analysis and interpretation has to be done manually.

6.3 Comparison to other tools

Comparing the solutions introduced in section 6.2.1 with the tools that were mentioned in chapter 4, all approaches that have been shown are supporting the optimisation of web pages. In order to choose a specific tool, the scenario in which it should be applied and also the desired outcome has to be taken into account. While Google Analytics provides a lot of useful quantitative data about website visitors, it lacks of a meaningful analysis of mobile users. Clicktale also uses a heatmap based approach but the focus on Mouse Tracking also excludes mobile users. With the solutions discussed in section 6.2.1 both an analysis focusing on the users perspective as well as on mobile users specifically can be achieved.

6.4 Economic potential

The economic potential of Website Optimisation tools is very high. Many businesses having a web appearance are relying on such tools in order to be able to react fast on user behaviour changes and different needs. Google Analytics is very well established in this market segment, but a lot of similar tools are available. The growing rate of smartphone users creates an opportunity for approaches as *Mobile Viewport Tracking* to enter this market segment and compete with tools as Google Analytics. A possible problem for the *Mobile Viewport Tracking* approach is the rising number of mobile websites, which in fact limits the analysing capabilities of the system. This development has to be observed thoroughly when a decision to enter the market is made.

CHAPTER 7

Evaluation

7.1 Introduction

This chapter provides a short overview about the evaluations that were made during the project and also this master thesis. The section 7.2 covers evaluations in general, whereas section 7.3 explains an experiment that was made on the Vienna University of Technology.

7.2 Evaluation

The general strategy for evaluation was to have an iterative process of design, implementation and evaluation. This process was repeated in cycles during the *Mobile Viewport Tracking* project in order to constantly test the newest developments. At major milestones when new features reached a stable version, the implementations were presented at the DBAI institute at Vienna University of Technology within the working group in order to get feedback for further developments and improvements. The implementations that were made for the common project were performed within the team with the goal was to have a stable base of the *Mobile Viewport Tracking* system that everybody could use for the purpose of his specific thesis.

Regarding this thesis the evaluation was focused on the *SessionRecorder* component that has been introduced with the Web Interaction Archive in section 5.5.2. The critical component of the Web Interaction Archive was evaluated using the Selenium WebDriver [1] and also the Selenium AndroidDriver [2] which is a new version of Selenium running on Android phones or Android device emulators. Both versions allowed to replay user sessions to a certain degree but with drawbacks. The major problem was to replicate the zooms that were performed by the user. On a real smartphone a user zooms with a finger gesture using two fingers and moves them into opposite directions on the smartphone's screen. This gesture cannot be triggered at the moment by Selenium AndroidDriver. On a standard browser, using Selenium Webdriver, the zoom level

[1] http://seleniumhq.org/projects/webdriver/

[2] http://code.google.com/p/selenium/wiki/AndroidDriver

can only be adjusted gradually. No exact zoom levels can be replicated by code. The conclusion that was made is that Selenium is not capable of replaying user sessions as recorded by *Mobile Viewport Tracking* at the moment, but the AndroidDriver is still in development and it could be possible that the zooming gesture is supported in future versions. An alternative would be to use JavaScript for replaying by highlighting the area where the smartphone's viewport currently resides, and blurring the rest of the web page in order to visualise the sequence of actions of the user session. This approach is a feasible alternative to a native replay on a device or device emulator such as with the Selenium AndroidDriver, but it has not been further evaluated during this thesis.

7.3 Studies

7.3.1 Introduction

In order to test the *Mobile Viewport Tracking* system and get a reasonable amount of data for further analysis and evaluation, an experiment was made during the summer term of the course 'Applied Web Data Extraction and Integration' at the Vienna University of Technology. An optional task was given as homework that contained instructions for using the *Mobile Viewport Tracking system*. A more detailed description of the task follows in section 7.3.2. 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.

7.3.2 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 section A.2.1)] 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 [3]
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 websites 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 on different (non-SSL) websites.
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 choose, the 3 newspaper sites you did use, comparison between these 3 sites regarding usability, how fast you found 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.

7.3.3 Results

The exercise was very successful and the participating students contributed a lot of data to the database.

[3] `http://heatmaps.noip.org/MVTRegistration`

- 14 students participated in the exercise
- About 15000 actions were triggered during the surf sessions
- The total surfing time of all students was about 8 hours
- About 100 different domains have been visited

The data that was produced during the experiment gave the opportunity to evaluate the *Mobile Viewport Tracking* system regarding usability, plausibility of the resulting heatmaps and also the resources that are needed in order to operate the system. A detailed evaluation of the *Mobile Viewport Tracking* system is made in the thesis "Case Studies, Automatic Reporting, Evaluation and Optimization of Usability using *Advanced User Behaviour Analysis*".

Concerning the creation of the Web Interaction Archive, the experiment in fact proved that a reasonable amount of data could be retrieved by tracking random user traffic. The scenario that was chosen for the exercise was perfectly suited for an evaluation as it is very similar to track users in public wireless networks. The low number of 14 participants already resulted in over 150 user sessions that were created covering over 100 different domains. The average surf duration for each participant was about 35 minutes. This could be compared to 35 minutes of tracking within a wireless hotspot with 14 active users on average. The Web Interaction Archive would gather more than enough data when having a dozen of well frequented public hotspots as contributors. When planning the architecture of the archive, the number of contributors that is expected and also the number of user sessions that is intended to be stored per domain has to be taken into account when thinking about resources in order to cope with this huge amount of data.

Concerning evaluating Website Optimisation based on user-centric data, the experiment did not provide enough meaningful data for single web pages. In order to achieve this, the participants would have to be instructed to surf on a specific page only. Most likely also a higher number of participants would have been needed. A meaningful evaluation could only be done by embedding the system within a real website for a specific period of time and analyse the results.

CHAPTER 8

Conclusion

This thesis introduced a new and innovative way to track users surfing the World Wide Web, visualising the recorded results in different forms such as heatmaps, with the goal to improve websites according to specific user groups' needs, and also in order to suggest a new approach for the discipline of Web Archiving which intends to preserve the history of the web. In chapter 2 the general term of *Advanced User Behaviour Analysis* was introduced, describing disciplines that focus on user behaviour analysis using sophisticated technical solutions and measurement techniques. Within this context, concrete solutions such as Mouse Tracking and Eye Tracking systems were explained as they can be assigned to this term. Also common web analytics solutions were mentioned such as Google Analytics, but excluded from the term of *Advanced User Behaviour Analysis* as they only focus on the path the users followed on a website and lack information about the viewing behaviour on the pages itself. At the end of chapter 2 also the concept of heatmaps was explained, as it defines the main strategy for visualising recorded user data within this project. With chapter 3 a new approach for *Advanced User Behaviour Analysis* was introduced, that represents the common part of the project work performed during these theses. The solution is called *Mobile Viewport Tracking* and implements the idea of making use of the main limitations of smartphones, the small screen size, in order to track user behaviour during surf sessions in the World Wide Web. The system tracks all kind of actions that can be performed on a smartphone during surfing the web, such as scrolls, zooms, clicks, double taps, rotations, key presses etc., by injecting a client-side script on the website the user is visiting. The script is capable of recording all of these actions, and periodically submits the data to a Restful Web Service which stores it in a database. It can be injected by using two different ways. Firstly by embedding it directly into a website that a website owner wants to have tracked. In this case the owner of the website would have to decide to use the tracking system. The second possibility is to use a proxy server that injects the script in every web page a user requests. In this case the proxy has to either be configured manually by the users on their smartphone, or it has to be an invisible proxy that could be used e.g. on a public wireless hotspot. After storing the data within the database, they can be visualised using heatmaps. The heatmap basically uses a screenshot of the website the user did surf on as background image, and creates an overlay that shows the intensity of data, in this case the areas which were viewed more often than the others. In order to be able to create such a heatmap, a weighting matrix is used to define the weightings of each spot on the website based on the recorded data. Every value of the weighting matrix is projected on a colour gradient from very high/hot (white or red), to very low/cold (blue). The weightings are defined by using different algorithms to determine the importance of a specific area. Factors such as the time a user remains on a certain area and also the zoom level, which is an indicator that a specific area was read, are taken into account. Another important factor for analysing the recorded data is the identification of users. The ability to pin certain actions to a specific user is the key for interpreting the data that is gathered by *Mobile Viewport Tracking*. Therefore, the tracking of users is made on several levels, mainly using cookies. When using a proxy based approach, it is even possible to track one user over multiple websites and record complete user sessions.

