W3Touch
Crowdsourced Evaluation and Adaptation of Web Interfaces for Touch

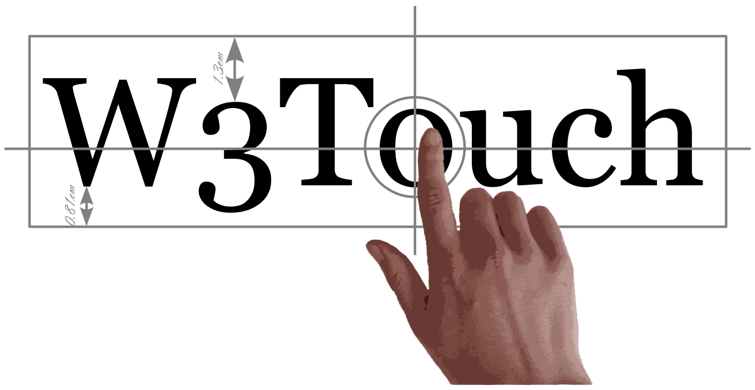
Master Thesis

Author(s):
Speicher, Maximilian

Publication date:
2012

Permanent link:
https://doi.org/10.3929/ethz-a-007139741

Rights / license:
In Copyright - Non-Commercial Use Permitted



W3Touch
Crowdsourced Evaluation and Adaptation of Web Interfaces for Touch

Master's Thesis

Maximilian Speicher
<speichem@student.ethz.ch>

Prof. Dr. Moira C. Norrie
Michael Nebeling

Global Information Systems Group
Institute of Information Systems
Department of Computer Science

1st March 2012

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

globis

Abstract

The increasing range and diversity of novel touch devices makes it cumbersome for web developers to cater for the wide variety of browsing contexts. While it is work-intensive to create mobile versions of a website, non-optimised websites that were designed with desktop computers in mind are usually scaled by mobile browsers so that text is rendered unreadable and details are difficult to view. Moreover, hyperlinks typically yield too small touch areas, as many interfaces are not designed for novel input modalities. Interacting with such pages in an effective manner therefore requires extensive zooming and scrolling, which can become major usability issues.

This thesis presents *W3Touch*—a lightweight website plug-in making use of crowdsourcing to collect context-aware activity data specific to touch input. Based on this data, potentially critical webpage components and suitable adaptations for different device characteristics are inferred. Thanks to W3Touch, developers are provided with visualisations of these data for identifying interface design issues related to touch input as well as an adaptation catalogue for defining and fine-tuning adequate adaptations.

In two user studies, we investigate W3Touch's capabilities to help providing touch-optimised web interfaces. Results suggest that crowdsourced interfaces do not only improve user experience, particularly on small-screen mobile touch devices, but can also compete with existing mobile websites to a certain extent. Moreover, a set of touch-specific usability metrics is introduced that can complement the features of W3Touch for assessing web interfaces with respect to touch input.

Acknowledgements

First of all, I would like to thank *Prof. Moira Norrie* and *Michael Nebeling* for their support and giving me the opportunities to, not only write this thesis at the GlobIS group, but also participate in projects that were very interesting and challenging to work on. Moreover, I would like to thank Michael Nebeling again for being a demanding but fair supervisor and for his valuable feedback.
This Master's thesis is dedicated to *my family* and *grandma*, who always supported me throughout my studies, and especially *my parents*, who believed in the plans I've had and never said "No!" when I came and wanted to change university (again). Furthermore, this thesis is also dedicated to *Dénise*, who always supported me ever since we met and came with me to Zurich to study at ETH♡
Finally, I would like to thank *Tim Church* for his good company throughout the three semesters at ETH and for all the pizza days.

Contents

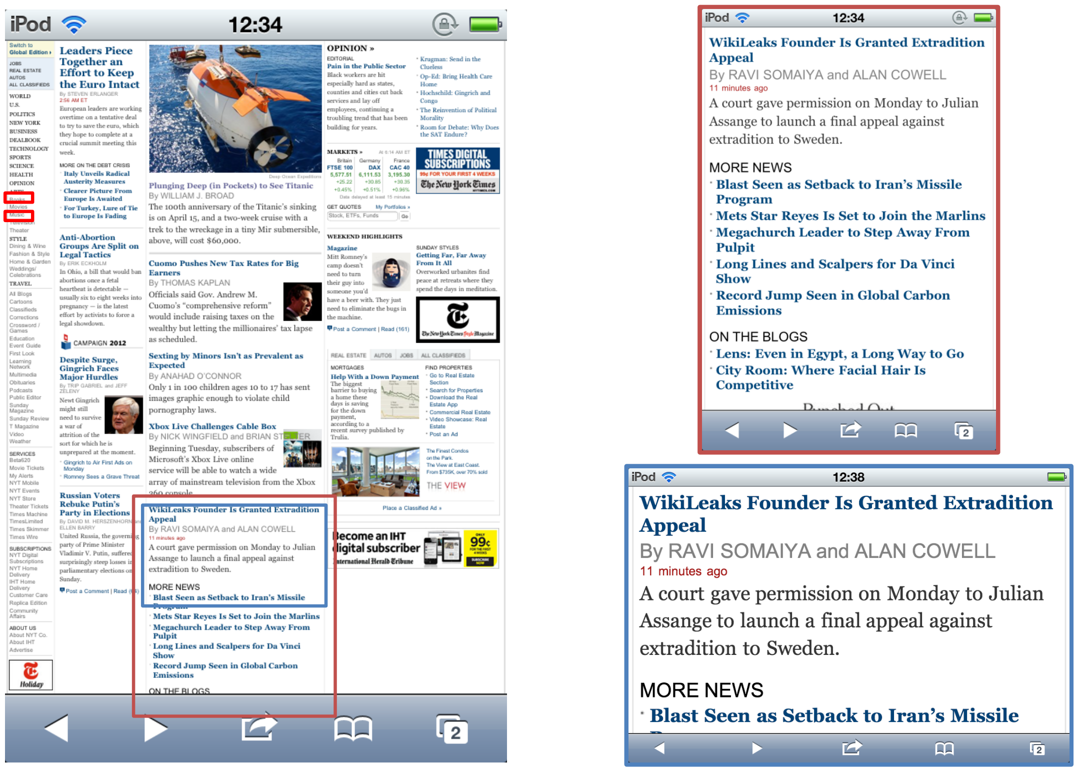
1 Introduction

Nowadays, we experience an increased proliferation of novel mobile devices—such as smartphones or tablet computers—featuring touchscreens for operation. The most prominent examples for these might be Apple's iPhone and iPad, which have been sold more than 146 million times (since April 2007) and more than 39 million times (since April 2010) respectively, according to official press releases.[1] But also other devices, e.g. those running the Android OS, are becoming more and more popular, which adds to the great diversity. Since this new generation of touch-operated devices typically offer Internet connectivity, e.g. via WiFi or UMTS, and also feature a range of browsers including Safari Mobile, Mobile Firefox or the Dolphin Browser, the user is able to access arbitrary websites from practically anywhere. However, the mobility and diversity of devices poses numerous challenges to web developers, not only in terms of the smaller screen real estate available, but also in terms of the novel input modalities and generally many differences between devices.
The majority of websites still appears to be designed with desktop computers in mind, i.e. they are optimised for keyboard and mouse input and feature column widths oriented at popular desktop screen dimensions (in particular 1024×768; Nebeling et al., 2011). Concerning the adaptation of such websites for mobile devices, Bickmore and Schilit (1997) mention four general approaches:

- *Device-specific authoring,* which means designing a page for a specific (family of) device(s). Current examples are dedicated *apps* for different devices and mobile operating systems or special, separate versions of a site (Charland and LeRoux, 2011). Legacy examples include technologies like WML[2] or cHTML[3].

- *Multiple-device authoring,* where a single source document is mapped to a range of different devices, as can be done with the help of CSS3 media queries[4] (used by frame-

[1]`http://www.apple.com/pr/library/`
[2]`http://www.wapforum.org/what/technical.htm`
[3]`http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/`
[4]`http://www.w3.org/TR/css3-mediaqueries/`


iPod 12:34

WikiLeaks Founder Is Granted Extradition Appeal

By RAVI SOMAIYA and ALAN COWELL

11 minutes ago

A court gave permission on Monday to Julian Assange to launch a final appeal against extradition to Sweden.

MORE NEWS

- Blast Seen as Setback to Iran's Missile Program
- Mets Star Reyes Is Set to Join the Marlins
- Megachurch Leader to Step Away From Pulpit
- Long Lines and Scalpers for Da Vinci Show
- Record Jump Seen in Global Carbon Emissions

ON THE BLOGS

- Lens: Even in Egypt, a Long Way to Go
- City Room: Where Facial Hair Is Competitive

iPod 12:38

WikiLeaks Founder Is Granted Extradition Appeal

By RAVI SOMAIYA and ALAN COWELL

11 minutes ago

A court gave permission on Monday to Julian Assange to launch a final appeal against extradition to Sweden.

MORE NEWS

- Blast Seen as Setback to Iran's Missile

Figure 1.1: A website designed for desktop computers, as seen on a small-screen touch device. The screenshots on the right show viewports in portrait and landscape mode, when zoomed in for a readable size. The tiny red boxes at the very left indicate the touch areas of exemplary text hyperlinks.

works such as *Foundation*[5]) as well as the entire range of model-based approaches (e.g. Paternò et al., 2009).

