ANALYSIS OF LOW-LEVEL INTERACTION EVENTS AS A PROXY FOR FAMILIARITY

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences

2016

By
Aitor Apaolaza Llorente
School of Computer Science
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>10</td>
</tr>
<tr>
<td>Declaration</td>
<td>12</td>
</tr>
<tr>
<td>Copyright</td>
<td>13</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>14</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>15</td>
</tr>
<tr>
<td>1.1 Research questions</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Original Contributions</td>
<td>19</td>
</tr>
<tr>
<td>1.3 Publications</td>
<td>21</td>
</tr>
<tr>
<td>1.4 Discussion</td>
<td>23</td>
</tr>
<tr>
<td>1.5 Thesis Structure</td>
<td>25</td>
</tr>
<tr>
<td><strong>2 Background</strong></td>
<td>26</td>
</tr>
<tr>
<td>2.1 Longitudinal studies</td>
<td>27</td>
</tr>
<tr>
<td>2.2 Web Behaviours</td>
<td>30</td>
</tr>
<tr>
<td>2.2.1 Profiling</td>
<td>31</td>
</tr>
<tr>
<td>2.2.2 Single purpose</td>
<td>32</td>
</tr>
<tr>
<td>2.2.3 General browsing</td>
<td>33</td>
</tr>
<tr>
<td>2.3 Data capture</td>
<td>36</td>
</tr>
<tr>
<td>2.3.1 Task driven</td>
<td>36</td>
</tr>
<tr>
<td>2.3.2 Eye tracking</td>
<td>38</td>
</tr>
<tr>
<td>2.3.3 User feedback</td>
<td>39</td>
</tr>
<tr>
<td>2.3.4 Remote solutions</td>
<td>40</td>
</tr>
<tr>
<td>2.3.5 Privacy concerns</td>
<td>46</td>
</tr>
<tr>
<td>2.4 Interpretation of the data</td>
<td>48</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.4.1 Heuristics</td>
<td>48</td>
</tr>
<tr>
<td>2.4.2 Web analytic tools</td>
<td>50</td>
</tr>
<tr>
<td>2.4.3 Privacy and biometrics</td>
<td>53</td>
</tr>
<tr>
<td>2.4.4 Task models</td>
<td>55</td>
</tr>
<tr>
<td>2.4.5 Visual analysis</td>
<td>56</td>
</tr>
<tr>
<td>2.4.6 Longitudinal analysis</td>
<td>57</td>
</tr>
<tr>
<td>2.5 Presentation of the information</td>
<td>58</td>
</tr>
<tr>
<td>2.6 Weaknesses of current common techniques</td>
<td>65</td>
</tr>
<tr>
<td>2.7 Summary</td>
<td>70</td>
</tr>
<tr>
<td>3 Analysis Framework</td>
<td>73</td>
</tr>
<tr>
<td>3.1 Naturalistic capture of longitudinal Web interaction</td>
<td>74</td>
</tr>
<tr>
<td>3.1.1 Formative study</td>
<td>74</td>
</tr>
<tr>
<td>3.1.2 Web interaction capture solution</td>
<td>81</td>
</tr>
<tr>
<td>3.1.3 Data processing</td>
<td>87</td>
</tr>
<tr>
<td>3.1.4 Summary of interaction capture solution</td>
<td>97</td>
</tr>
<tr>
<td>3.2 Longitudinal analysis</td>
<td>97</td>
</tr>
<tr>
<td>3.2.1 Analysis input</td>
<td>98</td>
</tr>
<tr>
<td>3.2.2 Identified requirements</td>
<td>98</td>
</tr>
<tr>
<td>3.2.3 High-level construct extraction</td>
<td>102</td>
</tr>
<tr>
<td>3.2.4 Analysis techniques</td>
<td>103</td>
</tr>
<tr>
<td>3.2.5 Summary of longitudinal analysis</td>
<td>111</td>
</tr>
<tr>
<td>3.3 Micro Behaviours</td>
<td>112</td>
</tr>
<tr>
<td>3.3.1 Definition</td>
<td>113</td>
</tr>
<tr>
<td>3.3.2 Event input</td>
<td>113</td>
</tr>
<tr>
<td>3.3.3 Mouse micro behaviours</td>
<td>115</td>
</tr>
<tr>
<td>3.3.4 Scroll micro behaviours</td>
<td>116</td>
</tr>
<tr>
<td>3.3.5 Thresholds</td>
<td>117</td>
</tr>
<tr>
<td>3.3.6 Summary</td>
<td>122</td>
</tr>
<tr>
<td>4 Application of the Analysis Methodology</td>
<td>123</td>
</tr>
<tr>
<td>4.1 Familiarity</td>
<td>124</td>
</tr>
<tr>
<td>4.2 Design decisions for analysis</td>
<td>126</td>
</tr>
<tr>
<td>4.3 Analysis of Survey participants</td>
<td>128</td>
</tr>
<tr>
<td>4.4 Analysis of population</td>
<td>130</td>
</tr>
<tr>
<td>4.4.1 Consistent micro behaviours</td>
<td>134</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Inconsistent micro behaviours</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Usable micro behaviours</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Conclusion</td>
</tr>
<tr>
<td>4.5</td>
<td>Prediction of familiarity</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
</tr>
<tr>
<td>5</td>
<td>Design of a Web application to support interaction analysis</td>
</tr>
<tr>
<td>5.1</td>
<td>Interactivity</td>
</tr>
<tr>
<td>5.2</td>
<td>Longitudinal emerging sequences</td>
</tr>
<tr>
<td>5.3</td>
<td>Evolution of micro behaviours</td>
</tr>
<tr>
<td>5.4</td>
<td>Summary</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions and Future Work</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions</td>
</tr>
<tr>
<td>6.2</td>
<td>Limitations</td>
</tr>
<tr>
<td>6.3</td>
<td>Future Work</td>
</tr>
<tr>
<td>A</td>
<td>Additional Models</td>
</tr>
<tr>
<td>A.1</td>
<td>Survey participants</td>
</tr>
<tr>
<td>A.2</td>
<td>Homepage excluded</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
</tr>
</tbody>
</table>

Word Count:
47426
## List of Tables

2.1 Weaknesses of current common techniques ............................................. 72

3.1 Sample of events captured by the manual annotation ............................. 75

3.2 All events contain generic information. * indicates that this event is partially supported at the time of design ............................................. 86

3.3 Captured mobile events ........................................................................... 87

3.4 Top 10 of URL sequences found to be longer than 1 extracted from the use of cs.manchester.ac.uk out of 30023 sequences. There is a total of 10838 sequences longer than 1, from which 8117 sequences occurred only once .................................................. 101

3.5 Extracted Micro Behaviours ...................................................................... 121

4.1 Analysis of Micro Behaviours for Survey Participants ............................ 130

4.2 Distribution of users for different active time limits for the analysis of the controlled scroll micro behaviour ...................................................... 132

4.3 Analysis of Micro Behaviours for Phase 1 data in cs.manchester.ac.uk Web page .................................................................................................. 132

4.4 Analysis of Micro Behaviours for Phase 2 data in cs.manchester.ac.uk Web page .................................................................................................. 133

4.5 Comparison of the population correlations for survey participants, phase 1 data, and phase 2 data for the cs.manchester.ac.uk Web page ........................................ 133
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Diagram of the different levels of logging techniques</td>
<td>42</td>
</tr>
<tr>
<td>2.2</td>
<td>Diagram of common latency features employed in research into keystroke dynamics [Banerjee and Woodard, 2012]</td>
<td>54</td>
</tr>
<tr>
<td>2.3</td>
<td>Tool for definition of expected behaviour [Propp and Forbrig, 2009]</td>
<td>56</td>
</tr>
<tr>
<td>2.4</td>
<td>Path laid over a Web site in the VISVIP tool [Cugini and Scholtz, 1999]</td>
<td>60</td>
</tr>
<tr>
<td>2.5</td>
<td>WELFIT generated Markov chain of an interface object [de Santana and Baranauskas, 2010]</td>
<td>61</td>
</tr>
<tr>
<td>2.6</td>
<td>Visualisation of interaction events in a timeline by Carta et al. [Carta et al., 2011]</td>
<td>62</td>
</tr>
<tr>
<td>2.7</td>
<td>Dense pixel displays of database records [Shneiderman, 2008]</td>
<td>63</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of the KUPB Web application interaction. The user performs a basic search, selects results to be seen in more detail, and adds a filter. Each interaction’s number indicates its correspondence to the graph in Figure 3.2</td>
<td>78</td>
</tr>
<tr>
<td>3.2</td>
<td>Transitions between relevant actions in KUPB Web application. “Advanced Search” is being currently selected. Red lines indicate what actions are performed after “Advanced Search”, and green lines indicate what actions are performed before it.</td>
<td>79</td>
</tr>
<tr>
<td>3.3</td>
<td>Cumulative distribution function of inactive times (in seconds) on cs.manchester.ac.uk site.</td>
<td>88</td>
</tr>
<tr>
<td>3.4</td>
<td>Distribution of resulting episode lengths on cs.manchester.ac.uk site. A threshold of 40 minutes has been employed to calculate the episodes. The plot is bounded from 10 to 80 minutes.</td>
<td>90</td>
</tr>
<tr>
<td>3.5</td>
<td>Unbounded distribution of resulting episode lengths using a threshold of 40 minutes on cs.manchester.ac.uk site.</td>
<td>91</td>
</tr>
<tr>
<td>3.6</td>
<td>URL clusters according to their height and count of block elements from all accessed URLs on the cs.manchester.ac.uk site.</td>
<td>93</td>
</tr>
</tbody>
</table>
3.7 User distribution over accumulated active time in all URLs on the cs.manchester.ac.uk site. .......................... 95
3.8 Interface of the KUPKB Web application www.kupkb.org .................. 99
3.9 Timeline of a single user’s entire scroll interaction with cs.manchester.ac.uk homepage. .......................... 104
3.10 Plot of the scroll duration over time of 100 users on the cs.manchester.ac.uk homepage. .......................... 105
3.11 Density plots of users’ episode length corresponding to different active time bins of 30 seconds. .......................... 107
3.12 General linear models applied to each individual user showing evolution of users’ episode length over active time. .......................... 108
3.13 Plot of the linear mixed model for episode length time. .......................... 109
3.14 Information reported from repeated execution of a linear mixed model for different user samples, showing the evolution of the duration of users’ episode over time. .......................... 110
3.15 Example of analysis of Controlled Scroll Duration and speed using active time as the temporal metric. The use of two different temporal thresholds is compared for each feature. .......................... 119
3.16 Example of analysis of Mouse Inactive Time Median and Mean using active time as the temporal metric. The use of two different temporal thresholds is compared for each feature. .......................... 120
4.1 Responses to the questionnaire shown to first-time visitors. .......................... 125
4.2 Responses to the questionnaire shown to recurrent visitors. .......................... 125
4.3 Histogram of the residuals of a mixed linear model for episode duration. .......................... 127
4.4 Q–Q plot of the residuals of a mixed linear model for episode duration. .......................... 127
4.5 Visualisation of the resulting fitted values by the model against the residuals for a mixed linear model for episode duration using active time as the temporal metric. .......................... 128
4.6 Analysis of Mouse Click Speed for the survey participants. Limited to users with at least 2 occurrences. .......................... 130
4.7 Analysis of Mouse Idle After Click for the survey participants. Limited to users with at least 2 occurrences. .......................... 131
4.8 Analysis of Mouse Click Speed Time for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time. .......................... 139
4.9 Analysis of Mouse Median Inactive Time for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.10 Analysis of Mouse Mean Inactive Time for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.11 Analysis of Mouse Time To Click for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.12 Analysis of Mouse Idle After Click for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.13 Analysis of Mouse Click After Load for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.14 Analysis of Mouse Move Distance for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.15 Analysis of Controlled Scroll Duration for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.16 Analysis of Controlled Scroll Speed for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.17 Analysis of Episode Duration for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.

4.18 Percentage of episodes above each threshold for the values of the speed of the controlled scroll micro behaviour.

4.19 Percentage of episodes above each threshold for the values of the median inactive mouse time micro behaviour.

4.20 Percentage of episodes above each threshold for the values of the inactive mouse time micro behaviour and the speed of the controlled scroll micro behaviour.

5.1 Diagram comparing usage between different stages of expertise.

5.2 Example of the developed Web application to explore the evolution of micro behaviours.

A.1 Analysis of Mouse Inactive Time Median for the survey participants. Limited to users with at least 2 occurrences.

A.2 Analysis of Mouse Inactive Time Mean for the survey participants. Limited to users with at least 2 occurrences.

A.3 Analysis of Mouse Time To Click for the survey participants. Limited to users with at least 2 occurrences.
A.4 Analysis of Mouse Click After Load for the survey participants. Limited to users with at least 2 occurrences. .......................... 169
A.5 Analysis of Mouse Move Distance for the survey participants. Limited to users with at least 2 occurrences. .......................... 169
A.6 Analysis of Controlled Scroll Duration for the survey participants. Limited to users with at least 2 occurrences. ......................... 170
A.7 Analysis of Controlled Scroll Speed for the survey participants. Limited to users with at least 2 occurrences. ......................... 170
A.8 Analysis of Episode Duration for the survey participants. Limited to users with at least 2 occurrences. .......................... 171
A.9 Analysis of Mouse Click Speed Time excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 172
A.10 Analysis of Mouse Median Inactive Time excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 173
A.11 Analysis of Mouse Mean Inactive Time excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 174
A.12 Analysis of Mouse Time To Click excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 175
A.13 Analysis of Mouse Idle After Click excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 176
A.14 Analysis of Mouse Click After Load excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 177
A.15 Analysis of Mouse Move Distance excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 178
A.16 Analysis of Controlled Scroll Duration excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 179
A.17 Analysis of Controlled Scroll Speed excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 180
A.18 Analysis of Episode Duration excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time. 181
This thesis provides insight into long-term factors of user behaviour with a Web site or application using low-level interaction events (such as mouse movement, and scroll action) as a proxy. Current laboratory studies employ scenarios where confounding variables can be controlled. Unfortunately, these scenarios are not naturalistic or ecologically valid. Existing remote alternatives fail to provide either the required granularity or the necessary naturalistic aspect. Without appropriate longitudinal approaches, the effects of long-term factors can only be analysed via cross-sectional studies, ignoring within-subject variability. Using a naturalistic remote interaction data capturing tool represents a key improvement and supports the analysis of longitudinal user interaction in the wild. Naturalistic low-level fine-grained Web interaction data (from URLs visited, to keystrokes and mouse movements) has been captured in the wild from publicly available working live sites for over 16 months. Different combinations of low-level indicators are characterised as micro behaviours to enable the analysis of interaction captured for extended periods of time. The extraction of micro behaviours provides an extensible technique to obtain meaning from long-term low-level interaction data. 18 thousand recurring users have been extracted and 53 million events have been analysed. A relation of users’ interaction time with the site and their degree of familiarity has been found via a remote survey. This relation enables the use of users’ active time
with the site as a proxy for their degree of familiarity. Analysing the evolution of extracted micro behaviours enables an understanding of how users’ interaction behaviour changes over time. The results demonstrate that monitoring micro behaviours offers a simple and easily extensible post hoc approach to understand how Web-based behaviour changes over time. Results of the analysis have identified key aspects from micro behaviours that are strongly correlated with users’ degree of familiarity. In the case of users scrolling continuously for short periods of time, it has been found that the speed of the scroll increased as users’ become more familiar with the Web site. Users have also been found to spend more time on the Web site without interacting with the mouse. Understanding long-term interaction factors such as familiarity supports the design of interfaces that accommodate users’ interaction evolution. Combining found key aspects enables a prediction of a user’s degree of familiarity without the need for continuous observation. The presented approach also allows for the validation of hypothesis on longitudinal user interaction behaviour factors.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see [http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487](http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487)), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see [http://www.manchester.ac.uk/library/aboutus/regulations](http://www.manchester.ac.uk/library/aboutus/regulations)) and in The University’s policy on presentation of Theses.
Acknowledgements

First of all, I would like to thank my supervisors Simon Harper and Caroline Jay. Their continuous support, guidance, and more importantly, patience, were crucial for the research and the writing of this thesis. I also want to thank Markel. I would not be here if it weren’t for him, and I owe him his support as a colleague, mentor, and friend.

To my friends from the CDT, the drinks during periods of doubt and frustration have been vital to remain as positive as I could be, particularly during the final years.

I especially want to thank my father and family for their support while I was abroad. Although I couldn’t go back home as much as I wanted to, I always felt that they were proud of what I was doing.

I always describe doing a thesis as a sequence of good and bad moments. Enjoying the good moments, and persevering through the bad ones, has been important to maintain enough motivation to get to the end of this thesis. Having someone with whom to enjoy the good times, and to provide perspective during the bad periods made the endurance possible. For that, I have to thank my girlfriend, Aileen. Even though we were far away, she has always supported me to do what I wanted, come what may.
Chapter 1

Introduction

Understanding users’ interaction has always been an intrinsic part of user interface design. The necessity of adapting interfaces to users’ behaviour arose when computing became available to the general public and stopped being a niche speciality. To accommodate technology adoption by non-experts, the use of designs intuitive to users was required. Rather than forcing users to learn non-familiar interaction techniques, designers try to make interfaces that feel second nature. Factors such as consistency, familiarity, and self-description support the design of a more intuitive interface. Ultimately, the only way to design an entirely intuitive interface is adapting it to real users’ interaction behaviour. Adaptation of interfaces requires an understanding of the target audience motivation. Without an understanding of how the target audience interacts designers resort to speculation and prejudgement of users’ interaction behaviour. Long-term analyses are necessary to determine the adoption rate of an interface. Designers need tools providing them information about real usage with their interface and how it changes over time.

Many different tools and approaches have appeared in the last years. Some of them gather data from controlled environments providing detailed observations of participants’ interaction. Remote observations are an alternative that allow a more naturalistic capture from participants’ homes. The Web domain can benefit greatly from the use of remote observations. Web interfaces can be remodelled without interfering with the user. Therefore an iterative design of the interface based on a continuous observation of naturalistic usage of the Web site or application is possible.

The understanding of users has been explored by other domains in the context of ergonomic factors. Its relevance in computer science started when complex computers appeared in the 70s. Computers were difficult to handle and required the few people
using them to be experts. As they became more popular the necessity for systems
which were easier to learn was evident. This need can be noticed in software engineer-
ing research [Buse et al., 2011]. User evaluations in software engineering publications
have grown 500% at the top venues. Observing and measuring users’ interaction is
becoming increasingly important in the academic domain, as can be seen in major
conferences like CHI. The percentage of submitted papers presenting technologies and
not including evaluation with users has dropped from nearly 40 in 1983 to less than 3
in 2006 [Barkhuus and Rode, 2007]. New design paradigms focused on users’ needs
can also be found. User centred design is an application designing paradigm in which
an iterative cycle involving usability evaluation helps improve the final product of a
development before it gets released. It takes into account user participation in the pro-
cess, so it ensures users’ expectations are met. Agile development is common in these
approaches, as it enables the introduction of usability testing of prototypes in the it-
erative process. Even when the product has been released, debugging tools take into
account usability problems. For example, time lags in the interaction can be discovered
and analysed to determine if they hinder users’ interaction [Jovic et al., 2011].

Looking for possible improvements in user centred design is a complex task. Ob-
jective metrics as effectiveness and efficiency can be used to measure interaction with
an interface. Effectiveness relates to task completion and quality, including among
others task completion success ratios, the number of errors, and quality of outcome.
Efficiency relates to the amount of resources spent, such as time and mental effort.
Subjective measurements, on the other hand, rely on users’ opinion. Insight into users’
motivations and feelings helps understand their interaction. Users’ satisfaction and at-
titude toward the interface can be measured via the use of surveys [Hornbæk, 2006].
Surveys can give qualitative and quantitative data and valid feedback about users’ opinion
and usability errors. Questions can include users’ self-reports of their degree of
skill or familiarity making comparisons between different stages of interaction pos-
sible. However, user feedback is not perfect. What users report can be different from
what they did as they need to rely on their memory [Hassenzahl and Sandweg, 2004].
Certain observational techniques such as the think-aloud protocol and involving evalu-
ators help to provide a less biased view of the interaction, but these techniques are not
feasible for a longitudinal study.

Techniques have evolved from using manual methods, which imposed a high cost
in time and resources, to automatic ones. There has also been a shift towards remote
evaluation. It allows more participants to take part in the studies for a small price of
lack of control. Another advantage is that it leads to a more naturalistic approach, avoiding usual laboratory environments with specific set ups. Although remote evaluation poses practical benefits, it still has its disadvantages, caused by the lack of control over the experiment. Certain situations remain out of the control of the researcher, such as users leaving the computer unattended to perform some other unrelated task.

Rather than relying on access to users’ opinions and motivations, low-level events can be analysed. Low-level interaction relates to direct user input, providing raw access to how users interact with the Web. Analysis of users’ input has been found to be useful to predict moods [Zimmermann et al., 2006]. Thus, internal experiences manifest in the way users interact. Particular aspects of the interaction can be extracted from low-level events. Finding a correlation between particular interaction aspects and users’ internal experiences makes the use of low-level events as a proxy possible.

Long-term factors can also be explored via the discovery of subtle differences between different points of interaction. This way interaction aspects correlated with temporally sensitive factors, such as familiarity, can be discovered.

1.1 Research questions

The goal of this thesis is to explore the analysis of interaction behaviour as a scalable and unobtrusive alternative to existing methods. Laboratory studies and qualitative methods are not feasible approaches to longitudinal analysis. The traditional way of measuring skill, familiarity, and other long-term factors is through user interviews. These factors are then commonly used to group users accordingly. Self-reports rely on users’ own perception, which might be flawed, and laboratory studies are necessarily bounded in time. Longitudinal analysis of interaction events can provide insight into how interaction behaviour changes over time. Low-level Web interaction data can provide insight into subtle changes in users’ behaviour over time, serving as a proxy for higher level concepts, such as moods or skill.

This thesis aims to answer a set of questions, tackling current gaps identified in existing approaches to longitudinal analysis of Web interaction. The possibility of exploring long-term factors of users’ Web interaction through the analysis of longitudinal low-level interaction data is explored. Long-term factors can then be linked to higher level concepts concerning users’ motivation. Supporting such analysis requires a way to process and aggregate low-level interaction data into comparable units. Temporal
correlations across those interaction units need to be found, and mapped to the explored higher-level concept. To support this process a steady evolution over time is necessary so the longitudinal analysis of the data is justified. Even if such evolution exists, the data employed for the analysis need to cover the evolving aspects.

**R1** Can longitudinal low-level interaction data be analysed without losing its fine-granularity? Longitudinal low-level interaction has the potential to provide insight into subtle changes in user behaviour. Avoiding the use of coarse aggregation is challenging due to the extent of the observation. A way of aggregating and analysing low-level interaction without losing its fine-grained features complements existing techniques.

Current approaches provide the means to analyse low-level interaction data captured for short-periods of time. Task-driven approaches employ predefined metrics and prejudgements of the interaction in order to obtain comparable units of interaction. Prejudgements introduce bias into the analysis, and the use of coarse interaction statistics – such as page-view sequences from Web logs – cannot provide the granularity necessary to discover subtle changes in interaction over time.

*Micro behaviours* are presented as an extensible approach to obtain comparable fine-grained interaction units. These interaction units are formed by a combination of low-level events. Section 3.3 presents the design of micro behaviours and the features that can be extracted from them. Longitudinal analysis of the extracted features supports a scalable approach to obtaining insight into subtle changes in user Web interaction over extended periods of time.

**R2** Does users’ interaction behaviour change over time? Can these changes be inferred from low-level interaction data? Analysis of low-level Web interaction data enables the detection of subtle interaction behaviour changes over time. High-level aspects of interaction behaviour, such as motivation, can only be explored using obtrusive techniques, such as controlled environments. Changes in behaviour can also affect the way users interact with input devices causing a change in low-level interaction data, such as scroll and mouse movement.

Controlled observations are limited in time, or introduce bias in the observation. Naturalistic observations for extended periods of time are necessary to answer this question. Existing techniques employ high-level constructs to simplify the analysis and define optimal interaction models, thus prejudging users’
interaction. A data-driven analysis of low-level data prevents prejudgements and enables the emergence of indicators of interaction behaviour evolution.

The capture solution described in Section 3.1 has been deployed in public Web sites and applications. This solution has captured low-level Web interaction events over a period of 16 months. Application of the analysis presented in Section 3.2 identified particular aspects of user interaction that change over time. Results of this analysis are presented in Chapter 4.

R3 Can discovered interaction changes be mapped to higher-level concepts of user interaction? Previous research question describes how high-level concepts of interaction might affect the way users interact. If so, low-level interaction aspects linked with high-level concepts exist and can be found. Changes over time are user dependent, requiring continuous observation of users. Low-level interaction aspects can be used as a proxy for the explored high-level concept without the need for continuous observation.

Previous work fails to provide naturalistic longitudinal observations of user interaction. Extended observations are necessary to provide insight into long-term factors. A longitudinal analysis can identify the key aspects of interaction that change over time. Found key aspects can then be used as a proxy for the explored high-level concept if it is also found to increase over time.

Familiarity has been selected as a long-term high-level usability concept to analyse. Obtaining insight into familiarity is challenging when using existing approaches as continuous observation of users over extended periods of time is impractical. It has been hypothesised that users’ degree of familiarity is positively correlated with the amount of time they spend on the Web site or application. Answers from users to a questionnaire deployed on a publicly available Web site have supported this hypothesis. Based on this hypothesis, temporally correlated micro behaviours have then been used as a proxy for familiarity. Details on this approach can be found on Chapter 4.

1.2 Original Contributions

The design of a methodology to extract micro behaviours from longitudinal low-level interaction data. One of the key challenges of processing low-level interaction
data captured for extended periods of time is finding a way of converting it into meaningful data. Usual approaches resort to splitting the data into episodes and reporting aggregated statistics such as the number of clicks. Such approaches provide a manageable summary of the data but lose meaningful low-level aspects of the interaction. Section 3.3 presents the use of micro behaviours as an alternative to make the analysis of meaningful aspects of low-level interaction possible.

**The design of an entirely naturalistic capture solution of longitudinal data.** Existing interaction capture solutions either support short-term observations or provide aggregations of interaction data. The designed solution presented in Section 3.1 provides a remote, unobtrusive, longitudinal, low-level interaction capture system. It can be seamlessly deployed on any Web site or Web application and it has been successfully employed in research. This capture solution can easily be extended to capture tailored events (or new standard Web events) and is scalable to thousands of different users. Existing interaction capture solutions either support short-term observations or provide aggregations of interaction data.

**A technique to analyse longitudinal low-level interaction.** Processing millions of low-level interaction events is challenging, particularly when a heterogeneous collection of events is being processed. The uneven distribution of events across users represents additional challenges, as the main objective of the analysis is shedding light into how single users’ interaction behaviour changes over time. The implemented analysis presented in Section 3.2 takes into account these challenges, and enables the discovery of changes in interaction behaviour in a reliable manner.

**The discovery of low-level interaction aspects that serve as indicators of users’ degree of familiarity with a Web site.** Discovered interaction aspects are related to extracted micro behaviours, as well as other aspects of the interaction such as episode durations. In the case of sequences of continuous scroll interaction, the speed of the scroll action has been found to be strongly correlated with users’ degree of familiarity. Episode duration and the length of periods without mouse activity have also exhibited strong correlations with users’ degree of familiarity. These results have been presented in Chapter 4.
1.3 Publications

The work presented in these thesis has been published in peer reviewed publications. The work can be categorised according to their stage in the process of achieving the overall goal of this thesis:

1. **W4A’13 [Apaolaza et al., 2013]** A way of capturing suitable data for the projected longitudinal analysis is designed and implemented. Section 3.1 is based in this work. The Web interaction capture solution design and implementation are described.

2. **UIST’13 [Apaolaza, 2013]** Captured data and its possibilities are explored. Some prospective results are also presented, introducing possible future work. Section 3.1.3 is based on this work where challenges encountered when processing interaction data captured over extended periods of time and the possibilities of the analysis of such data are explored. Chapter 5 presents different visualisation options considered for the visual analysis of the data. The Web application presented in that chapter includes enhanced versions of the visualisations presented in this publication. In Section 3.1.1 the emergence of tasks guided by identified high-level events presented in this paper is considered. The Section 5.2 explores the longitudinal possibilities of the analysis of emerging tasks.

3. **HT’15 [Apaolaza et al., 2015]** A longitudinal analysis methodology is designed to provide insight into long-term factors. Initial results fulfilling the goal of this thesis are presented. Section 3.2 introduces the design of these algorithms. The advantages of the designed analysis techniques are presented, along with their possible applications. Encountered challenges are also mentioned, such as the original drawbacks of the analysis and how they are tackled. The application of this analysis produced the results presented in Chapter 4.

The content of each paper is presented below. For each publication, the core contribution is presented, along with a summary of the work.

**W4A’13** *Understanding users in the wild* in the Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility. An appropriate capture solution is required to gather data supporting a fine-grained longitudinal analysis. In this publication, the design of such solution is presented, along with the motivation for its creation. This capture tool has been shown to be useful not
only for the work presented in this thesis but also for other research in similar domains.

Summary: Laboratory studies are a well-established practice that present disadvantages in terms of data collection. One of these disadvantages is that laboratories are controlled environments that do not account for unpredicted factors from the real world. Laboratory studies are also obtrusive and therefore possibly biased. A tool that is easily deployable in any Web application and captures longitudinal interaction data unobtrusively is presented. It enables the observation of accessibility-in-use and guides the detection of emerging tasks.

UIST'13 Identifying emergent behaviours from longitudinal web use in the Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology. The inference of high-level events from low-level interaction enabled the exploration of emerging task-models. This work presents possible visualisations and applications of the data captured by the capture solution designed for this project. Although the project is focused on the temporal aspect of users’ interaction behaviour, these visualisations helped on the design of the capture solution. The utility of unobtrusive interaction capture to improve Web design is supported and enabled the later design of visualisations providing insight into how users’ interaction behaviour changes over time.

Summary: Low-level interaction data captured in-situ from real world Web applications is visualised through an interactive Web application. The transition between interaction options can be explored, enabling the discovery of emerging task models. As opposed to traditional interface design, focused on the design of an idealised task model, this approach supports the design for real usage of the interface. Similarities among users can be found and solutions that are effective for real users can be designed. An example of the exploration results with one Web application is presented.

HT’15 Longitudinal Analysis of Low-Level Web Interaction through Micro Behaviours in the Proceedings of the 26th ACM Conference on Hypertext & Social Media. Results obtained from the analysis of micro behaviours are presented. The presented work focuses on providing insight into how users’ interaction behaviour changes with respect to their degree of familiarity. Results support the premise that low-level interaction behaviour can be used as a proxy for users’ degree of familiarity with a Web site or application.
Summary: To truly understand how people learn to navigate and use a Web site or application, real usage data needs to be collected over extended periods of time. Detailed Web interaction data is gathered in the wild, providing an in-depth, ecologically valid view of interaction, and enabling an understanding of how behaviour evolves over time. At the core of this approach is the aggregation of low-level interaction data into micro behaviours. A longitudinal data-driven analysis of fine-grained interaction data captured from 14,000 recurrent users over 12 months. Monitoring micro behaviours has been found to be a suitable approach to understanding how Web-based behaviour evolves over time.

1.4 Discussion

I argue that even if the use of task-driven approaches provides designers with knowledge about how users’ interact with their interfaces in a fast and cheap way, they only take into account a subset of all the possible known interactions in a Web site or application. They show a prejudgement of the user interaction, limiting the evaluation to the task models identified beforehand. In the cases where fine-grained user interaction is analysed temporal aspects are not taken into account. In the case of longitudinal approaches, high-level interaction data is summarised, losing important low-level interaction behavioural aspects. The approach proposed in this thesis provides understanding on how users’ interaction behaviour changes over time. It employs low-level interaction data (from URLs visited, to keystrokes and mouse movements) unobtrusively captured from the users remotely while they interact with the Web. Being unobtrusive avoids capturing biased data from the users, allowing them to interact freely. The remote character of the tool allows to observe users in the ecological environment of their homes. The inclusiveness of this approach enables the observation of any kind of users, including the ones with special requirements, like physically disabled users whose mobility is reduced, or users requiring special assistive technologies to be able to access the Web site.