After explaining the basic concepts of *Advanced User Behaviour Analysis* and *Mobile Viewport Tracking*, some research work was made concerning Digital Preservation, Web Archiving and Website Optimisation in chapter 4. The term Digital Preservation was introduced at first, as it is the basic concept which also applies to Web Archiving. The general problem of Digital Preservation was explained and also various strategies and known challenges and difficulties were outlined. A more detailed analysis was made in section 4.3 on Web Archiving. The reasons and also the benefits for archiving the web were outlined. Common strategies and approaches were described in detail, also based on examples such as The Internet Archive. In the end, various different projects that focus on the preservation of the history of the web were discussed and compared. The last topic that was mentioned in chapter 4 was Website Optimisation. In this section the basic idea of Website Optimisation was explained and the goals, target groups and existing tools were described.

After the research chapter, chapter 5 then introduced a completely new approach for archiving the web. In this chapter a connection was made from classic Web Archiving to a new type of Web Archiving, using the data that is collected by *Advanced User Behaviour* Analysis, more specifically from the *Mobile Viewport Tracking system*. The approach is called *Web Interaction Archiving*. The main idea of Web Interaction Archiving is not to preserve the Internet itself, but to preserve the user experience of people surfing in the web. The assumption behind that is that the history of the World Wide Web does not only consist of the content of websites, but also of the users' interaction with the content. The way of how the content is actually perceived by people using different devices, inputs or navigation techniques, is also a very interesting and therefore an important part of our history. The idea is to use the crowd represented by users utilizing mobile devices, such as smartphones in combination with the technology of *Mobile Viewport Tracking*, in order to create a user-centric Internet archive. The approach refers to a known method to cope with complex or time consuming problems which is called "crowd sourcing". The archive is structured in a similar way as *The Web Interaction Archive* and receives its data from contributors. The possible contributors are basically providers of free wireless hotspots that use an invisible proxy server in combination with the *Mobile Viewport Tracking* system, in order to track user sessions and submit them regularly to the Web Interaction Archive. The archive reproduces the recorded user sessions in the form of videos, showing the actual interactions that happened during the session. Chapter 6 tried to explain how the approach that was introduced in chapter 5, could also be applied for Website Optimisation. The method of taking a user-centric view into consideration when thinking about optimising web pages, is compared to already mentioned approaches such as Google Analytics or Clicktale. Chapter 7 covered evaluations made during the project and also this thesis. It also described an experiment that was made at the Vienna University of Technology in order to test and evaluate the *Mobile Viewport Tracking* system.

Future work that was not covered within this thesis and the project could be the implementation of a prototype for the Web Interaction Archive. Of course lots of resources would be needed in order to perform a serious evaluation concerning feasibility. The problem of recording and replaying user sessions could not be fully solved within this thesis. Concerning this topic, also further research and evaluation is needed in order to provide a way of creating accurate replications of actual user sessions. The *Mobile Viewport Tracking* system also requires further developments, mainly for providing an easy to use and powerful analysis application which could compete against reporting tools such as Google Analytics. Another challenge is to cope with existing mobile websites. A strategy has to be found of how the tracking and analysis of mobile websites could be improved, since the number of mobile web pages is rising.

Although the preservation of the World Wide Web seems to be a necessity for mankind, as it has already become an important part of our history, the implementation of a uniform solution still remains a problem. The question is, whether such a solution is even feasible? It seems that the archiving of the web is a task that is too complex and time-consuming for one institution. It is likely that it is even too complex to be coped by using one single approach or solution. The web is a global network, therefore also the preservation of it is a task that has to be solved globally. In order to succeed with it, many people, organisations, institutions, countries and also investors have to work together and join the forces. This thesis gave an overview of the challenges, strategies, problems and current approaches for Digital Preservation and Web Archiving. It even introduced a completely new approach that could cover a part of the Web Archiving process, and would fit into a number of different approaches in order to deal with this task. Although, many institutions are working on the problem, constantly developing new solutions, the future of Web Archiving is uncertain. In the end a partial solution, covering specific artifacts or areas of the web, will be more feasible than a holistic approach. Concerning Website Optimisation, it has to be said that this is also a task that gains more and more importance these days. The revolution of the smartphones and tablets was only the start of a new era that breaks the borders of classic Internet computing. A person is no longer bound to a desktop computer when surfing the web. Devices such as laptops, tablets, smartphones or e-book readers are offering their user base the freedom to be online on the go. Apart from new devices also other user groups, such as elderly or handicapped people, have to be considered when designing modern web pages. The optimisation of websites according to those many different types of users is a huge challenge. Very similar to the archiving of the web this problem can also not be solved by using one single solution. *Mobile Viewport Tracking* introduced a new approach, in order to analyse web pages and their users especially focusing on mobile devices. This approach could contribute to a set of already existing solutions, with the goal to improve the user experience of web pages for all existing user groups.

APPENDIX A

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

```java
package at.tuwien.viewport.utils.heatmapdrawer;

import java.awt.image.BufferedImage;

import at.tuwien.viewport.persistence.entities.Heatmap;
import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;

public interface IHeatmapDrawer {

    public BufferedImage drawHeatmap(Heatmap heatmap);

    public boolean drawAndSaveHeatmap(Heatmap heatmap, String outputDirectoryPath,
         String filename)
            throws MVTScreenshotReadException;

}
```

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

```java
package at.tuwien.viewport.utils.heatmapdrawer;

import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.logging.Logger;

import javax.imageio.ImageIO;

import at.tuwien.viewport.persistence.entities.Heatmap;
import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
import at.tuwien.viewport.screenshot.MVTPropertyNotFoundException;
import at.tuwien.viewport.screenshot.ScreenshotLoader;

public abstract class AbstractMapDrawer implements IHeatmapDrawer {

    protected final static Logger log = Logger.getLogger(AbstractMapDrawer.class.getName());

    protected String fileExtension = ".png";

    protected String filePostfix = "";

    public AbstractMapDrawer(String filePostfix)
    {
        this.filePostfix = filePostfix;
    }

    @Override
    public boolean drawAndSaveHeatmap(Heatmap heatmap, String outputDirectoryPath, String filename)
            throws MVTScreenshotReadException {
        File file = new File(outputDirectoryPath + filename + "-" + heatmap.getHeatmapVersion() + filePostfix
                + fileExtension);
        if (heatmap.getImage() == null || heatmap.getImage().getImage() == null)
        {
            boolean alternativeScreenshotTaken = false;

            if (heatmap.getImage() != null && heatmap.getMetadata() != null && heatmap.getMetadata().getScreenshots() != null)
            {

                try {
                    for (String s : heatmap.getMetadata().getScreenshots())
                    {
                        if (ScreenshotLoader.exists(s))
                        {

                            heatmap.getImage().setImage(ScreenshotLoader.loadScreenshot(s));
                            alternativeScreenshotTaken = true;
                            break;
                        }
                    }
                    if (!alternativeScreenshotTaken)
                    {
                        for (String s : heatmap.getMetadata().getScreenshots())
                        {
                            if (ScreenshotLoader.exists("RECREATED_" + s))
                            {
                                heatmap.getImage().setImage(ScreenshotLoader.loadScreenshot("RECREATED_" + s));
                                alternativeScreenshotTaken = true;
                                break;
                            }
                        }
                    }
                } catch (MVTPropertyNotFoundException e) {
                    e.printStackTrace();
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }


        }
        BufferedImage img = this.drawHeatmap(heatmap);
        if (img == null || file == null)
        {
            return false;
        }
        try {
            ImageIO.write(img, "png", file);
        } catch (IOException e) {
            throw new MVTScreenshotReadException("Could not write Screenshot");
        } catch (NullPointerException e) {
            throw new MVTScreenshotReadException(e);
        }
        return true;
    }


}
```