- *Client-side navigation,* which offers users the possibility to interactively alter the portion of a website that is displayed. Examples are Fishnet (Baudisch et al., 2004a), where focus regions of a page are scaled to a readable size, and Collapse-to-Zoom (Baudisch et al., 2004b), where users can collapse parts of a page so that the remaining contents are given more space.
- *Automatic re-authoring,* where a website designed for desktop computers is dynamically transformed in a way that it can be appropriately displayed on a certain target device, taking into account given device characteristics. This has been realised in systems like PUC (Nichols et al., 2002) or Supple (Gajos et al., 2008).

However, in practice, only the first two of these principles—i.e. creating specific mobile versions of a website or using multiple stylesheets for one website that are selected based on the current context—are commonly used by developers to also cater for mobile and touch-based browsing contexts. But since it is cumbersome and time-consuming to maintain different

[5] `http://foundation.zurb.com/`

versions of the same website or to develop pages with several stylesheets, the websites to offer versions that are specifically optimised for touch-operated mobile devices are mostly popular websites with sufficient resources, such as *Facebook*[6], *Twitter*[7] or *FAZ*[8]. In contrast, the vast majority of regular websites are usually scaled by mobile browsers to fit the width of the screen, thus often rendering text unreadable (cf. Figure 1.1). Moreover, text hyperlinks are commonly not defined with sufficient horizontal and vertical spacing and, if turned into touch-sensitive areas without further adjustments, result in targets that are difficult to hit on touch-based small screens when using a finger instead of a mouse or pen (cf. Figure 1.1). Using such websites in an effective manner therefore involves huge amounts of zooming as well as both horizontal and vertical scrolling, which are common examples of client-side navigation, but require considerable effort by the user and can lead to a bad website overview. This is underpinned by a survey concerned with browsing on touch devices and carried out by Nebeling et al. (2012b), where participants pointed out that touch input is often inaccurate because of typically too small input elements, websites are often not optimised for touch and there are issues regarding the amount of scrolling and zooming necessary. All of these problems account for both viewing and reading contents as well as navigating through the website, e.g. being able to easily find and activate desired hyperlinks.

1.1 Aims of Research

One new way of addressing the challenges for web developers resulting from the proliferation of novel and diverse mobile touch devices is to use a crowdsourcing approach (Nebeling and Norrie, 2011). In this case, lightweight end-user tool are proposed that are part of a dedicated extension to common web application architectures and offer users the possibility to adjust websites to their needs. That means, the web developer provides an initial web interface that evolves over time through user emancipation while taking into account the different browsing contexts the website is accessed with. Therefore, developers do not need to maintain different versions of webpages or stylesheets since different adaptations are stored for different use contexts (e.g. *tablet PC* or *smartphone*) and applied automatically for users of according devices.
However, regarding the fact that this thesis focuses on touch-operated devices with smaller screens than desktop computers, on which it is already considerably difficult to only interact with regular websites, we present *W3Touch*—a lightweight website plug-in based on crowdsourcing ideas that is meant to complement the work by Nebeling and Norrie (2011). Our tool aims at automatically inferring required adaptations for different browsing contexts based on the users' behaviours rather than providing means for direct manipulation, which would be difficult to use on small-screen touch devices. Therefore, the aims of this thesis are as follows:

1. Develop a set of methods for collecting input data relevant for web browsing on touch devices, such as taps and zoom gestures. These data shall be used to identify the most important interaction components of a website as well as to aid developers identify key design issues based on analyses and visualisations.

[6]`http://m.facebook.com/`
[7]`http://mobile.twitter.com/`
[8]`http://m.faz.net/`

2. Conduct expert interviews in order to gain a better understanding of potential usability issues when browsing regular websites on touch devices and to inform the design of adequate adaptations.

3. Develop a set of context-aware layout adaptation techniques based on main interaction components and tracking data, taking into account recommendations gained from the expert interviews.

4. Design and carry out user evaluations in order to show that the developed adaptation techniques can be successfully applied to existing websites. Goals are to identify the main limitations and to demonstrate usability improvements over regular websites concerning tasks that are potentially cumbersome to perform on touch devices. Additionally, take into account existing mobile versions for comparison.

5. Develop a set of specific metrics based on *jQMetrics* (Nebeling et al., 2011), that support developers with assessing websites on mobile touch devices.

1.2 Structure of this Thesis

This thesis starts in Chapter 2 by discussing the background of this project and reviewing related work. Next, in Chapter 3, we describe the concepts and implementation of W3Touch—the proposed tool for automatic usability evaluation and context-aware adaptation based on user activity tracking. Chapter 4 describes expert interviews that give insight into potential usability issues of current websites and how they could be addressed as well as a user study that has been carried out to evaluate W3Touch. Moreover, the adapted website based on usage data collected during the study is compared to an existing mobile and touch-optimised version. This happens in terms of a second user study and specific metrics. Finally, we provide concluding remarks in Chapter 5.

2 Background

W3Touch aims at the adaptation of web interfaces for mobile devices featuring touch input. The application of context-aware adaptations shall thereby be based on a crowdsourcing scenario and automatic usability evaluations involving client-side interactions and components of interest inferred from these. Therefore, this project is related to a wide variety of existing research that has been focusing on the corresponding aspects of adaptation and evaluation as well as similar topics. The following chapter gives background information about the specifics of touch input and the principles of crowdsourcing. As well as providing a structured overview of existing solutions and propositions dealing with adaptation to mobile devices and usability evaluation, we point out differences to the approach presented in this thesis.

2.1 Touch Input

In addition to small screens, mobile devices nowadays often feature direct touch input. Since interacting with websites using one or more fingers involves novel gestures and paradigms, it is significantly different from mouse or pen input and thus poses specific requirements to web developers. In particular, while it is possible to precisely target a link using a mouse or pen, fingers produce a much larger *contact area* on a touchscreen (Holz and Baudisch, 2011). Today's capacitive touchscreens, as e.g. used for the iPhone, assume that the centroid of this area is the intended target, but users rather align visual features on the top of their finger with the link to be touched, which leads to offset errors (Holz and Baudisch, 2011). These errors—with an average offset of 4 mm—are especially problematic if a user faces areas with a huge number of densely packed, small links. To give an example, on an iPod touch with a resolution of 163 pixels per inch[1], these 4 mm (0.157 in) correspond to approximately 26 pixels, which is already as large as font sizes commonly used on websites (e.g. 12 pt $\approx$ 27 px, at 163 pixels per inch), or even larger if a website has been scaled to fit the width of a small screen (cf. Figure 1.1).

[1] `http://support.apple.com/kb/sp496`

Besides these conceptual differences regarding input modalities, direct touch also requires changes concerning the implementation of websites. Since it is, e.g. not possible to hover elements using a finger, the `mouseover` and `mouseout` DOM events are not suitable for touch input. This means that developers cannot rely on interactions building on such mouse-specific events, but rather have to employ novel events, including `touchstart`, `touchmove` and `touchend`, as proposed by the W3C[2]. The *Safari API*[3] introduces additional events to be able to easily handle higher-level gestures as well. Additionally, since touch input also triggers default browser behaviours, such as scrolling and zooming, developers have to ensure that these do not interfere with any events or gestures that are necessary for interaction with the web interface as well as to guarantee that no ambiguous gestures exist (cf. Nebeling and Norrie, 2012).

2.2 Crowdsourcing

Rather than *out*sourcing tasks to dedicated employees in a company, *crowd*sourcing (Howe, 2006) means involving a huge number of (unknown) people—typically over the Internet—in solving tasks through open calls. Howe (2006) describes crowdsourcing as a result of the increased connectedness of people, which leads to huge amounts of contributions created for fun or in one's free time, which are of good enough quality—as compared to a professional's work—at a significantly lower price. In this way, it is e.g. possible to purchase a high-quality amateur photograph at *iStockphoto*[4] for $1, instead of paying a professional a hundred times the amount. Even R&D departments of companies can make use of crowdsourcing by posting unsolved problems to a platform called *InnoCentive*[5], where anyone can try to contribute to these research tasks. Although solvers receive comparably high payments ($10,000 upwards), this is still cheaper for companies than costly in-house solutions (Howe, 2006).
In the context of computer science, crowdsourcing is especially useful for small tasks that are difficult for computers, but can be easily solved by humans, such as photo recognition or transcriptions (Howe, 2006). To give a famous example, Amazon took advantage of this by developing *Mechanical Turk*[6]—a platform that is less specialised than the above mentioned iStockphoto and InnoCentive and aimed at a larger amount of potential contributors. Anyone can post tasks (along with a reward) and ask the crowd to solve them, while an API even makes it possible to receive solutions programmatically. Because of this "simulation" of computing power by crowdsourcing small problems from within running programs, as e.g. done in *Soylent* (Bernstein et al., 2010), Amazon call their approach "Artificial Artificial Intelligence". While the described platforms realise explicit forms of crowdsourcing, it is as well possible to employ it in an implicit way. As described in Section 3.1, W3Touch collects interaction data and informs the design of suitable adaptations from these. Therefore, interacting with a website can be seen as the crowd's contribution to solving the problem of adapting it for different browsing contexts.

[2]`http://www.w3.org/TR/touch-events/`

[3]`http://developer.apple.com/library/IOs/#documentation/AppleApplications/Reference/SafariWebContent/HandlingEvents/HandlingEvents.html`

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

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

[6]`https://www.mturk.com/mturk/welcome`

An example for a crowdsourcing solution whose architecture is similar to the one proposed by Nebeling and Norrie (2011) is *Adaptable GIMP* (Lafreniere et al., 2011), which supports the customisation of the *GIMP*[7] user interface for specific tasks, such as creating a sepia effect. However, the major difference to crowdsourcing web interfaces is that the use context (i.e. the task a GIMP user wants to perform) cannot be determined automatically. Therefore, the user has to use a search form to find and choose an adequate customisation for their desired task before applying according adaptations.