As opposed to cross-sectional studies, presented approach follows the same users for extended periods of time. Changes in interaction behaviour are analysed to understand how interaction changes over time. The naturalistic interaction capture tool has been deployed in publicly available working live sites. 53 million low-level fine-grained Web interaction events have been extracted from 18 thousand recurring users. As a way to process that amount of interaction events, these have been grouped into
micro behaviours. Micro behaviours are higher level abstractions of interaction data representing small-scale behaviours, such as sequences of scroll, or short periods of mouse interaction. These micro behaviours have been based on past research [Vigo and Harper, 2013] as well as manual analysis of captured low-level interaction logs. Different interaction aspects from these micro behaviours have been analysed to find correlations with the time users spent on the Web site. Extracted interaction aspects depict low-level interaction behaviour, such as the speed at which users scroll down during a page scan and periods of lack of mouse activity. Discovered temporal correlations are explored as a way to obtain insight into higher-level concepts. Familiarity is studied as a prospective high-level concept that has long-term effects. A remote survey has been conducted, supporting the hypothesis that users’ interaction time with the site and their degree of familiarity are positively correlated. Results of the analysis have identified key aspects from evolving micro behaviours that are strongly correlated with users’ degree of familiarity. In the case of users consistently scrolling for short periods of time, it has been found that the scroll speed increases as users’ become more familiar with the Web site. Users have also been found to spend more time on the Web site without interacting with the mouse.

Monitoring micro behaviours offers a simple and easily extensible post hoc means of understanding how Web-based behaviour evolves over time. Additional micro behaviours can be easily designed, and different hypothesis about temporal factors in user interaction can be tested. Aggregation of interaction into high-level concepts, such as page-views, can mask underlying subtle features of interaction. Micro behaviours provide features concerning low-level interaction supporting scalable analysis of highly detailed interaction data. Therefore subtle changes in user interaction can be discovered through the analysis of micro behaviours. High-level concepts of user interaction have an effect on how users interact and this effect is reflected in the captured low-level interaction data. The work presented in this thesis employs micro behaviours to discover temporal correlations over extended periods of time. An analysis methodology has been designed to identify robust consistent temporal correlations in a conservative manner. Identified micro behaviours exhibiting temporal correlations are then mapped to the high-level concept of familiarity. Values from the discovered micro behaviours have then be used to predict a user’s degree of familiarity. Presented analysis provides a scalable and cost-effective approach to obtaining insight into the effect users’ degree of familiarity has in the way they interact.
1.5 Thesis Structure

Chapter 2 The goal of this thesis is filling the existing gap in longitudinal analysis of user interaction. Existing techniques are not feasible approaches to perform longitudinal studies, lacking the required granularity or not providing long-term observations. Existing analysis approaches rely on prejudgements of the interaction, and do not account for longitudinal fine-grained analysis that would highlight subtle changes in user interaction.

Chapter 3 An analysis framework is presented to capture low-level interaction data over extended periods of time and analyse it. An scalable and easy to deploy observation tool for Web sites and applications is implemented. A suitable analysis methodology is proposed to obtain insight into how interaction changes over time. Micro behaviours are presented as technique to aggregate interaction into comparable units. The use of micro behaviours enables a longitudinal analysis without disregarding subtle interaction changes.

Chapter 4 To test if the proposed analysis framework provides insight into long-term high level behaviour concepts the analysis is applied to the long-term factor of familiarity. The use of low-level interaction data as a proxy to the explored factor was found to be valid. Various micro behaviours were found to be temporally correlated with familiarity.

Chapter 5 Application of the analysis framework is complex, and exploration of the different micro behaviours with the different possible conditions is cumbersome. A Web application is implemented to ease this process and provide researchers a tool to easily explore captured longitudinal data. Interactive exploration of provided visualisations of longitudinal interaction data helps to explore long-term factors. Extraction of high-level events enables the discovery of emerging behaviours and their evolution over time.
Chapter 2

Background

One of the key characteristics of the work presented in this thesis is about tackling the lack of naturalistic longitudinal studies of user interaction. The definition of longitudinal is not clear as published research presents longitudinal studies of various lengths. This issue is discussed further in Section 2.1. The solution proposed in this thesis makes use of micro behaviours as a way to support fine-grained longitudinal analysis of Web interaction. Related work on inferring behaviour from user interaction has been reviewed, and is presented in Section 2.2.

Similar to any other experiment set-up, the presented approach includes a way to capture the data, a way to interpret the data and finally a way of presenting the results. With that structure as a premise, the rest of the chapter is organised into three sections covering research done for each part in the domain of the project. Section 2.3 presents existing techniques on capture of interaction data. Section 2.4 presents existing research on interpreting and analysing interaction data. Section 2.5 presents existing approaches to visualising the results of the interpretation and analysis of the data.

Section 2.6 presents a summary of the existing techniques to obtain insight into Web interaction followed by a discussion of the main disadvantages identified in existing research. These disadvantages need to be tackled to enable the proposed naturalistic longitudinal analysis of Web interaction data. Finally, related work has been classified in Table 2.1 indicating for each technique the particular identified weaknesses.
2.1 Longitudinal studies

Longitudinal studies are understood to last for extended periods of time. However, the length of these periods has not been clearly defined as it might need to vary depending on the purpose of the study. Longitudinal studies have been found not to have a generic length measure across domains. Some studies can take decades with large population samples – national census statistics – while others can take few weeks, following a specific group of people – such as the health of firemen after being exposed to noxious fumes [Coggon et al., 2009].

In the domain of anthropology, longitudinal studies are of paramount importance. They rely on long-term studies based on repeated visits as short periods cannot give enough insight into complex cultural patterns. The observation needs to be long enough to observe culturally relevant events such as weddings, or ceremonial burials. Researchers are likely to start with a long initial observation period of 1 or 2 years, to then come back for shorter visits in subsequent years [Holland et al., 2006].

Longitudinal studies are common in the domain of medicine. Longitudinal observations in medicine employ studies following hundreds of patients over decades. The length of these studies can cause problems, such as the death of participants during the study, reducing the available sample over time [van Weel et al., 2006]. Shorter studies can focus on temporally bounded observations, such as observing social phenomena during a particular period. For example, how the limited visibility of newly published medical research in local newspapers affects the creation of new policies [Bartlett et al., 2002].

In the domain of epidemiology, cross-sectional studies can be employed to identify possible factors of a particular problem or disease. Longitudinal studies are then commonly employed to find the relationship between the identified factors and the persistence of the studied problem. However, no rationale for the length of the study is usually given [Morphy et al., 2007]. Particular scenarios allow constraining the observation, so comparisons between varying factors are possible. For example, in the case of HIV-infected patients is common to follow patients for 48 weeks after the initiation of a particular treatment [Brenchley et al., 2006]. Instead of defining a predefined length, a longitudinal measure system can be put in place, without any a priori knowledge of the duration of the study. The continuity of the data collection provides the means to analyse longer consequences of studied factors [Buck, 2002].

In the domain of HCI a wide variety of work self-labelled as longitudinal can be found. These studies differ dramatically with respect to the length, ranging from as
CHAPTER 2. BACKGROUND

Little as five days [Sporka et al., 2007] up to several years [Krishnamurthy and Wills, 2009]. Longitudinal studies were commonly found to be repeated experiments over a period. Most of them use the same participants and are based on observations carried out across several laboratory sessions.

The variety of lengths found among longitudinal studies can be categorised according to their duration [von Wilamowitz-Moellendorff et al., 2006]. **Micro studies** consist in short-term studies of several days. **Meso studies** are longer, and take weeks or months. **Macro studies**, the longest, take years to finish. The length of the observation has been mentioned to be dependent on the purpose of the study. Nevertheless, finding the appropriate length to fulfil the goal of the study can help shortening unnecessarily long observations. For example, three months has been considered to be the most discriminative threshold to differentiate between experts and novice [Novick et al., 2012]. Rather than users becoming experts after three months, this threshold represents the boundary by which the data could be split into two groups. If three different degrees of skill had been defined, different boundaries would have arisen.

Longitudinal studies have gained importance over time, leading to the organisation of several workshops at the CHI conference. The nature of longitudinal studies was explored, and a unified concrete definition for longitudinal was discussed [Vaughan and Courage, 2007, Vaughan et al., 2008, Courage et al., 2009, Jain et al., 2010, Karapanos et al., 2012]. It was agreed that the goal of longitudinal research is to look beyond the initial user experience, with any qualitative or quantitative method. To categorise a study as a longitudinal study a plan to analyse data collected over time is necessary. A set of dimensions must remain constant in order to make comparisons over time possible – such as using the same set of participants with a similar observation set-up. The method and length of the study depend on the research questions, rather than the longitudinal nature. For example, to determine the impact of the introduction of a new tool, the study should occur before the novelty of that tool has ended [Vaughan et al., 2008].

In the case of interfaces in which the performance can be quantified, the longitudinal study can stop when the desired level has been reached. Lengthy observations are expensive, and obtaining long-term observations poses difficulties that may not be worth the results. One of the main concerns raised in the mentioned workshops discussing longitudinal studies was convincing stakeholders of the value of longitudinal analyses. The lack of a certain outcome can make longitudinal studies appear risky.
to stakeholders. Participant drop-out is another risk, which can be tackled by over-recruiting candidates and providing incremental incentives. The majority of laboratory studies, on the other hand, are limited to short-term observations of a set of users for a few days, with a clear outcome.

Laboratory studies have shown that users’ evolution is not constant over time. Acquired knowledge fades during inactive periods and needs to be refreshed between sessions. A way to obtain a virtually continuous observation of user interaction is discarding the first sessions from each day [Card et al., 1987]. It is known that the learning curve is a power function, so less change is expected towards lengthier observations.

Traditional longitudinal analysis considers each observation as a “wave of data”, and considers three waves the minimum to discern non-linear temporal correlations [Singer and Willett, 2003]. In the case of user interaction a user’s episode can be considered a wave of data. To do so, a way of determining an interaction episode is necessary, as its length lacks a unified definition. The length of the episodes can be either enforced periods on users – interact with a particular interface for a certain amount of time, or until certain task is fulfilled – or precise measurements of interaction without temporal constraints – users interacting freely with an interface for as long as they want. In laboratory experiments an episode can be defined as the interaction period in which the user is fulfilling a given task. In remote continuous approaches, the necessity for splitting captured data apart arises. Data cannot be securely split in discrete episodes, as users might interact on several occasions throughout the same day.

The need to split interaction data can be seen in Google Analytics[1], one of the most common tools employed for Web site analytics. In this tool the term session is used as “a group of interactions that take place within a given time frame”. The concept of session is similar to the introduced “interaction episode”, so techniques employed to split sessions can be applicable to discern episodes. Google Analytics employs three different rules to separate sessions: session time-out after 30 minutes, expiration of sessions at midnight, and sessions starting via the use of a different campaign – e.g. via a new search or accesses through a different link [Google, 2014]. Session timeout is a measure generalisable to Web interaction, but determining its length is not trivial. Determining the particular duration for the time-out can be challenging, as using predefined threshold values can bias captured data. High values group different

episodes together while short values might split long episodes. If the session time-out is bigger than one day, then users returning everyday would end up producing unrealistic long episodes \cite{Arlitt2000}. The episode time-out can be approximated by using a heuristic specific to the task being performed.

In the case of Web search tasks, the optimal time threshold has been found to be between 10 and 15 minutes as a result of the analysis the distribution of the episodes’ lengths \cite{He2000}. The use of task specific features, such as similarity between queries, can also be used to train a predictor. That way the use of a threshold is discarded, and instead the queries’ syntactic and temporal cues are used to determine when a task, subtask, or goal has ended \cite{Jones2008}. In the wild, where no tasks are known, the use of a set threshold remains common practice, as it has been found that differences in values between 5 to 60 minutes are negligible \cite{Thomas2014}.

One common problem when selecting a discriminative episode time-out is that resulting episode lengths show a bias around the selected value. These biases can be appreciated when different time-out values are used, looking for a drop around the selected value in the distribution of the resulting episode lengths. This effect is minimised when higher thresholds are employed \cite{Zakay2013}. Instead of a generic threshold, user-specific thresholds can be employed \cite{Mehrzadi2012}. As the threshold varies between users, introduced bias is not visible among all episode lengths. However, the lack of bias among users remains to be tested. The lack of ground-truth is the main problem in the validation of these thresholds.

\section{2.2 Web Behaviours}

The use of micro behaviours is suggested as a way to analyse longitudinal interaction data without losing its fine-grained aspect – more details on Section 3.3. Previous work on extraction and analysis of behaviours employed resulting behaviour to classify users – see Section 2.2.1. These classifications provide profiles that can be used to optimise Web pages to the discovered behaviours. A deeper insight into user behaviour can be obtained in the cases of single purpose Web sites or applications. Section 2.2.2 presents work where the analysis is narrowed down to a set of tasks to understand the way users achieve their goals. In the case of general browsing, analysis of sequences of Web pages requests help to understand and predict the way users interact with Web
sites. This way Web behaviours can be extracted and users’ motivation can be understood. Work exploring the possibilities of predicting users’ motivation of various granularity levels – from the fulfilment of goals to pinpointed situations of frustration – are presented in Section 2.2.3.

2.2. Profiling

The use of raw Web logs provides access to the series of page requests from users. Combining the sequences of Web pages with the information contained in them provide information about usage trends. That way user visits can be classified according to the topics of the visited pages, so visitors’ usage of the Web site can be analysed [Heer and Chi, 2002].

As an alternative to between-user clustering, variability within users can also be used to quantify how consistent visitors’ browsing behaviour is. On the one hand, visitors with low variance have consistent interaction patterns and tend to follow a direct path from query submission to problem resolution. On the other hand, high variance indicates that interaction patterns are variable, with visitors accessing many new domains to complete their goal [White and Drucker, 2007]. Additional categories have been found based on users’ browsing patterns. [Tossell et al., 2012] identified 8 of them from existing research, before presenting a classification of mobile users based on their preference for native mobile applications or Web browser when accessing the internet.

Instead of classifying users according to their browsing behaviour, the purpose of such browsing can be analysed to determine how it fits their everyday life [Lindley et al., 2012]. Five models of use have been identified: respite (procrastination), orienting (everyday routine activities), opportunistic (being online encouraged participants to perform opportunistic tasks), purposeful (open the browser for a specific task) and lean-back (watch or listen to online content).

Data sources for behaviour analysis can either be site-centric or user-centric. In the former case, data from a single site is analysed. It prevents privacy concerns such as being able to identify users across different sites for profiling. In the latter case, data for the same user from various sites is available. Proxies can be employed to generate Web logs to all Web requests from the users connected to the proxy. These data can then be used to predict users’ requests, for example, to enhance the data preprocessing [Xu et al., 2013]. Another advantage of user-centric approaches is that it can help predicting the behaviour of users in Web sites even if it is the first time they visit them.
A user’s behaviour during a first visit to a site can be predicted from data captured from data captured from other sites [Park and Fader, 2004].

### 2.2.2 Single purpose

Behaviour extraction can be simplified by narrowing the analysis down to a **single purpose Web site or application**. One common task in the Web is the search of information, via search engines. Research in this domain looks for ways to get the best results for the queries submitted by users. Users of *Search Engine Result Pages* (SERP) have been coarsely classified into three categories [Broder, 2002] depending on their intention when using a SERP. Navigational users look for a particular Web site, informational users look for information about a certain topic, and transactional users perform some Web-mediated activity. An example of the latter is downloading certain types of files or shopping. Identifying the task at hand can be useful to accommodate the interface to it, but it can be challenging without interviewing the users. Queries can be reviewed and manually annotated, as they usually provide enough information [Rose and Levinson, 2004]. For example a query of “facebook” obviously denotes a navigational intent, while “what do turtles eat” is clearly informational. Even if the same tools are employed to fulfil both purposes, the information requirements are clearly different, allowing for optimisations. The information provided along with the found results of the query can mislead the users and impact the efficiency of the search. In informational queries, the snippet can be relevant, as it can give the answer to the query. In navigational queries, on the other hand, it can take users’ attention away from the result’s URL, which may have been more meaningful [Cutrell and Guan, 2007].

Studying individual users’ behaviour can give hints about what result the user may click, allowing the ease of the loading process by pre-caching prospective Web pages. Query sequences can serve as guidance to discern what the user would most probably select [Downey et al., 2007]. A finer picture of a user’s behaviour can be obtained by analysing the focus of attention. Mouse capture provides information about the user inspecting certain results without clicking them, or using the mouse to guide the exploration of the results page. Users’ interest can be inferred from this information, which could be employed as another ranking criterion for the results, or to infer elements of interest in the results page, such as advertisements [Guo et al., 2009]. One disadvantage of mouse capture is that different users may use it in a different way. Some users may use the mouse as a guide for reading, providing at all times their attention focus. Alternatively, users could be using the mouse to directly select the
desired result from the results page, only providing the information equivalent to page requests. Eye tracking, on the other hand, ensures that the focus point of the user is provided at all times. It provides a detailed view of how users scan the results, showing if the users carry on scanning results even after clicking one of them [Rodden et al., 2008].

Behaviour analysis focused on single purpose Web applications outside the SERP domain has explored how users interact with particular Web interfaces. Predefined interface specific high-level events can be extracted to provide insight into how users employ available tools. Use frequency of different menus and interaction with page elements can be extracted as behaviours. Survey specific interaction, such as selection of different options from a dropdown menu, can help identify interface errors such as questions with confusing wording [Stieger and Reips, 2010]. Other interactive functionalities that may be considered useful can in reality decrease users’ confidence level, resulting in a higher rate of incorrect answers [Breslav et al., 2014]. If the number of identified high-level functionalities is small, sequences of actions can be clustered to categorise different tasks [Lafreri et al., 2010]. Unfortunately, such clustering is only applicable when approximation of individual tasks is possible.

### 2.2.3 General browsing

Extracting behaviours that can be generalised to free interaction can be challenging. Generic behaviours such as browser use and recommended loading times [Weinreich et al., 2006b] can serve as recommendations to accommodate the majority of users, but provide limited information about how users behave while browsing specific pages. In the same way as in SERPs, there have been approaches to classifying users according to their purposes, such as everyday tasks in a routine fashion, or specific individual tasks. Depending on the browsing intention, users may use different devices. Users who start browsing the Web to perform a specific task usually make use of the device that poses the lowest barrier of entry. For example, for tasks such as looking for a particular information or to occupy themselves while waiting, users may use a mobile phone rather than turning on the computer [Lindley et al., 2012]. Classifications of such generic behaviours are difficult to perform and require user input to determine users’ intention. In the case of Web interaction instead of exploring users’ behaviour with task-specific Web sites, interaction with elements that are common to different Web pages can be analysed. This way a model generalisable to every Web site containing those elements can be obtained. Users’ behaviour with generic banner ads can
be considered. This way relevant generalisable features that have an effect on users’ attention can be found [Jay et al., 2013]. In the case of banners, it was determined that the way the banner was triggered was the most relevant feature to infer if the element would be noticed.

Analysis of the sequences of Web pages followed by individual users enables the discovery of browsing strategies, such as orienteering, where users use the search engine to reach the desired page, and then use the information contained on the landing page as a guide [O’Day and Jeffries, 1993].

Information scent is the subjective perception of visitors of the value of particular paths of information [Chi et al., 2000]. can be used as a predictor of user motivation. Determining the task being performed by the visitor is challenging when user feedback is not available. Web page sequences can be interpreted to infer the goal the user is trying to fulfil. The semantics of the Web pages can be taken into account, so patterns of applications events can be obtained to ease the interpretation of page sequences [Oberle et al., 2003]. Even though sequences of pages can be interpreted as visitors fulfilling a goal, a change in the task being performed can be difficult to detect. Physiological information can be used as an indicator so that page sequences can be adequately split. For example, it has been found that pupil size drops rapidly at the end of each task [Chen et al., 2013].

Rather than predicting the overarching goal of the visitors’ visit, sequences of fine-grained interaction can be analysed. Pattern search in sequences can be employed when data is simplified down to a sequence of events without any timestamps. It uses sequential pattern mining algorithms to look for similar patterns following certain requirements. In algorithms like the IPM2 [El-Ramly et al., 2002] certain options can be introduced, like the minimum support and maximum error criterion thresholds for the patterns to be selected. Minimum support specifies the number of times the pattern has to be found in the set. Maximum error criterion specifies the degree of difference allowed between patterns – the number of additions or subtractions that would require one pattern to match another. Getting these smaller frequent patterns using automatic techniques does not give much information about the usage. Without any context, sequence patterns provide little insight into what the users did [Fern et al., 2010]. Context around the patterns is required to understand the users and obtain meaningful information from patterns.

Some patterns are known beforehand. Small actions and behaviours have been employed as indicators of misuse or usability problems. These patterns can be as
small as a single action. For example, a common behaviour when an action has an unwanted result is trying to undo it. Undo and erase events have been used as markers for detecting usability problems [Akers et al., 2009]. Taking user feedback into account prevents losing the context and helps to discard false positives. One example of false positives is epistemic actions – actions the user perform to change the environment to search for a solution for a certain task [Kirsh and Maglio, 1994] – that would be falsely treated as errors when the user was deliberately exploring different possibilities.

Single actions can only be used as indicators if the action has meaning in itself as in the case of undo. Other actions like mouse clicks cannot be used on their own, and combinations of actions producing a more meaningful event must be considered. Combinations of repetitive actions have been used as indicators of user-visible failures in AJAX Web applications. When users find an error, which usually involves not getting the result they expect, they tend to repeat the same action. Successive Interaction Repetitions (SIR) might not always indicate an error. Sirana [Li et al., 2010] is a tool that employs a Bayesian model to classify failing SIRs – they indicate an error – and successful SIRs – users performed them on purpose and there was no error. Looking for certain patterns can also help locating among all the collected data the behaviour that needs to be understood. It can be used as a fast way of filtering the data, to leave only the data concerning those episodes. Focusing on switches between applications during pair programming sessions – a technique in which one programmer codes and another one reviews the code while working on the same station – can give insight into the efficiency of the coding task [Vlasenko, 2011]. These switches can be analysed to explore how users alternate between different tools while performing developing tasks.

Episodes of user activity behaviour can be inferred from low-level interaction data. The existing relation between mouse and eye movement allows the identification of different user behaviours while interacting with a search engine result page [Huang et al., 2012] including: reading, when the mouse moves horizontally; inactive, when the mouse does not move; and clicking and examining, the rest of the time. Gaze position can be predicted using the future position of the mouse. This prediction is only possible in some particular scenarios, as the strength of the correlation between eye and mouse is variable. Using mouse input as a heuristic for eye activity enables a cheaper, scalable and more comfortable eye tracking technique. Research has not provided the perfect proxy yet, but they have the ability to identify when the estimations are close. If the technique is refined, it will allow a precise estimation of the eye position, even if some information gaps are unavoidable. The only required input would be the mouse
movement log. User distraction and frustration can also be predicted, using mouse
dwell time and the number of mouse positions on text as features [Navalpakkam and
Churchill, 2012].

High-level patterns can be more difficult to design and find but they are also more
self-explanatory. Rather than predicting users’ attention focus, or the use of the mouse,
these patterns take into account users’ motivation. For example, some abstract behavi-
oural patterns can occur when users try to cope with certain situations. The necessity
of coping mechanisms increases in the case of disabilities, such as visual impairment
[Lunn et al., 2011]. In past research, the interaction of users while freely browsing
online was observed and various abstract behavioural patterns used for coping with
different situations were discovered. The adaptation of the Web page taking these be-
haviours into account is beneficial: these abstract patterns enhance the understanding
of users’ motivation and provide information about the usage. Adaptation of the sys-
tem to discovered patterns is possible, but obtaining these patterns can be challenging
without direct individual observation of users. Web applications that automatically
identify indicators of navigation difficulty, pointing them out to designers, ease the
process of adapting the interface to improve the navigation [Thomas, 2014].

2.3 Data capture

One of the main characteristics of the data capture is the physical location where the
capture is taking place. Laboratory environments provide more control over the obser-
vation, while capturing information in the wild offers a more naturalistic environment.
Some techniques’ special requirements can require laboratory capture, making their
use outside of them difficult.

2.3.1 Task driven

When the analysis is focused on a task or set of tasks, a laboratory study can be car-
rried out where a bespoke computer is used for the procedure. The task to be evaluated
is identified beforehand, and the users perform it while an observer records their ac-
tions. The observer can either be a researcher, who takes notes of the users’ behaviour
and their opinions; physical recordings, like audio, video or eye-tracking systems; or
software which records the interaction with the interface.

A task driven example is the think-aloud protocol, which requires an experimenter
2.3. DATA CAPTURE

being present [Ericsson and Simon, 1980]. This protocol requires the user to perform a series of tasks while expressing vocally what he is doing at all times. It was presented as a way of getting more reliable data than the one obtained from the users trying to remember what they did. Requesting information at the time of the action makes the users focus their attention on it. Asking users to recall this information from their experience risks that the users did not notice it at the time, so they necessarily make the information up. A variation of this protocol is the coaching protocol [Kato, 1986], in which an expert in the system to be evaluated sits beside the user. This way the users are encouraged to talk about what they are doing because the expert will reply with suggestions that will help them complete the task. A variation of the coaching protocol called question-suggestion protocol allows the expert to freely make suggestions to the user [Grossman et al., 2009]. This way the protocol accelerates the learning process – in which a novice user becomes an expert of the system – and enables the identification of learnability problems.

Automating the capture of interaction data is an alternative. The addition of a piece of JavaScript code can capture various interaction events from a Web page and can take into account DOM changes induced by AJAX. Showing a timeline of the captured interaction events using different colours for each type of event makes the identification of repetitive events intuitive, easing the discovery of usability errors [Carta et al., 2011]. Another possibility offered by the automatic capture of interaction is recording an optimal use of the interface to use it as a model to look for discrepancies with the real recorded use. Other measures like queries, time lapses and sequences of actions can be obtained from these recordings. These measures allow the researchers to get objective observations such as the time required by a user to perform an action, or the frequency of certain keywords in queries.

These task driven techniques are focused on the evaluation of certain predefined tasks. These tasks are usually identified by the designer as the most common, or most critical tasks to perform on the Web site. The results are a measurement of the usability of those tasks, and an identification of the weak points to be fixed. For example, this approach can help to enhance the usability of precise tasks such as buying a product in a shopping Web site. In that case it can help to detect if the users had problems searching for the object they were looking for or if it took them a long time to find an item in a list. Combining a task driven approach with remote data capturing enables a flexible evaluation of predefined tasks. Remote data capture widens the range of scenarios and devices to be considered for the evaluation of a particular task, as well
as broadens the pool of participants [Nebeling et al., 2013].

However, the use of predefined tasks poses a prejudgement of the user interaction by the designer. This prejudgement results in an adaptation of the user to what the designers is expecting them to do. Enforcing tasks on users can be alienating, causing them to interact differently to the way they would normally, when completing tasks that are not necessarily representative of real world interaction [Cordes, 2001].

2.3.2 Eye tracking

Insight into users’ attention focus can be obtained by identifying the eyes’ point of gaze via eye tracking [Duchowski, 2007]. Its technology is usually too cumbersome or too expensive to carry around so is bounded to laboratory environments. It uses a system that tracks and records the eye position, providing information on where the user is looking at the screen at all times. This technique provides the means to detect users’ attention priorities among the elements of an interface. Eye gaze is very different from the mouse as it is a subconscious action that gives more information, as a continuous indicator of users’ focus. Due to its subconscious aspect some of the eye gaze data may not truly represent users’ attention. Some stimuli such as changes in the interface may be observed by users, but not reach their awareness. This phenomena is known as inattentional blindness and gets aggravated if the tasks performed by the user require a high perceptual load [Beanland and Pammer, 2010]. Certain aspects of the interface can increase the likelihood of interface changes being recognised. The chance of an interface change being recognised has been found to mainly depend on the way the change was triggered. User requested interface changes, such as clicking on a dynamic element, are more likely to be perceived [Jay et al., 2013]. As opposed to eye gaze, the mouse is only operated when the user intentionally focuses on something and interacts with it [Jacob and Karn, 2003]. Compared to other sources of interaction data eye tracking can provide larger amounts of useful data, as sight interaction is faster than any other computer interaction. As an alternative to estimating where in the screen the user is looking at, distinctive features can be found in direct recordings of eye activity. Features such as pupil size and blinking have been found to be indicative of users’ cognitive load [Chen et al., 2013].

Observations of eye activity introduce artificial elements to the interaction. Equipment for eye tracking requires expensive set ups in laboratories which need the user to be physically available. The eye tracking device requires a precise calibration and the participant’s movements are restricted to ensure accurate readings. Therefore their use
outside a laboratory environment is challenging.

2.3.3 User feedback

Asking users for feedback is a popular technique due to its low cost both in time and personnel involved. A list of questions is presented to the users to collect their opinion about the interface. Answers from users provide information about usage and possible problems.

The way to apply this technique may differ, depending on the characteristics to be evaluated. Questionnaires are flexible allowing any question to be included, and they can be carried out at any time. For example, using questionnaires before, during, and after the interaction provides the researcher insight into user experience surrounding the interaction [Bargas-Avila and Hornbæk, 2011]. This way the expectation towards the product, the feelings during the use of it, and the fulfilment of the expectations can all be measured. Analysing the experiences reported by the participants over longer periods of time provides understanding about how these experiences relate to different adoption stages. These stages can then be grouped into coarser categories: anticipation, prior to any actual experience of use; orientation, initial experiences; incorporation, when the product becomes meaningful in users’ daily lives; identification, when the product is accepted in users’ daily lives and becomes part of their self-identity [Karapanos et al., 2009]. Asking participants to self-report their experiences over time can be cumbersome, and sets limits in the amount of available participants. A more flexible approach involves gathering information from users’ retrospective assessment. Day Reconstruction Method requires asking participants to recall experiences from the previous day [Kahneman et al., 2004]. For periods longer than a day, questionnaires can ask users to chronologically order their memories. Even if users are not able to place an experience in a particular point in time, the patterns of change between them can be analysed to provide insight into the evolution of the experience [Karapanos et al., 2010]. The use of these reporting techniques over extended periods of time provides rich longitudinal qualitative data and can be used as a tool to understand the factors that cause user experience to improve or deteriorate over time [Kujala et al., 2011]. Some observational approaches can enhance the captured quantitative data from the observation of the interaction with qualitative information collected in the survey [Kim et al., 2008].

Performing low-scale surveys is an easy way to gather answers to specific questions and get qualitative information about the usability and the usefulness of certain
products. Technologies have eased the effort of carrying out surveys, allowing designers to conduct them online. Free solutions like Google Forms, and commercial tools like SurveyMonkey allow remote surveys, capable of reaching a high number of people at a very low cost. Furthermore, these surveys can be taken outside laboratory environments, like users’ homes, to provide a more ecological environment.

Surveys provide information based on the replies to the specified questions. Sometimes new knowledge obtained from received answers can give hints about new questions that could be helpful. Finding out the answers to those new questions is not possible without carrying out a new survey, thus requiring questions in surveys to be carefully planned in advance. Ambiguities in the questions need to be avoided, as they can lead to misunderstandings with users. Although collecting answers to particular questions is useful, users’ self-assessment might not be reliable as the perception of what they did may not reflect reality. The way humans’ memory prioritise events according to the time they happened is also a bias in their assessment. Humans remember most recently experienced events in a clearer way. Lastly, this technique is impractical for longitudinal studies, as it would require the user to fill in a survey periodically.

2.3.4 Remote solutions

To obtain more naturalistic observations, remote solutions can be employed. They enable the observation of users in their environment, avoiding biases generated by the setting of laboratory studies and preventing the displacement of the users. Thus increasing the range of the observation and reducing the obtrusiveness towards the users.