Listing A.3: The Colorful Heatmap Drawer

```
package at.tuwien.viewport.utils.heatmapdrawer;

import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.net.URL;
import java.util.Arrays;
import java.util.logging.Level;

import javax.imageio.ImageIO;

import at.tuwien.viewport.persistence.entities.Heatmap;
import at.tuwien.viewport.persistence.exceptions.MVTPersistenceException;
import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
import at.tuwien.viewport.utils.HeatmapManager;

public class HeatMapDrawer extends AbstractMapDrawer {

    /** half of the size of the circle picture. */
    private static int HALFCIRCLEPICSIZE = 64;

    /** path to picture of circle which gets more transparent to the outside. */
    private static String CIRCLEPIC;
    private static String SPECTRUMPIC;
```

```java
    private static String EMPTYIMAGE = "/empty.png";

    private float multiplier = 1.0f;

    private int[] circles = new int[] { 256, 128, 96, 64, 48, 32, 24, 16, 8 };

    public HeatMapDrawer() {
        super("heat");
        SPECTRUMPIC = "/colors.png";
        CIRCLEPIC = "/circle_128.png";
    }

    /**
     * @param multiplier
     *            calculated opacity of every point will be
     *            multiplied by this value. This leads to a HeatMap that is easier to read,
     *            especially when there are not too many points or the points are too
     *            spread out. Pass 1.0f for original.
     */
    public HeatMapDrawer(float multiplier, int circleSize) {
        super("heat-" + circleSize);

        if (!Arrays.asList(circles).contains(circleSize))
        {
            log.log(Level.SEVERE, "Could not create drawer because circle size not available.");
            return;
        }
        this.multiplier = multiplier;
        SPECTRUMPIC = "/colors.png";
        CIRCLEPIC = "/circle_" + circleSize + ".png";
        HALFCIRCLEPICSIZE = circleSize / 2;
    }

    @Override
    public BufferedImage drawHeatmap(Heatmap heatmap) {

        BufferedImage mapPic = heatmap.getImage().getImage();

        if (mapPic == null) {
            log.log(Level.INFO, "Drawing Heatmap with empty image for URL: " + heatmap.getUrl());
            // return null;
            mapPic = loadImage(EMPTYIMAGE);
        }

        int maxXValue = mapPic.getWidth();
        int maxYValue = mapPic.getHeight();

        BufferedImage circle = loadImage(CIRCLEPIC);
        BufferedImage heatMap = new BufferedImage(maxXValue, maxYValue, 6);
        paintInColor(heatMap, Color.white);

        double maxWeight = heatmap.getMaximumWeightOfImportanceMatrix();
        double lowerLimit = 0.0;

        for (int i = 0; i < heatmap.getImportanceMatrix().length; i++) {

            int positionY = HeatmapManager.TILE_SIZE * i;

            for (int j = 0; j < heatmap.getImportanceMatrix()[i].length; j++)
            {
                int positionX = HeatmapManager.TILE_SIZE * j;
                // now we are on the pos i j
                double valueAtCurrentPos = heatmap.getImportanceMatrix()[i][j];

                if (valueAtCurrentPos > lowerLimit)
                {
                    float weight = (float) (valueAtCurrentPos / maxWeight) * multiplier
                        ;
                    addImage(heatMap, circle, weight,
                            (positionX - HeatmapManager.TILE_SIZE / 2 -
                                HALFCIRCLEPICSIZE),
                            (positionY - HeatmapManager.TILE_SIZE / 2 -
                                HALFCIRCLEPICSIZE));
                }
            }
        }

        heatMap = colorImage(heatMap);
        addImage(mapPic, heatMap, 0.4f);
        return mapPic;
    }

    /**
     * First it negates the image.
     * Then remaps black and white picture with colors.
     * It uses the colors from SPECTRUMPIC. The whiter a pixel is, the more it
     * will get a color from the bottom of it. Black will stay black.
     *
     * @param heatMapBW
     *            black and white heat map
     */
    private BufferedImage colorImage(BufferedImage img) {
        BufferedImage colorGradiant = loadImage(SPECTRUMPIC);
        int gradientHeight = colorGradiant.getHeight() - 1;
        int width = img.getWidth();
        int height = img.getHeight();
        for (int x = 0; x < width; x++) {
            for (int y = 0; y < height; y++) {

                // (1) Negate
                int rGB = img.getRGB(x, y);

                // Swaps values
                // i.e. 255, 255, 255 (white)
                // becomes 0, 0, 0 (black)
                int r = Math.abs(((rGB >>> 16) & 0xff) - 255); // red inverted
                int g = Math.abs(((rGB >>> 8) & 0xff) - 255); // green inverted
                int b = Math.abs((rGB & 0xff) - 255); // blue inverted

                // get heatMapBW color values:
                rGB = (r << 16) | (g << 8) | b;

                // (2) Remap

                // calculate multiplier to be applied to height of gradiant.
                float multiplier = rGB & 0xff; // blue
                multiplier *= ((rGB >>> 8)) & 0xff; // green
                multiplier *= (rGB >>> 16) & 0xff; // red
                multiplier /= 16581375; // 255f * 255f * 255f

                // apply multiplier
```

```java
                int colorY = (int) (multiplier * gradientHeight);

                // remap values
                // calculate new value based on whitenes of heatMap
                // (the whiter, the more a color from the top of colorGradiant
                // will be chosen.
                int mapedRGB = colorGradiant.getRGB(0, colorY);
                // set new value
                img.setRGB(x, y, mapedRGB);

            }
        }
        return img;
    }

    /**
     * changes all pixel in the buffer to the provided color.
     *
     * @param buff
     *            buffer
     * @param color
     *            color
     */
    private void paintInColor(BufferedImage buff, Color color) {
        Graphics2D g2 = buff.createGraphics();
        g2.setColor(color);
        g2.fillRect(0, 0, buff.getWidth(), buff.getHeight());
        g2.dispose();
    }

    /**
     * prints the contents of buff2 on buff1 with the given opaque value
     * starting at position 0, 0.
     *
     * @param buff1
     *            buffer
     * @param buff2
     *            buffer to add to buff1
     * @param opaque
     *            opacity
     */
    private void addImage(
            BufferedImage buff1, BufferedImage buff2, float opaque) {
        addImage(buff1, buff2, opaque, 0, 0);
    }

    /**
     * prints the contents of buff2 on buff1 with the given opaque value.
     *
     * @param buff1
     *            buffer
     * @param buff2
     *            buffer
     * @param opaque
     *            how opaque the second buffer should be drawn
     * @param x
     *            x position where the second buffer should be drawn
     * @param y
     *            y position where the second buffer should be drawn
     */
    private void addImage(BufferedImage buff1, BufferedImage buff2,
            float opaque, int x, int y) {
```

```
        Graphics2D g2d = buff1.createGraphics();
        g2d.setComposite(
                AlphaComposite.getInstance(AlphaComposite.SRC_OVER, opaque));
        g2d.drawImage(buff2, x, y, null);
        g2d.dispose();
    }

    private BufferedImage loadImage(String ref) {
        BufferedImage b1 = null;
        try {
            URL url = getClass().getResource(ref);
            b1 = ImageIO.read(url);
        } catch (IOException e) {
            log.log(Level.SEVERE, "error loading the image.", e);
        }
        return b1;
    }

    public static void main(String[] args) throws MVTScreenshotReadException,
        MVTPersistenceException
    {
        HeatmapManager.getInstance().drawAllHeatmapsFromDatabase(new HeatMapDrawer());
    }
}
```

Listing A.4: The Rastermap Drawer

```
package at.tuwien.viewport.utils.heatmapdrawer;

import java.awt.AlphaComposite;
import java.awt.BasicStroke;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics2D;
import java.awt.image.BufferedImage;
import java.text.NumberFormat;

import at.tuwien.viewport.persistence.entities.Heatmap;
import at.tuwien.viewport.persistence.exceptions.MVTPersistenceException;
import at.tuwien.viewport.persistence.exceptions.MVTScreenshotReadException;
import at.tuwien.viewport.utils.HeatmapManager;

public class RasterMapDrawer extends AbstractMapDrawer {

    public RasterMapDrawer() {
        super("raster");
    }

    @Override
    public BufferedImage drawHeatmap(Heatmap heatmap) {
        BufferedImage img = heatmap.getImage().getImage();
        if (img == null)
        {
            return null;
        }

        Graphics2D g2d = img.createGraphics();

        BasicStroke bs = new BasicStroke(1);
        g2d.setStroke(bs);
```

```java
        Font font = new Font("Serif", Font.PLAIN, 8);
        g2d.setFont(font);

        // needed for scaling, making it bigger than the maximum to have it always a bit transparent
        double maxWeight = Math.log(heatmap.getMaximumWeightOfImportanceMatrix()) * 1.2;

        boolean drawRaster = true;
        for (int i = 0; i < heatmap.getImportanceMatrix().length; i++) {

            int positionY = HeatmapManager.TILE_SIZE * i;
            if (positionY < img.getHeight())
            {
                // Drawing horizontal line
                if (drawRaster)
                {
                    g2d.setColor(Color.GRAY);
                    g2d.drawLine(0, positionY, img.getWidth(), positionY);
                }
            }

            for (int j = 0; j < heatmap.getImportanceMatrix()[i].length; j++)
            {
                int positionX = HeatmapManager.TILE_SIZE * j;
                if (positionX < img.getWidth())
                {
                    // Drawing vertical line
                    // Because this is a nested loop has to be checked that it is only done once
                    if (drawRaster && i == 0)
                    {
                        g2d.setColor(Color.GRAY);
                        g2d.drawLine(positionX, 0, positionX, img.getHeight());
                    }
                }
                // now we are on the pos i j
                double valueAtCurrentPos = heatmap.getImportanceMatrix()[i][j];

                double loggedValue = Math.log10(valueAtCurrentPos);
                if (loggedValue < 0)
                {
                    loggedValue = 0;
                }
                setColorForCurrentPos(loggedValue, maxWeight, g2d);
                drawRectToTile(i, j, loggedValue, g2d, drawRaster);
            }
        }
        g2d.drawImage(img, null, 0, 0);
        g2d.dispose();
        return img;
    }

    private void drawRectToTile(int y, int x, double valueAtCurrentPos, Graphics2D g2d, boolean drawRaster) {
        int tileSize = HeatmapManager.TILE_SIZE;
        int positionX = tileSize * x;
        int positionY = tileSize * y;
        g2d.fillRect(positionX + 1, positionY + 1, tileSize - 1, tileSize - 1);

        // Lighter text color
        g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 0.5f));

        if (drawRaster)
        {
            g2d.setColor(Color.GRAY);
            NumberFormat nf = NumberFormat.getInstance();
            nf.setMaximumFractionDigits(1);
            g2d.drawString(nf.format(valueAtCurrentPos), positionX + 2, positionY + 13);
        }

        // Reset composite
        g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 1));

    }

    private void setColorForCurrentPos(double valueAtCurrentPos, double maxWeight, Graphics2D g2d) {
        float r = 1.0f, g = 0f, b = 0f;
        float opacity = (float) (valueAtCurrentPos / maxWeight); // between 0 and 1

        g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, opacity));
        g2d.setColor(new Color(r, g, b, opacity));
    }

    public static void main(String[] args) throws MVTScreenshotReadException, MVTPersistenceException
    {
        HeatmapManager.getInstance().drawAllHeatmapsFromDatabase(new RasterMapDrawer());
    }
}
```