2.3 Adaptation of Websites for Mobile Devices

Existing techniques concerning the adaptation of websites to mobile devices can be roughly divided into three categories. The first one includes automatic adaptation of actual layouts shown to the users as well as the generation of more device-specific layouts from these. The second category comprises of techniques that allow users to directly adjust web interfaces to their needs, while solutions extending the input capabilities of the user rather than focusing on actual layouts fall into the third category.

2.3.1 Automatic Layout Adaptation/Generation

Summarisation

One approach to changing layouts in order to provide a better user experience on small-screen devices is based on *summarisation* techniques. Buyukkokten et al. (2000, 2002) have developed solutions for pen-based PDAs that generate single-column summary views taking into account link structures of websites and semantic units within pages, thus giving users a better overview of a website when searching for specific pieces of information. Users can then “zoom into” more details if necessary. Although an improvement in user experience in terms of less scrolling, less pen movements and better browsing speed Buyukkokten et al. (2000) as well as better information discovery Buyukkokten et al. (2002) has been shown, these approaches do not reflect the complexity of today’s websites and devices, e.g. images are ignored to cater for monochrome displays.
In contrast to generating summary views, Lam and Baudisch (2005) propose “summary thumbnails”, which means optimising the *thumbnail view* users see when a website is zoomed out beyond readability in order to fit the available width on small screens (cf. Figure 1.1). This means that font sizes of text fragments are increased, while the larger text is then shortened to preserve the original line count. This can lead to lower error rates (as compared to single-column views) and less zooming (as compared to original thumbnail views), while not significantly increasing vertical scrolling efforts. However, user tests showed that, while participants preferred summary thumbnails for tasks relying on the layout of a page (e.g. maps or schedules), they did not for linear reading tasks (e.g. news websites or search results). Moreover, since this approach is focused on adjusting text for small screens, it does not take into account issues with novel input modalities, such as direct touch (cf. Section 2.1).

[7]`http://www.gimp.org/`

Segmentation

Chen et al. (2003) propose to *segment* websites rather than generating summaries of different types. To achieve this, they split a page into semantically related units that fit the small screens of mobile devices, taking into account visual separators and assumptions about commonalities in the structures of different websites. The user is then presented a standard thumbnail view highlighting the identified content blocks with different colours and can zoom into those units. However, although this approach shows improvements in navigation and reading efficiency, it does neither reflect the increasing complexity of today's websites nor does it take into account specific issues related to touch input (cf. Section 2.1)—as already mentioned for the above solutions focusing on summarisation.
Hattori et al. (2007) propose a similar approach to segment a website, but employ a different technique to discover semantically related content units. In particular, they use a hybrid solution that combines a higher-level structural analysis of the layout with lower-level segmentation via "content distances", i.e. a method based on "the strength of connections between content elements of the Web page based on the structural depth of HTML tags" (Hattori et al., 2007), that is more robust against invalid HTML. Moreover, instead of providing a thumbnail overview, they rely on "title lists" that are generated from the page segmentation results. Regarding the similarity of this approach to the one proposed by Chen et al. (2003), it accounts for the same strengths and has similar weaknesses.

Interaction-based

In contrast to all of the approaches described above, Leiva (2011) proposes to infer web interface adaptations from user interactions and thus describes the approach that is most similar to W3Touch so far. The developer of a website has to define page elements and associated CSS properties that shall be adapted based on a user's activities. These properties are then adjusted according to a given function that calculates a weighting based on touch interactions with the respective page element. For example, a weighting of 0.2 would increase the font size of an element by 20%. However, the proposed solution also shows several differences as compared to W3Touch. First of all, the gathered tracking data relies on hovering (which is not even available with touch) and clicking elements, while not taking into account other interactions that might indicate usability issues, such as zooming gestures. Second, adaptations are stored individually for each user, thus not making use of crowdsourcing techniques or propagating adaptations to users with similar browsing contexts. And third, since interaction data are not expressed as metrics, but rather translated into a list of page elements (ordered by browsing time), and the same given weighting function accounts for all adaptations, the developer can only define numerical CSS properties that shall be adapted, but not in which way. In particular, this stands in contrast to W3Touch's *adaptation catalogue* described in Section 3.1.4, that provides developers access to interaction metrics for individual elements as well as with richer means for influencing actual adaptations.

2.3.2 Direct Manipulation

In contrast to automatic methods, direct manipulation techniques allow users to adjust websites to their specific needs and preferences by taking advantage of end-user tools provided on

the client side. In this way, users can, e.g. move, remove and resize page elements or change the font sizes of certain paragraphs of text.
Bila et al. (2007) propose a technique called REUC (Reusable End-User Customisation) that has led to a prototype application developed for PDAs. It provides users a toolbar within their mobile browser, allowing them to manipulate parts of the displayed webpage. The customisations are then applied on subsequent visits to the same or similar pages within the same website. Similarity between pages is thereby determined based on commonalities in DOM structure. User tests have shown that customisations are reasonably accurate and can be successfully reapplied to an average of 75% of pages of the same website which had been found to be similar.
Unlike the approach just described, Nebeling and Norrie (2011) propose a more general framework taking advantage of crowdsourcing techniques. It also employs visual tools that give users the opportunity to create context-aware adaptations by, e.g. collapsing content or changing the column count of text passages. These adaptations are then shared among users with similar browsing contexts, involving a review and rating system to cater for quality.
However, as already mentioned in Section 1.1, this thesis aims at automatically inferring adaptations from users' activities since visual tools for direct manipulation would be cumbersome to use on touch-operated devices with small screens, where it is already difficult to interact with non-optimised websites. Nevertheless, we make use of parts of the framework proposed by Nebeling and Norrie (2011) since it describes useful principles to share context-aware adaptations among users as well as sharing many similarities in implementation.

2.3.3 Extended Input Techniques

In contrast to adapting existing interfaces or generating new ones from these while relying on available input techniques, other research has focused on extending input capabilities to improve efficiency and effectiveness when working with standard, unadapted websites. This can be achieved by, e.g. introducing novel paradigms for navigation (Baudisch et al., 2004b) or multi-step selection techniques (Watanabe et al., 2011).
Baudisch et al. (2004b) have developed Collapse-to-zoom, a tool that enables users to collapse irrelevant tiles of a webpage, while the remaining contents expand in size to fill the newly available space. To be able to perform this, users are presented a so-called "marquee menu" that consists of simple single-stroke gestures also allowing, e.g. to directly expand certain parts of a webpage to fill the whole browser window. When re-accessing a webpage, it is then presented in the state at which it was bookmarked. While Collapse-to-zoom can help users to identify relevant parts of a webpage, it has been developed with pen-based devices in mind, which makes it possible to offer the convenient marquee menu. However, when applying this concept to mobile devices featuring direct touch input, the single-stroke gestures could interfere with other standard gestures, such as scrolling and zooming. Moreover, it would be more difficult to collapse and expand desired tiles of a webpage using a finger, given that they might be too small and a pen is the more precise input method.
Watanabe et al. (2011) follow an entirely different approach by showing enlarged versions of text hyperlinks when those are being touched by a user. This gives users the possibility to ensure that the selected link is correct before releasing their finger to navigate to the link target or moving their finger away to cancel the interaction. Although this solution can decrease the number of erroneously clicked links, it does not directly address the problem of unread-

able text and that of touch areas that are too small. In particular, a high number of densely packed (and maybe too small) links would still require a decent amount of time-consuming and inefficient "trial-and-error" swiping.
In general, the above and similar methods usually treat issues related to direct touch as a matter of input technique rather than of web interface design, which constitutes a major difference to the approach presented in this thesis.

2.4 Automatic Usability Evaluation

Since the adaptations applied by W3Touch shall be inferred from user interactions, our system builds on existing methods for automatic usability evaluation (Atterer et al., 2006; Carta et al., 2011). The following shall give a rough overview of some approaches in this field, which we divide into client-side and server-side solutions.

2.4.1 Client Side

Client-side approaches to usability evaluation focus on users' interaction with web interfaces. However, in this way it is only possible to evaluate individual webpages. In case the tracking of a user shall span several pages of a website, additional server-side components (e.g. session management) are necessary.
Atterer et al. (2006) propose JavaScript-based interaction tracking to ease the realisation of usability tests, i.e. they gather mouse movements, clicks, keyboard input, elapsed time etc. using a proxy server, so that no modification of an investigated website is necessary. In this way, neither specific machines have to be set up for user tests, nor is it necessary to invite participants or make use of cameras to collect information. The gathered events can then be mapped to the respective page elements, which makes it possible to visualise, e.g. interaction paths of users on a webpage and identify main components of interest. Although this approach provides developers with useful data, it is not possible to automatically infer implicit information from these, as is intended by W3Touch. In addition, possible adaptations based on issues identified from the data have to be applied manually. While Atterer et al. (2006) focus on evaluations on single webpages, Leiva and Vidal (2010) follow a different approach also based on client-side interaction. Their solution makes use of the collected data to form clusters of documents that trigger similar user behaviours. In this way it could be possible to identify usability issues on one document and apply the corresponding adaptations on multiple similar pages, which is, however, not considered in their work.
Carta et al. (2011) extend the principles described above by providing a complete framework for remote usability testing. Their approach involves "optimal" logs defined for specific tasks by the evaluator, which are then compared to the logs produced by participants. This solution enables evaluations across websites and captures standard mouse and keyboard events as well as additional *jQuery* events[8] and the touch and gesture events present in the Safari API. While novel input modalities are partly addressed here, the identification of usability issues—supported with timeline visualisations of user interactions—is left to the developer.
Although the above solutions may provide good starting points for the aims of this thesis, they still need to be enhanced with means for capturing higher-level touch events, such as scrolling

[8]http://api.jquery.com/category/events/

or zooming, as well as automatically inferring usability issues specific to touch input from the collected interaction data.