Mouse tracking enables a remote objective observation, enabling a more naturalistic environment while capturing all mouse movement on the screen. The recreation of the mouse movement on screen is possible this way, identifying where the interest of the user resides. The problem of relying solely on mouse tracking is that a user’s attention can be aimed at a point far from the mouse position. Even if a relation between mouse location and focus of attention exists, it is not always explicit. Research indicates that places visited by the eye are usually visited by the mouse, and places not visited by the mouse are usually not visited by the eye either. This relation gets stronger during saccades – fast movements of the eye. In the same way as

https://drive.google.com/
http://www.surveymonkey.com/
2.3. DATA CAPTURE

eye tracking, analysing mouse movements can provide insight into where user attention is focused. Additionally, the analysis of mouse interaction can serve as a proxy to user distraction and frustration using mouse dwell time and the number of mouse positions on text as indicators [Navalpakkam and Churchill, 2012]. This analysis should take into account that as opposed to eye movement, mouse movement can stay stationary, giving no information about user attention.

The Enhanced Restricted Focus Viewer is a creative solution that avoids the use of expensive equipment [Tarasewich and Fillion, 2004]. Instead of locating eye gaze, it blurs all the screen except the area surrounding the cursor. This way the users are required to use the mouse to explicitly point out what they want to see. The tool then records mouse movements, which are assumed to be correlated with users’ attention. It is a very obtrusive and tedious technique for the user. Constraining the users by this technique may not represent how they would navigate through a Web page.

Instrumentation of interaction data other than mouse movements in an application is possible. Applications can be designed in a way so they can log interaction events. In the case of Web applications these events can include page loads, mouse and button clicks, and mouse movements, such as hovering over elements. This process is not scalable, as every new feature would have to contain extra code in order to log the new events. Another problem is deciding a threshold between how much interaction data to capture and the complexity of the subsequent analysis. The manual instrumentation of the application enables the selection of the events to be logged. Reducing the number of captured events simplifies later analysis, but there is a risk of missing relevant data. The way the collection of data takes place affects directly the data analysis possibilities, as well as the degree of insight into users’ behaviour.

To avoid overload of data the logging can be reduced to meaningful events – they have a meaning on their own and are interpretable. The definition of these meaningful events depends on the purpose of the analysis. In the case of purpose specific applications, the design of the events to be captured depend on the tasks available. Instrumented software for image manipulation contains bespoke data such as the count of layers, or image size, to provide the necessary context information [Terry et al., 2008]. Other meaningful events can be composed of other events, as some low-level events can be meaningless on their own. If the events were broken down to their lowest level, a mouse movement could be composed of an event stating the mouse moved, and another one describing the new mouse coordinates. The former indicates the mouse position changed, without any information about its location, while the latter states the
location of the mouse. Depending on the needs of the analysis, knowing that the mouse moved can suffice, for example if the frequency of the mouse movements is required. To obtain both the occurrence of a mouse movement and its position a composition of events is needed. A mouse movement event can be combined with the mouse co-ordinates to create a new event informing where the mouse moved and when. Another way is the inference of events from higher frequency ones. In the case mentioned, mouse movement event has a lower frequency than a periodic mouse coordinates update. If the events are compared, a difference of coordinates indicates a movement of the mouse. Depending on the logging tool employed, the inference of the events can be carried out in the logging tool itself – when mouse movement is detected an event describing the movement, including the coordinates, can be logged – or in the analysis step – looking for changes between mouse coordinates.

The frequency of the events can vary depending on their type. It has been stated that some events can be composed of others, so one way to avoid overload is recording only lower frequency events. Lower frequency events provide the abstraction required to visualise events without the need of preprocessing the interaction data. On the other hand, if all low and high frequency events are recorded, a deeper analysis of the data is possible. Richer data enables the possibility of reanalysing the captured data for different purposes supporting the exploration of post-hoc hypotheses.

Tools to log interaction data from Web sites can be deployed in three different levels with respect to the user [Srivastava et al., 2000]. The figure 2.1 shows the three different levels.

![Diagram of the different levels of logging techniques.](image)

- **Web logs** have been used for a long time in research. They are the most unobtrusive way to capture interaction data from the users. Every HTTP request is
Data Capture

Logged so the path of Web pages the user is following in the Web site is recorded. This information can be analysed to see if the users are capable of finding the most appropriate path to their destination, and to get timing information that will point out usability problems that hinder users’ progression in the Web site.

- **Proxy solutions** consist of a computer standing between the Web site server and the user. It requires a set up by which the client connects to the Web site via the proxy. This way the proxy receives all the HTTP requests from the client to then redirect them to the Web site server, to get the HTML of the desired Web page.

- **Client solutions** log all the interaction data via an application installed on the client machine. It has direct access to the computer, so it can log a broader range of interaction data.

Web logging solutions are the least obtrusive solution. They do not require any modification, to the Web site or to the users’ equipment, as the data is available in the Web server. They offer enough data to obtain basic information about Web site’s usage at a very low cost. One disadvantage is that the list of HTTP requests can be missing data if the Web pages are cached on the users’ computer. In those cases, the HTTP petition will not be sent, leaving gaps in the recorded sequence of HTTP requests. Another disadvantage of this technique is that it requires direct access to the server where the Web site is deployed.

Client level collection is the most obtrusive of the mentioned approaches. It is capable of logging more data than any other configuration, at the cost of requiring a direct configuration of the user’s computer or the software itself. Users can be reluctant to install additional software on their computer, and modifying existing software can be cumbersome making the software more difficult to maintain. Controlled experiments in a laboratory are a feasible solution. When large controlled usability studies are carried out, the evaluated software can be thoroughly modified to get all the interaction data. For example in TRUE [Kim et al., 2008], a video game usability evaluation tool is presented, where different kinds of data – such as number of deaths or favourite weapons – are continuously measured, frequent surveys were used to complement captured data, and video playbacks of critical spots were possible. In the case of Web applications, a browser extension can be used.

Modifying the Web pages so they can capture data is a popular approach. Even though the location of the logging tool is on the server, but the script to log events is executed on the client side. Client side execution provides the advantages of a logging
CHAPTER 2. BACKGROUND

Early approaches like WebVip [Scholtz et al., 1998] help the designers with the task of manually annotating the links of a Web site, so selected events are logged along with identifiers and timestamps. Despite the effort of manually modifying the events to be recorded, this customisation can be beneficial. As it has been mentioned before, the definition of when an event has semantic meaning can change depending on the purpose. In some cases knowing that the user clicked might be enough, while in others the context of the click might be required. If the definition of a semantic event is clear, selecting what events to log can provide a fast and simple way of logging only the events that seem important for the researcher. WET [Etgen and Cantor, 1999] eases the effort of recording selected events like mouse overs, clicks and page loads by adding a piece of JavaScript code to the Web page. A cookie is employed to record the captured data, so the memory available is limited. WebRemUsine [Paganelli and Paternò, 2002] resolves the memory problem by sending the collected data at the end of the task, only using cookies to identify users. It uses a manually inserted JavaScript code as well, but instead of logging predefined interaction events, it requires the input of a task model. Only the actions involved in the execution of that particular task are logged, to then be compared to the optimal execution of the task. The execution of the tasks by the users can be classified according to their effectiveness – success or failure – as well as efficiency – time spent and number of unnecessary actions – so evaluators can identify problematic tasks [Paganelli and Paternò, 2003]. More recent tools like WELFIT [de Santana and Baranauskas, 2010] enable the automatic capture of several low-level interaction events by manually adding a piece of JavaScript code to all Web pages. A Markov Chain is then employed to show the transition between different low-level events. Analysis of these transitions enables the detection of interaction problems such as double clicks on not clickable elements.

The use of a proxy can give access to at least as much information as the server logs. It is more obtrusive, though, as it requires some modification of the user environment, or direct access to an access point already in use by the user. One advantage is that it overcomes the need of having direct access to the Web server, while giving the possibility of logging the interaction with all Web sites accessed via the configured computer. Data is noisy, as artefacts such as advertisements, HTML frames and automatic reloads will provoke requests that should be filtered [Weinreich et al., 2006a]. Even without being able to discern unique users, proxies have been exploited to show paths and measure timings at each Web page [Cugini and Scholtz, 1999]. They also
give the possibility of carrying controlled experiments, were the users of the proxy are known. This way the usage can be tagged with unique identifiers for each user [Hong et al., 2001].

Another advantage of the proxy is that it allows to modify Web pages before sending them to the user. Modifying Web pages allows researchers to inject JavaScript code in the HTML files. It is the same approach as directly modifying the Web pages, except that it eases the process of capturing the data. A proxy server can be configured to inject a piece of JavaScript code in every Web page before sending it to the user removing the need of manually modifying all Web pages in a Web site. This code can automatically log interaction events in the Web page. These interaction events can include: page loads, mouse clicks, key strokes, mouse movement coordinates, mouse hovering over elements and any data requests – like the ones caused by page loads and Ajax techniques. Indiscriminate capture of interaction events can lead to an overload of information but ensures all interaction events likely to be captured are recorded. The analysis module required for this technique needs to cope with a high number of different interaction events.

The practicality of injecting JavaScript code has fostered its use in many studies on browsing behaviour. The proxy can be configured to gather interaction data from selected participants. Identification of observed participants makes analyses of particular demographics possible. Insight into children’s browsing behaviour can be obtained recording particular browsing usage metrics, like the speed of typing and number of clicks [Hollande et al., 2010]. WebinSitu [Bigham et al., 2007] looked for the differences between sighted users and blind users while navigating a Web page. These observations last for short periods – one week – and present the results in the form of summaries. Reports of the percentage of users who performed a particular action from each group highlight differences in behaviour. For example, blind users were found to be more likely to click on images with alternate text.

Most studies using interaction capture tools are limited in time. Information is easier to analyse or visualise, it can be pinpointed in time, and more confounding factors can be controlled. Longitudinal data poses more difficulties, as ruling out every factor that partially accounts for the observed effect on the dependent variable is challenging [Hutto et al., 2013]. Still, longitudinal studies shed light on factors that are difficult to observe in laboratory studies. Observing how users’ behaviour evolves over time when interacting with an interface enables a design that adapts better to users’
learning process. Longitudinal studies have been successfully employed to predict expertise changes by determining what factors affected their increase and decrease over time [Huang et al., 2013].

However, the implementation of a longitudinal capture system is challenging and poses additional burden to the design of a longitudinal study. The use of existing interaction data capturing frameworks relieve the researcher from common issues, such as scalability and the transfer of data. Depending on the needs of the study and the employed framework, thorough modification of the application events to be captured might still be required [Jensen, 2009]. Alternatively, longitudinal capture of user feedback can provide in situ information about users’ day-to-day issues [Gerken et al., 2010]. Combination of automatic interaction capture and user feedback provides both qualitative and quantitative information about user behaviour over extended periods of time. Qualitative input enriches captured interaction data providing context to help understand user motivation during particular interaction periods [Voida and Mynatt, 2009].

2.3.5 Privacy concerns

When capturing interaction data it needs to be considered that any user observation can pose privacy concerns. Personally Identifiable Information (PII) is a particular kind of sensitive information that allows to establish a connection with or even identify a particular person. For example, raw Web logs contain basic information that is commonly collected by any Web server, and IP addresses can be obtained without the necessity of processing those logs. Even if IP addresses are the most basic information that can be obtained from users, their collection has raised concerns on whether they constituted PII. To consider information to be reasonably linked to a person both the required information to form that link and how that information could be obtained should be considered [Lah, 2008]. A simple example would be collecting usernames for a certain Web site that contains sensitive information about them, such as shopping Web sites, where credit card information and addresses are stored. As long as only usernames are collected, and there is a certainty of the impossibility of accessing the information linked to those usernames, this information alone would not be considered PII. Sometimes privacy problems may not come from the sensitive nature of the collected information, but from the combination of information considered non-personal. Profiles for users visiting a particular Web site over time can be created, and linked to
that user via IP address or cookie. The links clicked by users or their system configuration are not sensitive information on their own. However, this information can be used to profile users and classify them into groups surreptitiously [Velásquez, 2013].

More detailed information allows to identify users in the Web with a higher degree of precision. Different user fingerprinting techniques rely on the uniqueness of combination of features, such as browser versions and the list of fonts installed in a system [Acar et al., 2013].

Narrowing down users from regular data is still possible. Even without identifiable information. For example, using the timestamps from a log of accessed URLs, users from a particular time-zone can be identified by comparing inactive times with the night time in their area. This coarse narrowing can be combined with other techniques to end up identifying individuals. Activity logs among different social networks can be combined, to determine accounts pertaining to the same users under the assumption that users tend to use various social networks at the same time [Korayem and Crandall, 2013]. To prevent this kind of inferences data can be distributed across different databases. The distribution can be horizontal, with different records in different databases or vertical, in which the values for the attributes are distributed.

Original data can be altered to avoid inferences that would pose a privacy concern. Modifications can involve blocking or perturbing certain values. Sensitive data disclosure can also be prevented by releasing only a sample of the original data. Using modification techniques data can be downgraded so that the accuracy of the rules that would disclose sensitive data is reduced. Utility of the data can inadvertently be diminished if the applied alteration affects non-threatening rules. The entropy of the sensitive value before and after the modification can be compared with a measurement of the loss of data utility to decide if enhancing the security is worth the diminished utility [Verykios et al., 2004]. Another way to increase the anonymity of the data is ensuring that the granularity is coarse enough so all records map onto at least a minimum K number of other records. For example, timestamps can be generalised from a granularity of days to years. The disadvantage of this K-anonymisation approach is that the problem is NP-hard. If the data querying is controlled, this same mechanism can be applied, so only the queries that respect the set K-anonymisation rule are allowed [Aggarwal and Yu, 2008].
2.4 Interpretation of the data

Interpretation of the data involves transforming obtained data into useful information. The obtained data can either be summative or formative [Hilbert and Redmiles, 2000]. Summative techniques provide metrics summarising the interaction. A common metric is measuring the time required to perform an action or task. Formative techniques act as an informational tool, providing feedback about the interaction. One example would be the use of surveys to discern what interface changes would please the users.

2.4.1 Heuristics

A special case of analysis are techniques based on heuristics. These techniques omit the capture of the data, and instead of using real usage they use models that predict users’ behaviour. The lack of need for users allows a cheaper and faster evaluation of the interface, although the accuracy of the prediction needs to be considered. Depending on the quality and complexity of the model the results will be closer to the outcome of real user testing. Cognitive constraints and psychological research can be taken into account by these techniques to provide a more realistic model.

Cognitive walkthroughs enable the discovery of prospective interaction problems without the need for users. A group of experts is required who will design the optimal sequence of actions to perform a task. All the actions in the sequence are then analysed to look for possible problems. The use of a checklist helps experts to take into account important aspects of the interaction. This checklist can include questions to help make sure that the users know at all times what action to perform to carry on with the sequence of actions that will let them achieve their goal. The motivation for this technique is that users prefer exploring a software than learning it. Therefore, every action needs to be sufficiently clear for inexperienced users [Wharton et al., 1994]. This technique ensures first-time users will be able to intuitively carry out the analysed task.

User cognitive models have been a subject of research in HCI for a long time. In the early eighties, the Keystroke-Level Model (KLM) was presented [Card et al., 1980], showing a method to predict the interaction time of a user. It employs the GOMS model [Card et al., 1983], which represents the user interaction considering four factors: Goals, the objective the user is expecting to achieve; Operators, the actions the user will perform in order to get to the goal; Methods, the different ways to achieve the same effect in the interface; and Selection rules, the method the user selects
among all existing ones. All user actions are represented in pseudocode by assigning them a letter: K is a keystroke (not the production of a character, so writing a capital letter by pressing SHIFT plus a letter counts as 2 keystrokes), P is pointing with the mouse, H is homing the hands to the keyboard or another device, D represents drawing, M is the mental preparation of the user, and R is the response time of the system (the time the user has to wait to perform the next action).

To evaluate a task with the KLM method all the necessary actions to be performed need to be described in pseudocode. The necessary time to perform the task is then calculated using a set of rules defining how long each action takes. The keystrokes (K) can vary to represent different degrees of typing skill. Pointing the mouse (P) uses Fitts’ law [Fitts, 1954] to determine how long a mouse movement takes. This law established in 1954 calculates the time to reach an object with a cursor based on both the distance to the object and its dimensions. Homing (H) the hands have a preset timing that represents the time required to move the hand from one device to another. Drawing (D) takes into account the length of the drawn segment. Mental preparation (M) is also a preset time obtained from experimental data. Finally the response time (R) is specific to each system, requiring external input.

Writing the pseudocode corresponding to the sequence of actions to be performed to achieve the goal is required to use this method. More advanced tools have been developed through the years to ease this task, like Cogtool [John et al., 2004]. Instead of using pseudo algorithms, the task can be performed and the tool considers the resulting interaction as input. This tool requires creating the interfaces in HTML and executing the same actions a user would during real interaction. The tool automatically calculates the necessary time to perform the tasks, using a sophisticated model and introducing mental operators automatically. It makes use of advanced heuristics, like reducing the time required to perform an action if the same action has been performed recently, simulating the user preparation.

Not all user factors may be considered in these simulations, resulting in an incorrect prediction. Real users employ information scent to decide which link to follow, relying on their perception of the value of the information sources available. Not taking these user factors into account can lead to a different behaviour than the one predicted by GOMS models [Card et al., 2001]. Cognitive models can consider information scent by simulating text comprehension. Semantic similarities between user goals and link text, among other factors, can be used to infer what link would be chosen by a real user.

http://cogtool.hcii.cs.cmu.edu/
Eye gaze can also be considered. Eye behaviour has special characteristics apart from predicting how fast a user can focus on one object when coming from another. Eye movement tends to go to closer objects, and even if the gaze stops over an object, the participant may still fail to recognise it. Laboratory studies can be used to discover eye behaviours to be included in users’ gaze models [Halverson and Hornof, 2007].

Examples of computer models simulating human constraints can also be found outside the HCI domain. For example, a model for unmanned vehicle human pilots can be obtained by “hobbling” a pilot assistant software [Pickett et al., 2013]. Pilot assistant software is designed to be a perfect pilot so its “hobbled” behaviour simulates the imperfect behaviour of human pilots. The resulting hobbled model can then be used to test other unmanned vehicle pilot assistants. This way poor assistants can be discarded so only the successful ones are tested with real users.

Heuristics remove the need for participants at the cost of risking the validity of the obtained data. Validity depends directly on how realistic the used model is. The result of the simulation also depends on the user profile simulated by the system. New users can be simulated using cognitive walkthroughs, testing if the site is intuitive enough. However, cognitive walkthroughs follow a list of the predefined problems that the users might encounter. There can be problems not taken into account in the list and therefore omitted. As opposed to a cognitive walkthrough, cognitive models simulate expert users. Cognitive modelling is useful to compare the efficiency of two different sites based on timing measurements. Both techniques are task oriented and consider single user profiles. This prejudgement of the interaction results in a biased evaluation that covers a subset of all possible interaction with the Web site. A longitudinal approach would be possible, but only to measure how the different changes in the Web site affect users’ adaptation or its time-effectiveness according to the model.

2.4.2 Web analytic tools

Real usage data is required to get real interaction information as opposed to the result of a simulation. Data gathered at the server level provides basic information in a straightforward manner. The information is varied and includes demographic data of the visitors to the Web site, counts of visited Web pages, followed paths, and search queries. The first Web log analysis solutions fed from database analysis techniques. Database analysis has a long history in computer science, and when Web log data is
2.4. **INTERPRETATION OF THE DATA**

transformed into a database, techniques like Online Analytical Processing (OLAP) enable the interactive analysis of this multidimensional data [Zaïane et al., 1998]. OLAP enhances the analysis of multidimensional data by enabling subsequent filtering of the data combining various dimension restrictions. In this way, summaries relating any dimensions are possible using familiar database queries.

**Web site analytic** tools gather sophisticated data from visitors. They usually require the addition of a piece of JavaScript code to every Web page, so the traffic data is sent to the correspondent server. Google Analytics\(^5\) is a popular Web site analytics tool from Google. Various information can be gathered from the users’ system. It records language, network provider, browser and location among others. It also records if the users are first-time visitors or recurring ones, and what was the source of their visit – e.g. a referral from another Web page or the result of a search engine.

The main purpose of Web site analytic tools is to help attract more visitors and improve conversion rates. Conversion occurs when a visitor performs an action that has some benefit for the Web site, like making a purchase or accessing a particular Web page. To analyse these conversion rates, funnels and visualisations of different statistics can be employed. Funnels are graphs picturing the sequence of actions required to perform that conversion and how many users drop out at each step, pointing out major user leaks. Keywords from search queries that lead to the Web site can be analysed to provide insight into users’ motivation when accessing the site. Understanding what search terms lead users to the different Web pages makes the adaptation of these pages to future visitors possible.

The commercial purpose of Web analytic tools makes them focus on increasing the number of visits as this will increase the company’s Web site revenue. Although users’ behaviour is an important aspect of this task, these tools do not provide the means to understand users’ interaction behaviour. Their economically pragmatic approach only considers the mainstream users relevant. Rather than providing insight into a user’s behaviour these tool focus on identifying prospective sources of visits and increasing a Web site’s online presence.

Other **commercial alternatives** provide a deeper insight into users’ behaviour. HP’s Real User Monitor software (HPRUM\(^6\)) is focused on reducing time delays caused by the performance problems in the system. Any time delay can harm user experience, hence the importance of locating performance problems in the software.

\(^5\)[http://www.google.com/analytics/]
In the case of complex software, such as Web applications, complex transactions can take place. For example, in database-based Web applications, a single transaction has different stages. From the user performing a query to the Web page to the Web server receiving and processing the content of the query and requesting the information to the database. Time delays can occur during any of those steps so listing the time delays for each step can be useful to find performance problems. Similar tools like NewRelic\footnote{http://newrelic.com/} also allow to narrow down low-level transactions causing time delays, showing the SQL query responsible for it.

Other approaches take a more individualistic view of the users. Instead of showing the overall trend of users, tools such as TeaLeaf (IBM)\footnote{http://www-01.ibm.com/software/marketing-solutions/tealeaf/} show individual visualisations of users’ HTTP requests. Users’ interaction can then be partially reconstructed. Interaction with input forms is recorded, so the order in which a user fills in the records in those forms can be recreated. The occurrence of particular situations can also be tracked, such as the appearance of “product not available” messages. In the case of mobile devices, additional interaction information such as the use of pinching, zooming, scrolling and device rotation is provided, which can be combined with other demographic statistics.

Mentioned commercial tools offer a higher degree of insight into users’ behaviour than usual Web analytic oriented tools. They go beyond the pragmatic approach of increasing the number of visits and look into what could affect the user experience in Web applications. Still, some of them are mainly focused on time delays, which can be overcome by increasing the performance of the correspondent steps in the transaction that misbehaved. Others take into account user interaction, but are limited to certain cases – input forms – and do not provide the means to perform longitudinal studies beyond providing usage statistics.

Search queries\footnote{\texttt{http://newrelic.com/}} can also be extracted from Web server data. They provide information about what the users look for while navigating a Web site. They can help identifying the purpose of their visit, and provide quantitative data about what product or service is the most popular one. Analysing search queries from the main search engines help identify what problems the users are having with a product. In CUTS (Characterizing Usability Through Search)\footnote{Fourney et al., 2011}, the keywords used in conjunction with a tool’s name are analysed under the hypothesis that users look for
2.4. INTERPRETATION OF THE DATA

solutions online when they encounter problems. Frequent problems with that particular tool are likely to be found this way, covering the mainstream sector of the users. Although useful to detect critical problems, it only detects a small subset of all problems.

2.4.3 Privacy and biometrics

Users’ interaction analysis can pose privacy issues, as it can provide information that helps to identify users. In the same way, that same information can be employed to increase security in sensitive applications. Measures like tracking IP addresses are commonly employed by major Web applications, such as social networks. They can help to detect abnormal connections by comparing geographical locations. More intrusive techniques that provide a finer identification can be employed in more sensitive domains. Fingerprints and retinal scans are examples of the most intrusive identification techniques, non-viable for the security level of the majority of Web applications. Electronic IDs are an alternative that only requires a special card reader and a personal ID card. Even if in different degrees, these systems are all quite intrusive and require an active participation from the users. An unobtrusive passive authentication system presents two advantages: keeping prospective attackers unaware of the security measures decreasing the chances of an attack, and not involving the user reducing the burden and ensuring the security measures are taking place. Analysing keystroke dynamics is a simple way of performing an automatic authentication. Common features are latencies between key presses or key releases, and time spent pressing a key or between key presses. Banerjee et al. provided an extensive review of various keystroke analysis techniques with authentication purpose [Banerjee and Woodard, 2012], and a very clear diagram of the different latency features employed by these techniques, which can be seen in Figure 2.2.

Keystroke analysis can be static, pinpointed in time; or continuous, taking place throughout the interaction. Patterns among the extracted characteristics from the keystroke sequences can be analysed and used to classify different users. There are two ways to obtain these keystroke sequences, using fixed or free text. Fixed text refers to the typing of a particular text piece while free text analyses any text the user types. Even if the free text provides a higher amount of data, it has been shown that the use of fixed text provides a higher degree of precision [Monrose and Rubin, 2000]. Fixed text can be used when users are typing their passwords, for example, as this corresponds to
CHAPTER 2. BACKGROUND

Figure 2.2: Diagram of common latency features employed in research into Keystroke dynamics [Banerjee and Woodard, 2012].

a known timing sequence – there is no need to store the actual characters of the password – and users are likely to type it at a similar speed every time. Mouse features can be combined with keystrokes to increase the amount of information. This information can be collected in the background, and can be employed in a reactive way. This way, even if a perpetrator manages to login into the system, a background collection of information can take place until the impersonation is detected. At that point, all executed actions can be reverted, and countermeasures can take place [Traore et al., 2012].

Work on affective computing have made use of keystroke analysis as a proxy to users’ emotions. Users’ affective state can refer to users’ emotions or moods. Emotions can be defined as a reaction while moods are longer-lasting. Although the particular analysed aspect of affective states (mainly referring to users’ emotional state) is not always clarified, moods are the common objective due to the difficulty of analysing short reactions using these techniques [Zimmermann et al., 2006].

As opposed to obtrusive methods of emotion self-reports, low-level events such as keyboard presses offer a low-cost solution to infer users’ emotional state. Users’ reported state is compared with features extracted from keyboard and mouse movement to discover correlations. Key presses can be grouped into combinations of 2 and 3 keys, and key duration – time elapsed from the first key press to the last key release – and keystroke latency – time elapsed from a key release to the next key press. Various stimuli can also be used to trigger particular emotional states. Changes in interaction while users transition between emotional states are analysed [Salmeron-Majadas et al.,]
The use of fixed text provides notably better results for this analysis. The variance found in free text affects the analysis negatively, making a fully unobtrusive inference of emotional state challenging [Epp et al., 2011].

### 2.4.4 Task models

Analysing interaction sequences from users performing a task is a way of determining the efficiency of the interface. One of the objectives of interface design is achieving a design that lets real users be as efficient as an expert user of the system. Some techniques consider task recordings of designers’ usage as the task models and compare recorded data with them. Problems in what the designer considers to be the crucial tasks can be found this way. Analysing a predefined set of tasks gives the possibility of having a task model that enables a straightforward comparison. One way to do this is to compare the sequences of the actions performed to carry out a task. Markov chain representations can be employed, showing what percentage of users did not follow the optimal path.

When timestamps are taken into account, usage data can be plotted in the form of a timeline. These representations provide more insight into timing between events, making efficiency problems easier to detect. There are tools that allow the definition of what actions are expected at each point of time when performing a task [Propp and Forbrig, 2009]. The resulting task model as shown in figure 2.3 is similar to a common Gantt chart.

Other sophisticated timelines plot all usage data in a single timeline [Carta et al., 2011]. They use different colours for each kind of interaction event and use the height to represent the frequency of the event. Interaction from the execution of the task by a designer can be plotted and used as the task model.

Enforcing the task to be performed in remote observations allows to increase the pool of participants while retaining control of the execution. Comparisons of the interaction between users and the different tasks to be performed are then possible [Rzeszotarski and Kittur, 2011].

Task oriented recordings are usually obtained through formal tests explicitly asking the users to perform particular tasks while they are being recorded. The additional constraint of the user not being able to interact freely with the interface is unavoidable in those cases. Instead, the user can be encouraged to interact freely with the interface while the interaction data is captured. Not enforcing any predetermined task makes it less obtrusive and provides a more ecological environment. Gathered data is then
analysed and the tasks to be evaluated are isolated. Identifying the starting and ending actions of a task allows the retrieval of the sequence of all the actions performed inside the task. The most common sequence can then be identified and evaluated for usability errors [Vargas et al., 2010]. After logging users’ interaction for any period, it allows to perform comparisons among the captured data, as all the data corresponds to identified tasks. Using predefined task models still poses a bias in the analysis. Even if the user is allowed to interact freely, all interactions outside the predefined tasks to be evaluated will be omitted.

2.4.5 Visual analysis

When neither the task model nor the relevant features from the data to be analysed are clear, developing an analysis module to identify the key points can be challenging. In those occasions when the objective of the study is not clearly defined, initial exploration of the data helps to narrow down the analysis possibilities and define the study to be conducted. Human analysis of visual data helps to find patterns among large
amounts of data if the data is presented in a way that salient events are highlighted [Tory and Möller, 2004]. A popular example is Google Analytics in which the user is presented with visualisations of the metrics corresponding to the visitors to their site. Although the user might be mainly concerned about the number of visitors over time, the initial objectives are not certain, so Google Analytics presents all available metrics in a suitable way. When the user identifies what features are of interest, the tool can be customised so selected features are visualised.

Deeper insight into particular use cases can be obtained visualising the usage corresponding to certain tasks. The use of predefined tasks reduces the complexity of the visualisation. Task boundaries narrow down the amount of data to consider making the visualisation more understandable. Task oriented visualisations had been found useful in literature, although they still put boundaries in the data to be visualised. A non-task oriented visualisation technique would overcome these problems.

### 2.4.6 Longitudinal analysis

Insight into temporal factors can only be obtained through longitudinal analysis. Laboratory approaches allow to control confounding variables, reducing the inherent noise from uncontrolled environments. If enough laboratory sessions are carried out, evolution in the use of novel input devices can be evaluated [Harada et al., 2009]. When users’ have fully learnt how to interact with these devices, efficiency comparisons can be carried out under the assumption of these users being expert users [Card et al., 1987]. However, analysing the evolution of novice users into experts is a challenging task when the observation is bounded to a laboratory environment. This learning process follows a power function [Newell and Rosenbloom, 1993], rapidly increasing at first and then flattening as it moves away from the origin. Taking into account the initial increase a particular degree of skill metric can be set as an objective in laboratory studies. Although still limited in time, these laboratory studies provide insight into learning metrics such as the amount of time to reach the set objective from a novice state [Gerken et al., 2009].

Longer periods of time can be taken into account if the different laboratory observations are set apart in time. Systems with a professional purpose – e.g. health care information systems – are known to be used daily. Therefore laboratory observations of these systems can assume that selected professional users interact for comparable periods of time between laboratory sessions. The progression of novice users into experts and the differences in the encountered problems can be analysed providing information
to differentiate between critical problems and cosmetic problems that can be overcome with time [Kjeldskov et al., 2010]. When longer studies with the same users are not possible, taking a *cross-sectional* approach is a possibility. Cross-sectional approaches combine data coming from different studies, removing the need of following the same set of users for extended periods of time. Studies covering various time spans can be combined to extend the length of the study. If the study aims to determine differences between different stages of interaction, studies of participants with varying degrees of skill or experience can be combined to widen the spectrum of possible interaction stages. Variation between users cannot be controlled increasing the risk of reaching false conclusions about the effects of time. However, there are situations that can lead to similar problems with extended observations. Uncontrolled factors may affect users between observation points, particularly if the different in time between observations is large. For example, users might learn while not being observed, affecting the results of the following observation. Cross-sectional studies can tackle such problems by relying on users’ self-reported expertise, and analysing the differences between users with varying degrees of expertise [Novick et al., 2012]. Such an approach still falls under the assumption that users can self-report their degree of expertise to an acceptable level of precision.

### 2.5 Presentation of the information

After obtaining information from the captured data, meaning needs to be inferred from it. Textual summaries in the form of tables can be useful, and *statistical data* provide insight into what aspects of the Web sites need to be tackled. When demographical data is available, presenting statistical data can help adapting the site to certain target groups. WebInSitu [Bigham et al., 2007] for example discovered that blind users click more often on images annotated with alternative text.