A.1.2 Action Type Determiner

Listing A.5: The Action Type Determiner

```
package at.tuwien.viewport.utils;

import java.lang.reflect.Method;
import java.util.logging.Logger;

import at.tuwien.viewport.persistence.entities.ActionType;
import at.tuwien.viewport.persistence.entities.ViewportEntry;
import at.tuwien.viewport.persistence.entities.ViewportEntry.Fields;

public class ActionTypeDeterminer {

    private ViewportEntry oldViewport;
    private ViewportEntry newViewport;
    private static final int tolerance = 5; //Tolerance in pixels

    private enum Change {
        BIGGER,
        EQUAL,
        SMALLER;
    }

    private final static Logger log = Logger
```

```
            .getLogger(ActionTypeDeterminer.class.getName());

    public ActionTypeDeterminer(ViewportEntry oldViewport,
            ViewportEntry newViewport) {
        this.oldViewport = oldViewport;
        this.newViewport = newViewport;
    }



    public ActionType getActionType() {

        // ZOOM-IN
        if (is(ViewportEntry.Fields.innerHeight, Change.SMALLER)
                && is(ViewportEntry.Fields.innerWidth, Change.SMALLER)) {
            return ActionType.ZOOM_IN;
        }
        // ZOOM-OUT
        if (is(ViewportEntry.Fields.innerHeight, Change.BIGGER)
                && is(ViewportEntry.Fields.innerWidth, Change.BIGGER)) {
            return ActionType.ZOOM_OUT;
        }
        // SCROLL_UP_LEFT
        if (is(ViewportEntry.Fields.pageOffsetX, Change.SMALLER)
                && is(ViewportEntry.Fields.pageOffsetY, Change.SMALLER)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_UP_LEFT;
        }
        // SCROLL_UP_RIGHT
        if (is(ViewportEntry.Fields.pageOffsetX, Change.BIGGER)
                && is(ViewportEntry.Fields.pageOffsetY, Change.SMALLER)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_UP_RIGHT;
        }
        // SCROLL_DOWN_LEFT
        if (is(ViewportEntry.Fields.pageOffsetX, Change.SMALLER)
                && is(ViewportEntry.Fields.pageOffsetY, Change.BIGGER)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_DOWN_LEFT;
        }
        // SCROLL_DOWN_RIGHT
        if (is(ViewportEntry.Fields.pageOffsetX, Change.BIGGER)
                && is(ViewportEntry.Fields.pageOffsetY, Change.BIGGER)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)
                && is(ViewportEntry.Fields.innerHeight, Change.EQUAL)) {
            return ActionType.SCROLL_DOWN_RIGHT;
        }

        // SCROLL_LEFT
        if (is(ViewportEntry.Fields.pageOffsetX, Change.SMALLER)
                && is(ViewportEntry.Fields.pageOffsetY, Change.EQUAL)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_LEFT;
        }
        // SCROLL_RIGHT
        if (is(ViewportEntry.Fields.pageOffsetX, Change.BIGGER)
                && is(ViewportEntry.Fields.pageOffsetY, Change.EQUAL)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_RIGHT;
        }
        // SCROLL_UP
        if (is(ViewportEntry.Fields.pageOffsetX, Change.EQUAL)
                && is(ViewportEntry.Fields.pageOffsetY, Change.SMALLER)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_UP;
        }
        // SCROLL_DOWN
        if (is(ViewportEntry.Fields.pageOffsetX, Change.EQUAL)
                && is(ViewportEntry.Fields.pageOffsetY, Change.BIGGER)
                && is(ViewportEntry.Fields.innerWidth, Change.EQUAL)) {
            return ActionType.SCROLL_DOWN;
        }


        return null;
    }


    private Boolean is(Fields field, Change change) {
        String methodName = "get"
                + field.toString().substring(0, 1).toUpperCase()
                + field.toString().substring(1);
        try {
            Method getter = oldViewport.getClass().getMethod(methodName);
            int oldValue = (Integer) getter.invoke(oldViewport);
            int newValue = (Integer) getter.invoke(newViewport);

            switch (change) {
            case BIGGER:
                if (newValue > oldValue + tolerance) {
                    return true;
                }
                break;
            case SMALLER:
                if (newValue < oldValue - tolerance) {
                    return true;
                }
                break;
            case EQUAL:
                if (newValue == oldValue) {
                    return true;
                } else if (newValue > oldValue) {
                    if (newValue <= oldValue + tolerance) {
                        return true;
                    }
                } else if (newValue < oldValue) {
                    if (newValue + tolerance >= oldValue) {
                        return true;
                    }
                }
                break;

            }
            return false;

        } catch (Exception e) {
            log.severe("Exception: " + e.getMessage());
        }

        return null;
    }
}
```