2.4.2 Server Side

Contrary to the above, Hong et al. (2001) present WebQuilt, that is a logging and visualisation tool based on server logs. Rather than tracking users' interactions with an interface, as possible with client-side solutions, this approach records the communication between client and server, i.e. navigation paths across websites, clicked links etc., which can then be visualised in a graph showing webpages as nodes and actions as edges. In this way, it is also possible to infer certain user interactions, such as clicking the browser's back button. Concerning the aims of this thesis, WebQuilt is of minor interest, since it is focused on detecting issues with the document structure of websites rather than issues related to touch input. Moreover, although the developers state that their framework is extensible, it has difficulties handling dynamic actions based on JavaScript, which is a great disadvantage given the increased popularity of libraries such as *jQTouch*[9] or *Sencha Touch*[10]—that are mainly based on dynamic scripting—for developing mobile web applications.

[9] `http://jqtouch.com/`
[10] `http://www.sencha.com/products/touch`

3 W3Touch

According to the aims defined in Section 1.1, this chapter presents W3Touch—a lightweight website plug-in that features context-aware tracking of user activity on touch devices, identification of main interaction components of webpages as well as the automatic inference and application of suitable adaptations for different browsing contexts based on the crowdsourced interaction data and an adaptation catalogue provided by the developer.

3.1 Concept

Conceptually, W3Touch consists of two high-level components (cf. Figure 3.1), i.e. 1) means for tracking user activity on the client side and storing corresponding input data in a server-side database, and 2) adaptive mechanisms aimed at fixing potential interface design issues at run time. The first component employs the crowdsourcing ideas presented by Nebeling

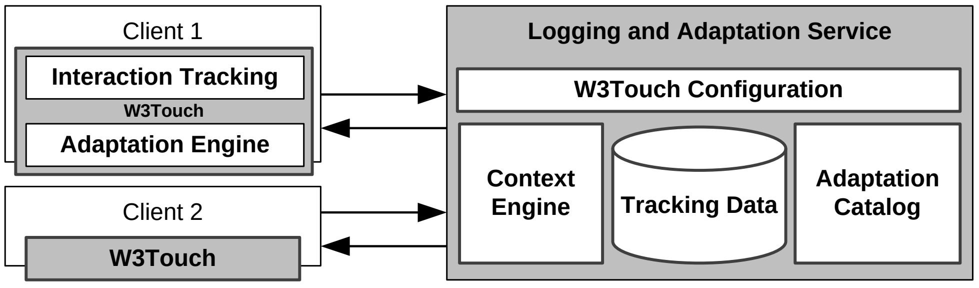


Figure 3.1: The W3Touch architecture, including core components and interaction between two clients indicating the underlying crowdsourcing scenario.

and Norrie (2011), extended with means for tracking touch-specific interaction on different mobile devices. The second component makes use of the gathered data—associated with context information—to support the inference of critical parts of webpages and automatically apply adaptations, which are based on a set of design rules defined by the web developer through an adaptation catalogue. The following will discuss the specifics of the W3Touch architecture in more detail.

3.1.1 Context-aware Interaction Tracking

The techniques for user activity tracking used by W3Touch are based on the principles described by Atterer et al. (2006). However, rather than relying on legacy DOM events designed for keyboard and mouse input, these have been enhanced to cater for the specifics of mobile touch devices. This means, in addition to considering raw touch events—as defined by the W3C and e.g. introduced by Mozilla[1]—our system supports semantically richer types of events. This is possible by keeping track of the position and dimensions of viewports above a given zoom level—as triggered by double-tap and pinch-to-zoom gestures—, scrolling gestures and orientation changes of the device. Moreover, while ignoring any single taps that belong to higher-level gestures, W3Touch records, not only taps on interactive elements of a webpage (e.g. hyperlinks and input fields), but is also able to recognise single taps that potentially missed their intended targets. To achieve this, our system prepares a webpage by adding *underlays* that span a specified range around interactive elements and record any taps which missed the corresponding hyperlink, input field or similar. Additionally, W3Touch can combine subsequent scrolls in the same direction in a single event, in order to reduce redundant information and the amount of data to be processed.
On the client side, all of the collected information are associated with the URL of the webpage as well as with the user's browsing context—including device orientation, screen dimensions and user agent string. Subsequently, the server-side context engine (cf. Figure 3.1) additionally infers the type of device (e.g. *tablet PC* or *smartphone*) and the data is sent to a database for future processing.

3.1.2 Interaction-based Page Segmentation

Based on the recorded user interaction data, W3Touch aims at partitioning webpages into the main interactive components that contain *critical elements* in order to manage appropriate scopes of adaptation (e.g. adjusting a complete navigation bar based on data collected for the individual hyperlinks contained). To achieve this, we take into account the visual structure of a webpage and—as a first simple approach—consider critical elements to be either interactive elements which have been potentially missed (see above), or text and images that appeared in viewports above a certain threshold, which could mean that a considerable amount of zooming was necessary to be able to easily read a piece of text or view certain details. What is then marked as a *critical component* by W3Touch is either the critical element itself—if associated with an `id` attribute—or the closest parent element in the DOM tree that has an ID set, provided that it visually contains the critical element (cf. Figure 3.2). The idea behind this segmentation into critical components is that wrapper elements (such as a navigation bar, header or footer), which are often the components driving the structure of a

[1]`https://developer.mozilla.org/en/DOM/Touch_events`

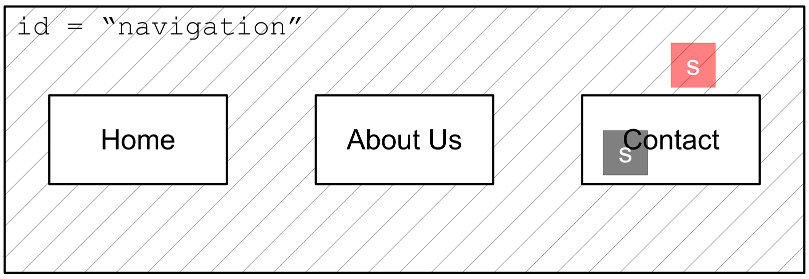


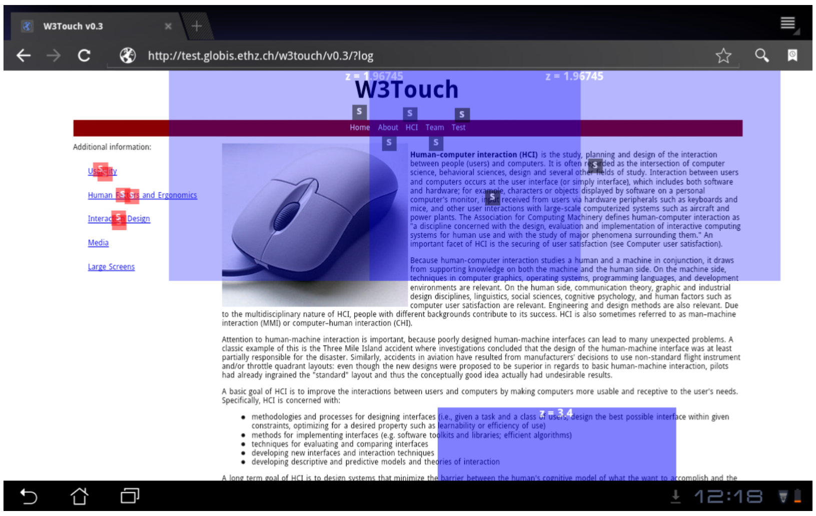
Figure 3.2: The "Contact" link is an example for a critical element, with the red square representing a single tap that missed its target. The critical component is the entire navigation bar since it is the closest parent (including the individual links themselves) associated with an ID. Therefore, adaptations inferred from the tracking data collected for the "Contact" link can be applied to all hyperlinks in the `navigation` container.

webpage, are commonly assigned `id` attributes for dynamic scripting and styling rules based on CSS. Therefore, relying on `id` attributes is a fairly effective and robust way for W3Touch to uniquely identify interactive components in need of adaptation and access them within the DOM tree. In the remainder of this thesis, "components" refers to all wrapper elements which are associated with an `id` attribute, while "critical components" refers to only those which are considered critical based on the associated tracking data.
In contrast to the approach just described, Bila et al. (2007) argue that `id` attributes are only rarely used by web developers and it is not ensured that they refer to the same content over time. Therefore, they propose an approach based on the position of an element within the DOM tree. However, since the structure of a DOM tree tends to change more frequently than the assignment of IDs, Bila et al. (2007) extend their approach with aspects of neighbourhood search if an element cannot be found at the expected place, and also consider additional context information of the DOM elements to be tracked. Although this solution has proved to be working well—also when trying to find similarities between several pages of the same website—, we have consciously decided to use the ID-based approach described above for several reasons:

- The assignment of `id` attributes is nowadays evolving as a state-of-the-art practice to mark the driving components of a webpage.
- Today's websites increasingly emphasise a consistent user experience and branding, which means that designs, and therefore also IDs, do not tend to change very frequently (except for major redesigns).
- The approach by Bila et al. (2007) does only cater for tracking of elements over time, but not for the page segmentation aspect. However, this is easily possible with `id` attributes while otherwise, additional techniques for page segmentation would be necessary (see Section 2.3.1 for examples).
- The necessity for neighbourhood search and additional context leads to a considerable amount of overhead, which decreases the efficiency of the approach by Bila et al. (2007).