A common visualisation technique for data obtained from eye and mouse tracking tools are heat-maps. This technique summarises all recorded data points and plots them on the corresponding screenshot of the Web page. The result is a drawing in which the different areas are colour coded according to the density of data points. Common colours to use are gradients from blue to red, being the red the hot – more crowded – points. This visualisation helps identify the parts of an interface with which the users interacted the most. This technique has been employed to detect what visual elements sighted users focus on for each different purpose [Yesilada et al., 2008].
The visualisations of data from Web logs can give insight in terms of demographic data, or time spent on the site. These visualisations can be found in Web analytics tools. Common representations are statistical data in the form of tables and timelines showing longitudinal information about the change of a certain metric over time.

Information about sequences of states or performed actions allows Markov chains to be plotted, which are directed graphs between different states representing the probability of going from one state to another. In the case of Web logs, these states can be the different Web pages inside the Web site, and the probability of users to move from one to the other, based on the percentage of users that took that step. A similar approach called VISVIP [Cugini and Scholtz, 1999] shows a 2D graph representing the structure of the Web site with Web pages as the nodes. Lines represent the links between Web pages, a curved line represents a user’s path through the Web site, and additional vertical lines of varying heights next to each node represent the time spent by the user in that Web page. The figure 2.4 was obtained from the project’s Web site on August 2012. In the figure, a line describing the path followed by the user is described, and the dotted lines next to each visited node represent the time spent in each Web page. The location of the nodes in the tree can be represented in a different way, depending on the links between them and the content. A 3D representation of the trees enables the observation of changes in travelling trends through the Web site [Chi, 2002].

WebQuilt [Hong et al., 2001] employs Markov chains as well, showing thicker arrows for the most traversed paths via an interactive interface. Representations of Web sites in the shape of trees are a common way to show how users navigate through the various Web pages. WebQuilt improved the way to collect data by using a proxy that distinguishes unique users. It also infers user actions from the path they take, as the backwards button when the users come back to the previous Web page. This way it generates a log per user and shows a task driven visualisation. It displays the starting and ending point, and makes use of task models showing the optimal path as a thick black arrow. Changing the zoom is possible so the level of detail changes from an overview of the Web site map to a more detailed view with screen shots of the Web pages. Unfortunately, it gathers the same information as the Web log, therefore lacking detailed interaction information.

Interaction data visualisation techniques are usually more sophisticated than Web log ones. If only one kind of interaction data is recorded, like mouse movement, it
Figure 2.4: Path laid over a Web site in the VISVIP tool [Cugini and Scholtz, 1999].

can be plotted on heat maps or lines accompanied by vectors [Arroyo et al., 2006]. If the variety of data to visualise is higher, it increases the visualisation possibilities but also makes providing an understandable overview of all the data more challenging. As different kind of interaction data can be captured, the data will be multivariate. Various interaction events can be treated as various states of a Markov chain [de Santana and Baranauskas, 2010]. To reduce the size of an otherwise too big and non-understandable Markov chain events are clustered related to the same element.

Understandable visualisations of different metrics can be achieved disregarding the rest of them [Fisher et al., 2012]. One of the possible examples is a heat map of the number of player deaths in a multiplayer game. Selecting this metric among all the data available enabled the designers to understand the problems in the map. Another alternative is allowing the selection of metrics to be compared. The required interface
2.5. PRESENTATION OF THE INFORMATION

Figure 2.5: WELFIT generated Markov chain of an interface object [de Santana and Baranauskas, 2010].

needs to be able to visualise a limited amount of dimensions that would then be set by the designer. An example is 2D Geographic Information Systems (GIS), in which different variables can be chosen to be shown on the map. In these cases, even when the collected information has more than one variable, it can be presented using pointers that combine colours with shapes [Keim, 2002].

When timestamps are available timelines are a common way of depicting usage. The actions the user performed are located on a horizontal axis that represents time. The Web page the user was in can be added as context information, resulting in visualisations similar to the common Gantt charts. Other approaches are focused on the representation of different interaction events or sensor inputs in the case of capture systems for mobile devices [Carta et al., 2011]. The Figure 2.6 shows an example with various interaction events. Assigning a different colour to different features, or different colour intensity depending on their value, can help finding relations between the various variables. Different participants can be compared in the same way, so similar behaviours appear salient in the visualisation [Yu et al., 2009].
The overload of information from data loggers can be difficult to handle and visualise. Both the nature of the screens and users pose boundaries in the amount of data that can be shown at once. Humans do not make comparisons in the same way if they have to recall information from their memory compared to having both datasets to be compared visible. Screen size has a significant role in that aspect. When datasets are small, this is not a problem, as they can be scaled and shown in the same screen so the comparison is straightforward. They can also be shown using different overlays, which can facilitate the comparison only if the number of overlays is adequate, which will vary depending on the analysis purpose. For example in the case of timelines overlaying more than two at the same time can make inferring the different values over time confusing. However, overlaying many visualisations appropriately helps to find salient situations.

Bigger screens, or more than one, can be a solution but they are cumbersome and hinder the performance of the analysis and the scalability of the solution. Therefore presenting big amounts of data in a cognitively comprehensible way is a challenge [Shneiderman, 2008]. “Overview first, zoom and filter, the details on demand” is considered the visual information seeking mantra [Shneiderman, 1996]. The overview gives an overall perspective of the data, giving importance to the context around the important points. Zooming provides more information about an interesting part of the data while maintaining the notion of position and context. One special zooming technique is the Fisheye, in which the area surrounding the selected point is zoomed in a lesser degree relative to the distance to the selected point. In that way, the selected point receives the biggest focus. The immediate surroundings are zoomed as well to keep the information about its context, leaving the farthest areas untouched. If the elements are presented in a grid, and the grid bends alongside the zoom of the elements, it can explicitly show what kind of distortion is being applied [Carpendale et al., 1997]. Filtering the irrelevant information helps to give a clearer image of
only the appropriate data. In the case of longitudinal data, selecting an appropriate time range is helpful, or filtering data by a particular demographic setting. Finally, the selection of a particular item or group gives details about it.

**Dense Pixel Displays** help to make the most of the screen space, as this technique summarises each data point into a pixel, to maximise the presented data [Keim, 2002]. Using colour codes and ordering the records according to a certain parameter helps to interpret the data. In the example shown in figure 2.7 the records are ordered according to age in a spiral way. The colour coding represents days of hospitalisation last year, showing a clear correlation between this and the age of the patient. The use of **colour coding** can help not only to discern different variables of the same data point but to make it easier for humans to comprehend and make sense out of the visualisation. When the data has no unique features different combinations of colours, shapes and representations can help locate combination of features. Gradients of colours can also be shown to represent quantitative data if the employed colour scale enables a sufficiently wide representation of noticeable colours for the data to present [Levkowitz and Herman, 1992].

![Figure 2.7: Dense pixel displays of database records](Shneiderman, 2008).

When the size of the data or the size of the screen does not allow the display of
all the data at once, the Screen Real State Problem [Tory and Möller, 2004] is encountered. Direct comparisons without relying on memory are not possible. **Coordinated multiple views** (CMV) [Roberts, 2007] provide the means to handle and compare big amounts of data. They consist of a set of different views in the same interface that visualise different data and are coordinated between them. An overview of the data can be shown, where a data point is highlighted while indicating the details for that element in a different view. This technique enables the detection of patterns by allowing to have different points of view from the same data. Differing coordinated points of view help understand complex relations between datasets. Showing various points of view results in a more efficient way to analyse the data, as zooming in and out forces the designer to step back and forth to compare different views of the data [Plumlee and Ware, 2006]. The different views can be customised enabling the detection of patterns and relationships between variables not possible in any other way. However, having multiple views can be confusing if certain guidelines are not followed [Baldonado et al., 2000]. These guidelines make sure important factors are not missed, like enhancing salient features – e.g. pointing out changes occurring in one of the views ease the effort of monitoring all the views for changes. Although practical, this kind of interfaces greatly increases the complexity of the design.

Research in the domain of **video evaluation** has offered alternative ideas of how to visualise behavioural data. Behavioural data cannot be summarised to one image like photographs, or presented in a serial style like text. It poses the challenge of presenting it in a way so no information is missing but remains understandable. Video data is similar to interaction data, as both are multivariate data recorded for extended periods of time.

Different video visualisation techniques were compared using video recordings of a group of people acting freely in a house equipped with cameras [Romero et al., 2011]. The techniques presented included automatically generated techniques, like a VideoCube, which piles up cuts of video, or a regular video player. The generation of other more complex visualisation required a manual intervention, as the Activity Cube. The activity Cube presents the obtained data in three dimensions. It is a combination of GIS visualisations with a third dimension that represents time and enables the detection of salient behaviours.

Visualisation techniques can be employed to combine human flexibility and general knowledge with the storage capacity and processing power of computers to tackle vague data mining objectives. When the hypothesis is not clear, automatic rules cannot
be set, making an automatic analyser of the data infeasible. There have been extensive research into how visualisation techniques can offer exploratory capabilities [de Oliveira and Levkowitz, 2003].

Event and video recording with surveys can be combined [Kim et al., 2008]. This way the drill-down from general statistics to video playback to detect usability problems in video games is possible. Although proved valid this technique did not enable neither longitudinal nor unobtrusive observation.

To make the visualisation interfaces cognitively comprehensible, there are problems that need to be tackled. Humans cannot notice a causation relation between events that are more than 100ms apart [Card et al., 1983]. The consequence is that every visualisation interface must carry out all interaction tasks in a time below that threshold. Both the speed of the data processing and the efficiency of the storage format are critical to optimise the analysis process.

### 2.6 Weaknesses of current common techniques

Due to the exploratory nature of the work done in this thesis, a wide variety of approaches to obtain insight into user behaviour has been covered in this chapter. Alternatives to interaction observation – such as cognitive models – and obtrusive approaches – such as eye tracking – have also been presented. The approach presented in this thesis focuses on the analysis of low-level Web interaction events. Capture and analysis of such events have evolved over the years. Early tools eased the process of instrumenting Web pages for remote usability testing [Scholtz et al., 1998]. The inclusion of temporal metrics allowed researchers to visualise individual users’ paths through the Web site, along with the time spent on each page [Cugini and Scholtz, 1999]. The addition of JavaScript code to the page allows researchers to automatically capture particular events [Etgen and Cantor, 1999]. Task driven approaches enable the remote capture of interaction data so differences with a predefined task model can be found and identified as errors [Paganelli and Paterno, 2002].

The use of proxies provides enough information to visualise individual user’s path, removing the need to access Web logs. Using a proxy can also simplify the process of adding a piece of JavaScript code to every Web page but requires configuration of the participants’ system. Such JavaScript can be configured to track a user’s mouse movement so that mouse interaction can be used as a proxy for attention. Usaproxy [Atterer et al., 2006] is deployed as a proxy and automatically captures interaction events with
several Web page elements. Employing UsaProxy makes the remote capture of interaction data from particular users possible, enabling behaviour analysis of particular demographics for short periods of time [Bigham et al., 2007; Hollande et al., 2010]. Sequences of low-level events can be extracted and visualised to discover unnecessarily long sequences that might indicate a usability error [de Santana and Baranauskas, 2010]. Comparison of low-level interaction between users is also possible if the task being performed is identified [Carta et al., 2011].

Modifying the page to manually insert the piece of JavaScript to capture interaction widens the observation to all visitors of the Web site. In the case of single purpose Web sites – such as Search Engine Result Pages – particular rules can be used to infer users’ behaviour from their mouse activity [Huang et al., 2012]. A wider range of scenarios can also be observed. A flexible approach can capture information such as device type so comparisons between various contexts are possible [Nebeling et al., 2013]. If the task is predefined, a set of events can be captured remotely to compare and even recreate entirely users’ interaction [Breslav et al., 2014]. Different visualisation techniques – such as heatmaps and timelines – enable a thorough analysis of individual interaction sessions.

Several commercial approaches to Web interaction analysis provide scalable longitudinal insight into users’ Web interaction. Google Analytics[10] focuses on supporting the conversion rate of users – conversion occurs when a visitor performs a predefined action that benefits the Web site in some way. HP’s Real User Monitor software (HPRUM[11]) identifies time delays that might affect user interaction. Delays in complex system operations can be identified, and drill down to the particular process causing the delay is possible. TeaLeaf (IBM)[12] offers a more individualised view of users, providing visualisations of HTTP requests sequences for each user. A partial recreation of users’ interaction is also possible, such as the action of filling out a form.

Weaknesses

All techniques presented in this chapter pose various advantages depending on the purpose of the study. Both their advantages and disadvantages have been described, and despite the variety of the techniques, some common weaknesses that have not been tackled yet were found.

---

• *Current common techniques are task oriented.* The tasks to be analysed need to be designed beforehand. Then the environment is tailored for the chosen technique. This prejudgement of the possible interactions narrows down the set of user interactions to be analysed. Another problem this approach poses is that the user has to fit the evaluation technique, instead of the other way around. Even if some techniques enable the gathering of data in a naturalistic and unobtrusive way, if the user does not perform the predetermined task, the analysis will fail. The only way to tackle this problem is to analyse user behaviour without specifying any task, neither before the user interaction nor after. In this way, previously unknown emerging tasks can be identified from the interaction information, and previously unknown task models emerge. Such analysis is challenging as it cannot rely on the prejudgements that would guide the exploration of the data. Furthermore, the use of a naturalistic setting for the observation prevents control over the interaction environment increasing the risk of variability in the captured data.

• *They conform to an established norm.* Using a predefined metric or rule poses a hindrance to the development of innovative interfaces [Greenberg and Buxton, 2008]. The use of traditional metrics – such as task completion time – benefits interfaces which have been designed considering those predetermined norms. Prospective new non-traditional interfaces are evaluated in a very early state when they cannot compete with other more conservative approaches. Unsuitable judgement provokes an early rejection of what could have been a useful and innovative solution.

• *They do not allow longitudinal analysis.* Presented approaches make a longitudinal study of the user interaction challenging. They are limited to the duration of the study, and prolonging it or repeating the experiment is too expensive or cumbersome to be considered practical. One possible approach would be combining different recordings to get a broader cross-sectional view of the user behaviour. This combination is sensitive to the differences in the scales used, as some tools may record in days and others in minutes. Combining data from different studies is also challenging due to the differences in the captured interaction information. As long as the studies stay in laboratory environments and last for short periods naturalistic longitudinal studies will not be possible.

• *Obtrusive observation.* Most of the techniques involving users are obtrusive, in
CHAPTER 2. BACKGROUND

the sense that users are aware that they are being studied. This awareness can provoke unwanted effects that will affect the validity of the results [Webb, 2000]. For example the common guinea pig effect, or the role selection effect, where users alter their behaviour to fit what they consider to be the optimal subject for the study. Studies show that the effect dissipates over time [Webb, 2000]. If the users are told they are being observed at the start, and they are not reminded throughout the study, their behaviour becomes more natural over time. Usual laboratory studies tend to be short, so they may not take advantage of this effect.

A longitudinal study would allow to take a more unobtrusive approach even if the user needs to be aware for ethical reasons that he is part of a study at the start.

The table 2.1 relates identified problems with techniques presented. It has been argued obtrusiveness to be an important characteristic of the observation, as it can pose a significant bias in the study. The use of Web logs and JavaScript in current research has increased the level of unobtrusiveness. Solutions like WELFIT [de Santana and Baranauskas, 2010] can work unobtrusively if changes are made in the deployment. The original solution required user feedback, and privacy issues were tackled reminding the users their actions were being recorded. Although privacy issues should be considered, the highest degree of unobtrusive level should be achieved to avoid as much bias as possible. Some of the presented work is task driven, instructing the users what tasks to perform. It can happen either because the technique is designed that way, or because technical difficulties impede removing this boundary. Other approaches allow the users to interact freely, but the analysis is still focused on a subset of tasks, making them task oriented. These approaches avoid employing formal tests with predetermined tasks so users interact in the same way they would normally do. The execution of particular tasks can then be extracted from the captured free interaction to obtain task oriented metrics, such as task completion time, enabling the detection of possible time delays among those predefined tasks [Jovic et al., 2011]. Established norm is still a problem, as many approaches use a predefined metric as a measurement of usability. Metrics can be fixed, such as the comparisons to optimal paths and task models, or they can be multivariable metrics like in Web analytics tools and the TRUE [Kim et al., 2008] technique, in which different metrics are employed to locate problems. The metrics used to describe observed interaction need to be adjusted to the purpose of the study. Longitudinal analysis of user interaction can complement existing predefined metrics with measurements concerning user interaction evolution. The set up
has not been found to be a problem. Apart from special situations where physical equipment or modified software needed to be used on site, the use of proxies allows to easily set up a natural observation setting. However, there is a lack of longitudinal solutions. Several solutions allow a longitudinal approach but they are impractical. When a longitudinal approach is taken, usually summaries of the data are presented. Such approach does not allow a deep introspection into the data to understand the event in its context.

Techniques for understanding users’ behaviour ultimately serve to guide the design of interfaces that support effective and accessible interactions. These techniques can be used to evaluate interfaces according to different usability principles. The direct observation of the users during the interaction may be required to evaluate some of them. If the designers’ subjectivity is involved, it needs to be contrasted with users’ opinion. The concept of familiarity, in which users’ previous experiences are employed to make a more intuitive interface, can only be tested by directly observing users and their reactions to the interface. The evaluation of other principles requires a long term observation that would be difficult without a remote solution. The concept of progressive disclosure describes an interface that remains simple for novice users but provides all the complex interaction necessary to satisfy expert users. A longitudinal approach would be needed to find out what functionalities novice users value the most, and if the rate at which they discover new functionalities is optimal. Interfaces should also be designed with the intention of being usable by as many users as possible. Users have individual needs, and individual problems that may not be common to all users. A flexible interface would enable tailoring the system to specific necessities. A longitudinal approach helps to find what those necessities are and how the designed solutions tackle them in the long term, as long as it enables the identification of the different users.

The majority of design principles for a better usability try to optimise the interface for novice users, so they can become experts sooner. The interface should be understandable, easy to interact with and accessible to novice users. These factors – among others – speed up their learning process, minimising the need for support, thus reducing costs in the long term. The difficulty of performing longitudinal studies in laboratory studies was mentioned, as well as how any obtrusiveness should be avoided. Therefore, a remote observation solution, which is both unobtrusive and longitudinal, is necessary.
2.7 Summary

Even though longitudinal studies are extensively employed in a wide variety of domains they lack a clear definition of what they constitute. Their duration can vary from days to decades depending on the purpose of the study. To obtain a concrete description, a set of requirements that these studies should comply to were gathered. A longitudinal study needs to present a plan to analyse data collected over time. To make comparisons over time possible at least one dimension needs to remain constant. In the case of Web interaction Web pages from the same Web site might differ. Therefore user interaction evolution might be different between the various Web pages and needs to be considered. As opposed to cross-sectional studies, a continuous observation of the same users enables the comparison of the interaction of the same user over time. Duration of the study can vary, enabling the categorisation of studies according to their length as micro, meso, and macro. Several studies present a plan for capturing data over a predefined length of time, arguing the set period is enough to fulfil the needs of the corresponding analysis. In the case of observing how user interaction changes over time this length cannot be defined. The rate of change is unknown and possibly different for each user. Instead of predefining a concrete duration for the study, a way of collecting the data will be deployed, and data will be analysed and collected iteratively.

A clear definition of interaction episodes is necessary to enable comparisons over time. Time-outs are commonly used in remote approaches, but previous work does not provide a valid technique to determine their duration. Found approaches were either based on arbitrary rules to split data into episodes, or only applicable to single purpose Web sites or applications. An empirical approach, based on analysis of interaction data captured in a naturalistic way will be employed.

Reviewed approaches to user interaction observation fail to support a detailed naturalistic longitudinal analysis. Laboratory studies provide high-level of detail, and control over confounding variables but are inherently bounded in time. Users cannot interact in their environments, possibly introducing bias to the observation. Remote approaches are coarse or limited in time. Other approaches such as studies based on self-reports from participants can be biased due to lack of accuracy from users’ recollection. Enforcing the tasks to be performed provides comparable executions. Comparisons of the execution of the same task over time are possible, simplifying analysis and enabling the extraction of metrics. Such analysis introduces prejulgements based on non-validated assumptions of the interaction. Although valid to analyse specific tasks in single-purpose interfaces, they are not extensible to freely captured data.
2.7. **SUMMARY**

Weaknesses found among the reviewed work were presented. The goal of this thesis cannot be achieved without tackling these weaknesses. Inspired by the reviewed work introduced in this chapter, a set of requirements were identified for the capture and analysis of Web interaction data. In the following chapter, the designed naturalistic observation solution is presented. This solution tackles the weaknesses found in existing observation tools, enabling a longitudinal study supporting the understanding of user interaction evolution.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Think-aloud protocol</td>
<td>1996</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Possible</td>
<td>No</td>
<td>No</td>
<td>Audio recording agent</td>
</tr>
<tr>
<td>Coaching protocol</td>
<td>1996</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Possible</td>
<td>No</td>
<td>No</td>
<td>Expert present</td>
</tr>
<tr>
<td>Question-suggestion protocol</td>
<td>2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Possible</td>
<td>No</td>
<td>No</td>
<td>Expert present</td>
</tr>
<tr>
<td>Eye tracking</td>
<td>2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Not necessarily</td>
<td>No</td>
<td>Yes</td>
<td>Specific physical set up</td>
</tr>
<tr>
<td>User feedback</td>
<td>2009</td>
<td>Yes</td>
<td>Natural</td>
<td>Not necessarily</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Mouse tracking</td>
<td>2009</td>
<td>Yes</td>
<td>No</td>
<td>Yes (not can change it)</td>
<td>No</td>
<td>Yes</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Website analytics</td>
<td>1978</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Evaluation of Mouse Rate Controlled (…)</td>
<td>Card et al., 1987</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>OLAP</td>
<td>1998</td>
<td>No</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>WebVip</td>
<td>1998</td>
<td>No</td>
<td>Natural</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>VISVIP</td>
<td>2000</td>
<td>No</td>
<td>Natural</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>WebRemUsine</td>
<td>2002</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Classifying search queries</td>
<td>Rose and Levinson, 2004</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>JavaScript and Eye tracking</td>
</tr>
<tr>
<td>UsaProxy</td>
<td>Atterer et al., 2006</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special software</td>
</tr>
<tr>
<td>MouseTrack</td>
<td>Arroyo et al., 2006</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Analysing page requests and interviews</td>
<td>Weinrich et al., 2006</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>WebStim</td>
<td>2007</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Eye tracking for SERP search analysis</td>
<td>Censusi and Gau, 2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special software</td>
</tr>
<tr>
<td>Classifying errors via variability in SERP</td>
<td>Winter and Druker, 2009</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special software</td>
</tr>
<tr>
<td>TRIS</td>
<td>Kim et al., 2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special software</td>
</tr>
<tr>
<td>Engaging Story</td>
<td>2006</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Eye-gaze coordination</td>
<td>SERP</td>
<td>Reddy et al., 2008</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Unders and cue events</td>
<td>Akam et al., 2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special software</td>
</tr>
<tr>
<td>Displaying advertisement link via mouse cursor</td>
<td>Guo et al., 2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special software</td>
</tr>
<tr>
<td>User experience over time (…)</td>
<td>Katrinaos et al., 2009</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Longitudinal study of people learning (…)</td>
<td>Kaltenbach et al., 2009</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>User-interaction with Input Device Evaluation (…)</td>
<td>Kurten et al., 2009</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>“It is Better than Filling (…)”</td>
<td>Madsen et al., 2009</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>InteractiveTracker Benchmark and Report</td>
<td>2011</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>WELFIT de Sanabria and Fernandez, 2011</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Measuring Children watch</td>
<td>Holldave et al., 2012</td>
<td>No</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Hearing problems on screen</td>
<td>Nghi et al., 2013</td>
<td>No</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>User-visible failures in AMM</td>
<td>2013</td>
<td>No</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>A longitudinal study of usability of front end (…)</td>
<td>K{&quot;u}ld{&quot;o}r{&quot;o}ö et al., 2013</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>On the Reliability of Assessment (…)</td>
<td>Katzenbach et al., 2009</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Pocket Box</td>
<td>Kretz et al., 2020</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CUPS cookie bypass or detection</td>
<td>2011</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Support for remote system monitoring</td>
<td>Carta et al., 2011</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Activity cube</td>
<td>Romoser et al., 2011</td>
<td>Yes</td>
<td>Laboratory</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cycles in Performance Problems</td>
<td>2011</td>
<td>Yes</td>
<td>Laboratory</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Looking for behavioral patterns</td>
<td>Jain et al., 2011</td>
<td>Yes</td>
<td>Laboratory</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bug detection in the wild</td>
<td>Ho et al., 2011</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>CredibleScope</td>
<td>Krasnov and Stang, 2012</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Identifying behaviors from mouse</td>
<td>Huerta et al., 2021</td>
<td>Yes</td>
<td>Laboratory</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bug detection in the wild</td>
<td>Novak et al., 2011</td>
<td>Yes</td>
<td>Laboratory</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Short-term Methodology for Long-term Testing</td>
<td>2012</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Task switch detection via Eye-tracking</td>
<td>Chen et al., 2013</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Task identification via Eye-tracking</td>
<td>Chen et al., 2013</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Inferring user click through via Markers models</td>
<td>Novak et al., 2013</td>
<td>Yes</td>
<td>Natural</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Predicting users’ experiences to dynamic content</td>
<td>Sahu et al., 2013</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>CrossStudy</td>
<td>Niebling et al., 2013</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Mirro</td>
<td>Dhillon et al., 2014</td>
<td>Yes</td>
<td>Natural</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Cognitive monitoring</td>
<td>Gual et al., 1980</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Cognitive 8G</td>
<td>Jain et al., 2001</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>ColloDeK</td>
<td>Kholodenko et al., 2005</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
<tr>
<td>Eye gaze cognitive model</td>
<td>Henschen and Hornet, 2007</td>
<td>Yes</td>
<td>Laboratory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Special input device</td>
</tr>
</tbody>
</table>

Table 2.1: Weaknesses of current common techniques
Chapter 3

Analysis Framework

The previous chapter explored related work providing insight into users’ Web interaction behaviour. Research results based only on users’ self-reports have been found to be unreliable, due to subjectivity and memory factors. Using low-level interaction data as a proxy to higher level concepts offers an objective and unobtrusive alternative. Longitudinal analysis of low-level interaction data has the potential to provide detailed insight on how users’ interaction changes over time. However, the capture and process of such fine-grained interaction data over extended periods of time is challenging. The fine-granularity of the data can easily be lost if information is not aggregated appropriately. Unfortunately, none of the reviewed existing techniques provided data suitable for such longitudinal study.

Different interaction observation alternatives have been compared to extract a series of requirements for the captured interaction data. Based on those requirements an appropriate interaction data capture solution has been designed and implemented. Challenges derived from the extent of the study, such as the processing of large amounts of interaction events and missing or erroneous data are identified and tackled.

Implemented capture solution has been deployed in real world Web applications and sites for over sixteen months. To obtain meaningful information from the captured low-level interaction events a scalable analysis methodology has been designed. Challenges derived from the lack of control of the observation and the uncertainty about users’ motivation needed to be tackled. Micro behaviours are presented as a way to obtain comparable units of interaction. Temporal correlations among features extracted from selected micro behaviours act as proxies for higher level concepts. An analysis methodology has been designed capable of discovering correlations from uneven data coming from different users.
3.1 Naturalistic capture of longitudinal Web interaction

A solution to capture longitudinal Web interaction data in a naturalistic way has been designed. This solution fills existing gaps in current capture solutions contributing to the set of tools available for researchers to study user behaviour. Prior to the design of the solution, requirements needed to be defined based on the required characteristics of the captured data. A formative study has been carried out, comparing two different ways of capturing Web interaction data. The results of the study shaped the subsequent design of the capture solution.

Captured data needs to be processed before it can be used as input for analysis. Temporal attributes and context are included to support a longitudinal analysis. Problematic aspects of the data need to be considered. For example, missing data is completed and invalid data filtered out. This awareness of the limitations of the captured data contributed to the design of a robust and reliable analysis.

3.1.1 Formative study

Different interaction data capture techniques are compared to gather the requirements of the tool and inform the design and implementation. The feasibility of their deployment in various Web sites and applications and the analysis possibilities of the captured data with each of them are taken into account. Both capture techniques have been deployed in the Kidney and Urinary Pathway Knowledge Base (KUPKB) Web application. This Web application is designed to help researchers, physicians, and students allowing them to query and browse published datasets from scientific publications and other related renal databases.

The manual annotation technique entailed manually extending the code of an already running Web application. The modification consisted on adding mechanisms to listen to specific events and logged them to a text file for analysis. This technique is not generalisable as it is hard-coded, but enables the customisation of the events to gather, avoiding information overload.

KUPKB has been modified to capture 22 different application specific events, including additional information unique to particular events, such as keywords from search queries and selected filters. A sample of the captured events is listed in Table

[http://www.kupkb.org/](http://www.kupkb.org/)
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

<table>
<thead>
<tr>
<th>KUPB events</th>
<th>Action description</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>A01</td>
<td>Basic search for single molecule</td>
<td>Search term</td>
</tr>
<tr>
<td>A02</td>
<td>Basic search for multiple molecule</td>
<td>Search term</td>
</tr>
<tr>
<td>A03</td>
<td>Basic single selected search</td>
<td>Selected term</td>
</tr>
<tr>
<td>A04</td>
<td>Basic multi selected search</td>
<td>Selected terms</td>
</tr>
<tr>
<td>A05</td>
<td>Sort result by gene</td>
<td>Empty</td>
</tr>
<tr>
<td>A10</td>
<td>Sort result by molecule type</td>
<td>Empty</td>
</tr>
<tr>
<td>A11</td>
<td>Basic result single value filter</td>
<td>Filter options</td>
</tr>
<tr>
<td>A12</td>
<td>Basic result multi value filter</td>
<td>Filter options</td>
</tr>
<tr>
<td>A13</td>
<td>Advanced search expression</td>
<td>Search option</td>
</tr>
<tr>
<td>A14</td>
<td>Advanced search single location</td>
<td>Search option</td>
</tr>
<tr>
<td>A18</td>
<td>Advanced search filter multiple</td>
<td>Filter option</td>
</tr>
<tr>
<td>A19</td>
<td>Basic search results found</td>
<td>Empty</td>
</tr>
<tr>
<td>A20</td>
<td>Basic search no result</td>
<td>Empty</td>
</tr>
</tbody>
</table>

Table 3.1: Sample of events captured by the manual annotation

Some of these events provide “additional information” captured from the data layer of the Web application, such as the queries employed to retrieve information from the Web application’s database.

The JavaScript injection technique consisted in a modification of UsaProxy [Alterer et al., 2006]. The original tool relies on a proxy that sits between the user and the Web site server and inserts a piece of JavaScript code in all the Web pages served to the user. This JavaScript code records all low-level events from the user, from mouse movements and key presses to HTTP requests.

UsaProxy has been modified so it could be used without requiring the configuration of a proxy. The modification entailed that JavaScript code has been inserted into each Web page to make the deployment possible without configuring the client side. Additionally, avoiding the use of proxies tackles the privacy concern of recording all user interactions with non-registered Web sites, limiting the recording to the modified Web site. It also ensured that the Web site kept working even if the server receiving the interaction data failed, as opposed to what would happen when using a proxy.

Comparison of capture approaches has showed that JavaScript injection provides finer interaction data and a simpler deployment. Instead of high-level events, specific to the Web application, low-level events such as mouse movement and keyboard interaction can be captured. These low-level events could also be obtained through manual annotation, but the wide range of possible events makes it impractical. The main advantage of high-level events is the context information they provide.
lists a sample of the events registered in the Web application, along with the additional information that serves as the context. If the context from the mentioned high-level events can be inferred from low-level events the resulting data would provide a more complete observation of user interaction. Therefore, JavaScript injection would provide an simpler, more complete solution to capture Web interaction data, removing the need of manually annotating each Web page.

Context information mainly relates to the current state of the Web page’s Document Oriented Model (DOM). The DOM state of a page defines the logical structure of the document and the way a document is accessed and manipulated [W3.org, 2015]. Manual annotation behaviour can be emulated by modifying the injected JavaScript, so it extracts information from the generated DOM. For each high-level event an alternative listener has been implemented extracting the necessary information on the fly from the user’s browser DOM state. Depending on the emulated functionality, this alternative listener observed different aspects of the Web page elements. List of activated checkboxes or computed Cascading Style Sheets (CSS) colours have been used to infer the different states and contexts in the Web application.