A.1.3 Client Side - Viewport.js

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

```
//Version 2.1

// MVT namespace to avoid conflicts
var MVT = {};
// Basic configuration
MVT.Config =
{
        SERVER_URL : null // Set within "init"
};

MVT.DOMEvents =
{
        CLICK : "CLICK",
        SUBMIT : "SUBMIT",
        INPUT_CHANGED : "INPUT_CHANGED",
        KEY_PRESS : "KEY_PRESS",
        WEBSITE_ACTIVATED : "WEBSITE_ACTIVATED",
        WEBSITE_DEACTIVATED : "WEBSITE_DEACTIVATED",
        ROTATION_TO_HORIZONTAL : "ROTATION_TO_HORIZONTAL",
        ROTATION_TO_VERTICAL : "ROTATION_TO_VERTICAL",
        DOUBLE_TAP : "DOUBLE_TAP",
        HOLD: "HOLD"
};

//Variables
MVT.currentX = -1;
MVT.currentY = -1;
MVT.innerWidth = -1;
MVT.innerHeight = -1;
MVT.outerWidth = -1;
MVT.outerHeight = -1;
MVT.scrollWidth = -1;
MVT.scrollHeight = -1;

MVT.queuedData = [];
//check for race conditions
MVT.post_id;
MVT.OSName;
// Cookie within this page
MVT.pageLoadToken;
// Cookie within same domain
MVT.domainUserToken;
MVT.userToken;

//filter Keywords - Do not report Viewport of Ads
MVT.AD_KEYWORDS = new Array("/uim.html", "partner=", "banner.html",
"/AdServer/", "/MetaAdServer/", "a.ligatus.com", "doubleclick.net",
"plugins/like.php", "plugins/activity.php", "likebox.php", "Banner.php", "ads.", "heatmaps.no-ip.org"
);

MVT.lastEventTimeStamp = 0;
MVT.scrolling = false;
MVT.pauseDetection = false;
MVT.scrollTimer = false;
MVT.blockPasswordTransmission = true;

MVT.init = function(serverIP)
{
        if(!serverIP)
        {
                console.error("Server IP of MVTService needed.");
                return;
        }
        MVT.Config.SERVER_URL = "http://" + serverIP + "/MVTService/rs/dataCollector/storeData";

        //Setting, getting cookie
        var found = false;
        var cookies = document.cookie.split(";");
        for (var i=0; i<cookies.length; i++) {
                var keyValue = cookies[i].split("=");
                if (keyValue[0].match("viewport")) {
                        MVT.domainUserToken = keyValue[1];
                        found = true;
                        break;
                }
        }
        if (!found) {
                MVT.domainUserToken = MVT.getRandom();
                //expiration is 365 days
                document.cookie = "viewport=" + MVT.domainUserToken + "; expires=" + MVT.getExpireDate(365, 0) + "; path=/";
        }

        //Setting One-Time-Variables
        MVT.post_id = 1;
        MVT.pageLoadToken = MVT.getRandom();
        MVT.OSName = navigator.appVersion;
        if(!MVT.filterDocuments(document.URL))
        {
                var iDevice = navigator.userAgent.match(/iPad/i) != null || navigator.userAgent.match(/iPhone/i) != null;
                var androidDevice = navigator.userAgent.match(/Android/i) != null;

                window.addEventListener(’message’, function (e)
            {
                  if (e.origin == ("http://" + serverIP))
                  {
                          var data = e.data;
                          if(data)
                      {
                                  MVT.userToken = data.substring(data.indexOf("=")+1);
                      }
                  }
                }, false);

                window.addEventListener(’load’, function()
                {
                        MVT.registerDOMEvents();

                        //alert("w:"+document.body.scrollWidth + ",h:" + document.body.scrollHeight);

                        window.addEventListener(’pageshow’, function()
                        {
                                MVT.queueData(MVT.DOMEvents.WEBSITE_ACTIVATED, null, "window-pageshow");
                        }, false);

                        window.addEventListener(’focus’, function()
                        {
                                MVT.queueData(MVT.DOMEvents.WEBSITE_ACTIVATED, null, "window-focus");
                        }, false);

                        // Android fires "blur" but iPhone the "pagehide"
                        if(androidDevice)
                        {
                                window.addEventListener(’blur’, function(){ MVT.queueData(MVT.DOMEvents.WEBSITE_DEACTIVATED, null, "window-blur"); }, false);

                        }
                        else if (iDevice)
                        {
                                window.addEventListener(’pagehide’, function(){ MVT.queueData(MVT.DOMEvents.WEBSITE_DEACTIVATED, null, "window-pagehide"); MVT.postData(); }, false);
                        }

                        window.addEventListener(’scroll’, function( event )
                        {
                    if (!MVT.scrolling)
                    {
                        MVT.scrolling = true;
                    }

                    clearTimeout( MVT.scrollTimer );
                    MVT.scrollTimer = setTimeout(function()
                    {
                        MVT.scrolling = false;
                        if(MVT.pauseDetection)
                        {
                                MVT.pauseDetection = false;
                        }
                    }, 100 );
                }, false);

                        window.addEventListener(’orientationchange’, function()
                        {
                if ( window.orientation == 0  || window.orientation == 180 )
                {
                        MVT.queueData(MVT.DOMEvents.ROTATION_TO_VERTICAL);
                }
                else if ( window.orientation == 90 || window.orientation == -90 )
                {
                        MVT.queueData(MVT.DOMEvents.ROTATION_TO_HORIZONTAL);
                }
                        }, false);

                        var hammer = new Hammer(document.body, {
                                swipe: false
                        });

                        hammer.ondragstart = function(ev) {
                                MVT.pauseDetection = true;
                        };
                        //hammer.ondrag = function(ev) { };
                        hammer.ondragend = function(ev) {
                                if(!MVT.scrolling)
                                {
                                        // No "lazy" scrolling, directly capture data
                                        MVT.pauseDetection = false;
                                }
                        };

                        //hammer.ontap = function(ev) { };
                        hammer.ondoubletap = function(ev, data) {
                                MVT.queueData(MVT.DOMEvents.DOUBLE_TAP, null, null, ev.position[0].x + ";" + ev.position[0].y);
                        };
                        hammer.onhold = function(ev) {
                                MVT.queueData(MVT.DOMEvents.HOLD, null, null, ev.position[0].x + ";" + ev.position[0].y);
                        };

                        hammer.ontransformstart = function(ev) {
                                MVT.pauseDetection = true;
                        };
                        //hammer.ontransform = function(ev) { };
                        hammer.ontransformend = function(ev) {
                                MVT.pauseDetection = false;
                                //MVT.queueData();
                        };

                        //hammer.onrelease = function(ev) { };

                        setInterval("MVT.detectChanges()", 200);
                        setInterval("MVT.postData()", 1500);
                }, false);
        }
};

MVT.registerDOMEvents = function()
{
        var fnRegisterListener = function(aElement, sListenerName, aDOMEventEnum, bStoreValue)
        {
                aElement.addEventListener(sListenerName, function (event)
                {
                        var aValue = bStoreValue ? aElement.value : null;
                        // Click is performed for an element and all underlying elements
                        // therefore the timestamp is compared to just check the first element
                        if(MVT.lastEventTimeStamp == event.timeStamp)
                        {
                                return;
                        }
                        var actionTypeData = null;
                        if(aDOMEventEnum == MVT.DOMEvents.CLICK)
                        {
                                actionTypeData = event.x + ";" + event.y;
                        }
                        if(aDOMEventEnum == MVT.DOMEvents.INPUT_CHANGED || aDOMEventEnum == MVT.DOMEvents.KEY_PRESS)
                        {
                                if(event.target && event.target.type && event.target.type == "password" && MVT.blockPasswordTransmission)
                                {
                                        var aLength = aValue.length;
                                        var aFirstPWChars = "";
                                        if(aLength > 0)
                                        {
                                                aFirstPWChars = aValue.substring(0,1);
                                        }
                                        aValue = "Password: ";
                                        if(aLength > 0)
                                        {
                                                aValue += aFirstPWChars;
                                        }

                                        for(var i=1; i<aLength; i++)
                                        {
                                                aValue += "*";
                                        }
                                }
                        }

                        MVT.lastEventTimeStamp = event.timeStamp;
                        MVT.queueData(aDOMEventEnum, MVT.getElementXPath(this), aValue, actionTypeData);
                        MVT.postData();
                }, false);
        };

        MVT.forAllTags("a", MVT.bind(fnRegisterListener, ['click', MVT.DOMEvents.CLICK, false], true));
        MVT.forAllTags("area", MVT.bind(fnRegisterListener, ['click', MVT.DOMEvents.CLICK, false], true));
        MVT.forAllTags("span", MVT.bind(fnRegisterListener, ['click', MVT.DOMEvents.CLICK, false], true));
        MVT.forAllTags("div", MVT.bind(fnRegisterListener, ['click', MVT.DOMEvents.CLICK, false], true));
        MVT.forAllTags("p", MVT.bind(fnRegisterListener, ['click', MVT.DOMEvents.CLICK, false], true));

        // button, checkbox, radio
        MVT.forAllTags("input", MVT.bind(fnRegisterListener, ['change', MVT.DOMEvents.INPUT_CHANGED, true], true));
        MVT.forAllTags("textarea", MVT.bind(fnRegisterListener, ['change', MVT.DOMEvents.INPUT_CHANGED, true], true));
        MVT.forAllTags("input", MVT.bind(fnRegisterListener, ['keydown', MVT.DOMEvents.KEY_PRESS, true], true));
        MVT.forAllTags("textarea", MVT.bind(fnRegisterListener, ['keydown', MVT.DOMEvents.KEY_PRESS, true], true));

        MVT.forAllTags("select", MVT.bind(fnRegisterListener, ['change', MVT.DOMEvents.INPUT_CHANGED, true], true));
        MVT.forAllTags("form", MVT.bind(fnRegisterListener, ['submit', MVT.DOMEvents.SUBMIT, false], true));
};

MVT.forAllTags = function(sTagName, fnCallback)
{
        if(!fnCallback)
        {
                return;
        }
        var elements = document.getElementsByTagName(sTagName);
        for(var i = 0; i < elements.length; i++)
        {
                fnCallback(elements[i]);
        }
};

/**
  * Possibility to specify arguments which are always appended when a function is called
  */
MVT.bind = function(fn, args, appendArgs)
{
        var method = fn,
                slice = Array.prototype.slice;

        return function() {
                var callArgs = args || arguments;

                if (appendArgs === true) {
                        callArgs = slice.call(arguments, 0);
                        callArgs = callArgs.concat(args);
                }
                else if (typeof appendArgs == ’number’) {
                        callArgs = slice.call(arguments, 0); // copy arguments first
                        Ext.Array.insert(callArgs, appendArgs, args);
                }

                return method.apply(window, callArgs);
        };
};

MVT.getRandom = function()
{
        return Math.floor(Math.random()*10000000000001);
};

MVT.getElementXPath = function(element) {
  // TODO: Also handle className and name attributes to optimize xpath
  if (element && element.id)
  {
          return ’//*[@id="’ + element.id + ’"]’;
  }
  else
  {
          return MVT.getElementTreeXPath(element);
  }
};

MVT.getElementTreeXPath = function(element) {
  var paths = [];

  // Use nodeName (instead of localName) so namespace prefix is included (if any).
  for (; element && element.nodeType == 1; element = element.parentNode)  {
          var index = 0;
          // EXTRA TEST FOR ELEMENT.ID
          if (element && element.id) {
                  paths.splice(0, 0, ’/*[@id="’ + element.id + ’"]’);
                  break;
          }

          for (var sibling = element.previousSibling; sibling; sibling = sibling.previousSibling) {
                  // Ignore document type declaration.
                  if (sibling.nodeType == Node.DOCUMENT_TYPE_NODE)
                        continue;

                  if (sibling.nodeName == element.nodeName)
                          ++index;
          }

          var tagName = element.nodeName.toLowerCase();
          var pathIndex = (index ? "[" + (index+1) + "]" : "");
          paths.splice(0, 0, tagName + pathIndex);
  }

  return paths.length ? "/" + paths.join("/") : null;
};

MVT.getExpireDate = function(tage, stunden)
{
  var jetzt = new Date();
  var zeit = jetzt.getTime();
  var zukunft = zeit + (((tage * 24) + stunden) * 3600 * 1000);
  jetzt.setTime(zukunft);
  var haltbarkeit = jetzt.toUTCString();
  return haltbarkeit;
};

MVT.filterDocuments = function(url) {
        ow = window.outerWidth;
        oh = window.outerHeight;

        // Filter for iframes
        if (window.top != window.self)
        {
                return true;
        }
        if(ow < 80 && oh < 80){
                //filter small add-frames or faceook-like-frames or something like that
                return true;
        }
        for (var i=0; i< MVT.AD_KEYWORDS.length; i++)
        {
                if(url.match(MVT.AD_KEYWORDS[i])){
                        return true;
                }
        }
        return false;
};

MVT.detectChanges = function() {
        if (!MVT.pauseDetection && (MVT.changed(window.pageXOffset, MVT.currentX)
                        || MVT.changed(window.pageYOffset, MVT.currentY)
                        || MVT.changed(window.innerWidth, MVT.innerWidth)
                        || MVT.changed(window.innerHeight, MVT.innerHeight)
                        || MVT.changed(window.outerWidth, MVT.outerWidth)
                        || MVT.changed(window.outerHeight, MVT.outerHeight))
                        )
                {
                        MVT.queueData();
                }
};

MVT.changed = function(val1, val2)
{
        //TOLERANCE for equals
        var tol = 5;
        if (val1 > val2) {
                if(val1 <= val2 + tol)
                {
                        return false;
                }

        } else if (val1 < val2) {
                if(val1 + tol >= val2)
                {
                        return false;
                }

        } else {
                return false;
        }
        return true;
};

MVT.queueData = function(aEventType, sXPath, sElementValue, actionTypeData)
{
        MVT.currentX = window.pageXOffset;
        MVT.currentY = window.pageYOffset;
        MVT.innerWidth = window.innerWidth;
        MVT.innerHeight = window.innerHeight;
        MVT.outerWidth = window.outerWidth;
        MVT.outerHeight = window.outerHeight;
        MVT.scrollWidth = document.body.scrollWidth;
        MVT.scrollHeight = document.body.scrollHeight;

        var recorded_millis = new Date().getTime();

        var aData = {
                                  pageLoadToken : MVT.pageLoadToken,
                                  URL : document.URL,
                                  OS : MVT.OSName,
                                  pageOffsetX : MVT.currentX,
                                  pageOffsetY : MVT.currentY,
                                  innerWidth : MVT.innerWidth,
                                  innerHeight : MVT.innerHeight,
                                  outerWidth : MVT.outerWidth,
                                  outerHeight : MVT.outerHeight,
                                  scrollWidth : MVT.scrollWidth,
                                  scrollHeight : MVT.scrollHeight,
                                  domainUserToken : MVT.domainUserToken,
                                  recordedTime : recorded_millis,
                                  postId : MVT.post_id++,
                                  actionType : aEventType ? aEventType : null, // explicit set null to avoid "undefined"
                                  actionTypeData : actionTypeData ? actionTypeData : null,
                                  domXPath : sXPath ? sXPath : null,
                                  domElementValue : sElementValue ? sElementValue : null,
                                  referrer : document.referrer != "" ? document.referrer : null,
                                  domain : window.location.host
                                };

        // Message from iframe not yet received
        MVT.waitForUserToken(aData);
};