Moreover, the approach employed by W3Touch gives web developers additional means to optimise and fine-tune the components of a webpage to be adapted (cf. Figure 3.2). It should also be mentioned that it is technically possible to extend the segmentation process to support `class` attributes in addition to IDs, which enables greater flexibility (cf. Nebeling and Norrie, 2011).

3.1.3 Data Visualisation



http://test.globis.ethz.ch/w3touch/v0.3/?log

W3Touch

Home About HCI Team Test

Additional information:

Usability

Human Factors and Ergonomics

Interaction Design

Media

Large Screens

Human–computer interaction (HCI) is the study, planning and design of the interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioral sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface (or simply interface), which includes both software and hardware; for example, characters or objects displayed by software on a personal computer's monitor, input received from users via hardware peripherals such as keyboards and mice, and other user interactions with large-scale computerized systems such as aircraft and power plants. The Association for Computing Machinery defines human-computer interaction as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." An important facet of HCI is the securing of user satisfaction (see Computer user satisfaction).

Because human-computer interaction studies a human and a machine in conjunction, it draws from supporting knowledge on both the machine and the human side. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human factors such as computer user satisfaction are relevant. Engineering and design methods are also relevant. Due to the multidisciplinary nature of HCI, people with different backgrounds contribute to its success. HCI is also sometimes referred to as man–machine interaction (MMI) or computer–human interaction (CHI).

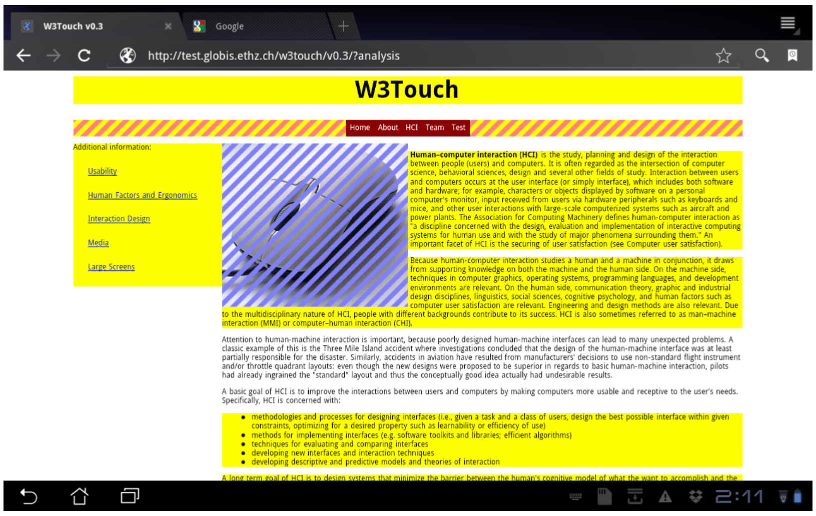
Attention to human-machine interaction is important, because poorly designed human-machine interfaces can lead to many unexpected problems. A classic example of this is the Three Mile Island accident where investigations concluded that the design of the human-machine interface was at least partially responsible for the disaster. Similarly, accidents in aviation have resulted from manufacturers' decisions to use non-standard flight instrument and/or throttle quadrant layouts: even though the new designs were proposed to be superior in regards to basic human-machine interaction, pilots had already ingrained the "standard" layout and thus the conceptually good idea actually had undesirable results.

A basic goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs. Specifically, HCI is concerned with:

- methodologies and processes for designing interfaces (i.e., given a task and a class of users, design the best possible interface within given constraints, optimizing for a desired property such as learnability or efficiency of use)
- methods for implementing interfaces (e.g. software toolkits and libraries; efficient algorithms)
- techniques for evaluating and comparing interfaces
- developing new interfaces and interaction techniques
- developing descriptive and predictive models and theories of interaction

A long term goal of HCI is to design systems that minimize the barrier between the human's cognitive model of what the want to accomplish and the

Figure 3.3: An example visualisation of actual tracking data collected by W3Touch for a single user on a tablet PC.



http://test.globis.ethz.ch/w3touch/v0.3/?analysis

W3Touch

Home About HCI Team Test

Additional information:

Usability

Human Factors and Ergonomics

Interaction Design

Media

Large Screens

Human–computer interaction (HCI) is the study, planning and design of the interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioral sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface (or simply interface), which includes both software and hardware; for example, characters or objects displayed by software on a personal computer's monitor, input received from users via hardware peripherals such as keyboards and mice, and other user interactions with large-scale computerized systems such as aircraft and power plants. The Association for Computing Machinery defines human-computer interaction as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." An important facet of HCI is the securing of user satisfaction (see Computer user satisfaction).

Because human-computer interaction studies a human and a machine in conjunction, it draws from supporting knowledge on both the machine and the human side. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human factors such as computer user satisfaction are relevant. Engineering and design methods are also relevant. Due to the multidisciplinary nature of HCI, people with different backgrounds contribute to its success. HCI is also sometimes referred to as man–machine interaction (MMI) or computer–human interaction (CHI).

Attention to human-machine interaction is important, because poorly designed human-machine interfaces can lead to many unexpected problems. A classic example of this is the Three Mile Island accident where investigations concluded that the design of the human-machine interface was at least partially responsible for the disaster. Similarly, accidents in aviation have resulted from manufacturers' decisions to use non-standard flight instrument and/or throttle quadrant layouts: even though the new designs were proposed to be superior in regards to basic human-machine interaction, pilots had already ingrained the "standard" layout and thus the conceptually good idea actually had undesirable results.

A basic goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs. Specifically, HCI is concerned with:

- methodologies and processes for designing interfaces (i.e., given a task and a class of users, design the best possible interface within given constraints, optimizing for a desired property such as learnability or efficiency of use)
- methods for implementing interfaces (e.g. software toolkits and libraries; efficient algorithms)
- techniques for evaluating and comparing interfaces
- developing new interfaces and interaction techniques
- developing descriptive and predictive models and theories of interaction

A long term goal of HCI is to design systems that minimize the barrier between the human's cognitive model of what the want to accomplish and the

Figure 3.4: An example visualisation of potentially critical components inferred from actual tracking data for a single user on a tablet PC.

In contrast to other approaches for automatic usability evaluation (Atterer et al., 2006; Carta et al., 2011; Hong et al., 2001) that provide visualisations of the gathered data in forms of user navigation paths, timelines of events or graph representations of websites, W3Touch focuses on spatial aspects specific to touch input since we consider according usability issues with respect to interface design. Therefore, our goal is to provide web developers with context-relevant visualisations that support the processes of manually identifying core interaction components of webpages, discovering the need for adaptations on different touch devices and informing the design of appropriate adjustments. To achieve this, W3Touch supports two additional views for web pages on which user interaction data have been collected, in which these data—or interpretations thereof—are visualised using semi-transparent overlays and colouring of affected elements.

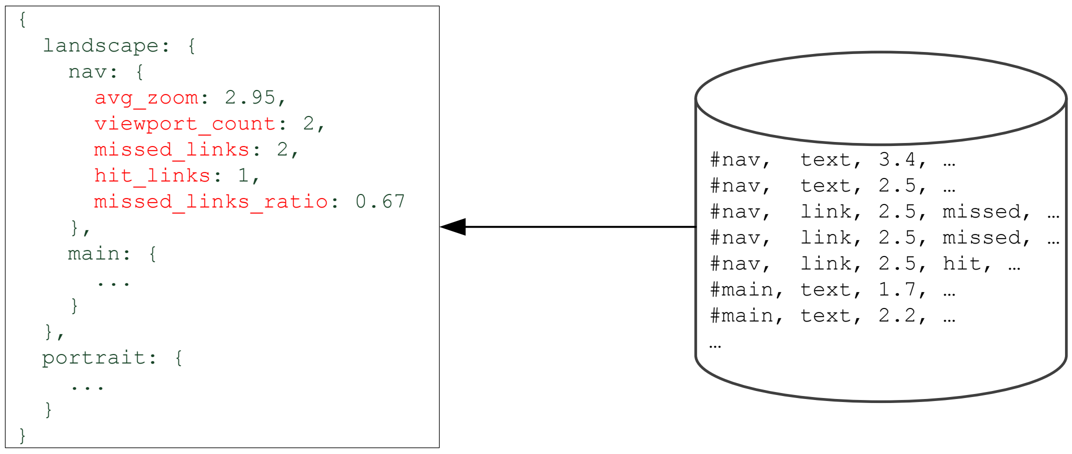
To give an example, Figure 3.3 presents a possible visualisation of actual tracking data collected by W3Touch. In this view, single taps that successfully hit an interactive element are highlighted using a red "S" symbol (as can be seen in the left navigation bar), while single taps that potentially missed their target are highlighted in grey (as can be seen in the top navigation bar). Additionally, the blue rectangular overlays represent the exact viewports of areas that have been zoomed in, while also showing the respective zoom factors. All of this information can be used by a developer to, e.g. determine which hyperlinks are most popular on a webpage or which parts of the content are zoomed in very frequently, thus indicating that they might be of major interest and therefore need adjustments in size.

Accordingly, Figure 3.4 presents a visualisation of possibly critical components, as inferred from the user activity data shown in Figure 3.3 and described in Section 3.1.2. In particular, all coloured components are associated with `id` attributes, whereas a yellow background indicates small text that required zooming, a red hatching indicates components containing links with small touch areas that have been potentially missed and a blue hatching indicates a potentially too small image. Rather than visualising the low-level W3Touch tracking data, this view draws the developer's attention directly to a higher-level analysis of the parts of a webpage in need of adaptation, thus removing the necessity to manually discover main interaction components and potential usability issues. However, this approach relies on mechanisms for automatic analysis and might therefore be not as precise as manual inspection.