Implementing the corresponding high-level events has been found to be possible, although not scalable. High-level events are too specific and cannot be generalised to all Web sites and applications. Nevertheless, it has been determined that the final Web interaction capture solution would include the computed state of the Web page’s DOM to make extraction of Web page specific events possible. Additionally, user’s interaction can be fully recreated by reconstructing the context through the state of the DOM at that particular instant. Changes to the DOM derived from dynamic Web updates are also stored in the database. The analysis performed in this thesis is limited to the set of events common to all Web sites and applications to keep the procedure generalisable.

**Possibilities of the captured data**

As a way to explore the possibilities of the interaction data captured by the JavaScript approach, a Web application visualising the interaction with interface elements has been designed. Web applications are commonly designed to fulfil a set of use cases based on an assumed usage of the Web application interface. Design and improvements of the interface are driven by scenarios based on those interaction assumptions. Even if real users’ interaction patterns fall outside those assumptions, they will not be
considered, preventing an interface adaptation to the real users. If the available interaction options within a particular application are limited a tailored approach looking at transitions between those known interactions can be useful. Different executions can be clustered according to these transitions, so different executions can be categorised according to the task being performed [Lafreniere et al., 2010].

Data captured from KUPB\footnote{http://www.kupkb.org/} using the JavaScript injection approach has been used for the analysis. KUPB is a highly dynamic Web application, with self-updating Web elements. The interactive elements from the Web application have been identified, and transitions between them extracted. A visualisation tool based on D3\footnote{http://d3js.org/} has been designed to enable an interactive exploration of these transitions. D3 has been chosen to be able to provide easy online access to the visualisations with a high degree of interactivity. The processed data contains only the necessary information to be visualised, minimising the privacy risks.

An easily generalisable way of obtaining relevant high-level events from the already captured data has been employed. Mouse click events are extracted, as well as the text content of the nodes in which the users clicked. A list of the text content from identified interface elements is then introduced to identify high-level interactions.

This analysis considers all buttons and the selection of different tabs as relevant high-level events. Pages are becoming more dynamic, offering page content updates without the need of page requests. Regular Web logs and systems based on page-request analysis only consider events causing a page request, ignoring events triggering Web page updates. Events from dynamic pages are therefore ignored and the related behaviours overlooked.

An example of a dynamic interaction is shown in Figure 3.1. In this example, the user performs a search in the site by pressing the button “Search”. The Web application shows a pop-up listing the results, where the user presses the button “Continue”, which triggers a more detailed list of results. The option of adding a filter is available by pressing “Add Filter”. These three interactions with buttons are an example of high-level interaction happening within the same Web page. A graph showing the transition between the selected relevant actions is shown in Figure 3.2. In this graph, the interaction examples mentioned before are numbered. When hovering over one of the options, the transitions starting from that action are coloured red, and the transitions ending in it are coloured green. This visualisation shows the overall trend, without
Figure 3.1: Example of the KUPB Web application interaction. The user performs a basic search, selects results to be seen in more detail, and adds a filter. Each interaction’s number indicates its correspondence to the graph in Figure 3.2 disregarding outliers.

In the example, an unexpected behaviour while hovering over “Advanced-Search” was found. The interface is divided into tabs (see Figure 3.1), with different interfaces and interaction options. The visualisation shows that the user had selected the second tab (“Advanced-Search”), but afterwards interacted with an interface element only available in the first tab (“Search” button). This transition was not initially considered in the Web application design, as to make this option available the corresponding tab should have been selected. An alternative way of getting to this option is by reloading the Web page, which changes the tab to the default one (“Molecule Search”).
Figure 3.2: Transitions between relevant actions in KUPB Web application. “Advanced Search” is being currently selected. Red lines indicate what actions are performed after “Advanced Search”, and green lines indicate what actions are performed before it.

**Identified requirements**

The formative study enabled the choice of an appropriate technique to capture Web interaction data. *JavaScript injection* has been found to be an appropriate solution to capture interaction data. Captured low-level events need to be processed due to their fine granularity, but the resulting data leads to more possibilities for analysis. If necessary, the recreation of Web page specific events has been found to be possible using a combination of the captured low-level events and the state of the page.

Throughout the formative study different requirements for the design of the capture tool have been identified. Several of these requirements had already been identified as weaknesses of existing approaches – see section 2.6 – because otherwise bias could be introduced into the observation. The data obtained through existing techniques cannot cover all the identified needs so the final interaction observation solution has been
designed taking these requirements into consideration.

**Remote** Constraining users to a physical location can be alienating, particularly if the user is not familiar with the environment. Users might interact in a different way than they would in their homes or workplaces. Different remote approaches have been mentioned in the related work. However, they are either pinpointed in time or cannot provide a naturalistic observation.

**Unobtrusive** Obtrusive observations introduce bias into the obtained data. Although an initial notification to the users about the observation is necessary to tackle privacy concerns, the capture of the data cannot cause any interference with their interaction. This way the interaction remains natural, and real usage of the Web site or application can be obtained.

**Longitudinal** Several longitudinal techniques have been mentioned in the related work. In the case of laboratory studies, periodic sessions with the same users are necessary to provide insight into temporal factors. The start of the process of skill acquisition can be observed if participants are novice users. To extend the analysis longer observations or remote approaches are necessary. Unfortunately, observations for extended periods of time can be challenging in laboratory studies. In the case of existing longitudinal remote approaches, results provided are based on summaries of the captured data, missing the appropriate depth to provide insight into users’ interaction behaviour. In the case of cross-sectional studies, a virtual longitudinal study can be carried out. However, these approaches do not account for variability between users and require the gathering of demographic information.

**Fine-grained** Low-level events depict a detailed picture of the interaction and allow to understand subtle changes in users’ interaction behaviour. Task-driven approaches capture low-level events but are used to compute predetermined metrics, such as completion times and deviation from optimal models. Fine-grained interaction data freely captured in the wild provides the base needed for a detailed analysis of users’ interaction behaviour. A longitudinal data-driven analysis avoids prejudgements of the interaction and provides the means to obtain an unbiased understanding of how users’ interaction changes over time.
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

3.1.2 Web interaction capture solution

UsaProxy has been chosen as the starting point for the design of the capture solution. This tool has been extensively modified, so that the identified requirements are met. This way a scalable long-term capture of interaction data is supported. The capture solution has been deployed in the School of Computer Science Web site so captured data could be used for the iterative improvement of the tool. Captured events have been revised, and where necessary, additional data to provide enough context for each event has been captured.

Design of interaction capture tool

UsaProxy has been previously employed in research to perform short-term observations. Participants’ equipment needs to be modified to configure the required proxy for the capture of the data. Unfortunately, the capture tool cannot support long-term interaction capture in its original state, and does not provide a scalable deployment solution. The number of participants is limited by the necessity of a proxy, and the storage of the data is only suitable for data captured over short periods of time. Different design decisions have been taken, affecting the way the tool is deployed and how the data is stored.

UsaProxy relies on the configuration of a proxy on the client side. Any modification to the client side increases the obtrusiveness of the solution, and limits the number of prospective participants. Making Web servers rely on external proxies poses additional critical issues. If the proxy fails, the client loses access to the Web site. Additionally, the proxy server would need to be able to handle all incoming traffic from users, making a scalable solution challenging.

The deployment methodology has been modified by allowing the designed solution to be deployed through the manual modification of the Web pages of the Web sites or applications. A piece of JavaScript code is added to all pages, in a similar way to other commercial Web analytics tools, such as Google Analytics. This approach is inconspicuous to the users, thus requiring a way of notifying new participants of the observation. Participants are uniquely identified via the use of a cookie which is deployed the first time a user accesses the Web site. Although useful for a remote Web setting, the use of cookies cannot guarantee an entirely reliable identification of a user. If a particular user employs different Web browsers, or systems, each instance
is considered as a different user. This situation can also arise if the user deletes or modifies the cookies stored in the browser. If no pre-existing cookies are found in the user’s browser a message is shown notifying the participant about the collection of interaction data. In those cases, no interaction data is captured until the user purposely continues with the interaction.

Several privacy concerns are also tackled by changing the way the interaction capture tool is deployed. When using a proxy, all pages accessed by the users are tracked. Users are uniquely identified by a cookie accessible from all sites, thus enabling the recreation of a single user’s interaction throughout all Web sites and applications. The proposed alternative limits the capture of the interaction to the Web pages where the JavaScript code has been added. Users are uniquely identified through a cookie, but its scope is limited to the Web domain where the interaction is taking place effectively preventing profiling users across different sites. IP addresses are captured but they do not offer a certain reference of unique users and are not used in the analysis. Additional privacy options have been included for sensitive applications, including opt-out options, and a blacklist of page elements – so fields like password inputs could be excluded from the interaction capture.

The capture solution is hosted on a server different from the server hosting the Web pages. For this study, interaction data captured from cs.manchester.ac.uk and kupkb.org is sent to a server hosted in wel-experimental.cs.man.ac.uk. Even if the researcher has no direct access to the Web server, having access to the capture solution server is enough to modify the contents of the JavaScript to be injected. Apart from the aforementioned advantages of scalability, and preventing the capture tool from causing any access problems to the Web pages, it also allows iterative improvements. In the case of problems, such as incompatibilities with particular browsers, the code can be fixed, making it immediately accessible to all Web servers where the capture tool has been deployed. During the design and implementation of the solution, this feature has been found particularly useful. New events and features could be added seamlessly, contributing to the completeness of the observation.

Employed data storage needs to be scalable and support longitudinal analysis. Original UsaProxy makes use of text files to store captured events. Interaction data captured over extended periods of time from a high number of users makes the use of text files difficult to manage for longitudinal analysis.

The long-term capture of low-level interaction leads to large amounts of data, comprised of millions of small records. The appropriate structure to store this data depends.
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

greatly on the kind of events to be captured. The solution should be able to handle the fact that the type of events observed may change over time (something that is particularly an issue on the Web), meaning a database may need to be non-structured. The amount of data needs to be assumed to be ever increasing, particularly in studies where the end is not predefined. Dealing with this longitudinal aspect of the capture requires careful thought.

The selection of the database obviously affects how the data is processed in the future. A document-oriented database [Mongodb.org, 2015] has been chosen to store data for several reasons. It does not require a predefined structure, allowing it to contain different events that store various types of information. Lack of a rigid structure removes the need of knowing in advance what attributes to store, enabling the addition of new attributes without major modifications. Collecting data comprehensively, and allowing the inclusion of attributes of which the usefulness has not been tested in advance is strongly recommended [Dumais et al., 2014] and has been found to be particularly helpful. One example of this advantage is the addition of a timestamp to all events recording when the Web page is opened. Although not initially planned, it has proven useful for discerning when a user is using more than one tab and helps to differentiate events from different tabs.

Event compatibility needs to be considered as users’ setup cannot be controlled. One of the identified requirements specifies that the capture solution needs to be as unobtrusive as possible from the perspective of the user. Achieving this unobtrusiveness is an essential aspect of the observation, in order not to affect users’ interaction. Selected interaction events need to ensure this unobtrusiveness across all platforms and browsers. One of the biggest technical challenges has been caused by the fact that browsers can differ greatly with respect to event compatibility: whilst some browsers are highly compliant with the latest World Wide Web Consortium (W3C) event recommendations [W3.org, 2015], others consider them only to be in their roadmap. Browsers’ compatibility is also continually changing, so it needs to be appropriately documented at the start of the study, and kept up to date. Table 3.2 indicates which events are only partially supported by certain browsers. A failure to document the compatibility of events risks misinterpreting the data. If certain events only occur on some platforms or browsers – and this situation is not clearly documented – the researcher might assume conditions are equivalent for different users when they are not.
CHAPTER 3. ANALYSIS FRAMEWORK

Collected data

Specific users are tracked via a unique, anonymised code stored in a cookie. Captured events include all mouse and keyboard interaction, as well as browser window events, changes to the state of the elements on the page, and other system information. The state of the DOM and its changes during interaction are also stored, allowing the context of interaction to be potentially recreated. All the data is stored in JavaScript Object Notation (JSON) format in a Non Structured Query Language (NoSQL) database, which allows for extensibility and query scalability, and means that further events can easily be included as required. The design also supports the modification or removal of existing events should they become deprecated.

Window events, indicating when the page is loaded or when the page has lost focus, combined with user specific information, provide context for lower-level events. This information allows the grouping of events according to the specific environment in which they took place, such as the browser tab – identifiable via the page load timestamp – and URL. Logging the platform and browser version supports the control of event compatibility according to a system. The ever-changing nature of the Web and the possibility of enabling partially supported events makes this information useful for discerning problems with the way a given system handles interaction events.

The following information is collected for all events:

- **Web site ID**: identifier of the Web domain generating the data (e.g. 10002, 10006).
- **User ID**: identifier for each user, stored in a cookie (e.g. yAubjmZkSrD9).
- **IP**: user’s Internet Protocol address (e.g. 130.88.99.220).
- **URL**: address of the Web page (e.g. cs.manchester.ac.uk, www.kupkb.org/#tab0).
- **Timestamp**: time stored in ms based on the capture server (e.g. 1448466686).
- **Load timestamp**: time when the page is loaded, same for all events occurring within that page (e.g. 1537250757).
- **Time offset**: difference in minutes of the user’s local time with respect to the Coordinated Universal Time (UTC) (e.g. 0 for UK, +60 for Spain).
- **Platform**: details of the operating system (e.g. WinNT, win32, linux.x86).
- **Browser**: details of the browser (e.g. Chrome46.0.2490.86., Firefox 42.0).

Certain low-level events (see Table 3.2) are a combination of events automatically triggered by the browser during user interaction, such as Mouse down and Mouse up,
3.1. **NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION**

and manually triggered ones. In the case of automatically triggered events, the extent to which they function correctly depends entirely on the browser. Some of the events – marked with an asterisk (*) in Table 3.2 – are only partially supported by a few browsers and are therefore captured for possible use, but usually discarded in subsequent analyses to avoid an inaccurate interpretation of an unsuccessful interaction event.

Manually triggered events rely on periodic queries at predefined intervals or queries triggered under certain conditions. To obtain the value of *Window scroll*, the state of the browser’s window scroll is queried every 200ms. In the case of *Mouse move*, the position of the mouse is compared every 150ms. A *Mouse select* event is triggered at every *Mouse up* event, on checking if the result of that action is the selection of any content.

For some events, additional information is captured. *Mouse coordinates* are stored for events concerning mouse interaction. The reported *offset* coordinates are relative to the Web page’s node element where the interaction is taking place. The stored information is the following (the example used has coordinates 243x204 and offset coordinates 31x8):

- **X coordinates**: X coordinates relative to window (e.g. 243).
- **Y coordinates**: Y coordinates relative to window (e.g. 204).
- **OffsetX coordinates**: X coordinates relative to node element (e.g. 31).
- **OffsetY coordinates**: Y coordinates relative to node element (e.g. 8).

Additional events that stored other event-specific information included: the pressed key for *Keyboard* events; content of the selected text for text selection events; and a measurement of the scrolled distance for scroll events. If any event implied interaction with a node, information about that node is also stored. The following node information is captured to allow identification of the element the user is interacting with. An asterisk (*) indicates that this value is only captured if the DOM element contains this specific attribute. In the case of text value, it is only captured if the node is of type text.

- **DOM path**: path of the node in the DOM (e.g. BODY/DIV[5]/TABLE[1]/DIV).
- **Id**: id attribute of the node* (e.g. searchButton from node `<button id="searchButton">Search</button>`).
- **Name**: name attribute of the node* (e.g. div from node `<div>text</div>`).
- **Type**: same behaviour as name, added for increased compatibility.
Table 3.2: All events contain generic information. * indicates that this event is partially supported at the time of design.

- **Href**: destination of the link contained in the node* (e.g. url.com from node <a href="url.com">link text</a>).
- **Text content**: first 100 characters of the text content (e.g. text sample from node <p>text sample</p>).
- **Text value**: text value of the node* (e.g. textInput from <input value="textInput">).

**Mobile events** are also captured. Third party libraries provide access to sensors commonly available in mobile devices. To ensure compatibility, simplify maintenance, and allow future extensions, the use of such libraries has been avoided. Instead, the set
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

<table>
<thead>
<tr>
<th>Event group</th>
<th>Events</th>
<th>Additional information</th>
<th>Coordinates</th>
<th>Node information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Touch</td>
<td>TouchStart</td>
<td># touch inputs</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>TouchEnd</td>
<td># touch inputs</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sensors</td>
<td>Gyroscope</td>
<td>Orientation values</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>Landscape/Portrait</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Motion</td>
<td>Acceleration values</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Captured mobile events

of mobile events to be captured has been based on W3C recommendations [W3.org, 2015]. Any processing of the captured events on the client side has been avoided. Therefore, composite events common to touch interfaces as pinch, swipe, and drag events have not been considered. Instead, all events regarding touch input have been captured, including context information such as the number of fingers in contact with the screen. This way extraction of complex touch gesture interactions is possible. The orientation of the device is recorded each time the difference in orientation exceeded a set threshold of 10 degrees. Acceleration of the device is also stored, whenever a change in acceleration is detected every 200ms. These thresholds have been set arbitrarily with the purpose of reducing the overload of mobile events. They have been considered appropriate to limit the amount of captured mobile events over extended periods of time but studies requiring more precise readings from mobile sensors would need to tune them accordingly. The sensitivity of the captured data depends on the values set for these thresholds so all analysis of mobile sensor data needs to take into account the conditions in which the readings have been obtained. All the mobile events contain the basic information previously mentioned, as well as additional information, such as the coordinates of the touch event and the number of concurrent touch inputs – see Table 3.3.

3.1.3 Data processing

The amount of data captured throughout the study posed difficulties that needed to be tackled. A way to extract meaningful information for analysis is necessary. To ease the processing of the data users’ interaction has been split into interaction periods, similar to the concept of sessions in controlled laboratory observations. Temporal grouping of events is a common practice for data captured over extended periods of time. Unfortunately, no concrete threshold has been found to define these temporal groups. The unobtrusiveness of the observation prevents asking for users’ feedback, leading to a
lack of information about users’ motivation. Interaction data is captured from users interacting freely with all Web pages in the Web site or application. Due to the lack of consistency between pages and without the information about the tasks users are performing, each Web page effectively represents a possibly different scenario. A way of measuring differences between these scenarios is necessary to make the data comparable. Other problems derived from the lack of control needed to be identified to prevent analysis errors and misinterpretations of the results. Identifying limitations of Web-based interaction capture and prospective problems for processing the data is critical.

Temporal metrics

The designed capture solution gathers low-level interaction events as individual entities. Subsequent analysis needs to determine how these events are related to each other depending on the purpose of the study. For example, in the case of studying changes caused by some external stimuli, interaction data could be grouped with respect to the correspondent stimulus. To obtain insight into how interaction changes over time for each user, interaction data needs to be temporally ranked relative to each user. Therefore suitable metrics to temporally order events and quantify time differences between them are necessary.
**Episode definition** is a necessary abstraction to make the processing of longitudinal data possible. Grouping events into episodes is a common practice in user observation. In the case of laboratory studies, the length of the episode is determined by the duration of the session. In remote approaches, episodes can be represented as executions of predefined tasks. When users interact freely a way of delimiting different interaction periods is necessary. Data can be segmented into episodes equating a single, continuous session of Web use. Applying an inactivity threshold to user interaction [Google, 2014, He and Göker, 2000, Thomas, 2014] is a common practice in remote uncontrolled environments. Unfortunately, there are differences among the values employed for such threshold. One of the risks of using an episode time-out is that it might bias the resulting episodes by merging different episodes together. As a way to obtain a data-driven threshold and minimise that risk, the length of inactive periods have been analysed. All inactive periods between events have been extracted from interaction data captured from the cs.manchester.ac.uk Web site over a period of 8 months. The cumulative distribution function of the periods between events has been calculated, setting one minute and 24 hours as the lower and upper boundaries respectively. Figure 3.3 shows the plot of the resulting cumulative distribution function. As expected, the plot follows a power function. The lowest point in which the change rate is small enough is then considered the threshold between episodes. Using as the threshold a point where the variance is small enough ensures that the number of episodes containing a longer interruption is minimal [Zakay and Feitelson, 2013]. Values have been compared with a precision of 1 minute, and the first value where the change is smaller than 0.01% has been selected. The resulting episode threshold is 40 minutes, which is depicted with a vertical red line in Figure 3.3.

The distribution of the length of the calculated episodes can be seen in Figure 3.4 with a vertical red line indicating where the threshold is set. The plot is limited to episodes with a length higher than 10 minutes to increase readability. If the selected threshold value were introducing a bias in the resulting episode lengths a drop would be seen at that point. The number of episodes decreases as their length increase, showing no sudden drop around the selected threshold.

Episode count is commonly used as a metric of users’ experience, where a higher number indicates a more experienced user. This approach can not be applied naively to data captured in the wild, however, for two reasons. Firstly, the basic length of each episode is not controlled and may vary considerably, although the majority last less than five minutes. Figure 3.5 shows this positive skewness, as well as the threshold...
chosen to distinguish between different episodes. Secondly, within two episodes of a given length the level of interaction can also vary. The real interaction could commonly be between mere seconds and several minutes; it is even possible for episodes to contain no interaction – in the case of an accidental page load that is closed immediately afterwards. Therefore episode length alone has been considered not to reflect real usage time and the new metric of active time has been introduced by aggregating the amount of time the users spent being active – time spent generating interaction data – on the page.

**Active time** represents the length of time the user has been active on the site throughout time. The calculation is similar to episode definition but a smaller threshold is defined. Instead of comparing the value of inactive time with a precision of minutes, they are compared with an accuracy of seconds, providing a smaller threshold of 50 seconds. If the user stops interacting for longer than the set threshold that time is not considered as active time. This metric represents a more accurate depiction of user interaction time. For example, even if an episode is 30 minutes long a user might have interacted for a total of a few seconds.

Both temporal metrics – episode count and active time – are included in all captured
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

Figure 3.5: Unbounded distribution of resulting episode lengths using a threshold of 40 minutes on cs.manchester.ac.uk site.

events. These metrics are calculated for both the site and each URL. If the analysis needs to tackle the existing variance between Web pages, metrics concerning individual URLs can be employed.

Web page variability

Web interaction data is captured from all Web pages on a Web site or application. Differences between Web pages is unavoidable, although depending on the kind of Web site the degree of these differences may vary. To support a page-dependant analysis information about the Web pages where the interaction takes place is needed. Among the collected data – see Subsection 3.1.2 – the DOM state of the pages is captured. The DOM can be processed to obtain relevant information of the content displayed, such as the number of words, links or images. Characteristics specific to users’ browsers are extracted from the captured events. All events capture users’ browser details and their operating system. In the case of load and resize window events the size of the Web page is obtained. Both total size of the Web page and users’ viewport – the size of the viewable portion of the Web page in the user’s browser – are collected. Recorded sizes have been extracted and the aggregated size measurements have been computed so this information is available for every accessed URL on the site.
Analysing Web page metrics allows to calculate a metric of page complexity. Different aspects can be taken into account including the number of words and links. Aside from a particular complexity metric, analysis of this data provides a measurement of similarity between pages. Inspired by an approach that counts the number of block elements – tagged elements used to construct documents in HTML – in each section of a page [Harper et al., 2013] a coarser approach has been implemented. Following the original approach of partitioning Web pages according to a particular viewport is not a feasible approach for the amount of captured data and the variety of possible viewport sizes. The total number of block elements per page has been used instead in combination with the height of each Web page. The original approach does not consider the height of the Web pages, as the analysis is limited to cropped snapshot of the Web page. The height needs to be accounted for to consider the effect the size of the page can have on the interaction – e.g. scroll interaction needed to view the rest of the page. Count of block elements and the height of the page have been normalised to be used as input for the affinity propagation clustering technique [Frey and Dueck, 2007]. The height is included in the analysis, but its weight has been halved to prioritise the count of block elements as the selected main metric of page complexity. This approach has been applied to the accessed Web pages on the cs.manchester.ac.uk Web site. The result is a set of 15 clusters that support the analysis of Web pages of various degrees of complexity as can be seen in Figure 3.6.

Capture scope

Depending on the kind of capture tool employed for the observation, the scope of the interaction it logs will vary. In laboratory studies third-party software can be installed on the computer, making the scope wide enough to capture all interaction with the operating system. In the case of Web interaction, as long as no other software is employed, capture is limited to interaction with elements inside the Web page. This approach is assumed to safely capture mouse, keyboard, and window interaction without any interruption. However, the capture of some events can be problematic under certain circumstances. For example, if captured correctly, page unload can be considered a way to discern when the user exited the Web page or closed the tab or browser. After reviewing captured data it has been found that the capture tool does not have time to send the event information before the connection to the server is interrupted. Any other unexpected interruption of the application also interfered with the transfer of interaction data, making it virtually impossible to notify its termination event. This problem
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

Figure 3.6: URL clusters according to their height and count of block elements from all accessed URLs on the cs.manchester.ac.uk site.

can also affect events that are triggered just before the user closes the browser or moves away from the page.

Misunderstanding the scope caused some initial confusion in interpreting the results of the behavioural analysis. An especially problematic situation can arise from assuming a series of events comprising a single behaviour is part of the same scope. For example, a user might click inside the browser window, therefore enabling the capture of that event, but move away before ending the second part of the click action – releasing the click button. This situation effectively moves the second part of the action to a different scope – outside our observed environment. Other situations, such as the user moving to a different Web page before ending the previous action, need to be accounted for. To tackle these situations, it has been found necessary to either discard incomplete actions or extend the analysis to include actions taking place on different Web pages.

Data extraction

Potential problems have been identified during the extraction and process of the captured data. Some have been derived from the lack of control over the environment, making it impossible to ensure the recurrence of users and the quality of the captured
data. Other problems consist of hurdles common to the processing of large amounts of data. Ways to process and understand low-level events captured over an extended period have been explored.

**Decreasing user pool** has been found to be a problem. Unlike controlled studies, remote observations cannot provide any guarantee that a given user can be identified each time he or she visits the site. A cookie is used to identify users, but it is possible that a particular user may access the site from different machines, or that different users may access it from the same machine. Even when assuming a person can be identified with a reasonable degree of reliability, there is no control over how many times they visit, or for how long.

Due to the size of the observed population a pool of recurrent users was expected. Identifying these users would allow to investigate how (or if), for a given user, behaviour changed over time. In fact, it has been found that there are only a few users interacting with a particular Web page over a long period, and the number of recurrent users reduces exponentially as the analysed timespan increases (see Figure 3.7). This skewness can be problematic when it comes to analysis. Data samples need to be thoroughly scrutinised to make sure the distribution of users is appropriate for the analysis methods. For example, visualising the distribution of aggregated interaction time per user helps to identify different user profiles, for a possible filtering.

**Highly scalable analysis techniques** are necessary due to the high number of captured events. Processing techniques used for data collected in controlled circumstances are not suitable for direct application to longitudinal data. Database query executions that may take hours with a relatively modest volume of data can increase to months. To make many aspects of data processing feasible, techniques designed for working with big volumes of data are necessary, such as MapReduce [Dean and Ghemawat, 2008]. Careful filtering of the data according to the research question can also ease processing demands. In the particular case of observing behavioural changes in individuals over time, all non-recurrent users are ignored – around 75% of the sample and around 18000 users.

Certain limits have been imposed by the employed database system, which have only been discovered when violations occurred, triggering critical errors. The document-oriented MongoDB [Mongodb.org, 2015] database is used, which has an object size limit that is sufficient for the majority of applications, but still requires the concise
3.1. NATURALISTIC CAPTURE OF LONGITUDINAL WEB INTERACTION

Figure 3.7: User distribution over accumulated active time in all URLs on the cs.manchester.ac.uk site.

The design of produced objects to ensure that they remain within size limits. MapReduce has been employed to process information in parallel, creating single objects that contain all the processed information for each user episode. Naturally occurring outliers can also cause problems. An especially long period of usage can arise accidentally, causing a size limit violation. For example, one user opened the Web site on their phone and left it opened, sending mobile device sensor data for more than a day. The use of indexes is required to speed up processing time, but it should be noted that this takes up even more storage space.

Missing or erroneous data is an unavoidable consequence of capturing data from an uncontrolled environment. Understanding the cause of these problems has helped to shape the tool, and has allowed for the addition of measures to prevent the occurrence of errors. Many of the following problems have only occurred after the tool had been deployed in the wild, and could be solved thanks to the possibility of updating the capture tool on the fly.

Sporadic episodes with an unusually high number of events can cause problems during the process of the captured data. These situations are scarce and have been discarded from the analysis after reviewing the content of the corresponding episodes.
Thorough analyses of the problematic episodes has revealed that the frequency of the events triggered by the mobile sensors could occasionally cause problems under unexpected scenarios. Although the capture solution has been tested before deployment, these unexpected situations caused the mobile sensors to trigger an unusually high number of events. For example, a user forgetting to close a browser window triggered a continuous stream of events indicating changes in the gyroscope sensor. Another exceptional situation arose when a particular user changed its system’s time during the interaction. The capture solution relies on the time provided by users’ systems to calculate the time difference from loading the page till the event to be captured takes place. If the user changes the time during the interaction the timestamp for following events will be incorrect. The designed system is robust against this error. Instead of identifying each page load as an episode it aggregates events occurring within set time periods. In this case, events would be processed as being part of different episodes making the analysis of this data possible. The designed temporal metrics would actually consider the resulting episodes as happening one immediately after the other, preventing any misinterpretation of the data.

Other instances of events containing erroneous data have been identified and discarded. The majority of cases are due to an unusually low number of events in random accesses to very concrete Web pages. This behaviour could be caused by a crawling bot searching for particular filetypes or accidental loads from users. Even though the compatibility of the events with major browsers had been tested, a small percentage of events have been found to be missing critical attributes for analysis. For example, a small percentage of events are missing the URL of the Web page where the interaction took place. This attribute is completed using events contained within the same episode, which are known to have taken place in the same URL. Although considered an isolated problem, analysis algorithms have been designed to be robust against such situations, and data is discarded if inaccuracies could not be fixed in a reliable way.

Finally, the iterative design of the capture solution needs to be taken into account during the extraction of the data. The possibility of extending the data is mentioned as a feature supporting iterative improvements of the captured data. If old data is taken into account for the analysis it needs to be ensured that all data contains comparable fields.
3.1.4 Summary of interaction capture solution

A longitudinal fine-grained Web interaction capture solution has been designed and implemented to fill the gaps found in existing approaches. A formative study has been carried out to choose a suitable design, helping to identify the requirements of the capture solution. These requirements have helped shape a longitudinal observation tool that provides naturalistic data concerning users’ low-level interaction.

The main objective of this thesis is supported through the capture of the required data to study changes in user interaction over time. Fine granularity enables the detection of subtle changes, and meeting the identified requirements helps to design a scalable solution. The deployed system makes the processing of a high number of events feasible, employing techniques designed to handle big volumes of data.

Lack of control over the environment and prolonged observations have posed hurdles during the processing and extraction of the data. The process of data cleansing took into account the necessary requirements to fulfil the objective of this thesis. Inadequate data – such as non-recurrent visitors – for a longitudinal analysis has been discarded to ease the subsequent analysis, and missing data has been completed in a reliable manner. Ways to temporally group, process, and extract meaningful information from isolated low-level events have been explored. In the following section, an analysis methodology for the interaction data obtained using the implemented observation solution is presented.

3.2 Longitudinal analysis

Once the requirements of the captured data have been specified, a capture solution that complies with the identified requirements has been designed and implemented. To obtain longitudinal interaction data in a naturalistic way the capture solution has been deployed in real world Web applications and sites. Data has been captured without setting a predefined end to the observation so results of the analysis of various periods of time with varying lengths can be compared. Due to the uncontrolled nature of the capture unexpected issues arose. Several of these, mainly involving the processing of data and issues related with the capture, have been dealt with in the previous section. Problems with the analysis of the data are mainly caused by the lack of knowledge of users’ motivation, the disparity between users, and the extent of the study. A way to analyse user interaction data without introducing any bias is necessary. It has been
hypothesised that user behaviour changes as users interact over time. Thus, a methodology to find evolving factors of interaction is necessary.