```

```
MVT.waitForUserToken = function(aData)
{
        if(!MVT.userToken)
        {
                setTimeout(function()
                {
                        MVT.waitForUserToken(aData);
            }, 300);
        }
        else
        {
                aData.userToken = MVT.userToken;
                MVT.queuedData.push(aData);
        }

};

MVT.postData = function()
{
        if(MVT.queuedData.length == 0)
        {
                return;
        }
        var sData = JSON.stringify(MVT.queuedData);

        MVT.queuedData = [];


  var xmlHttp = new XMLHttpRequest();
  xmlHttp.open("POST", MVT.Config.SERVER_URL, true);
  xmlHttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xmlHttp.send("jsonData=" + MVT.Base64.encode(sData));
};

MVT.postJSONP = function()
{
        var script = document.createElement("script");
        script.type = "text/javascript";
        script.src = MVT.Config.SERVER_URL + "?data=" + encodeURIComponent(MVT.Base64.encode(sData));
        var head = document.getElementsByTagName("head")[0];
    (head || document.body).appendChild(script);
};

MVT.callback = function()
{
        console.info("stored data");
};

//------------------------------------------------------------------------
//----------     Base 64 Encoding
//----------     http://www.webtoolkit.info/
//------------------------------------------------------------------------
MVT.Base64 =
{
                /* The sources for Base 64 encoding where included here but are not
                   * included in this listing. They can be found on the project's homepage:
                   * http://www.webtoolkit.info/
                   */
};

```

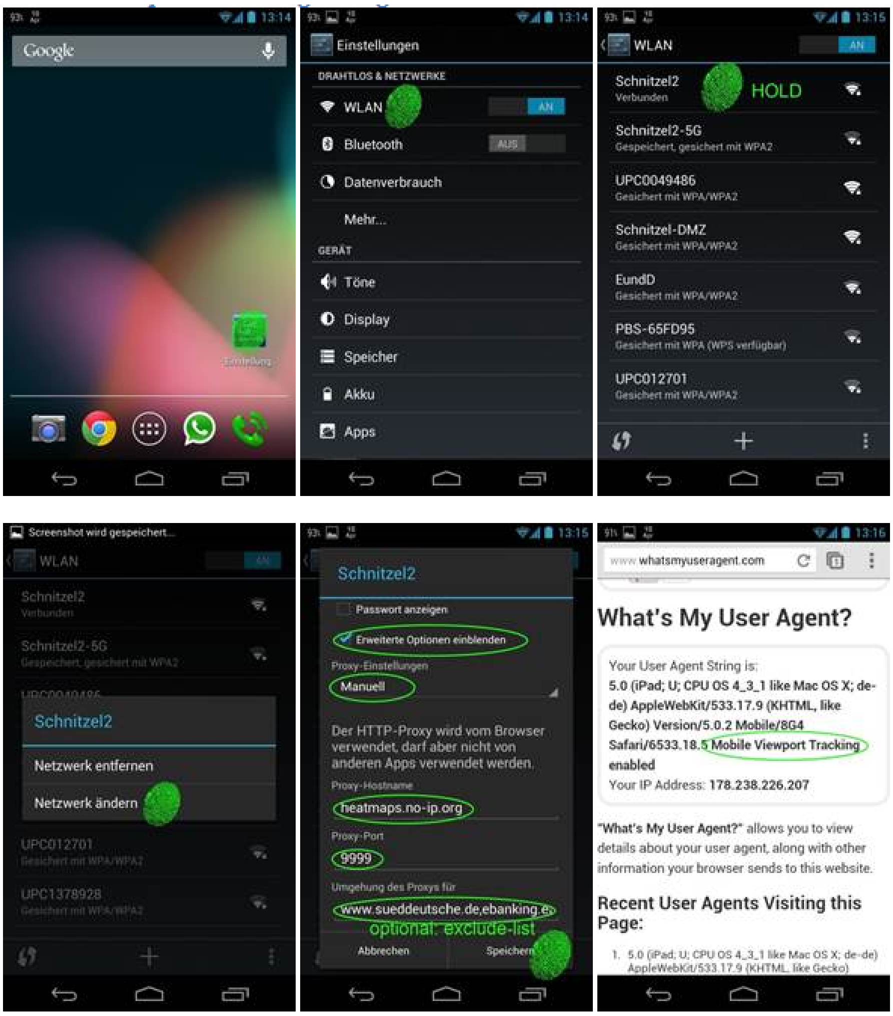
```




/*
 * Hammer.JS
 * version 0.6.4
 * author: Eight Media
 * https://github.com/EightMedia/hammer.js
 * Licensed under the MIT license.
 */
function Hammer(element, options, undefined)
{
  /* The sources of Hammer.js where included here (Version 0.6.4) but are not
   * included in this listing. They can be found on the project’s github:
   * https://github.com/EightMedia/hammer.js
   */
}
```