While the examples presented above make use of only a single user and browsing context to illustrate the possibilities given to developers and the inference of critical components, a real crowdsourcing scenario—such as described in Section 4.3—would generate a considerably larger amount of data for different contexts (between which the developer could switch) and therefore result in visualisations that are similar to heat maps, concerning the actual tracking data. This means that the most interactive parts of a page would have the greatest emphasis regarding the brightness of the colouring. Moreover, the identification of critical components would be more accurate since more information is available. It would also be easily possible to provide visualisation fine-tuning, such as considering only links that have been potentially missed at least X times or only paragraphs that appeared in viewports above a zoom level of Y.

3.1.4 Adaptation Catalogue and Engine

To provide a simple, yet effective way of fixing design and layout issues that have been identified using W3Touch's tools for usability evaluation, we support a server-side catalogue (cf. Figure 3.1) through which it is possible for developers to define adaptation rules for different components of a webpage. This adaptation catalogue provides several metrics, which are



```
{
  landscape: {
    nav: {
      avg_zoom: 2.95,
      viewport_count: 2,
      missed_links: 2,
      hit_links: 1,
      missed_links_ratio: 0.67
    },
    main: {
      ...
    }
  },
  portrait: {
    ...
  }
}
```

```
#nav,  text, 3.4, …
#nav,  text, 2.5, …
#nav,  link, 2.5, missed, …
#nav,  link, 2.5, missed, …
#nav,  link, 2.5, hit, …
#main, text, 1.7, …
#main, text, 2.2, …
…
```

Figure 3.5: Analysis of log entries to provide context-aware tracking data.

gathered from the tracking data associated with different critical components and contexts by the adaptation engine (cf. Figure 3.5):

- average zoom factor
- number of appearances in zoomed-in viewports
- number of potentially missed links
- number of successfully hit links
- missed links ratio

These can be used to formulate conditional rules and define appropriate adjustments.

```
if (W3T.getTrackingData('#navigation').missed_links_ratio >= 0.5) {
  $('#navigation a').css('padding', '2em');
} else {
  $('#navigation a').css('padding', '1em');
}
```

Listing 3.1: An exemplary absolute adaptation based on tracking data.

A simple example for a conditional rule is given in Listing 3.1, which refers to the navigation bar shown in Figure 3.2. In case the "missed links ratio" (i.e. number of taps that potentially missed a link divided by number of all taps) over all links within the navigation bar is greater than or equal to 50%, the padding of hyperlinks is increased to 2 em, in order to enlarge their touch areas, while it is set to 1 em otherwise. Concerning this rule, the padding would be set to 2 em since the missed links ratio of the example navigation bar is 50%, according to the gathered tracking data (2 taps in total of which 1 tap missed a link).
While the above example relies on *absolute* values, it is also possible to directly incorporate the given metrics into the values of *relative* adaptations, rather than using them in `if`-clauses only. In this way, it is possible to let a web interface evolve over time, as more and more user activity data are collected. An example for this is given in Listing 3.2, where the value

of the missed links ratio is part of the calculation of the touch area size. For the navigation bar shown in Figure 3.2, this would mean that the padding of links is set to 1.5 em, given the missed links ratio of 0.5.

```
$('#navigation a').css('padding',
  (1 + W3T.getTrackingData('#navigation').missed_links_ratio) + 'em'
);
```

Listing 3.2: An exemplary relative adaptation based on tracking data.

Once the developer has defined an adaptation catalogue for a webpage and saved it on the server, it is fetched by the *adaptation engine* (cf. Figure 3.1) along with the tracking data for the used browsing context when a user accesses the corresponding page. Subsequently, the engine evaluates the contained rules and applies the associated adaptations on the client side. We make use of the capabilities of the adaptation catalogue in Section 4.3, where we apply a set of simple absolute and relative adaptations for different browsing contexts based on the tracking data we gathered as part of a user study.
Contrary to the collected user interaction data, the rules defined in the adaptation catalogue are generally independent of the context. However, to give developers more flexibility in defining and fine-tuning rules, it is possible to incorporate aspects of the browsing context into conditional rules, such as `if(isSmartphone())` or `if(isPortrait())`.
In contrast to the examples shown above, it has to be noted that developers can make use of the whole range of capabilities of JavaScript and jQuery[2] within the adaptation catalogue. Therefore, it is possible to formulate rules which are much more complex than rather simple adjustments of, e.g. font size, line height or padding. In particular, a developer could rearrange the whole layout of a page based on the given tracking data and metrics, e.g. from multiple columns to a single column, although this would require additional work concerning the definition of the respective rules. Moreover, an adaptation catalogue does not have to be specific to a certain webpage since we make use of critical components which are identified based on `id` attributes. These are often shared between pages and therefore, whenever multiple pages assign the same IDs to certain components—e.g. if they use the same layout template—a single adaptation catalogue could be applied to all of these.
Finally, we want to mention that the adaptation engine also provides options for default behaviour, e.g. automatically scaling the font sizes of all components according to the collected zoom factors. It is either possible to have *common* adaptations, which are applied to all components before the more specific rules from the catalogue, or to adapt all components that remain unaffected by the rules defined in the catalogue. In this way, it is not required for developers to define a rule for each critical component. The definition of the default behaviour is incorporated into the server-side configuration (see below), where it is also possible to turn off default behaviour, so that only the components specifically used in the adaptation catalogue are considered.

3.1.5 Configuration

The server-side configuration (cf. Figure 3.1) is used to inform the client-side components of W3Touch about the location of the server-side installation, so that they are able to communicate. This would be especially important in cases where the website to be adapted resides

[2]`http://jquery.com/`

on a different server. Additionally, it is possible to provide options for default adaptation behaviour as well as the definition of according default adaptations. The configuration of W3Touch also provides more control over the tracking and adaptation process, i.e. it supports the exclusion of certain components from being tracked and/or adapted.

3.2 Implementation

The concepts and features of W3Touch described above have been implemented in terms of a fully working prototype that serves as a *proof of concept*. The development process has happened in four iterative cycles during which three W3Touch prototypes have been implemented. The first prototype (v0.1) was based on *jQMultiTouch*[3] (Nebeling and Norrie, 2012) for detecting taps and zooming gestures and featured user activity tracking as well as respective visualisations. However, since default browser behaviour led to partly unpredictable firing of events in some browsers, it was unreliable to detect zooming based on gestures. Thus, the second prototype (v0.2) introduced a different approach to detecting zoom level changes in addition to a first version of interaction-based page segmentation and visualisation of critical components. The third iteration involved the user study described in Section 4.3, for which the necessary subset of W3Touch (i.e. interaction tracking and page segmentation) was prepared and extended with the required server-side components for storing and retrieving context-enhanced data (cf. Figure 3.1). Also, first basic versions of the adaptation catalogue and engine were introduced. Based on the experiences from the user study, the third and final prototype (v0.3) has been developed during the last iteration. It provides all of the features described in Section 3.1 and serves as the basis for the following implementation description.
The presented prototype has been implemented for *WebKit*[4]-based browsers since at least one of these was provided on each of the devices available for testing: an Apple iPod touch 4G (running iOS 4), an Apple iPad 1G (running iOS 4) and an Asus EeePad Transformer TF101 (running Android 3.2). In particular, *Firefox Mobile*[5]—which requires different approaches in terms of, e.g. detecting zoom levels—was not available on Apple devices. The browsers used for testing were *Safari Mobile*[6], the *Android native browser*[7] and *Dolphin Browser HD*[8] v6.0.0 for Android.
The main technologies used to implement W3Touch were *HTML*, *CSS3* and *JavaScript* in combination with the *jQuery* library on the client side while relying on *PHP* and *MySQL* on the server side. Concerning the source code listings shown in the remainder of this chapter, omitted parts of the code are shown as `/* [...] */`.

3.2.1 Webpage preparation

For providing the desired features, W3Touch has to prepare certain elements of a webpage. That is, 1) underlays need to be added to all hyperlinks and 2) all text nodes need to be wrapped in `<span>` elements.

[3]http://dev.globis.ethz.ch/jqmultitouch/
[4]http://www.webkit.org/
[5]http://www.mozilla.org/en-US/mobile/
[6]http://www.apple.com/safari/what-is.html
[7]http://www.android.com/about/
[8]http://www.dolphin-browser.com/

Hyperlink Underlays and Exit Links

To be able to track potentially missed links, each hyperlink on a webpage is enhanced with a transparent <**div**> element that spans a pre-defined range (the *underlay margin*) around the link. For this, we select all <**a**> elements[9] and for each one append an according underlay to the respective parent element (see listing below, ll. 1–6).

```
   $('a').each(function(index) {
     $(this).parent().append(
       $('<div class="w3t-underlay"></div>')
         .css(/* [...] */)
 5       .data('number', index)
     );

     $(this)
       .addClass('link' + index)
10     .css(/* [...] */);

     /* [...] */
   });
```

Listing 3.3: Adding special underlays to hyperlinks.