Several requirements have been identified to guide the design of an appropriate analysis methodology. Any bias needs to be prevented, avoiding the use of task-driven approaches. Analysis of extended periods of time needs to be supported, which can be particularly challenging when the number of users is high. Micro behaviours as a way to obtain comparable units of interaction are introduced. Features are extracted from selected micro behaviours and a methodology to discern temporal correlations among them is presented.

3.2.1 Analysis input

The designed capture solution has been deployed in real world Web applications and sites over a 16 month period. The main source of interaction data has been the University of Manchester School of Computer Science Web site\textsuperscript{[5]} This site contains information about the School of Computer Science at the University of Manchester, covering news, events, teaching and research, and is used by a range of people (including current and prospective students and staff). As the site is frequently updated, the number of pages it contains varies; during this study data has been captured from a total of 1411 different URLs.

Interaction data has also been captured from users of the KUPKB\textsuperscript{[6]}. The KUPKB is a highly interactive Web application. It consists of a single Web page that dynamically updates its content to emulate different tabs and functionalities – see Figure 3.8. It provides an alternative environment, in which the usefulness of the designed analysis with more interactive scenarios could be tested. Unfortunately, the number of users is small, especially when compared to the high volume of users visiting the School of Computer Science Web site.

3.2.2 Identified requirements

The main objective of the analysis is providing insight into user interaction without introducing any bias. The work done in this thesis focuses on long factors of user interaction to cover the identified gap in current longitudinal studies. The identified

\textsuperscript{[5]} www.cs.manchester.ac.uk/
\textsuperscript{[6]} www.kupkb.org/
requirements for the analysis focus on the avoidance of bias, the longitudinal requirement of the analysis, and the existing variance between users.

Task driven approaches introduce bias into the analysis by prejudging what the user interaction is like. Although particularly useful for the analysis of use cases and to guide the analysis of the interaction, task driven approaches only consider a subset of the interaction with the Web site or Web application. A data driven approach prevents these prejudgements, enabling the emergence of evolving interaction aspects from users’ free interaction. Visualisation of data has been explored as a way of analysing data without introducing bias. Unfortunately visualisations of interaction data captured over extended periods of time are not scalable, making human interpretation of the data challenging. Analysis techniques able to handle extended periods of time are necessary to allow the identification of temporally correlated aspects of user interaction. The designed analysis technique needs to be robust against noise introduced by the uncontrolled aspect of the capture and its length. The unpredictability of user behaviour, as well as the variance naturally found between users, need to be taken into account.

Data driven

Task driven approaches simplify the analysis of the data guiding the discovery of predefined metrics and supporting the execution of particular tasks. Context for the interaction is provided and comparisons between different interaction periods are possible.
Task completion time and percentage of successful executions can be used as metrics to rank different executions. Detailed insight into the execution of selected tasks help improve their design, easing their execution and ensuring users achieve their goal. However, the use of task driven approaches cannot be directly applied to scenarios where users interact freely. Users’ motivation is unknown and the lack of context makes comparisons between interaction episodes challenging.

As interaction in the investigated situations has been uncontrolled (people are simply using the Web) information about the user’s motivation is unavailable. It is therefore entirely possible that users are performing different tasks every time they accessed a page. Nevertheless, it was believed there would be some instances of users performing the same tasks repeatedly, which would provide a basis for understanding how interaction evolves. This would also allow to understand whether the evolution of behaviour differed from user to user.

Determining task and context is extremely difficult, and runs the risk of circular reasoning, but it was hypothesised that a good starting point for inferring high-level commonalities between tasks or goals would be to look for patterns in the URL sequences performed in each visit. Visiting URLs in the same sequence does not necessarily imply the same task, but it can be argued that the user is at least following a similar procedure.

People may use [cs.manchester.ac.uk](http://cs.manchester.ac.uk) for a variety of purposes, but it can be assumed that people would frequently be seeking information, and that many people will be seeking the same, if not similar types of information. Whilst two users visiting URLs in the same sequence might not occur when analysing small numbers of users and visits, at least some common URL sequences were expected due to the high volume of traffic on the site. 30023 URL sequences have been extracted from all user visits, of which 10838 consisted of more than one URL, and 8117 are unique. A total of 7369 multi-URL sequences are executed only once. Table [3.4](#) lists the ten most frequent URL sequences with a minimum length of two [Gabadinho et al., 2011](#). It shows how many unique users have been found for each sequence, as well as how many perform it more than once – Recurrent Users – and the maximum number of times that sequence has been performed by a single user – Maximum Frequency.

The most frequent sequence occurred only 60 times. As the goal of the study is to understand the evolution of behaviour it is relevant to obtain how many times the same sequence is performed by a given user. It has been found that only nine users had performed the same sequence more than once.
3.2. LONGITUDINAL ANALYSIS

<table>
<thead>
<tr>
<th>Seq. code</th>
<th>Length</th>
<th>Frequency</th>
<th># Unique Users</th>
<th>Recurrent Users</th>
<th>Max. Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq. 1</td>
<td>2</td>
<td>60</td>
<td>41</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Seq. 2</td>
<td>2</td>
<td>51</td>
<td>45</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Seq. 3</td>
<td>2</td>
<td>50</td>
<td>36</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Seq. 4</td>
<td>2</td>
<td>49</td>
<td>40</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>Seq. 5</td>
<td>2</td>
<td>46</td>
<td>45</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Seq. 6</td>
<td>2</td>
<td>39</td>
<td>38</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Seq. 7</td>
<td>2</td>
<td>39</td>
<td>32</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Seq. 8</td>
<td>2</td>
<td>38</td>
<td>35</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Seq. 9</td>
<td>2</td>
<td>38</td>
<td>37</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Seq. 10</td>
<td>2</td>
<td>38</td>
<td>32</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3.4: Top 10 of URL sequences found to be longer than 1 extracted from the use of cs.manchester.ac.uk out of 30023 sequences. There is a total of 10838 sequences longer than 1, from which 8117 sequences occurred only once.

Although results appeared to indicate that – at a macro level – there is spectacular diversity in the usage of this particular site, reality might be different. Slight variations in behaviour could easily mask broad similarities in this naive approach to URL sequence analysis, and using more complex pattern recognition mechanisms [Fern et al., 2010] could provide different results. One promising future approach is n-gram analysis which would enable the discovery of sub-patterns within overall sequences [Vigo et al., 2015]. Nevertheless, it is believed it is important to remain cautious about the possibility of directly comparing sequences containing variation.

In the case of the cs.manchester.ac.uk Web site, due to the wide variety of URL sequences uncovered, and the existing uncertainty about the extent to which they are equivalent, future comparisons of behaviour are limited to interaction data logged within the same URL. It has also been shown how a task-driven approach is not suitable for analysis of freely captured interaction data. Variance between the followed paths in big Web sites makes assumptions of their interaction dangerous and poses a high risk of bias.

**Within user comparison to prevent user variability**

Laboratory studies enable the observation of a limited amount of users. Demographic information is commonly employed to group users into comprehensible categories. These categories allow to compare data captured from different users. Depending on the purpose of the study these differences can consist on different degrees of ability or academic background. Differences or commonalities in the way users interact can then be extracted from the results.
In an unobtrusive approach the collection of demographic information is not possible. Collection of interaction data from users can start at any point during their overall experience with the site, so any interaction before that point remains unknown. Furthermore, it has been argued that surveys and overall recollection of events by users cannot provide an accurate representation of their interaction or overall experience [Jansen et al., 2009]. Self-assessments of expertise levels have also been found to be misleading as they may not be correlated with external assessments by experts [Grossman and Fitzmaurice, 2015]. Due to the main goal of this thesis the lack of demographic information does not pose a problem. The objective is providing insight into how user interaction evolves as they interact over time. Rather than collecting information to define precisely each user’s initial state, differences between that initial state and subsequent ones are explored. To do that, analysis needs to focus on differences within users.

### 3.2.3 High-level construct extraction

An important initial step when dealing with large amounts of freely instigated low-level interaction data is to group it appropriately from a temporal perspective. Data has been segmented into ‘episodes’ which equate to a single, continuous session of Web use – the creation of episodes is detailed in 3.1.3.

For each episode, low-level events (individual mouse movements, keystrokes, etc.) have been grouped into sequences that represent what have been coined micro behaviours. These behaviours represent a small sample of user interaction, such as clicking a link, typing in a text field, or scrolling down the page. Interaction inherent to Web browsing can be modelled directly, such as mouse clicks or duration of episodes. **Manual observation of events** provides insight into interaction sequences, and possibilities for additional micro behaviours. Common information among events has been extracted and manually analysed using OpenRefine. Open Refine is an open source tool that supports the task of exploring and sanitising large data sets. Manual observation of the events enabled the discovery of low-level interaction patterns that supported the extraction of micro behaviours for the subsequent analysis. For example, this way it has been noticed that the time from hovering over an element to clicking on it could be extracted for every mouse click.

The discovery of micro behaviours can also be supported by previous research.

---

7 https://github.com/OpenRefine
3.2. **LONGITUDINAL ANALYSIS**

Literature related to the long-term factor to explore contributes to the selection of indicators. Previous work has shown that it is possible to identify certain behaviours that are common across individuals through behavioural observation and interaction data analysis, and also that these behaviours are important indicators of both what the user is doing, and how successfully they are completing their task [Lunn et al., 2011, Vigo and Harper, 2013].

As opposed to the aggregation of interaction data for each episode – e.g. total amount of scroll – extraction of micro behaviours enables the grouping of interaction into smaller units. Analysing finer grained groups of interaction data provides a more detailed insight into users’ behaviour and supports comparisons that would otherwise be possible. For example, if a user scrolls down fast for a short period it would go unnoticed if data were aggregated for each episode. However, if the scroll actions are combined into micro behaviours this interaction would stand out as a specific “fast scroll” micro behaviour. Further details about the extraction of these micro behaviours can be found in Section 3.3.

### 3.2.4 Analysis techniques

Features from micro behaviours provide access to meaningful aspects of user interaction. Comparisons of these aspects over time enable the analysis of how user interaction changes over time. Unfortunately, aforementioned requirements (Section 3.2.2) make the analysis of these micro behaviours challenging. Prejudgements need to be avoided through a data-driven analysis, impeding the use of predefined tasks. Differences between users are unavoidable, and categorisation of users via demographic information is not possible.

Various visualisations of the data have been explored. Descriptive statistics provides the means to obtain insight into the data without introducing any bias. Although useful to comprehend the nature of the data, the high number of users made the discovery of tendencies within single users challenging through traditional visualisations. General linear models provided a quantifiable method of discovering temporal correlations among features extracted from selected micro behaviours. Mixed linear models have been employed to tackle the variability between users. Mixed linear models provide the means to account for the mentioned variability, as well as uneven episode distributions between users. Although useful to discard invalid user inputs, and scarce datapoint sources, mixed linear models risk the occurrence of over significance of particularly user inputs. To provide a robust discovery of temporally correlated features, a random
Timeline visualisations

Timeline visualisations are particularly useful for understanding behavioural evolution, and it was thought they would be helpful to explore the temporal change in micro behaviours such as click duration or scroll distance. When working with a small number of users, changes to individuals’ behaviour could be seen clearly. With a greater number of users, however, spotting these changes became very difficult.

Figure 3.9 shows a single line clearly demonstrating one user’s mousewheel activity as a function of time. These visualisations are able to demonstrate relationships between features and the time spent by users on the page, although additional information that might affect user interaction, such as users’ inactivity periods or possible changes on the page, have been found to be necessary to understand fluctuations over time. Nevertheless, observing a single user provides a narrow and potentially unrepresentative view of interaction overall.

When the task is unknown, however, comparing timelines in this way becomes less meaningful. The graphs also become harder to read as the number of users grows. Figure 3.10 shows the 100 most active users – users with the longest aggregated interaction time – of the cs.manchester.ac.uk Web page and the duration of bursts of scroll action found over the whole period of interaction with that particular page.

The median duration and the variance are shown every 100 records with a red line, indicating the density of records over time. Although the figure seems to show that duration of scroll reduces over time there are in fact only a few users who interacted with the page for a long period. It cannot be assumed that the duration of scroll bursts got shorter over time, as the initial long bursts might have been caused by completely different users to those whose interaction is visible after a longer period.

Timeline visualisations provide detailed insight into user interaction over short
3.2. LONGITUDINAL ANALYSIS

Figure 3.10: Plot of the scroll duration over time of 100 users on the cs.manchester.ac.uk homepage

periods of time. Differences between comparable short interaction periods are highlighted and visual exploration of individual user’s interaction is possible. Task-oriented approaches can employ these visualisations to compare various executions of the same tasks, obtaining detailed insight into how particular tasks are performed by different users. Unfortunately analysis of interaction over extended periods of time through these visualisations is challenging. Traditional timeline visualisations are not scalable and their readability decreases as the number of users increases. Alternative scalable methods need to be employed to obtain insight into larger populations.

Descriptive statistics

Initially, raw captured data has been explored to get familiar with the data and its possibilities. Manual analysis of individual users’ interaction events inspired some of the micro behaviours. As a way to extend the analysis of interaction to prolonged periods of time statistics of the features extracted from micro behaviours have been
calculated. Averages and medians from user episodes have been plotted in timelines similar to the one shown in Figure 3.9. Although they depict a precise picture of how a particular feature changes over time temporal tendencies are not clear. These visualisations do not provide the means to efficiently compare different points in time as data points vary dramatically over time.

For some features a clear tendency might emerge. In the example shown it could be argued that “Total scroll” seems to decrease as the user spends more time on the page. Unfortunately, the strength of such tendency cannot be quantified, and validating it against the rest of users is costly.

When the number of users is high, instead of looking for within-user evolution over time, differences between different interaction stages can be compared. The values of the feature to explore are grouped into bins according to the user’s aggregated active time on the page. For each bin a smoothed density plot has been created describing the nature of the feature at that stage. Using density plots allows to see if the specific nature of the feature is changing at different interaction stages. For example, a positively skewed plot indicates the majority of values are small while a more evenly distributed plot would mean the values are higher. Due to the high skewness of the data a negatively distributed plot is extremely uncommon.

In Figure 3.11 density plots show the distribution of the length of users’ episodes over time for each accumulated active time bin. The sample of analysed users needs to be appropriately tuned so there is an even distribution of an equivalent number of users for each bin. In this case the figure seems to indicate that episode lengths increase as users spend more time in the page. Unfortunately, change within users is not considered in this kind of graphs. No assumptions can be made about within user evolution over time, as the observed evolution might be indicating that users with longer episodes interact for longer periods of time.

Descriptive statistics provide an overview of the interaction providing insight into the entire population of users. Inspection of particular aspects of interaction is possible without having to resort to more complex mechanisms but introspection into individual users is not directly possible. Rather than analyses of groups of users a technique that accounts for each individual user’s input is necessary.

**General Linear Models**

As a way to discover tendencies within users, **general linear models** have been considered. General linear models have been employed to represent linear relations between
3.2. **LONGITUDINAL ANALYSIS**

Figure 3.11: Density plots of users’ episode length corresponding to different active time bins of 30 seconds.

variables. They quantify changes in variables over time and allow for the inclusion of additional factors that may affect behaviour. For example, considering the time between episodes allows the model to account for possible memory effects. The advantage of using a model is that it allows for predictions: if data from one user is incomplete the model can be used to complete it. When many users are added, however, general linear models can become difficult to interpret as all data is considered as independent. Changes over time found by applying linear models to data coming from different users are not valid, as the model is assuming all the data is coming from a single user.

Ideally the model needs to be applied individually to each user. Figure 3.12 shows the correlations of users’ episode length with the accumulated active time in a particular URL. Unfortunately, there are discrepancies between the number of records per user. It is difficult to judge overall evolution using only a sample of users, as their data might be noisy or contain too few values, therefore not being representative of the entire population. If particular users provide noisy data or very few values, it is worth considering reducing their weight in the resulting judgement.
CHAPTER 3. ANALYSIS FRAMEWORK

Figure 3.12: General linear models applied to each individual user showing evolution of users’ episode length over active time

Linear Mixed Models

A more sophisticated take on linear models is **linear mixed models**, which allow the inclusion of random effects. If a linear mixed model is based on certain independent variables – in the presented example users’ accumulated active time and episode duration – the random effects represent the residual that cannot be accounted for by these variables. When random effects are added to the model they explicitly model the inter-subject variability. In this case, adding users as random effects tells the model to consider their values as different subjects. The result is a more sensible interpretation of the data. Furthermore, the model is robust against unbalanced data [Bates, 2010]. It evaluates how noisy the data from each user is, and calculates a user specific intercept and slope taking into account data from other users.

Still, using a predictive model blindly can lead to incorrect results. There are certain assumptions – such as absence of collinearity, and homoskedasticity – that need to be met to use the model appropriately [Winter, 2013]. Logarithmic transformations can help handle skewed data, and the resulting residuals from the model need to follow an appropriate distribution to ensure the model has been used correctly. One subtle problem with linear mixed models is that it assigns different weights depending on the
variability of data from a particular participant. If a person’s data is assigned too much weight compared to the rest its significance becomes over-inflated. These participants can be sought out with statistical methods. It is also important that the data included in the model follows an even distribution. Both filtering out users with scarce data, and limiting information from extremely active users can help to ensure the analysed data is representative. Furthermore, data must also be carefully prepared. Linear mixed models are robust against data missing at random, but if the missing data is not properly marked as not available the model will take the value into account. Although this sounds obvious it is easy to overlook such an issue when extracting large numbers of complex behaviours.

Figure 3.13: Plot of the linear mixed model for episode length time

Here the linear mixed model was applied to different user sample sizes chosen at random from the population. The distribution of the

Figure 3.13 shows a plot of the resulting linear mixed model of episode length time with users accumulated active time [Kuznetsova et al., 2014]. In this plot, an individual line is drawn for each user, red if it is increasing and green if not, with a black line indicating the resulting model for the entire population. Each line does not indicate each user’s individual change over time, but rather the resulting prediction for that user taking into account the rest of the data.

It has been mentioned that due to the linear mixed model’s noise handling behaviour, the application of these models can be sensitive to certain users exhibiting particularly strong correlations. To tackle this, the model has been applied to different user sample sizes chosen at random from the population. The distribution of the
Figure 3.14: Information reported from repeated execution of a linear mixed model for different user samples, showing the evolution of the duration of users’ episode over time.

model’s fixed effect values has then been observed – i.e. the value of the slope, indicating that particular feature’s tendency to change over time. This approach results in a more conservative and reliable prediction of temporal correlations. To avoid disparity in the number of data points per user thresholds in the sampling of users have been enforced. More precisely, a minimum amount of interaction has been set for the user to be included in the analysis, as well as removing long-term interaction beyond which users became scarce to prevent a small number of highly-recurrent users overpowering the model.

Figure 3.14 shows an example of the information reported by this technique. For this example data captured over a year from the cs.manchester.ac.uk page has been used. Users who have interacted for at least 4 minutes have been selected while any interaction beyond 20 minutes has been discarded. Instead of relying on the overall result of the model when all data is taken into account (298 users) the model has been applied to random samples of varying sizes. For each sample size the model has been applied 50 times, and the distribution of the results have been reported via whisker diagrams. Fixed effects are the most relevant aspect of the result, as they represent the positive or negative temporal correlation of the analysed feature. A confidence interval of the fixed effects distribution of 95% has been calculated for each sample size. Intercepts
3.2. **LONGITUDINAL ANALYSIS**

of the linear mixed model and p-values are also reported. P-value is inversely correlated with the distance of the fixed effect value from the origin – the farther away from ‘0’ the lower the p-value, and a horizontal red line indicates a value of 0.05. P-values have become controversial in research [Hubbard and Lindsay, 2008]. The objective of this analysis is discovering features exhibiting a reliable temporal correlation. To do that, rather than reporting p-values, consistency in the sign of the resulting fixed effects is explored. In the example shown, it can be seen that fixed effects follow a consistently positive value indicating a strong positive correlation of users’ episode duration and active time. Confidence intervals provide a less conservative interpretation of this phenomena.

### 3.2.5 Summary of longitudinal analysis

The interaction capture solution described in Section 3.1 has been deployed on the Web site of the School of Computer Science of the University [http://www.cs.manchester.ac.uk/](http://www.cs.manchester.ac.uk/) and in the KUPKB Web application [http://www.kupkb.org/](http://www.kupkb.org/). Interaction data has been captured from visitors over an extended period, and ways to analyse it have been explored. To support the overall goal of this thesis certain requirements needed to be met, such as the use of a data driven approach to prevent introducing bias into the data. Unfortunately, the use of traditional time-line visualisations is challenging due to the high number of users and the length of the study. The lack of demographic information prevents the grouping of users. Instead of searching for differences between users at different stages of interaction, changes within single users’ interaction over time have been analysed.

*Micro behaviours* have been presented as indicators of user behaviour to provide comparable units of low-level interaction. Features can be extracted from identified micro behaviours to support a longitudinal analysis and discover indicators of how user interaction changes over time. Section 3.3 provides details on how these micro behaviours have been designed, and the available features.

Insight into longitudinal aspects of interaction through a data-driven analysis is challenging. Descriptive statistics has been employed to visualise the data, and understand the nature of the extracted features. A more detailed insight can be obtained through general linear models, which provide a way to obtain quantifiable tendencies from individual users. Unfortunately, general linear models can only be applied to individual users, and ignore important problematic aspects of the captured data, such as
scarcity of available data points.

Proposed analysis makes use of *linear mixed models*. These models take into account input from individual users and provide a weighed prediction of the tendency. Unfortunately, due to the way these models assign different weights to inputs depending on the amount of data and quality their prediction can be biased towards particular individuals. To tackle this problem, and to obtain robust predictions of temporally correlated features, a random sampling methodology of varying sizes has been presented. The resulting analysis technique enables the robust discovery of temporal correlations among the features extracted from selected micro behaviours.

This analysis technique has been applied to the interaction data captured from real world Web applications. In Chapter 4 this methodology is applied to obtain insight into the high-level concept of familiarity. It serves as a validation of the presented methodology, as well as an example on how to extract micro behaviours relating to a particular higher level construct and their analysis.

### 3.3 Micro Behaviours

The objective of this thesis is providing insight into how Web interaction changes over time. Rather than coarse observations, such as page views, low-level interaction – from mouse movement to key presses – is analysed to provide insight into subtle changes in interaction. Analysis of low-level interaction data over extended periods of time is challenging. On the one hand, aggregated measures per episode can obscure possibly relevant interaction. For example, if the total amount of mouse movement is reported per episode, quick sudden movements of the mouse would not be detected. On the other hand, analysing changes over time is not possible if no comparable unit is provided.

Micro behaviours are presented as a way to aggregate captured events into comparable interaction units. They provide precise descriptions of low-level interaction from users providing the means to discover subtle changes in interaction over time. Design of these micro behaviours is based on manual observation of the captured interaction data, as well as existing literature. The list of prospective micro behaviours can be easily adapted to the needs of the study. In the case of the data captured for this study, the majority of the interaction is provided by mouse interaction. Therefore micro behaviours describing different mouse interaction aspects have been designed.

The objective of this thesis is providing insight into how user interaction changes
3.3. MICRO BEHAVIOURS

over time. Therefore designed micro behaviours focus on describing sets of interaction that might change as users come back to the Web site or application. Micro behaviours represent small units of interaction and their definition is flexible. Different events and inputs can be combined to form complex micro behaviours. In the case of the data obtained through the capture solution described in Section 3.1, the input is a series of Web events, representing user interaction and browser state events – see Table 3.2.

3.3.1 Definition

I define micro behaviours as an aggregation of low-level interaction events with a temporal aspect. Their purpose is providing a quantifiable and scalable view of an individual user’s Web interaction. The temporal aspect has been considered necessary for two reasons. First, although limited in time micro behaviours always provide a starting and ending timestamp which ensures the existence of a quantifiable aspect. Second, requiring a temporal aspect eliminates atomic behaviours that do not correspond to a combination of low-level events – e.g., the isolated event of a window closing.

The kind of event to be used as input may depend on the context of the study and the capabilities of the observation tool. In that sense, the definition is left open to include multiple combinations of input sequences. The purpose is providing small units of comparable data, so prospective micro behaviours should remain simple enough to enhance their occurrence – obtaining a higher count available data points for analysis. This way, quantifiable metrics depicting the low-level interaction of individual users can be obtained.

In the context of this study, the high occurrence of micro behaviours is considered as an additional requirement. To perform a longitudinal analysis, a minimum amount of data points is required – e.g., three data points is the minimum to study non-linear correlations. The occurrence of some of the micro behaviours defined in this chapter has been found to be accidental – i.e., the micro behaviour did not take place more than once per user and was only exhibited by a limited amount of users. Although those micro behaviours cannot support the analysis carried out in this study, they are presented as prospective micro behaviours for alternative studies.

3.3.2 Event input

The design of the micro behaviours depends greatly on the source of the interaction data. In the case of this study, Web interaction data from mouse, browser window, and
keyboard is available. Due to the nature of the main Web site to be analysed – cs.manchester.ac.uk – the majority of the interaction data has been produced by users interacting with the mouse. Combination of this data with additional data inputs, such as eye tracking, or image recognition for user movement, would allow for the design of more complex micro behaviours. For example, the action of fixating on a particular point on the screen for extended periods of time can be coded as a micro behaviour. The change in the duration of this fixation can then be analysed to determine how this micro behaviour changes over time. The design of the resulting micro behaviours has also been constrained by the available events – see Table 3.2. The majority of the captured interaction data have been generated by mouse and window events. The mouse has been found to be the users’ preferred interaction device so designed micro behaviours have focused on combinations of mouse and browser’s window events.

Behaviours concerning keyboard interaction have also been explored. Common features, inspired by existing work on typing behaviour [Dowland and Furnell, 2004], extract temporal metrics from digrams and trigrams. The analysis is complex, as no particular time-out has been found to extract typing behaviour from free interaction, making the definition of individual typing periods challenging. Data collected from keyboard interaction has also been found to be less frequent for the recurrent users of the cs.manchester.ac.uk site. In that case the availability of text input elements is limited, and their use by users accidental. Interaction data captured over 16 months shows that users have clicked with their mouse – 370400 mouse down events – more than twice the times they have typed a single letter – 141929 key press events. Events related with mouse movement and mouse scroll provide a finer granularity, with thousands of counts – 11153216 for mouse over events and 2471455 for mouse wheel events. Nevertheless, the frequency of designed micro behaviours depends on the type of Web site. The use of micro behaviours based on keyboard events can be suitable for studies in different environments or for studies with different purposes.

Designed micro behaviours have been grouped in two categories, mouse micro behaviours and scroll micro behaviours. The full list of designed micro behaviours is reported in Table 3.5. Some of the behaviours have been designed based on manual observation of the events. This manual observation consisted in a qualitative approach, in which selected users’ interaction has been observed in detail. Interaction captured over a period of two months from the three most active users – i.e. longest aggregated active time – have been selected. The tool OpenRefine\(^{10}\) has been employed to explore

\(^{10}\)https://github.com/OpenRefine
3.3. MICRO BEHAVIOURS

the resulting large datasets of interaction from these users.

3.3.3 Mouse micro behaviours

Manual observation of the data provided insight into low-level aspects of users’ interaction with the page. Coding interaction situations naturally inherent to the Web helped to obtain highly frequent micro behaviours. Several micro behaviours have been coded describing the different situations taking place when a user click on a page element. “Time To Click” as the time preceding the click, “Click Speed” as the length of the click action, and “Idle After Click” as the time preceding the next interaction with the mouse have been coded as micro behaviours. The change in the duration of these micro behaviours reflects how click interaction changes over time. Additionally, the time from loading the page to interacting with an element of the Web page has been designed as the “Click After Load” micro behaviour.

Manual observation has also shown periods of time in which a user would not interact with the page, inspiring the “Inactive Mouse Time” micro behaviour. This micro behaviour reports a measurement of all periods of inactive mouse time in an episode. Instead of reporting a unique measurement, both median and mean of the lengths of all periods has been reported. Mouse inactivity might be caused for a variety of reasons, such as mental operations – e.g. thinking where to move the mouse – or lack of attention – e.g. multitasking, indicating the user is paying attention to an external target.

Particular interaction situations have been found to occur sporadically. In several occasions a user has been found to hover over an element of the Web site several times before clicking on it. It has been considered to be an accidental interaction possibly indicating hesitance, and has been coded as the “Lack Of Mouse Precision” micro behaviour. If the occurrence of this accidental interaction is high, analysis of how the time taken to click and the number of times the user hovered over the element could provide insight into how user interaction with Web page elements changed over time. Past research has categorised various interaction errors as indicators of frustration. These errors have not been found to be a problem, but rather an entry barrier for novice users [Mendoza and Novick, 2005]. Therefore, analysing the occurrence of these particular indications can provide insight into how users’ interaction changes as they interact over time.

Other possible accidental interactions have been identified from the manual observation of the users’ interaction. Users have been found to occasionally click on the
same Web page element several times. The “Repeated Click” micro behaviour has been designed so repeated clicks on the same page element happening within a set period of time have been extracted as a prospective indicator of an interaction problem. This micro behaviour has not been found to be frequent enough to support the proposed longitudinal analysis. Various time thresholds have been tested to increase its frequency, and less strict variations have been designed. “Fail To Click Different Node” and “Fail To Click Ignore Node” have been designed varying the requirements concerning the target of the click. Another micro behaviour – “Lack Of Mouse Precision” – has been designed after observing that some users missed the target before a click action. None of the micro behaviours based on accidental interactions have been found to be frequent enough to support the proposed analysis methodology.

### 3.3.4 Scroll micro behaviours

Scroll interaction has been found to be frequent among the captured data. Past research has explored particular sequences of scroll interaction as a proxy for users’ emotion towards the Web site. For example, scrolling down quickly to the bottom of a page and then up again has been found to indicate that someone is struggling to find the information they are looking for [Vigo and Harper, 2013]. Inspired by that research, micro behaviours have been designed reflecting different types of scroll interaction. “Controlled Scroll” comprises consecutive mousewheel actions happening within a set period. “Fast Single Direction Scroll” is a single direction scroll action to the bottom of the page. “Fast Mouse Scroll Cycle” is a scroll action to the bottom, and up again and is a refined implementations of the former. Different limits have been compared to determine when a scroll behaviour would reach the bottom, and when the scroll behaviour came back to its original starting point (or similar). For the data captured for this analysis these specific scroll micro behaviours have been found not to be frequent enough.

The features obtained from each micro behaviour vary depending on the analysed interaction. In the case of mouse behaviours, features reflect distance between clicks or temporal metrics. In the case of scroll micro behaviours, a series of common metrics related with scroll interaction are provided. \textit{Delta} is the metric to measure the scroll distance. Negative and positive delta indicate the amount of downwards and upwards scroll respectively. The travelled scroll distance has been considered to be Web page specific, so in the application of the model, the speed and duration of the scroll micro behaviour are analysed. The full list of the provided features for all scroll micro
behaviours is the following:

- Positive delta,
- Negative delta,
- Number of mousewheel events,
- Total delta: subtraction between positive and negative delta,
- Total delta absolute value: addition of the absolute values of positive and negative delta,
- Speed: total delta divided by the duration of the micro behaviour,
- Speed absolute value: total delta absolute value divided by the duration of the micro behaviour,
- Duration.

### 3.3.5 Thresholds

In the particular cases of the Controlled Scroll and the Mouse Inactive micro behaviours, the use of specific temporal thresholds is reported. Based on manual observation of events conforming these behaviours, arbitrary thresholds of 10 seconds for Controlled Scroll and 0.5 seconds for Mouse Inactive Time have been set. To test if these values are introducing bias in the analysis, additional threshold values have been obtained analysing the distribution of all possible values for both behaviours: the periods between all scroll events in the case of Controlled Scroll, and the periods between all mouse movement events in the case of Mouse Inactive Time. The analysis of these distributions has been performed using the same technique employed to determine a valid timeout to split interaction data into episodes – see Section 3.1.3. The resulting threshold values are 6 seconds for Controlled Scroll and 3.7 seconds for Mouse Inactive Time. Examples of the application of the analysis technique presented in 3.2.4 to both micro behaviours are shown in Figures 3.15 and 3.16. For the purpose of this analysis, the ideal distribution for the correlations would be either consistently positive, or consistently negative. That ideal scenario would indicate that the correlation is consistent for all analysed user sample sizes. Figures 3.15 and 3.16 indicate that the consistency of the correlation distributions is higher when 10 seconds and 0.5 seconds thresholds are employed for the Controlled Scroll and Mouse Inactive Time micro behaviours respectively. These thresholds are less conservative, leading to a higher count of occurrences of the extracted micro behaviours. In the case of Controlled Scroll the difference is negligible – 493 vs 494 occurrences – but in the case of Mouse Inactive
Time the difference in the number of occurrences is noticeable – 765 vs 899. Due to the more consistent correlations, and the higher count of occurrences, 10 seconds and 0.5 seconds have been selected for the Controlled Scroll and Mouse Inactive Time micro behaviours respectively.