A.2 Setup of Mobile Viewport Tracking

A.2.1 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 A.1 and A.2 show the configuration tutorial as given to the participants of the case study.



Schnitzel2

□ Passwort anzeigen

☑ Erweiterte Optionen einblenden

Proxy-Einstellungen: Manuell

Der HTTP-Proxy wird vom Browser verwendet, darf aber nicht von anderen Apps verwendet werden.

Proxy-Hostname: heatmaps.no-ip.org

Proxy-Port: 9999

Umgehung des Proxys für: www.sueddeutsche.de,ebanking.e

www.whatsmyuseragent.com

What's My User Agent?

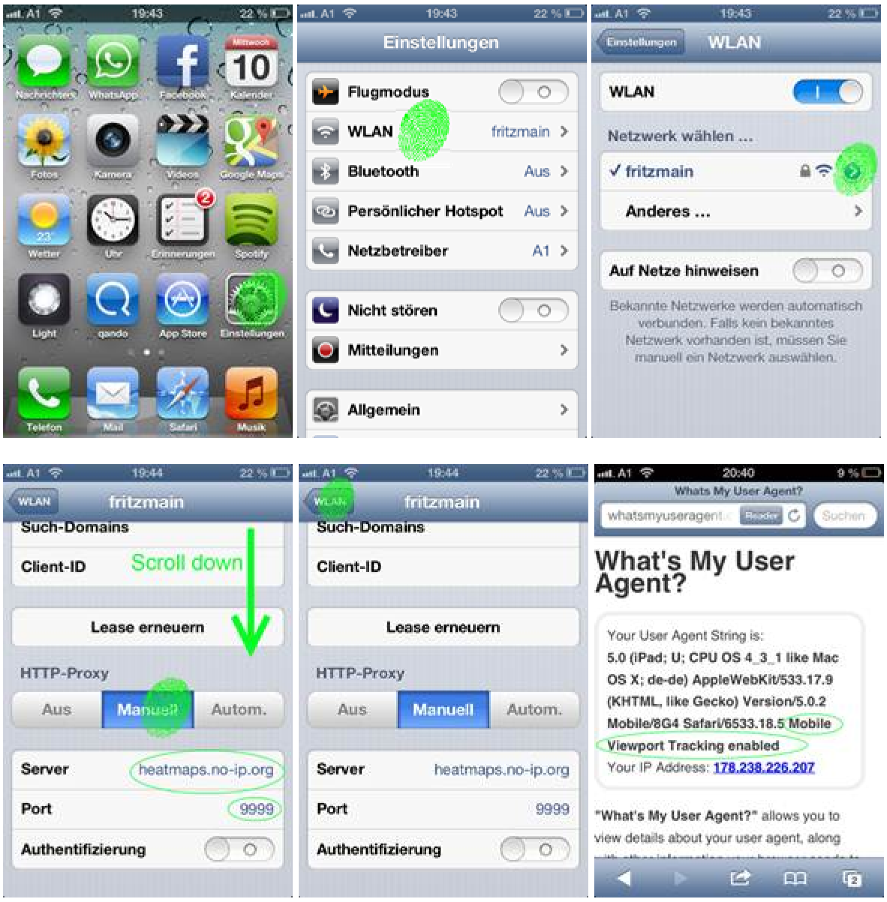
Your User Agent String is:
5.0 (iPad; U; CPU OS 4_3_1 like Mac OS X; de-de) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8G4 Safari/6533.18.5 Mobile Viewport Tracking enabled
Your IP Address: 178.238.226.207

"What's My User Agent?" allows you to view details about your user agent, along with other information your browser sends to this website.

Recent User Agents Visiting this Page:

1. 5.0 (iPad; U; CPU OS 4_3_1 like Mac OS X; de-de) AppleWebKit/533.17.9 (KHTML, like Gecko)

Figure A.1: Configuration Tutorial for Android 4.X



Flugmodus ○
WLAN fritzmain
Bluetooth Aus
Persönlicher Hotspot Aus
Netzbetreiber A1
Nicht stören ○
Mitteilungen

WLAN I
Netzwerk wählen ...
✓ fritzmain
Anderes ...
Auf Netze hinweisen ○
Bekannte Netzwerke werden automatisch verbunden. Falls kein bekanntes Netzwerk vorhanden ist, müssen Sie manuell ein Netzwerk auswählen.

Such-Domains
Client-ID
Lease erneuern
HTTP-Proxy
Aus [✓ Manuell] Autom.
Server heatmaps.no-ip.org
Port 9999
Authentifizierung ○

What's My User Agent?

Your User Agent String is:
5.0 (iPad; U; CPU OS 4_3_1 like Mac OS X; de-de) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8G4 Safari/6533.18.5 Mobile Viewport Tracking enabled
Your IP Address: 178.238.226.207

"What's My User Agent?" allows you to view details about your user agent, along

Figure A.2: Configuration Tutorial for iOS

A.2.2 Server Side Configuration

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

- *MVTProxy*: The MVTProxy 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 A.7: Proxy start command

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

  where 8080 has to be 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 hostname 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 A.8: Configuration file `reg_filter.xml`

```xml
<?xml version="1.0" encoding="UTF-8"?>
<reg-filters>
  <reg-filter type="replace" status="active">
    <name>MVTInjection</name>
    <description>Injects the Javascript for Mobile Viewport Tracking</description
        >
    <regexp>
      <match>&lt;\/[ ]*head[ ]*&gt;</match>
      <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>
    </regexp>
  </reg-filter>
  <reg-filter type="replace" status="active">
    <name>MVTUserInjection</name>
    <description>Injects the IFrame for Global Cookies</description>
    <regexp>
      <match>&lt;\/[ ]*body[ ]*&gt;</match>
      <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>
    </regexp>
  </reg-filter>
</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 A.9: Hibernate configuration file `persistence.xml`

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

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

```
      <class>at.tuwien.viewport.persistence.entities.ImportanceMatrix</class>
      <class>at.tuwien.viewport.persistence.entities.Image</class>
      <properties>
        <property name="hibernate.cache.use_query_cache" value="no"/>
                <property name="hibernate.cache.use_second_level_cache" value="no
                    "/>
                <property name="cache.provider_class" value="org.hibernate.cache.
                    NoCacheProvider"/>
        <property name="hibernate.dialect" value="org.hibernate.dialect.
            MySQLDialect"/>
        <property name="hibernate.hbm2ddl.auto" value="update"/>
                <property name="hibernate.connection.driver_class" value="com.
                    mysql.jdbc.Driver"/>
                <property name="hibernate.connection.url" value="jdbc:mysql://
                    localhost:3306/mvt?autoReconnect=true"/>
                <property name="hibernate.connection.username" value="mvtuser"/>
                <property name="hibernate.connection.password" value="
                    some_secret_password"/>
                <property name="hibernate.connection.pool_size" value="10"/>
      </properties>
   </persistence-unit>
</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 Driver[3] 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 A.10: The `config.properties` configuration file

```
# Use forward slashes
screenfolder = C:/MVT/Images/
firefoxExe = C:/MVT/Browser/Firefox15/firefox.exe
chromePath = C:/MVT/Browser/chrome/
# "FF" or "Chrome"
browser = Chrome
heatmapFolder = C:/MVT/Heatmaps/
heatmapUserSessionFolder = C:/MVT/UserSessions/
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 WAR[4]-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`

[4] Web Module - `http://docs.oracle.com/javaee/5/tutorial/doc/bnadx.html`

- *MVTRegistration*: The MVT Registration page is a decoupled part of the MVTAnlysis 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 MVTAnalysis application.
- *MVTAnalysis*: The MVT Analysis application which was developed as a simple analysis tool for case studies 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 A.11: Tomcat configuration file `tomcat_users.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users>
        <role rolename="webapp" />
        <role rolename="webappadmin" />
        <user username="exin" password="some_secret_password" roles="webapp" />
        <user username="admin" password="some_secret_password" roles="webapp,
            webappadmin" />
</tomcat-users>
```