These underlays are absolutely positioned within the links' parents and their CSS position and dimensions are determined by taking into account the computed size and offset (relative to the top left corner of the parent) of the associated hyperlink as well as the underlay margin defined by the developer. The `z-index` property of underlays is set to 9998 while that of an annotated link is set to 9999 in order to make sure that no link lies behind an underlay, but that also no underlay is covered by non-link contents. Since CSS positioning is only possible within elements that are CSS-positioned themselves, `position:relative` is added to the style of all non-CSS-positioned parent elements, which does not affect their original placing within the document. Any non-CSS-positioned links are given a relative CSS position as well since otherwise the `z-index` property would have no effect. Finally, the actual link and corresponding underlay are associated with the index number of the link (given as the function argument in Listing 3.3, l. 1) to establish the relationship.

```
   var href = $(this).attr('href');
 2
   if (href.indexOf('#') != 0 && href.indexOf('javascript:') != 0) {
     $(this)
       .addClass('w3t-exit')
       .data('href', href)
 7     .attr('href', 'javascript:;');
   }
```

Listing 3.4: Special treatment for links leading to other webpages.

Moreover, links that point to different webpages—i.e. their `href` attribute does not start with "#" or "javascript:"—are annotated with a special CSS class and the original value of the `href` attribute is stored and replaced with "javascript:;" (Listing 3.4). This has to be done to make sure that the activation of such links can be intercepted so that collected tracking data can be sent to the sever before the user leaves the current webpage.

[9]This can be easily extended to also take into account other interactive content, such as <**input**> elements or elements associated with certain DOM handlers, such as `touchdown` or `click`.

Wrapping Text Nodes

Since we want to determine pieces of text which have been zoomed in extensively and infer possibly critical components from these, it is necessary to compute their intersections with the browser viewport (i.e. the part of a document that is currently displayed by a mobile browser). However, text nodes do not feature the same set of DOM properties as regular HTML elements. In particular, they lack spatial information such as position and dimensions. Therefore, it is necessary to wrap all text nodes in `<span>` elements and annotate them with a special CSS class, so that we have the possibility to use `$('.w3t-textnode')[i].offset()`, `$('.w3t-textnode')[i].width()` etc. in order to compute intersections with viewports.

3.2.2 Zooming Detection

When trying to recognise zooming gestures based on streams of `touchdown`, `touchmove` and `touchup` events using jQMultiTouch (Nebeling and Norrie, 2012), we found out that the browsers used for testing showed different and partially unreliable behaviours in firing the corresponding touch events, which is most probably due to interferences with default browser behaviour. For example, the Dolphin Browser mostly triggered 3 `touchdown`, 2 `touchmove` and no `touchup` events for pinch-to-zoom gestures, where 2 `touchdown`, several `touchmove` and 2 `touchup` events would be expected. However, without the respective `touchup` events, it is not possible to determine where a zooming gesture ended and how much a user zoomed in. Therefore, also to ensure cross-browser compatibility, W3Touch follows a different approach in terms of zooming recognition.
In particular, the current zoom level is determined by dividing the fixed actual width of the displayed document (`document.documentElement.clientWidth`) by the current width of the mobile browser viewport (`window.clientWidth`). This means that the zoom level is equal to 1 if no horizontal scrolling is possible. It is stored in a global variable and within a pre-defined interval of 100 ms, W3Touch recurrently check for changes of this value. If the new zoom level is greater than the previous one (i.e. we only consider zoom-in events) and also exceeds a certain threshold of 1.5, W3Touch takes further action to handle the change in zoom level, thus triggering a “pseudo” *zoom changed* event. Finally, the new zoom level value is stored in the dedicated global variable for the next comparison. Following this approach, W3Touch is able to, not only detect pinch-to-zoom gestures, but also zooming actions that were triggered by double taps. Moreover, if the zoom level stays constant and lies above the defined threshold, W3Touch checks whether the position of the viewport (`window.pageXOffset` and/or `window.pageYOffset`) has changed, e.g. through scrolling or activating a page-internal link. If this is the case, the change in position is handled the same way as zoom level changes.

3.2.3 Logging

W3Touch provides a dedicated function for logging, whose arguments are the database to use (in terms of a string referring to a server-side PHP script that accesses the respective database), the data packet to be stored (as an array)[10] and an optional synchronous callback function that is executed after the data have been received by the server. It is possible to

[10]This array will always be referred to as the *data packet* in the following

choose between two databases, that are `'log'` for storing low-level W3Touch tracking data and `'critical_elements'` for registering potentially critical components.
The specific data packet passed to the logging function is automatically enhanced with standard context data, that in the current state of the prototype consist of *the current zoom level, device orientation, user agent string, screen size* and *window dimensions*. The device orientation is a property of the JavaScript `window` object, but not supported by all current mobile browsers. Therefore, we also consider the viewport dimensions, as they are a reliable way of determining the current orientation if no `window.orientation` property is available (e.g. if the viewport height is greater than the viewport width, the user is browsing in portrait mode). Moreover, a timestamp and the URL of the corresponding webpage are added to each log entry.
Once the data has been prepared, it is sent to the specified server-side PHP script. To enable cross-site requests in case the server-side components do not reside on the same domain as the webpage to be adapted, we make use of an `<iframe>` that is dynamically added to the page, prepare the data as hidden input fields within that `<iframe>` and submit the corresponding HTML form. Callback functions can be realised by preparing them as global variables and adding an `onload` handler to the `<iframe>` after it has been added to the page and before submitting the data. Finally, when the data is received by the PHP script, the type of device (*smartphone* or *tablet PC*) is additionally inferred from the user agent string making use of the *MobileESP* library[11], before storing all of the data and context information in the database.
It is possible to specify suitable intervals in which data will be sent to the server side. If the logging function is called and the elapsed time since the last data packet was sent is not greater than that interval, the data to be logged is cached while otherwise (or in case a callback function has been passed to the logging function) the next packet containing all of the cached data is sent to the server. However, it can be problematic to specify such an interval since data could get lost if a user leaves the webpage. Although hyperlinks leading to other webpages are prepared by W3Touch in a way that prevents data loss, a user could still leave the webpage by using the browser's address bar or closing the browser. Since not all mobile browser support the `window.onbeforeunload` feature[12] that makes it possible to ask a user if they really want to leave a page (e.g. in case there are unsaved data), only specifying an interval of 0 seconds would eliminate any possibility of data loss, but would also lead to communication overhead.

3.2.4 Context-aware Interaction Tracking

W3Touch makes use of the `touchable` function of jQMultiTouch (Nebeling and Norrie, 2012) to track touch interaction with hyperlinks or other parts of a webpage. Basically, to track any touch event within the given document, only the `<body>` element needs to be made *touchable:*

```
$('body')
  .touchable({
    touchDown: function(e, touchHistory) {
      /* [...] */
      log('log', ['down', '']);
    },
```

[11]http://blog.mobileesp.com/
[12]https://developer.mozilla.org/en/DOM/window.onbeforeunload

```
      touchMove: function(e, touchHistory) {
        // Necessary to distinguish taps from other gestures.
        /* [...] */
10      log('log', ['move', '']);
      },
      touchUp: function(e, touchHistory) {
        var coords = crossBrowserPageCoords(e);

15      /* [...] */
        log('log', ['up', 'missed,' + coords.x + ',' + coords.y]);
      }
    });
```

Listing 3.5: Tracking general taps.

That is, any touch interaction with any element of the document will be propagated to the <**body**> element, where it is registered as a stream of `touchDown`, `touchMove` and `touchUp` events, which are logged individually in terms of the event name (first element of the data packet) and additional information in string format (second element of the data packet). While no additional data are logged for `touchDown/Move` events, log entries of `touchUp` events are enhanced with event coordinates since these are relevant for data visualisations. The `crossBrowserCoords` function returns the *absolute* coordinates of an event, which are relative to the top-left corner of the document. This is necessary since different mobile browsers handle the coordinates contained in events in different ways (e.g. the `event.clientX/Y` properties already contain scrolling offsets in Safari Mobile, which is not the case in the Android native browser). Moreover, because the above code is intended to register general interactions, the additional data of `touchUp` events contains the keyword "missed" (Listing 3.5, l. 16) to be able to later distinguish these from successful taps on hyperlinks. To log the latter ones, the `touchable` function is called separately for all hyperlinks, whereas `touchUp` events are not propagated to the <**body**> element by calling `e.stopPropagation()`. Rather, the `log` function is called while this time indicating a successful "hit" within the additional data, which will result in a different visualisation of a corresponding tap (cf. Section 3.2.7). If a hyperlink has been marked as a link leading to an external page and prepared accordingly by W3Touch (cf. Listing 3.6), an additional callback function is passed to the logging function, which makes sure that all data is sent to the server before leaving the current webpage:

```
if ($(this).hasClass('w3t-exit')) {
  var href = $(this).data('href');

  log('log', ['up', 'hit,' + coords.x + ',' + coords.y], function() {
    /* [...] */
    location.href = href;
    /* [...] */
  });
}
```

Listing 3.6: Special treatment of external hyperlinks.

Moreover, zooming interaction is logged in terms of the corresponding viewports. That is, if W3Touch detects a *zoom changed* event (as described in Section 3.2.2), the logging function is called while passing "zoom" as the event name (instead of, e.g. "up", as can be seen in the listing above) and the position of the current viewport (`window.pageXOffset` and `window.pageYOffset`) as the second element of the data packet. The viewport dimensions do not need to be passed as part of the data since they are already contained in the standard context information that is added to each log entry.

3.2.5 Interaction-based Page Segmentation

Inferring possibly critical components from users' activities on a webpage does not happen by analysing low-level tracking data after they have been logged. Rather, it takes place at the same time that interactions are recorded and from within the same handler functions already mentioned above. In particular, our interaction-based page segmentation logic makes use of the same logging function, but rather than storing low-level tracking data (such as touch event or viewport coordinates) in the *log* database—that is used for data visualisation only—it sends data containing information about higher-level components to the *critical elements* database.

```
function logMissedLink(i) {
  var id = findIdOfClosestParent($('.link' + i));

  if (id != null) {
    log('critical_elements', ['link', id, 'missed']);
  }
}
```

Listing 3.7: Logging a potentially critical component based on a missed link.