Table 3.5 lists the designed micro behaviours with the employed thresholds when applicable. Several of the listed micro behaviours have not been found to be frequent enough, and have been discarded for the subsequent analysis – marked with an asterisk (*). They are reported as examples of prospective micro behaviours that could be useful under different circumstances in other studies.
Figure 3.15: Example of analysis of Controlled Scroll Duration and speed using active time as the temporal metric. The use of two different temporal thresholds is compared for each feature.
Figure 3.16: Example of analysis of Mouse Inactive Time Median and Mean using active time as the temporal metric. The use of two different temporal thresholds is compared for each feature.
<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Description</th>
<th>Threshold</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse</td>
<td>Click Speed</td>
<td>Time between clicks</td>
<td>No thresholds</td>
<td>Time between clicks</td>
</tr>
<tr>
<td></td>
<td>Inactive Time</td>
<td>Inactivity of the mouse</td>
<td>Records inactivity higher than 0.5 seconds and smaller than 2 minutes.</td>
<td>Periods between mouse events</td>
</tr>
<tr>
<td></td>
<td>Time To Click</td>
<td>Time from hovering over an element to actually clicking it</td>
<td>No threshold</td>
<td>Time between hover and click events</td>
</tr>
<tr>
<td></td>
<td>Idle After Click</td>
<td>Time between clicking and the mouse moving afterwards</td>
<td>No thresholds</td>
<td>Idle time after click</td>
</tr>
<tr>
<td></td>
<td>Lack Of Mouse Precision*</td>
<td>Repeated hovering preceding a click over an element</td>
<td>Various temporal thresholds tested, providing too sporadic results</td>
<td>Count of hover events, list of timestamps</td>
</tr>
<tr>
<td></td>
<td>Repeated Clicks*</td>
<td>Repeated clicks over the same element</td>
<td>Various temporal thresholds tested, providing too sporadic results</td>
<td>Count of clicks, list of timestamps</td>
</tr>
<tr>
<td></td>
<td>Fail To Click Different Node*</td>
<td>Repeated clicks at any page element with at least one repetition</td>
<td>Various temporal thresholds tested, providing too sporadic results</td>
<td>Count of clicks, list of timestamps</td>
</tr>
<tr>
<td></td>
<td>Fail To Click Ignore Node*</td>
<td>Repeated clicks at any page element</td>
<td>Various temporal thresholds tested, providing too sporadic results</td>
<td>Count of clicks, list of timestamps</td>
</tr>
<tr>
<td></td>
<td>Click After Load</td>
<td>First mouse click after the page is loaded</td>
<td>No thresholds</td>
<td>Time to first click</td>
</tr>
<tr>
<td></td>
<td>Mouse Move</td>
<td>Records all mouse movement.</td>
<td>No thresholds</td>
<td>Distance travelled with the mouse</td>
</tr>
<tr>
<td>Scroll</td>
<td>Controlled Scroll</td>
<td>Consecutive scroll events</td>
<td>More than one scroll movement within 10 seconds</td>
<td>Scroll information, count of scroll events</td>
</tr>
<tr>
<td></td>
<td>Fast Mouse Scroll Cycle*</td>
<td>Consecutive scroll events indicating a scroll down and up again.</td>
<td>Various temporal and scroll distance thresholds tested providing too sporadic results</td>
<td>Scroll information, count of scroll events</td>
</tr>
<tr>
<td></td>
<td>Fast Single Direction Scroll*</td>
<td>Consecutive scroll events indicating a fast scroll down</td>
<td>Various temporal and scroll distance thresholds tested providing too sporadic results</td>
<td>Scroll information, count of scroll events</td>
</tr>
</tbody>
</table>

Table 3.5: Extracted Micro Behaviours
3.3.6 Summary

Microbehaviours have been presented as a way to provide comparable interaction units. Low-level interaction data can be aggregated into microbehaviours to support analysis over extended periods of time. As opposed to coarse Web interaction metrics, such as page visits, microbehaviours retain the fine-grained aspect of the interaction. Analysis of how extracted microbehaviours change over time provide insight into subtle changes in user interaction. To support the longitudinal analysis proposed in Section 3.2 a high enough occurrence of microbehaviours is required. Only mouse microbehaviours based on interaction inherent to the use of the cs.manchester.ac.uk Web site have been found to be useful for the proposed analysis. These results are specific to the Web site being analysed. Several microbehaviours have been designed describing very specific interaction situations. Although these interaction situations have not been found to be frequent enough in this study, their occurrence might vary under different circumstances. Microbehaviours offer an extensible approach to analysing low-level interaction data. Additional microbehaviours can be designed, and existing ones can be augmented with additional information. The presented design of microbehaviours have focused on discovering temporal correlations. Alternatively, resulting microbehaviours design can focus on precision – e.g. describing the accuracy of mouse interaction – or particular measurements of skill – e.g. measuring typing speed.
Chapter 4

Application of the Analysis Methodology

The analysis methodology presented in Chapter 3 is applied to interaction data captured from the cs.manchester.ac.uk as a test to use users’ low-level interaction as a proxy for their degree of familiarity. The cs.manchester.ac.uk Web site contains information about the School of Computer Science at the University of Manchester. This Web site is used by a range of people (including current and prospective students and staff) and contains a high number of varying pages covering news, events, teaching and research. In this context familiarity with a Web site or application is considered a high-level subjective factor concerning how natural a user’s interaction is and the ability to recognise interface elements. I assert that a user’s familiarity with a Web site or application increases over time. Therefore temporally correlated low-level interaction factors can be employed as a proxy for familiarity. A longitudinal analysis of Web interaction is necessary to test this hypothesis. To apply the longitudinal analysis methodology presented in Chapter 3 the design of a set of micro behaviours related to familiarity is necessary. Following the proposed methodology, a combination of a review of existing research and manual observation of the captured events is employed to obtain prospective micro behaviours. Features from these micro behaviours are extracted and temporal correlations are discovered: episodes have been found to get longer over time, users tend to scroll faster, and mouse inactivity increases.
CHAPTER 4. APPLICATION OF THE ANALYSIS METHODOLOGY

4.1 Familiarity

In this context familiarity with a Web site or application is considered a high-level subjective factor concerning how natural a user’s interaction is and the ability to recognise interface elements. I have asserted that a user’s degree of familiarity with a Web site or application increases over time. Therefore temporally correlated low-level interaction factors can be employed as a proxy for familiarity. Familiarity cannot be measured through expert assessment, thus requiring user feedback, making its remote analysis challenging.

The purpose of the survey is to compare the state of users’ degree of familiarity at two different points in time. One of the key points of the proposed analysis methodology is the unobtrusiveness of the observation. Although showing a survey to the participants of the study can be considered obtrusive it is a necessary step for the validation of the methodology. The stipulated assertion needs to be supported to use changes in interaction over time as a proxy for familiarity. Two different questionnaires have been designed to be shown to different users at different stages of the interaction. Two of the questions are the same in both questionnaires – indicated with an asterisk (*) – so differences between the distribution of answers from both user groups can be compared.

The following questionnaire has been shown to first-time visitors to the Web site:

- How easy is the site to interact with?*
- How familiar do you currently feel with the site?*
- Did you think consistency between Web pages across the Internet helped you feel more familiar with the site?
- Was the Web site as easy to use as you expected?

The following questionnaire has been shown to users on their second visit to the site:

- How easy is the site to interact with?*
- How familiar do you currently feel with the site?*
- Do you think consistency across Web pages helped you feel more familiar with the site?
- Do you find the site easy to remember each time you come back?
- Did you find the site easier to interact each time you visited it?

When users visited the site for the first time, there was a 50% chance to be selected to partake in the first survey. The other 50% of users were selected as possible recurrent
4.1. FAMILIARITY

Figure 4.1: Responses to the questionnaire shown to first-time visitors.

Figure 4.2: Responses to the questionnaire shown to recurrent visitors.

users and were shown the second survey if they came back to the site. In both cases, the users have been asked to answer the survey at the end of their interaction.

101 answers have been collected from first-time visitors and 176 from recurrent visitors. The responses for both surveys are shown in Figures 4.1 and 4.2. A Wilcoxon test has been applied to test if the responses from recurrent users are significantly higher than the responses from first-time visitors. For the question “How easy is the site to interact with?” the median of first-time and recurrent visitors is 4, and the difference has not been found to be significant (W = 8616.5, Z = -0.45, p = 0.3253). However, for the question “How familiar do you currently feel with the site?” the medians of first-time visitors and recurrent visitors are 3 and 4, respectively and Wilcoxon test demonstrates that the familiarity of recurrent users is significantly higher than first-time visitors (W
The assertion that users’ familiarity increases over time is supported, hence providing a ground truth that makes it possible to use time spent on the site as a proxy for familiarity. The survey includes a question to test if the consistency between Web pages supported familiarity. This question has been asked for a possible study of familiarity based on Web page similarity, using the clustering of Web pages presented in Figure 3.6. The majority of users believed that consistency across Web pages helped them feel more familiar with the site – 76% of responses above 3 on a scale of 1 to 5.

### 4.2 Design decisions for analysis

Before the application of the analysis presented in Section 3.2, a series of design decisions need to be taken. The selection of an appropriate temporal metric for the analysis, as well as the preparation of the data need to be taken into account.

**Temporal metrics** Two different temporal metrics have been designed for the analysis: active time and episode count. Active time represents in a precise way the amount of time the user has been actively interacting with the Web site or application. Episode count, on the other hand, is defined by the amount of episodes accounted for a particular user up to that point. Episode count does not necessarily reflect a user’s total amount of interaction time, as the episodes’ length can vary from minutes to hours. Therefore, active time reflects a user’s accumulative interaction time in a more precise way.

The application of the proposed analysis methodology relies on the compliance to the requirements of linear mixed models. Among other requirements – such as independence in the input, see Section 3.2.4 – normality is recommended, even if these models are robust against violations of the assumptions of normality. Figure 4.3 shows the distribution of the residuals for the proposed temporal metrics: episode count and active time. The distribution of the residuals has been found to be similar for the application of all models with the correspondent temporal metric. Distributions for episode count and active time have been found to be bimodal and skewed left respectively. The Q–Q plot in Figure 4.4 supports these observations. Skewness has been found to be inherent to the captured interaction data, and it has been considered to be a less severe deviation from normality than a bimodal distribution. For this reason, as well as the additional precision that active time provides over episode count, active
4.2. DESIGN DECISIONS FOR ANALYSIS

(a) Using episode count as the temporal metric. (b) Using active time as the temporal metric.

Figure 4.3: Histogram of the residuals of a mixed linear model for episode duration.

(a) Using episode count as the temporal metric. (b) Using active time as the temporal metric.

Figure 4.4: Q–Q plot of the residuals of a mixed linear model for episode duration.

time has been employed for the analysis.

**Logarithmic transformation** Heteroskedasticity is a possible risk when applying mixed linear models. When heteroskedasticity occurs, the variability of the model is unequal across the range of values of the predictor. The result is an increasing loss of accuracy as the predictor’s value increases. Figure 4.5a shows an example in which the variance decreases for smaller values of the predictor. In this case, the mixed linear model has been applied to the data without any transformation to the data. The distance between the values fitted by the model and the resulting variance increases consistently. In Figure 4.5b a logarithmic transformation has been applied to the data. The distribution of the variance is more distributed, and the model shows less bias.

**Use of clusters** Due to the existing differences between Web pages from the same Web site discovered temporal correlations might be specific to a type of Web page, or even a single Web page. The similarity between pages has been explored to help
(a) Without applying a logarithmic transformation to the input.

(b) Applying a logarithmic transformation to the input.

Figure 4.5: Visualisation of the resulting fitted values by the model against the residuals for a mixed linear model for episode duration using active time as the temporal metric.

determine if discovered temporal correlations were unique to a set of similar pages, or all pages on the site. Accessed URLs from the cs.manchester.ac.uk site have been clustered based on the technique presented in Figure 3.6. The Web site’s homepage received more visitors than any other page on the site. Analysing the results from different clusters helped to identify if data from the homepage is obscuring possibly contradicting results from other Web pages. Each user and URL pair have been fed into the model as a unique user, to prevent effects from the variability between Web pages. When possible, results of applying the model to all URLs on the site, and the subset of URLs in the homepage’s cluster are reported.

4.3 Analysis of Survey participants

Interaction data captured from the participants of the survey was extracted for analysis. Based on the finding that familiarity increases over time, temporal correlations among extracted micro behaviours can be considered to act as a proxy for familiarity. The presented micro behaviours have been analysed, and problems derived from the small sample of employed users have been taken into account.

For the analysis of micro behaviours, only users exhibiting the micro behaviour at least twice have been selected. Two is the minimum required frequency of observations to find linear correlations. A maximum limit of 20 minutes of active time has been set to prevent bias towards unusually active users over time – it has been shown that the number of unique users with an active time greater than 5 minutes is already low, so 20 is a conservative limit, see Figure 3.7. Active time has been introduced in
4.3. ANALYSIS OF SURVEY PARTICIPANTS

Section 3.1.3 as a more accurate depiction of user interaction time. For example, even if an episode is long a user might have interacted for a total of a few seconds. Due to the limited amount of available users, analysis across different URLs can be problematic. There is a limited amount of datapoints per user, and when additional URLs are included, the distribution of users across URLs might be uneven. The duration of controlled scroll across all Web pages in cs.manchester.ac.uk was analysed. This micro behaviour was found to be exhibited at least twice by 91 different users who accessed 209 different URLs leading to 343 pairs of user/URL. Of those 209 URLs, only 50 received more than 4 visits. Furthermore, only 25 pairs of user/URL exhibited this behaviour at least 5 times. This diversification has been considered too severe to account for variability between pages. With this limited amount of users per page the risk of a few users exhibiting particularly strong correlations increases. The use of the introduced random sampling technique is not applicable either, which will require additional interaction data from the Web site.

Micro behaviours introduced in Section 3.3 have been analysed for the participants of the survey. Table 4.1 summarises the results of the analysis. For each micro behaviour the number of users, the number of resulting observations, and the resulting population correlation from the application of the linear mixed model is reported. Population correlation refers to the resulting prediction of the model for the entire population, depicted in each plot with a black line. The correlation indicates the trend of each feature as users’ active time increases. The number of occurrences varies between micro behaviours. The requirement of discarding users with fewer than 2 occurrences limits the number of observations, resulting in a very limited amount of users. The majority of the explored features showed a consistently positive correlation with the amount of active time on the page – the resulting models can be seen in A.1. Mouse Click Speed 4.6 and Mouse Idle After Click 4.7 show conflicting results among users. These results might be highly dependent on the employed user sample. However, the reduced number of survey participants makes the use of the random sampling technique presented in Chapter 3 challenging.

Due to the reduced number of participants, the observation might be biased. Consistent correlations discovered in the use of the homepage could not be tested against the rest of pages in the site. Therefore the effect of particularly active users could not be properly tested. The discovered temporal correlations might be overfitted to the reduced number of participants of the survey. The rest of the visitors to the page have been analysed to determine which correlations were caused by the limited amount of
CHAPTER 4. APPLICATION OF THE ANALYSIS METHODOLOGY

<table>
<thead>
<tr>
<th>Micro Behaviour</th>
<th>Number of Users</th>
<th>Number of Values</th>
<th>Population Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse Click Speed Figure 4.6</td>
<td>46</td>
<td>219</td>
<td>-0.03413739</td>
</tr>
<tr>
<td>Mouse Inactive Time Median Figure A.1</td>
<td>49</td>
<td>49</td>
<td>0.06588018</td>
</tr>
<tr>
<td>Mouse Inactive Time Mean Figure A.2</td>
<td>49</td>
<td>235</td>
<td>0.2091507</td>
</tr>
<tr>
<td>Mouse Time To Click Figure A.3</td>
<td>46</td>
<td>217</td>
<td>0.141652</td>
</tr>
<tr>
<td>Mouse Idle After Click Figure 4.7</td>
<td>30</td>
<td>140</td>
<td>-0.007105687</td>
</tr>
<tr>
<td>Mouse Click After Load Figure A.4</td>
<td>47</td>
<td>223</td>
<td>0.1921277</td>
</tr>
<tr>
<td>Mouse Move Distance Figure A.5</td>
<td>58</td>
<td>321</td>
<td>0.255083</td>
</tr>
<tr>
<td>Controlled Scroll Duration Figure A.6</td>
<td>18</td>
<td>79</td>
<td>-0.1859106</td>
</tr>
<tr>
<td>Controlled Scroll Speed Figure A.7</td>
<td>18</td>
<td>79</td>
<td>0.382929</td>
</tr>
<tr>
<td>Episode Duration Figure A.8</td>
<td>65</td>
<td>370</td>
<td>0.1607346</td>
</tr>
</tbody>
</table>

Table 4.1: Analysis of Micro Behaviours for Survey Participants

analysed users.

Survey Participants Mouse Click Speed

Figure 4.6: Analysis of Mouse Click Speed for the survey participants. Limited to users with at least 2 occurrences.

4.4 Analysis of population

To test if results were replicable the analysis methodology was applied to two independent datasets of data obtained from the same Web page. “Phase 1” contains data captured over a year from 14,000 unique recurring users. “Phase 2” occurred over
4.4. ANALYSIS OF POPULATION

Survey Participants Mouse Idle After Click

Figure 4.7: Analysis of Mouse Idle After Click for the survey participants. Limited to users with at least 2 occurrences.

the subsequent four months, and includes data from over 3,000 unique recurrent users. The same settings from the analysis of participant of the survey were maintained so all analyses use logarithmic transformations, active time as the temporal metric, and require a minimum of two occurrences per user. As opposed to the previous analysis, where particular users were selected, this analysis considers all visitors to the cs.manchester.ac.uk Web site. Widening the scope of the analysis provided more users, but also increased the skewness. Initial analysis using the same settings as with the participants of the survey failed to provide an appropriate model, leading to unstable predictions. A detailed analysis of the distribution of micro behaviour occurrences among users revealed a high skewness towards users with fewer occurrences. Table 4.2 shows the distribution of users according to the occurrence of the controlled scroll micro behaviour. The number of users with a minimum of 2 occurrences doubles the number of users with a minimum of 3 – 805 users vs 414, meaning there are 391 users who provided just two occurrences. Although two visits is considered the minimum amount of occurrences to observe a linear correlation, the distribution is clearly highly skewed. An additional limit of two minutes of active was introduced to obtain a more even distribution.

Tables 4.3 and 4.4 show the results of applying the model to the different phases
CHAPTER 4. APPLICATION OF THE ANALYSIS METHODOLOGY

<table>
<thead>
<tr>
<th>Number of occurrences</th>
<th>1 min</th>
<th>2 min</th>
<th>3 min</th>
<th>4 min</th>
<th>5 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum of 1</td>
<td>1313</td>
<td>579</td>
<td>345</td>
<td>241</td>
<td>174</td>
</tr>
<tr>
<td>Minimum of 2</td>
<td>805</td>
<td>494</td>
<td>320</td>
<td>232</td>
<td>170</td>
</tr>
<tr>
<td>Minimum of 3</td>
<td>414</td>
<td>330</td>
<td>248</td>
<td>198</td>
<td>151</td>
</tr>
<tr>
<td>Minimum of 4</td>
<td>233</td>
<td>209</td>
<td>178</td>
<td>156</td>
<td>130</td>
</tr>
<tr>
<td>Minimum of 5</td>
<td>153</td>
<td>147</td>
<td>136</td>
<td>123</td>
<td>102</td>
</tr>
</tbody>
</table>

Table 4.2: Distribution of users for different active time limits for the analysis of the controlled scroll micro behaviour.

<table>
<thead>
<tr>
<th>Micro Behaviour</th>
<th>Number of Users</th>
<th>Number of Values</th>
<th>Population Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse Click Speed</td>
<td>612</td>
<td>3710</td>
<td>0.01895399</td>
</tr>
<tr>
<td>Mouse Inactive Time Median</td>
<td>899</td>
<td>6191</td>
<td>0.06273081</td>
</tr>
<tr>
<td>Mouse Inactive Time Mean</td>
<td>899</td>
<td>6191</td>
<td>0.09498143</td>
</tr>
<tr>
<td>Mouse Time To Click</td>
<td>638</td>
<td>4061</td>
<td>0.008447066</td>
</tr>
<tr>
<td>Mouse Idle After Click</td>
<td>507</td>
<td>2588</td>
<td>0.01845398</td>
</tr>
<tr>
<td>Mouse Click After Load</td>
<td>652</td>
<td>4166</td>
<td>0.03362863</td>
</tr>
<tr>
<td>Mouse Move Distance</td>
<td>927</td>
<td>7979</td>
<td>0.03317005</td>
</tr>
<tr>
<td>Scroll Controlled Scroll Duration</td>
<td>494</td>
<td>2553</td>
<td>-0.03029693</td>
</tr>
<tr>
<td>Scroll Controlled Scroll Speed</td>
<td>494</td>
<td>2553</td>
<td>0.07060311</td>
</tr>
<tr>
<td>Episode Duration</td>
<td>1023</td>
<td>9599</td>
<td>0.07001057</td>
</tr>
</tbody>
</table>

Table 4.3: Analysis of Micro Behaviours for Phase 1 data in cs.manchester.ac.uk Web page.

of the data. More users for longer periods of time are being analysed, so the correlations were expected to be weaker than with the survey participants. Number of users between the different phases differ, due to the different lengths of the observations. A comparison of the population correlations for the three analysed datasets is reported in Table 4.5. Mouse Click Speed and Mouse Idle After Click show contradicting correlations across the datasets, and Mouse Time to Click reports a low correlation value compared to the rest of the obtained values.

The repeated sampling analysis technique presented in Section 3.2.4 has been applied to phase 1 data to obtain more insight into the validity of reported correlations. The use of analyses of random samples of varying sizes helps to identify correlations that are dependent on the input. Different sets of pages were analysed to discern Web page specific factors. The homepage of cs.manchester.ac.uk received more visitors than any other page, so analysis limited to this page is reported. A set of pages on the site were considered to be similar to the homepage, and the analysis for this particular
### 4.4. ANALYSIS OF POPULATION

<table>
<thead>
<tr>
<th>Micro Behaviour</th>
<th>Number of Users</th>
<th>Number of Values</th>
<th>Population Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse Click Speed</td>
<td>139</td>
<td>940</td>
<td>-0.00222147</td>
</tr>
<tr>
<td>Mouse Inactive Time Median</td>
<td>180</td>
<td>1158</td>
<td>0.0897745</td>
</tr>
<tr>
<td>Mouse Inactive Time Mean</td>
<td>180</td>
<td>1158</td>
<td>0.09527039</td>
</tr>
<tr>
<td>Mouse Time To Click</td>
<td>145</td>
<td>990</td>
<td>0.006678414</td>
</tr>
<tr>
<td>Mouse Idle After Click</td>
<td>119</td>
<td>637</td>
<td>0.006314002</td>
</tr>
<tr>
<td>Mouse Click After Load</td>
<td>148</td>
<td>1007</td>
<td>0.07103979</td>
</tr>
<tr>
<td>Mouse Move Distance</td>
<td>192</td>
<td>1580</td>
<td>0.04076208</td>
</tr>
<tr>
<td>Scroll Controlled Scroll Duration</td>
<td>83</td>
<td>417</td>
<td>-0.02822751</td>
</tr>
<tr>
<td>Scroll Controlled Scroll Speed</td>
<td>83</td>
<td>417</td>
<td>0.04159891</td>
</tr>
<tr>
<td>Episode Duration</td>
<td>203</td>
<td>1899</td>
<td>0.09282474</td>
</tr>
</tbody>
</table>

Table 4.4: Analysis of Micro Behaviours for Phase 2 data in cs.manchester.ac.uk Web page.

<table>
<thead>
<tr>
<th>Micro Behaviour</th>
<th>Survey</th>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse Click Speed</td>
<td>-0.03413739</td>
<td>0.01895399</td>
<td>-0.00222147</td>
</tr>
<tr>
<td>Mouse Inactive Time Median</td>
<td>0.06588018</td>
<td>0.06273081</td>
<td>0.0897745</td>
</tr>
<tr>
<td>Mouse Inactive Time Mean</td>
<td>0.2091507</td>
<td>0.09498143</td>
<td>0.09527039</td>
</tr>
<tr>
<td>Mouse Time To Click</td>
<td>0.141652</td>
<td>0.008447066</td>
<td>0.006678414</td>
</tr>
<tr>
<td>Mouse Idle After Click</td>
<td>-0.007105687</td>
<td>0.01845398</td>
<td>0.006314002</td>
</tr>
<tr>
<td>Mouse Click After Load</td>
<td>0.1921277</td>
<td>0.03362863</td>
<td>0.07103979</td>
</tr>
<tr>
<td>Mouse Move Distance</td>
<td>0.255083</td>
<td>0.03317005</td>
<td>0.04076208</td>
</tr>
<tr>
<td>Scroll Controlled Scroll Duration</td>
<td>-0.1859106</td>
<td>-0.03029693</td>
<td>-0.02822751</td>
</tr>
<tr>
<td>Scroll Controlled Scroll Speed</td>
<td>0.382929</td>
<td>0.07060311</td>
<td>0.04159891</td>
</tr>
<tr>
<td>Episode Duration</td>
<td>0.1607346</td>
<td>0.07001057</td>
<td>0.09282474</td>
</tr>
</tbody>
</table>

Table 4.5: Comparison of the population correlations for survey participants, phase 1 data, and phase 2 data for the cs.manchester.ac.uk Web page.
set of pages is reported as “homepage cluster”. Finally, analysis of all pages on the site is also reported. Additional analysis excluding the homepage from the mentioned sets of pages is also reported in Section A.2 Comparing the results of this additional analysis with the results presented in this section helps to interpret the effect the inclusion of the homepage can have in the analysis.

The interpretation of the results can vary depending on the focus of the analysis. In this case, a consistently positive or negative correlation is desired. Ideally, the distribution of the resulting correlations for each sample size would remain either positive or negative without crossing the origin – indicated in the figures with a red line. The distance from the median of the correlation distribution for each sample size to the origin is also considered as a metric of the strength of the correlation. The use of small sized samples (10 to 40) has been found to consistently exhibit a high variance. Due to the size of the populations – e.g. 4024 for Mouse Mean Inactive Time in Figure 4.10 Variance in this results have not been considered to be indicative of inconsistent correlations. Employed indicators for the interpretation have been the quartiles and the medians of the distributions, for sample sizes bigger than 40. Sporadically, the variance of the resulting correlation has been found to be bigger than surrounding sample sizes. In this interpretation of the visualisations I have considered the surrounding sample sizes to determine if such variance is negligible. A less conservative interpretation of the data could have employed confidence intervals – see Section 3.2.4 The use of such visualisation has been avoided to increase the reliability of the results, even though some prospective temporal correlations have been rejected as a result.

Based on the results of the analysis prospective micro behaviours have been classified into three different groups. Consistent micro behaviours exhibit a consistent positive or negative correlation across the different Web pages on the site. Inconsistent micro behaviours have been found to exhibit consistent correlations specific to a page or set of similar Web pages. Finally, non temporally correlated micro behaviours have been found not to exhibit any kind of consistent correlation under the circumstances in which this analysis took place.

4.4.1 Consistent micro behaviours

Mouse Inactive Time Both the median – Figure 4.9 and mean mouse inactive time – Figure 4.10 – show consistently positive correlations. Mouse inactivity has been designed as a measure of all mouse inactive periods per episode. It has been hypothesised that as users got more familiar with the Web page they would interact
quicker, resulting in shorter inactive times. Contrary to this hypothesis, it has been found that users’ mouse inactive time increased over time. It was initially believed that due to the particular interactivity of the homepage results could have been an isolated case. The homepage consists of a set of tiles and top menus that require hovering over them to reveal interaction possibilities. Such interaction can lead to higher dwelling times thus resulting in a unique interaction possibly different to other Web pages on the site. Further analysis excluding the homepage from the analysis shows that the occurrence of negative correlations increases slightly when the homepage is excluded. Although both median and mean report similar results, using the median was found to provide a more consistent positive correlation for the different sample sizes – see Figures A.10 and A.11. This phenomenon could be the result of users not having to explore a page they are already familiar with – e.g. first-time users might want to hover over elements to determine what the interaction options are.

**Mouse Click After Load** It has been hypothesised that the time from loading the page to a user clicking on an interface element would decrease over time. It would take users less time to interact if they had been to that Web page before. The correlation has been found to be consistently positive, although occurrences of negative correlations can be seen in the case of small samples when the homepage is included in the analysis – small samples up to 60, see Figure 4.13. Similarly to mouse inactive time, this positive correlation could be the result of users having to explore the page for the first time.

**Controlled Scroll Speed** It has been hypothesised that the speed of the controlled scroll micro behaviour would increase over time. Users coming back to the same page would scroll faster, allowing them to skip unnecessary parts of the page to reach their target. This hypothesis has been supported by the results of the analysis. Except from small sample sizes, this micro behaviour has been found to be consistently positive for all analysed Web page sets – see Figure 4.16. This correlation has been found to be maintained across pages, as the exclusion of the homepage from the analysis has a negligible effect A.17.

**Episode Duration** Initial hypothesis considered that episodes would get shorter over time. Users would spend less time in a page as they come back to it over time. Results of the analysis contradict this initial preconception showing a consistently positive
correlation of the episode duration with a user’s active time. It should be noted that
episode duration is calculated using a timeout of 40 minutes – see Section 3.1.3. If
a user returns to a particular URL within that time the entire interaction will be con-
sidered as being part of the same episode. Users could be coming back to a Web page
they are familiar with to use it as a reference for navigating or finding information.
Results presented in Figure 4.17 supports such theory, showing a consistently positive
correlation for all analysed Web page sets. Excluding the homepage from the analysis
had no effect in the resulting correlations A.18.

4.4.2 Inconsistent micro behaviours

The correlations for these micro behaviours have been found to be inconsistent, and
biased towards a particular type of Web page. Consistency between pages has been
reported to help visitors to the Web site feel more familiar. Temporal correlations
specific to particular Web page types could be the result of unique aspects to that set
of pages. Further analyses of the micro behaviours exhibiting these correlations could
reveal evolution in Web interaction specific to the type of Web page.

Mouse Idle After Click (Homepage Cluster)  It was hypothesised that the time from
a user clicking to moving the mouse would decrease over time. As users come back to
the same page, they would move the mouse quicker after clicking to perform the next
desired interaction. This hypothesis could not be supported as this micro behaviour has
been found to provide inconsistent correlations across the different datasets – see Table
4.5. Analysis of different sample sizes shows an inconsistent correlation in the case of
all pages. In the case of the homepage and the homepage cluster, the correlation has
been found to be more consistent, although the distribution of the resulting correlations
remains close to the origin with a very low median – see Figure 4.12. The exclusion
of the homepage from the analysis of all pages on the site increased the occurrence of
negative correlations A.13. This results might indicate that the homepage and pages
similar to it contain some unique interaction aspects that result in longer idle times
after clicking.

Controlled Scroll Duration (homepage)  I hypothesised that the duration of the con-
trolled scroll micro behaviour would decrease over time. Users would require less time
to find the desired target in the page over time, leading to shorter scrolling periods. This
hypothesis could not be supported by the analysis as the distribution of the correlations
4.4. ANALYSIS OF POPULATION

is inconsistent for all analysed sets of Web pages. The occurrence of negative correlations increases in the case of the analysis of all pages and the homepage cluster\textsuperscript{4.15}. Further analysis excluding the homepage from the analysed page sample – see Figure\textsuperscript{A.16} – reveals that the inconsistency across the analysed Web page sets increases. This might indicate that users scroll for shorter periods on time only in the case of the homepage compared to the rest of the pages.