A.2.3 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 A.12: Check out using Mercurial Console

   ```
   hg init
   hg pull pathToRepository
   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 MVTService 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 A.2.2 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 MVTService 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 A.13: Proxy start command

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

10. *Use*: Configure the smartphones as described in A.2.1. 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.

List of Figures

Bibliography

[1] Massimiliano Albanese, Antonio Picariello, Carlo Sansone, and Lucio Sansone. 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, November 2004. ACM Press.

[2] The Internet Archive. Website. `http://archive.org/`, accessed June 2013.

[3] Living Web Archives. Website. `http://liwa-project.eu/`, accessed June 2013.

[4] William Y. Arms, Selcuk Aya, Pavel Dmitriev, Blazej J. Kot, Ruth Mitchell, and Lucia Walle. Building a research library for the history of the web. *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries - JCDL '06*, page 95, 2006.

[5] Samuel Brylawski. Preservation of digitally recorded sound. *Building a National Strategy for Digital Preservation: Issues in Digital Media Archiving*, page 52, 2002.

[6] Georg Buscher, Ralf Biedert, Daniel Heinesch, and Andreas Dengel. 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, 2010.

[7] MC Chen, JR Anderson, and MH Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. *CHI'01 extended abstracts on ...*, pages 281–282, 2001.

[8] Clicktale. Homepage. http://www.clicktale.com/, 2013.

[9] World Wide Web Consortium et al. Web accessibility initiative (wai), accessed June 2013.

[10] DCC. What is digital curation? `http://www.dcc.ac.uk/digital-curation/what-digital-curation`, accessed June 2013.

[11] Alex J. DeWitt. Examining the Order Effect of Website Navigation Menus With Eye Tracking. *Journal of Usability Studies*, 6(1):39–47, November 2010.

[12] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[13] Archana Ganapathi and Steve Zhang. 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, October 2011. ACM Press.

[14] Jesse James Garret. Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications, 2005.

[15] Antonio Gulli and Alessio Signorini. The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903. ACM, 2005.

[16] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, March 2004.

[17] IIPC. Website. `http://netpreserve.org/`, accessed June 2013.

[18] Masanes J. *Web archiving*. Springer, 2006.

[19] M. Jones and N. Beagrie. Preservation Management of Digital Materials: A Handbook, Digital Preservation Coalition, York. `http://www.dpconline.org/advice/preservationhandbook/introduction/definitions-and-concepts`, 2002.

[20] S Josephson and ME Holmes. Visual attention to repeated internet images: testing the scanpath theory on the world wide web. *Proceedings of the 2002 symposium on Eye ...*, 0535(March), 2002.

[21] M Khoo, Joe Pagano, and AL Washington. Using web metrics to analyze digital libraries. *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 375–384, 2008.

[22] Bernhard Krüpl-Sypien, Ruslan R. Fayzrakhmanov, Wolfgang Holzinger, Mathias Panzenböck, and Robert Baumgartner. 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, September 2011. ACM Press.

[23] Kyong-Ho Lee, Oliver Slattery, Richang Lu, Xiao Tang, and Victor McCrary. The state of the art and practice in digital preservation. *Journal of Research-National Institute of Standards and Technology*, 107(1):93–106, 2002.

[24] Raymond A Lorie. Long term preservation of digital information. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 346–352. ACM, 2001.

[25] Peter Lyman. Archiving the world wide web. *Building a national strategy for digital preservation: Issues in digital media archiving*, pages 38–51, 2002.

[26] Julien Masanès. Web archiving methods and approaches: A comparative study. *Library Trends*, 54(1):72–90, 2005.

[27] The Internet Memory. Website. `http://internetmemory.org/`, accessed June 2013.

[28] Gordon Mohr, Michael Stack, Igor Rnitovic, Dan Avery, and Michele Kimpton. Introduction to heritrix. In *4th International Web Archiving Workshop*, 2004.

[29] Network Working Group and Douglas Crockford. JavaScript Object Notation (JSON). http://www.ietf.org/rfc/rfc4627.txt, 2006.

[30] Library of Congress: Digital Preservation. Warc, web archive file format. `http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml`, accessed June 2013.

[31] Bing Pan, Helene A. Hembrooke, Geri K. Gay, Laura A. Granka, Matthew K. Feusner, and Jill K. Newman. 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, March 2004. ACM Press.

[32] Pandora. Website. `http://pandora.nla.gov.au/`, accessed June 2013.

[33] Xiaoguang Qi and Brian D Davison. Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)*, 41(2):12, 2009.

[34] Gergely Rakoczi. Cast your eyes on Moodle: An eye tracking study investigating learning with Moodle. *... of the 4th International Conference Moodle. si*, (May):203–213, 2010.

[35] Herman Chung-Hwa Rao, Yih-Farn Chen, and Ming-Feng Chen. A Proxy-Based Personal Web Archiving Service. *Operating Systems Review*, 35(1):61–72, 2001.

[36] Andreas Rauber and Andreas Aschenbrenner. Part of our culture is born digital-on efforts to preserve it for future generations. *TRANS-On-line Journal for Cultural Studies*, 10, 2001.

[37] Andreas Rauber, Andreas Aschenbrenner, and Oliver Witvoet. *Austrian online archive processing: analyzing archives of the world wide web*. Springer, 2002.

[38] Andreas Rauber and Max Kaiser. Web Archivierung und Web Archive Mining : Notwendigkeit , Probleme und Lösungsansätze. *Communications*, 2009.

[39] Carl Rauch and Andreas Rauber. Preserving Digital Media : Towards a Preservation Solution Evaluation Metric. pages 203–212, 2004.

[40] Frank Romano. E-books and the challenge of preservation. *Microform & imaging review*, 32(1):13–25, 2003.

[41] Myriam Ben Saad and Stéphane Gançarski. Archiving the web using page changes patterns: a case study. In Glen Newton, Michael Wright, and Lillian N Cassel, editors, *JCDL*, pages 113–122. ACM, 2011.

[42] Steven M Schneider, Kirsten Foot, Michele Kimpton, and Gina Jones. Building thematic web collections: challenges and experiences from the september 11 web archive and the election 2002 web archive. *Digital Libraries, ECDL*, pages 77–94, 2003.

[43] Dietrich Schuller. Preserving the facts for the future: Principles and practices for the transfer of analog audio documents into the digital domain. *J. Audio Eng. Soc*, 49(7/8):618–621, 2001.

[44] Stephan Strodl, Christoph Becker, Robert Neumayer, and Andreas Rauber. How to Choose a Digital Preservation Strategy : Evaluating a Preservation Planning Procedure. 2007.

[45] Mike Thelwall. A Fair History of the Web ? Examining Country Balance in the Internet Archive 1. *Library and Information Science*, pages 1–12, 2004.

[46] Kenneth Thibodeau. Overview of technological approaches to digital preservation and challenges in coming years. *The state of digital preservation: an international perspective*, pages 4–31, 2002.

[47] Herbert Van de Sompel, Michael L Nelson, Robert Sanderson, Lyudmila L Balakireva, Scott Ainsworth, and Harihar Shankar. Memento: Time travel for the web. *arXiv preprint arXiv:0911.1112*, 2009.

[48] Jeffrey Van der Hoeven, Bram Lohman, and Remco Verdegem. Emulation for digital preservation in practice: The results. *International journal of digital curation*, 2(2):123–132, 2008.

[49] Maja Vukovic, Soundar Kumara, and Ohad Greenshpan. Ubiquitous crowdsourcing. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*, pages 523–526. ACM, 2010.

[50] W3C. Document Object Model (DOM). http://www.w3.org/DOM/, 2005.

[51] Fusheng Wang, Xin Zhou, and Carlo Zaniolo. Bridging relational database history and the web: the xml approach. In *Proceedings of the 8th annual ACM international workshop on Web information and data management*, pages 3–10. ACM, 2006.

[52] Donald Waters and John Garrett. *Preserving Digital Information. Report of the Task Force on Archiving of Digital Information.* ERIC, 1996.