4.4.3 Unusable micro behaviours

No consistent temporal correlations has been found for these micro behaviours. Distribution of the correlations was inconsistent for all analysed Web page sets indicating that the resulting correlation is highly dependent on the input. Therefore the hypothesis formulated for these micro behaviours could not be supported.

**Mouse Click Speed** I hypothesised this micro behaviour would show a decreasing value reflecting quicker full click interactions from users. The population correlation for this micro behaviour has already been found to differ across the different datasets, suggesting this correlation to be unstable, and dependent on the analysed sample – see Figure\textsuperscript{4.5}. Figure\textsuperscript{4.8} supports that conclusion showing a highly inconsistent correlation across all analysed Web page sets.

**Mouse Time To Click** Similarly to Mouse Idle After Click, it was hypothesised that as users got more familiar the interaction with the mouse would be quicker. The time from hovering over an element to clicking it was expected to reduce as users got more familiar with the Web site. Table\textsuperscript{4.5} shows that the population correlation has been found to be consistent across the three analysed datasets. However, the correlation has been mentioned to be particularly small compared to the rest of micro behaviours. Figure\textsuperscript{4.11} shows inconsistent correlations across the different sample sizes. The distribution of the resulting correlations crosses across the origin for the majority of the analysed sample sizes showing that the correlation is highly dependent on the input.

**Mouse Move Distance** It was hypothesised that mouse move distance would decrease over time. As users got more familiar with the Web site, they would become more efficient. Unnecessary mouse movements would be reduced as well as the resulting mouse travelled distance. Initial analysis of this micro behaviour failed to support this hypothesis, showing a consistent increase of mouse move distance over time for all
the analysed data sets – see Table 4.5. A more thorough analysis showed that the dis-
tribution of the correlations is inconsistent across the different explored sample sizes – see Figure 4.14. The exhibited correlation was not found to be consistent enough to
determine how the travelled distance with the mouse changes over time.
Figure 4.8: Analysis of Mouse Click Speed Time for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure 4.9: Analysis of Mouse Median Inactive Time for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
4.4. ANALYSIS OF POPULATION

Figure 4.10: Analysis of Mouse Mean Inactive Time for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
CHAPTER 4. APPLICATION OF THE ANALYSIS METHODOLOGY

Figure 4.11: Analysis of Mouse Time To Click for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
4.4. ANALYSIS OF POPULATION

Figure 4.12: Analysis of Mouse Idle After Click for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure 4.13: Analysis of Mouse Click After Load for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
4.4. ANALYSIS OF POPULATION

Figure 4.14: Analysis of Mouse Move Distance for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure 4.15: Analysis of Controlled Scroll Duration for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure 4.16: Analysis of Controlled Scroll Speed for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure 4.17: Analysis of Episode Duration for Phase 1 data. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
4.4. ANALYSIS OF POPULATION

4.4.4 Conclusion

Extracted micro behaviours have been analysed employing the random sampling ana-
lysis technique described in 3.2.4. This analysis has discovered consistent temporal
correlations across several micro behaviours. Initial exploration has compared the re-

tsults of applying a linear mixed model to two different datasets of varying lengths.
Unstable correlations, as well as particularly weak correlations, have been identified.
Application of the proposed repeated sampling analysis technique has presented a more
robust discovery of temporally correlated features. Analysis of just one of the datasets
has identified a wider range of unstable correlations. This analysis has been comple-
mented with a comparison based on page similarity. The effect of interaction data
captured from the homepage – the Web page with the higher count of visits – has on
the results of the analysis of all pages can be explored. This analysis has helped to
discard a set of temporal correlations specific to the homepage, or pages similar to it.

I have hypothesised that users scroll faster as they get more familiar with the Web
site. The speed of the controlled scroll micro behaviour has been found to be increasing
for all analysed Web page sets, therefore supporting the formulated hypothesis. The
hypothesis that episodes would get shorter over time has been contradicted. Results
indicate that episodes’ duration increase consistently over time for all analysed Web
page sets. It has been mentioned that this result might have been caused by the way
episodes are defined. A timeout of 40 minutes has been used to split interaction data
into episodes. If a user comes back to a Web page before this 40 minutes threshold
ends, the following interaction will be considered as part of the same episode. This
could be caused by users going back to pages they are familiar with to use them as a
reference. Laboratory observations, or a more detailed analysis of the captured data in
this study, are necessary to support this assertion.

The initial hypothesis that mouse interaction would get quicker over time leading to
shorter inactive periods has been contradicted. Results indicate that mouse inactivity
increases over time, which might be caused by the initial need of exploring unfamiliar
pages. One possible reason is that the majority of Web pages require particular inter-
actions in order to disclose additional information – such as hovering over elements.
Therefore, users would need to explore available interface elements to know where
information is, resulting in more frequent mouse interaction. In later visits, rather than
exploring the page, users would need to remember where the information is, leading
to longer idle periods. This theory is supported by the discovery that the time from the
page loading to users interacting with a page element increases over time.
Several micro behaviours exhibited temporal correlations specific to particular Web pages. It could be possible that these micro behaviours are related to particular aspects of certain Web page types. A more thorough analysis of the different Web page clusters would be required to support this relation. Under the particular circumstances of the performed analysis, the exhibited correlations of such micro behaviours have been found to be inconsistent. Therefore, no hypothesis about users’ Web interaction could be supported. A more thorough analysis of the rest of the Web page clusters would help to identify if these correlations are specific to a particular set of pages. Finally, the rest of micro behaviours have not been found to exhibit sufficiently consistent correlations. Initial preconceptions, such as users clicking faster over time, have been contradicted.

4.5 Prediction of familiarity

Based on the ground truth obtained through the survey, a way of predicting if a user is above a particular episode count has been explored. In Section 4.4 the relation of users’ episode count with different features has been explored leading to the discovery of several consistent correlations. Therefore, values obtained from episodes are expected to change according to the discovered correlations. Controlled Scroll Speed and Mouse Inactive Time have been chosen as two of the most discriminative micro behaviours. Figures 4.18 and 4.19 show the relation of these two micro behaviours with the percentage of users above the various episode counts. As expected, the lower the episode count, the higher the percentage of users’ on a higher episode count – e.g. 75% above episode count 1 and 28% above episode 6 when no thresholds are enforced. These plots support the findings from previous section, showing a relation of the given feature with users’ episode count.

Considering the ground truth obtained through the survey, which compared first-time visitors with users who had at least two episodes, a way to predict users’ third episode is designed. The two most discriminative features are selected: controlled scroll speed and inactive mouse time. Using the values of these features as filters, the percentage of episodes higher than three can be increased by a 10%. The heatmap shown in Figure 4.20 combines both features. In this figure the percentage of episodes above the third count are reported, dependant on the selected features. As the row count increases, a higher mouse inactivity threshold is applied – i.e. in the 3rd row, the percentage of episodes above the third count, out of all episodes with a mouse inactivity value of at least 2 seconds, are reported. As the column count increases, a
4.5. PREDICTION OF FAMILIARITY

Figure 4.18: Percentage of episodes above each threshold for the values of the speed of the controlled scroll micro behaviour.

Figure 4.19: Percentage of episodes above each threshold for the values of the median inactive mouse time micro behaviour.

higher controlled scroll speed threshold is applied. The percentage of episodes above the specified threshold increased from a minimum of 45% – no thresholds – to 67% – over 16 seconds of inactive mouse time and above 54 scroll units per second – the employed measure for scroll speed.

The reported increase of 22% supports the discovered temporal correlations. Selected consistent micro behaviours have been shown to be positively correlated with users’ episode count. Predicting a user’s degree of familiarity prevents the need for continuous observation. Additional micro behaviours can be used to design a more sophisticated predictor of familiarity.
4.6 Summary

Possible links between high-level interaction factors and changes in low-level interaction have been explored. Familiarity is a subjective factor, preventing its measurement through expert assessment. The use of current analysis techniques is impractical to provide insight into familiarity, due to its subjectivity and the necessity for continuous observation. The use of low-level events as a proxy for familiarity provides the means for a naturalistic longitudinal analysis. Such approach tackles existing gaps in current longitudinal analysis techniques.

It has been asserted that users’ degree of familiarity is correlated with their episode count. A survey has been conducted in a real Web site to test this assertion. Results have shown that users felt more familiar with the Web site at the end of their second visit, supporting the stipulated assertion. It has also been found that consistency across pages helps user feel more familiar. Therefore analysing how a user’s Web interaction changes over time can provide the means to use low-level interaction data as proxy for familiarity.

The possibility of using temporally correlated micro behaviours from Section 3.3 as a proxy has been explored. The same consistent temporal correlations discovered among the visitors of the cs.manchester.ac.uk site can be found among the participants of the survey. Users’ episode tend to be longer, possibly due to the use of more familiar pages as reference inducing frequent revisits. Scroll interactions get faster over time as users scroll faster to reach the desired point on the page. Inactive mouse periods and the time from loading the page to interacting with it have been found to
increase over time, possibly due to the lack of need to explore interaction possibilities with the Web page.

The observed correlations have been employed to decrease the uncertainty of a user’s familiarity with the Web site. The relation of the most discriminative features and the percentage of available episodes above a given episode count has been explored. Based on the ground truth obtained from the remote survey, a way to reduce the uncertainty about users having been to the site at least three times has been explored. Based on data captured over a year, uncertainty is reduced up to a 22\% – from 67\% to 45\% – by employing the values of Controlled Scroll Speed and Inactive Mouse Time as thresholds.

The discovered temporal correlations are validated, supporting the theory that micro behaviours can be used as a proxy for familiarity. Therefore micro behaviours have been shown to be an effective way of finding subtle interaction changes in user interaction. The analysis framework proposed in Section 3.2 has also been shown to be an appropriate technique to discover longitudinal tendencies among freely captured and highly skewed interaction data.
Design of a Web application to support interaction analysis

Discovering emerging interaction patterns supports short-term interaction design, enabling the adaptation of interfaces to real usage, rather than an idealised one. Furthermore, the observation of how these patterns change over time provides insight into how interaction with interface elements evolves.

The aggregation of sequences of low-level events – such as mouse clicks and scroll actions – into micro-level behaviours provides comparable interaction units. These interaction units are sufficiently fine-grained to allow the extraction of features that reflect low-level user interaction. Evolving features, such as users’ scrolling faster, provides a picture of how users’ interaction changes as they become more experienced with the interface. This knowledge enables the recommendation of techniques that accelerate the process by which users’ overcome initial interaction problems. Supporting their learning progress, and increasing the usability of the interface.

The application of the methodology presented in Section 3.2 is not straightforward. Chapter 4 has presented the analysis of 16 months of interaction data captured from the cs.manchester.ac.uk site to provide insight into the higher-level concept of familiarity. The analysis and interpretation of the extracted micro behaviours are cumbersome, requiring the preparation of the input data, and the creation of subsequent plots to identify features exhibiting temporal correlations.

A Web application has been created to ease the steps conforming the analysis. It provides the means to design an appropriate input for the employed analysis model and creates various plots to visualise and understand the results. This way a tool allowing an easy introspection into long-term factors of user interaction is provided. Extraction
of high-level events is also explored enabling the discovery of emerging behaviours. In this context high-level events are considered direct interactions with a Web page’s element, in order to fulfil a particular objective – e.g. clicking on a “Search” button to perform a query. Discovery of emerging behaviours and how they change over time supports interaction design for recurrent users of the Web site or application.

5.1 Interactivity

Easing the process of exploration of the results of the analysis is the objective of the implemented tool. The complexity of the longitudinal analysis of the extracted micro behaviours makes the presentation of all possible results challenging – see Chapter 4. Interactivity has been employed extensively in the data visualisation domain to enhance the possibilities of data exploration – see Section 2.5. Discarding predefined metrics enables the discovery of phenomena that would otherwise remain unknown. Designers can manipulate the inputs to get different insights into the results of the analysis. The complexity of this manipulation depends on the employed visualisation tool, ranging from the selection of users’ expertise to more complex data filtering to optimise the distribution of the analysed data. Input for the analysis needs to be appropriately prepared, so it conforms to the assumptions for the employed mixed linear models. Logarithmic transformations and the definition of particular temporal boundaries are required for the preparation of the input data. The interactivity of the tool enables the discovery of the appropriate transformation rules, guiding the users through visualisations that indicate the suitability of the input.

5.2 Longitudinal emerging sequences

The Web application presented in Section 3.1.1 has been augmented so longitudinal analysis is supported. Taking advantage of the long-term aspect of the collected data, visualisations between different stages of expertise with the Web application can be shown. The evolution between different visits of users with a similar degree of experience with the Web application can be shown, to analyse the evolution in users’ interaction. Figure 5.1 exemplifies the differences in usage between the same users with different levels of expertise with the Web application.
Applications to Interaction design  

The implemented tool provides a way to explore the real usage of an interface effortlessly. Results are compared with the original design of the interaction to improve the adaptation to users. In the provided example (Figure 3.1) the designer needs to take into account the found unexpected use to maintain the flow of the interaction.

Users found several functionalities of the Web application useful at the start of the interaction but they disregarded over time. Figure 5.1 shows how corrective functionalities such as “Cancel” and “Reset” are not used in later visits. Experimental functionalities that this Web application offered (“NetViz”) are found not to be used either, indicating that users tried them on their first visit but did not find them useful in later visits. This way adaptation of the interface to real users’ long-term interaction is possible.

Limitations  

Although useful to discover emerging behaviours such as the mentioned example, this approach is not scalable. High-level events have been automatically identified from a small Web application. In the case of bigger applications, automatic identification might lead to numerous nodes and transitions. Represented transitions might not be relevant, resulting in an overloaded visualisation, and requiring additional manual selection of events to provide a useful representation. Furthermore, this approach assumes all possible interaction sequences are known a priori, therefore prejudging how users interact. Any interaction outside this initial selection of sequences will be disregarded from the analysis.
5.3 Evolution of micro behaviours

In order to make the analysis of large volumes of interaction data over time possible, a way of splitting interaction into comparable units is necessary. Collected low-level events are aggregated into sequences which conform micro-level behaviours to support a longitudinal analysis of interaction data – see Section 3.3.

The aggregation of interaction events into micro behaviours is employed in order to obtain fine-grained units from which to extract relevant features. For example, in the case of the aggregation of short bursts of scroll interaction, speed and duration of the scroll action are extracted. In the case of mouse activity, saccade-like mouse movements are quantified. Analysing how these features change over time helps understand how user interaction evolves. One direct application of such technique is testing pre-judgements of the data. Assumptions about user interaction – e.g. mouse activity increasing as the user becomes more experienced – can be seamlessly tested on any Web interface – see Chapter 4.

Figure 5.2: Example of the developed Web application to explore the evolution of micro behaviours.

Although timeline visualisations can be useful to discern temporal changes, they can become difficult to read when the number of users to be analysed is big. In order to quantify the change of particular features over time, linear mixed models have been used [Kuznetsova et al., 2014]. Linear mixed models are a more sophisticated version...
CHAPTER 5. DESIGN OF A WEB APPLICATION TO SUPPORT INTERACTION ANALYSIS

of general linear models. They are particularly suitable for longitudinal analysis as they support the use of non-independent data (i.e. data collected from different participants). The model is robust against unbalanced data [Bates, 2010] and calculates different intercepts and slopes for each user. The model considers how noisy the data from each user is, and calculates a user-specific intercept and slope taking into account data from other users.

Details of the analysis implementation can be found in Chapter 3. The analysis was implemented using R [R Core Team, 2014], and Shiny R [Chang, 2015] was used to create the Web application. The interface of the implemented Web application is shown in Figure 5.2. Due to the size of the data, and the computational requirements the Web application has been designed to be run locally. During the analysis a skewness in the number of users over time was noticed. As longer periods of time were analysed, the amount of users decreased. Although expected, filtering methods that allowed a robust analysis of the data are necessary. The implemented Web application provides access to several filtering options. Filtering data enables a more distributed sample of participants, increasing the reliability of the resulted prediction.

The Web application provides different visualisation utilities to firstly understand the data, to then be able to apply linear mixed models appropriately. Coarse differences between different expertise stages can be observed visualising the data via density plots (see Figure 3.11). Although comparison within users is not possible using this visualisation, it provides insight into population-wide changes across different levels of expertise.

The analysis application also provides tools to evaluate the robustness of the model. Details on the distribution of residuals and statistical significance are reported. Even though linear mixed models are more robust than general linear models, their use still requires compliance to a set of assumptions [Winter, 2013]. The tool automatically generates visualisations to evaluate such compliance, as well as instructions for doing so. Finally, and in order to account for memory effects, additional factors can be included in the model. The visualisations presented only include two variables, in order to provide insight into longitudinal relations between a feature and a temporal variable. An additional model is generated, including the length of time between episodes as an additional variable. This way, memory effects can be accounted for. Both models are then compared, in order to evaluate if the introduction of memory effects accounts for a better model.
5.4 Summary

Commercial approaches such as Google analytics provide tools focused on Web pages as a factor, calculating number of page views, as well as to identify users’ landing page. Implemented tool provides additional information, with finer grained metrics, shifting the focus to users’ interaction evolution. Users’ visits are not processed as independent data-points, and low-level interaction is aggregated into temporally comparable interaction units. Changes of other micro behaviours’ features provide insight into changes of other aspects of the interaction.

A solution to unobtrusively capture low-level interaction data from Web applications and Web sites longitudinally has been implemented, which can be deployed in any Web page through minimal changes – see Section [3.1]. Particular sequences of events are aggregated into micro behaviours to obtain comparable units of interaction. An analysis methodology to discover temporal correlations across extracted micro behaviours is presented. The process of preparing the data and exploring the different micro behaviours is cumbersome particularly for users without programming experience – the analysis requires knowledge of the R programming language.

A Web application has been implemented to ease the process of exploration of the different micro behaviours and guide the preparation of the data. The tools necessary to make the input data comply to the necessary assumptions of the employed linear mixed models are provided. Several visualisations are provided, to support the discovery of temporal correlations. Density plots provide insight into the overall tendency of the micro behaviour with respect to different stages of expertise. Linear mixed models account for the variability between users and help to find how each user’s interaction changes over time.

A way of extracting meaningful high-level interaction from the captured low-level interaction data has been presented. Visualising transitions between high-level events help to discover emerging behaviours. Designers can make use of these behaviours to adapt the Web application to users’ real interaction. Taking advantage of the temporal aspect, an alternative visualisation that enables the comparison of these transitions at different stages of interaction is presented. Changes in the use of the different interface features have been identified, supporting the adaptation of the interface to more experienced users.

A data-driven approach to the analysis of longitudinal data is necessary to prevent introducing bias into the analysis – see Section [3.2.2]. Guiding the analysis of the data through bespoke tools risk introducing bias into the results. Even though one possible
idea is finding a proper way to visualise the data, and let the researcher find out the relevance of the data, statistical analysis has been found to be necessary. Therefore a tradeoff between how much data processing is necessary and respecting the researchers’ freedom to foster the discovery of unexpected information is needed. Rather than setting predefined options the researcher is offered a selection of several features to analyse and is encouraged to explore the different visualisations with various settings.
Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis presents an approach to providing insight into long-term Web interaction factors. I hypothesise that users’ behaviour changes over time and that this change is reflected in the way they interact. Therefore, captured interaction events have the potential to be employed as a proxy for higher-level factors of user interaction, such as a user’s degree of familiarity with the Web site or application. As opposed to coarse Web interaction information, low-level interaction data provides the means to discover subtle changes in the way users interact.

Micro behaviours are presented as an appropriate way of aggregating interaction data captured over extended periods of time without disregarding fine-grained aspects of the observation. A longitudinal analysis methodology is presented to discover temporal correlations across the features from extracted micro behaviours. This methodology supports the discovery of subtle changes in interaction over time taking into consideration variability between users. Users’ degree of familiarity is explored to test if identified changes can be linked to higher-level concepts of user behaviour. Micro behaviours strongly correlated with users’ degree of familiarity have been discovered. These correlations have been tested by constructing a predictor of a user’s degree of familiarity. However, the application of the analysis methodology is complex. The need to appropriately prepare the input data and the variety of possible visualisations makes the use of the analysis methodology challenging for non-experts – programming knowledge is required. A Web application guiding the process of data preparation and providing the resulting visualisations in an interactive manner has been implemented. This tool provides easy introspection into long-term factors of user interaction filling
the existing gap in available tools for analysis of user behaviour.

The work presented offers additional possibilities. The high level of detail of the captured data allows recreating user interaction accurately possible. Drill-drown into particular interaction episodes is possible, providing a deeper understanding of user behaviour. Unlike laboratory studies, unobtrusive remote studies are highly scalable but lack insight into user’s motivation. The combination of both remote and laboratory studies help to understand discovered phenomena. The use of micro behaviours is not limited to longitudinal approaches. Analysis of subtle changes in interaction between users or set-ups is possible. Studies comparing differences in low-level interaction can provide insight into additional high-level interaction factors, such as consistency.

**Low-level interaction recreation**

Implemented capture solution captures a wide range of Web interaction events, making the recreation of a particular user’s interaction possible. Such recreation is possible through existing techniques, such as the original UsaProxy [Arroyo et al., 2006]. An external viewer can be registered to the interaction session being captured by the tool, and can visualise the participant’s interaction as it takes place. The capture solution implemented in Section [3.1] captures the state of the Document Object Model (DOM) of the Web page, and subsequent modifications to it. Instead of live observation of a single participant’s interaction, the necessary information is stored for a future recreation.

Longitudinal analysis is supported, enabling the comparison of various interaction sessions at different points in time. Storage of the DOM states of the pages provides a history of the changes in the Web site. Taking into account changes in the DOM state helps to take into account the effect changes in the Web site have in user interaction.

**Combination with laboratory studies**

The presented approach helps understand users from a longitudinal perspective. Existing laboratory approaches provide valuable insight into users’ behaviour and detailed information about the interaction not possible through unobtrusive remote approaches. I believe existing qualitative approaches can be combined with the presented approach to providing greater insight into long-term factors and how a user’s Web interaction changes over time.
6.1. CONCLUSIONS

For example, in the case of this study, micro behaviours exhibiting consistent temporal correlations have been identified. For each micro behaviour, the expected correlation has been described. These expectations have only been supported for the case of the Controlled Scroll Speed micro behaviour. The rest of consistent micro behaviours contradicted the initial expectations. As opposed to my initial expectations, episodes get longer over time, and mouse inactivity increases. Possible reasons for these results have been explored. In the case of episode duration, interaction data is split using a timeout of 40 minutes. If a user comes back to the same Web page before this 40 minute timeout ends, the following interaction is considered as part of the same episode. Therefore, one possible explanation for episodes getting longer over time is that a user comes back to a familiar page more frequently, to use it as a reference. To verify this explanation, interaction data from the particular users exhibiting this temporal correlation can be analysed. Captured interaction data is fine-grained so individual users interaction can be recreated. However, the motivation for this behaviour can only be inferred by asking the user.

In the case of mouse inactive time increasing over time, one possible explanation is the need to hover over particular elements to disclose additional interaction. As users get more familiar with the page, the need to disclose this information decreases. Instead, the user needs to remember where the desired interaction option is located. Once again, the interaction can be fully recreated to explore if users’ interaction conforms to this possible explanation, but users’ motivation remains unknown.

Further analysis of the discovered temporal correlations through existing controlled techniques, such as laboratory studies, enables the collection of user feedback. Temporal correlations of consistent micro behaviours have been shown to remain consistent for smaller samples – see Section 4.3. Even if the number of available participants is limited in laboratory studies, I expect these results to be replicable.

**Insight into additional high-level interaction factors**

Familiarity has been chosen as an appropriate high-level interaction factor to analyse in this thesis. Familiarity has a temporal aspect, and current analysis techniques fail to provide a practical approach to analyse it. Micro behaviours can be employed to provide insight into additional high-level factors. For example, insight into the effect of consistency can be obtained by comparing the subtle differences in user interaction between two interfaces. These interfaces can be designed to vary according to varying usability factors, such as simplicity, or self-description. Combining the use of micro
behaviours with this A/B testing approach supports a scalable analysis the selected factor has on user interaction.

Supporting design hypothesis

Supporting particular behaviours is also possible. For example, designers might assume that mouse click time gets faster over time as the user get more familiar with their design. The use of the presented approach could help them discover that the mouse click time gets slower over time – or shows no change at all, as in the Web site examined in the presented study – and they could decide the rate of adoption of the interface can be improved by supporting faster click times. Interpretation of the strength of the correlations helps to discover the set-ups that support a faster adaptation to the interface. In the mentioned example, designers would look for interaction supporting a shorter transition to quicker click times – i.e. reported medians and quartiles of the correlation distributions are further away from the origin. Designers’ assertions can be easily tested supporting an iterative design of the interfaces based on the real usage of the Web sites or applications.

6.2 Limitations

I acknowledge the presented approach has limitations. If compared with traditional laboratory studies, the main limitation is the lack of control over the observed environment. Observing a physical participant provides a degree of awareness over the context not possible through remote observation techniques. Still, if compared with Web logs, the observation solution provides a finer collection of data. Taking advantage of the continuous interaction capture, the metric of active time is introduced, which provides a more precise metric of users’ interaction experience with the interface.

Micro behaviours provide the means to compare fine-grained interaction data over extended periods of time. The presented analysis methodology requires a high occurrence of extracted micro behaviours to make the longitudinal analysis viable. In the cases where the occurrence of these micro behaviours is not high, alternative approaches can be employed. Qualitative analysis of occasional micro behaviours is possible, and the interaction surrounding each particular micro behaviour can be analysed to obtain additional context information.
6.3 Future Work

Opportunities for additional research have been identified throughout the work performed in this thesis. Additional analysis of data captured from various Web sites will help determine if results from the analysis of the extracted micro behaviours are generalisable. Although interaction events specific to mobile devices have been captured analysis has been limited to desktop Web browser events to maximise the available interaction data. Finally, the fine granularity of the captured data provides the means to analyse additional longitudinal factors such as memory effects.

Extend analysis The implemented observation tool has been deployed in a Web site – cs.manchester.ac.uk – and a Web application kupkb.org. The proposed longitudinal analysis could not be applied to the Web application due to the low number of visitors this particular Web application received. Deploying the observation tool on other real world Web sites and Web applications would provide the means to test if the obtained results are generalisable.

Mobile usage analysis Mobile interaction differs greatly from desktop Web browser interaction. The mobile interaction may include complex gestures and specific ways to scroll pages. Analysis of how this usage changes over time requires the design of mobile specific micro behaviours. Mobile events are captured by the implemented observation solution, but they are not considered for the analysis. Micro behaviours specific to interaction with mobile devices can be designed based on the captured mobile events. Proposed longitudinal analysis methodology can then be applied to the resulting mobile micro behaviours in the same manner reported in Chapter 4.

Analysis of mobile events conveys additional challenges. The design of the capture solution aims to maximise the variety and frequency of captured events to support a wider range of analysis options. In the case of mobile sensors, their frequency has been found to be too high to be captured directly. Sensor data has been aggregated thus conditioning prospective analysis – see Section 3.2. The limitations of these aggregations need to be explored, and an optimal trade-off between granularity and information overload needs to be found.

Scalable recreation of low-level interaction Recreation of low-level interaction has been considered a suitable approach to drill-down into particular interaction episodes. Greater detail of a user’s interaction can be obtained through manual observation of
the low-level events, or recreating the interaction rebuilding its context from the corresponding DOM. This process is cumbersome, as it requires visualising individual users’ interaction. Existing scalable visualisations, such as heatmaps, aggregate interaction data from users masking underlying temporal aspects.

Prospective future work can focus on providing a scalable alternative. Visualisations supporting longitudinal analysis are possible, using the collected history of DOM states. The combination of various visualisation techniques would be necessary to provide insight into the multiple dimensions of the captured data. For example, heatmaps for different spatial dimensions of the data – such as where users clicked on the page – with Markov chains and timeline visualisations to describe fine-grained temporal interactions. The use of micro behaviours enables the visualisation of low-level interaction units. The design of additional micro behaviours would allow to quantify users’ interaction episodes and ease the visualisation by reporting these metrics.

The visualisation of interaction involving multiple tabs poses an additional challenge to these visualisations. A combination of fine-grained interaction data along with the navigational paths of users provides insight into tabbed browsing, but scalability is a concern.

**Memory effects** Previous work has explored the effect of relearning [Kim and Ritter, 2015] for mouse and keyboard related tasks, studying performance after controlled periods of inactivity. The effect of transfers between tasks has been found to depend on the kind of task and its level of difficulty [Gerken et al., 2009]. The addition of the lengths of the periods between episodes to the analysis helps to determine the effect inactivity has. Section 5.3 suggests the creation of an additional model including the time between episodes. This way the accuracy of the resulting models can be compared, to determine if prolonged periods of inactivity have an effect in how the interaction changes over time. The results of the application of this technique have not been tested, and no results are reported for it.

Further analysis of the effect memory factors have in the interaction enables a prediction of users’ recollection of the page. Interaction of users identified as novice, or with a low degree of familiarity, can be supported offering hints and tutorials. In the case of expert users or with a high degree of familiarity who return after an extended period, the interaction can be supported to refresh their existing knowledge about the interaction possibilities.
Appendix A

Additional Models

A.1 Survey participants

The following figures correspond to the analysis of micro behaviours extracted from the participants of the familiarity survey presented in Section 4.3.

Survey Participants Mouse Inactive Time Median

Figure A.1: Analysis of Mouse Inactive Time Median for the survey participants. Limited to users with at least 2 occurrences.
APPENDIX A. ADDITIONAL MODELS

**Survey Participants Mouse Inactive Time Mean**

![Graph showing the analysis of Mouse Inactive Time Mean for survey participants. Limited to users with at least 2 occurrences.]

Figure A.2: Analysis of Mouse Inactive Time Mean for the survey participants. Limited to users with at least 2 occurrences.

**Survey Participants Mouse Time To Click**

![Graph showing the analysis of Mouse Time To Click for survey participants. Limited to users with at least 2 occurrences.]

Figure A.3: Analysis of Mouse Time To Click for the survey participants. Limited to users with at least 2 occurrences.
A.1. SURVEY PARTICIPANTS

Survey Participants Mouse Click After Load

Figure A.4: Analysis of Mouse Click After Load for the survey participants. Limited to users with at least 2 occurrences.

Survey Participants Mouse Move Distance

Figure A.5: Analysis of Mouse Move Distance for the survey participants. Limited to users with at least 2 occurrences.
APPENDIX A. ADDITIONAL MODELS

Survey Participants Controlled Scroll Duration

Figure A.6: Analysis of Controlled Scroll Duration for the survey participants. Limited to users with at least 2 occurrences.

Survey Participants Controlled Scroll Speed

Figure A.7: Analysis of Controlled Scroll Speed for the survey participants. Limited to users with at least 2 occurrences.
A.2 Homepage excluded

The following figures correspond to the analysis of different combination of pages from the cs.manchester.ac.uk Web site. The homepage of that site has a particularly high count of visitors compared to the rest of the pages on the site. For this analyses the homepage was removed so the effect of the inclusion of that page in the model could be observed. Interaction data from phase 1 and phase 2 has been analysed – see Section 4.4 –, and results from all pages on the site, and from the pages considered similar to the homepage – see Section 3.1.3 – are reported.
Figure A.9: Analysis of Mouse Click Speed Time excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.10: Analysis of Mouse Median Inactive Time excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.11: Analysis of Mouse Mean Inactive Time excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.12: Analysis of Mouse Time To Click excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.13: Analysis of Mouse Idle After Click excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.14: Analysis of Mouse Click After Load excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.15: Analysis of Mouse Move Distance excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Controlled Scroll Duration

Phase 1 Homepage Cluster excluding homepage

Phase 1 All pages excluding homepage

Phase 2 Homepage Cluster excluding homepage

Phase 2 All pages excluding homepage

Figure A.16: Analysis of Controlled Scroll Duration excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.17: Analysis of Controlled Scroll Speed excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Figure A.18: Analysis of Episode Duration excluding homepage. Limited to users with at least 2 occurrences and 2 minutes of Active Time.
Bibliography


BIBLIOGRAPHY


Communication: Documenting & Designing for Pervasive Information, SIGDOC ’05, pages 151–158. ACM.