To give an example, Listing 3.7 shows the function that is called when a user has tapped on a hyperlink underlay. That is, the function is called from within the `touchUp` handler of the `touchable` function, given the case that the last touch event was a `touchDown`. If it was a `touchMove` instead, this would mean that a scrolling or zooming gesture happened instead of a tap.
The parameter `i` is the index number of the corresponding hyperlink that has been associated with the underlay and the hyperlink itself (cf. Listing 3.3). The helper function `findIdOfClosestParent` returns either the ID of the hyperlink—given the case that it has an `id` attribute set—or that of the closest parent element associated with an ID that visually contains the hyperlink (i.e. the coordinates of the link lie within the bounds of the parent element). If an ID could be found that is not part of the list of components to be excluded from tracking (see Section 3.2.8), a data packet is sent to the `'critical_elements'` database. This packet contains the keyword “link” to indicate the kind of interaction, the `id` attribute of the possibly critical component and the keyword “missed” as additional data to indicate that a missed link is being recorded. Accordingly, successful taps on hyperlinks are recorded as well to be able to provide the *missed links ratio* of a critical component, whereas the keyword “hit” is sent instead of “missed”. To cater for links which lead to a different webpage, the logging procedure is also incorporated into the code already shown in Listing 3.6 to make sure that both databases receive their corresponding data before leaving the current webpage:

```
if ($(this).hasClass('w3t-exit')) {
  var href = $(this).data('href');

  log('log', ['up', 'hit,' + coords.x + ',' + coords.y], function() {
    /* [...] */
    log('critical_elements', ['link', id, 'hit'], function() {
      location.href = href;
    });
    /* [...] */
  });
}
```
Listing 3.8: Special treatment of external hyperlinks involving both types of logging.

In terms of potentially too small text, whenever W3Touch detects a *zoom changed* event (as described in Section 3.2.2), it computes the intersections of all text nodes—which have been wrapped in `<span>` elements—with the current viewport. In case an intersection is found, a function similar to the one in Listing 3.7 is called. However, in contrast, the data packet sent to the *critical elements* database is `['text', id, '']`, where `'text'` denotes the kind of interaction (i.e. a piece of text that appeared in a viewport above a certain threshold) while `id` again refers to the potentially critical component. The third element is an empty string since we do not need additional information. Moreover, W3Touch also computes the intersections of all images with the current viewport. In case an image spans at least than 75% of the viewport area or at least 50% of the image are visible, it is logged in analogy to small text while only replacing `'text'` with `'img'` to indicate the different kind of interaction.

3.2.6 Context Engine

```
$ua = new uagent_info();

// use MobileESP to determine device context
if ($ua->DetectTierIphone()) {
  $device = "smartphone";
} else if ($ua->DetectTierTablet()) {
  $device = "tablet";
} else {
  $device = "?";
}

/* [...] */

$entries = mysql_query(
  "SELECT zoom, wnd_w, wnd_h, event, msg FROM log WHERE (
    url = '$url' AND
    device = '$device' AND
    wnd_w > wnd_h
  )"
);
```
Listing 3.9: Example request for getting context-aware log data for a particular webpage from the database, taking into account device type and orientation.

The context engine is incorporated into several PHP scripts on the server side that deliver log entries, information about potentially critical components and tracking data to the client-side parts of W3Touch. It ensures that appropriate conditional statements for retrieving context-aware data are included in the necessary SQL queries. Moreover, the context engine makes use of the MobileESP library to automatically detect the type of device with which a user is currently interacting with W3Touch (see listing above) and only delivers corresponding context-matching data. Although MobileESP is able to distinguish more detailed types of devices (e.g. *iPhone* vs. *Android phone*), we chose a broader distinction that is simple yet already very effective for our purposes, also regarding the user study described in Section 4.3.

3.2.7 Data Visualisation

W3Touch supports the separate visualisation of two types of data, that are the low-level tracking data stored in the *log* database (cf. Figure 3.3) and the higher-level information about potentially critical components stored in the *critical elements* database (cf. Figure 3.4). The client-side W3Touch script fetches the corresponding log entries for portrait and landscape mode upon start-up making use of PHP scripts on the server side that return the requested data in terms of JSON objects. The context engine ensures that only entries matching the current device context (*smartphone* or *tablet PC*) are delivered. Subsequently, W3Touch iterates through the received data and then adds appropriate overlays to the document and/or annotates affected components with special CSS classes for highlighting.

Tracking Data

In terms of visualising user activity data, we focus on general taps, taps which successfully hit a hyperlink and zoom level changes. In the first instance, to be able to distinguish taps from scrolling and zooming gestures, it is necessary to sequentially iterate through the log data received for the used device context, taking into account only those entries that match the current orientation. In case a `touchUp` event is being processed, W3Touch checks whether the previous event was a `touchDown` and adds an overlay for visualising a tap (i.e. a semi-transparent square containing an “s”) only if this is the case since a preceding `touchMove` would indicate a non-tapping gesture. If the log entry indicates a successfully hit hyperlink, the square is given a red background colour and a black background colour otherwise. Moreover, W3Touch parses the coordinates of the `touchUp` event from the data contained in the log entry and sets the absolute CSS position of the corresponding square accordingly.
For visualising zoom level changes, which are indicated as “zoom” events in the log, W3Touch parses the coordinates of the corresponding viewport from the log entry. Moreover, it gets the dimensions of the viewport and the related zoom level from the standard context information associated with the log entry. The blue overlay which is subsequently added to the document is placed at the exact position of the viewport and spans its exact dimensions. Furthermore, it contains a string indicating the zoom level, from which also the opacity of the overlay is inferred. That is, the opacity is proportional to the zoom level, up to a maximum of 3.4. For example, viewports with corresponding zoom levels ≥ 3.4 are given an opacity of 0.5 while viewports with a corresponding zoom level of 1.5 are given an opacity of 0.22.
All overlays added to the document are marked with two CSS classes: 1) `w3t-overlay` and 2) `w3t-portrait` or `w3t-landscape`, according to their context. If the orientation of the device used for viewing the visualisation changes, all overlays are hidden and the already fetched log entries associated with the new orientation are processed, or the related overlays are un-hidden, if they have already been added before, thus ensuring that each log entry is only processed once.

Critical Components

In terms of visualising potentially critical components, we focus on missed links, small text and small images. First, W3Touch fetches the components which have been found to be possibly critical for the current device context from the database in terms of pairs containing an `id` attribute and the kind of interaction (i.e. “link”, “text” or “img”), considering both portrait and landscape mode. Next, the elements holding the respective IDs are associated with two CSS classes, that are 1) `w3t-link`, `w3t-text` or `w3t-img`, according to the associated interaction and 2) `w3t-portrait` or `w3t-landscape`. Finally, depending on the current orientation, matching components which contain potentially missed links and/or potentially too small text are given additional CSS classes, that are `w3t-critical-link` and/or `w3t-critical-text`. These determine the actual visualisations. To ensure that a component can be marked with both, `w3t-critical-link` adds a background image that is a hatching with red and transparent stripes while `w3t-critical-text` sets the background colour to yellow. Therefore, a component with potentially missed links *and* small text would be highlighted with a red and yellow hatching. Components that contain potentially too small images are treated in a different way. In particular, a hatched `<div>` overlay with blue and transparent stripes is added to all images within the respective components. These overlays are again marked with either `w3t-portrait` or `w3t-landscape`.
In case the orientation of the device used for viewing the visualisation changes, all CSS classes that visualise missed links or small text are removed, while overlays are hidden. According to the new orientation, elements marked with the respective CSS class are again highlighted as described above while also new image overlays are added or existing ones are un-hidden. Again, each entry fetched from the database is only processed once.

3.2.8 Configuration

The W3Touch configuration is a JavaScript file that has to be included in a webpage along with the actual W3Touch script, but must be executed beforehand since it introduces global variables that are crucial for W3Touch to work properly. In particular, the configuration defines the following:

- The *path* to the server-side installation of W3Touch.
- An array of component IDs to be *excluded from tracking*.
- An array of component IDs to be *excluded from adaptation*.
- A boolean variable to determine whether the adaptation catalogue shall use *default behaviour*.
- A boolean variable to determine whether the default behaviour shall be applied as *common adaptations* (i.e. to all components of a webpage before the specific rules from the adaptation catalogue are applied) or to all components that remain unaffected by the adaptation catalogue.
- A function to define default behaviour. This function provides a variable `selector` that acts as a placeholder for the affected components (see Listing 3.10).

```
defaultBehaviour: function($, selector) {
  $(selector).css('background-color', 'lightblue');
}
```

Listing 3.10: An example for default behaviour.

An exemplary configuration file can be found in Appendix A.1.

3.2.9 Adaptation Catalogue and Engine

To provide developers with means for defining suitable adaptations, W3Touch first makes use of a server-side PHP script to obtain the tracking data for the current webpage and device context (*smartphone* or *tablet PC*) in terms of a JSON object, considering both portrait and landscape mode. In particular, the PHP script fetches all context-matching entries from the *critical elements* database and then computes the metrics introduced in Section 3.1.4 for each potentially critical component, further divided by orientation (cf. Figure 3.5). The tracking data which will be available from within the adaptation catalogue (through a global variable) is then set to the subset of the retrieved data that matches the current orientation of the device used to view the webpage. Next, W3Touch initialises an array called `W3T.unusedComponents` that contains all elements of the webpage associated with an `id` attribute that are not part of the list of components to be excluded from adaptation, which is defined in the W3Touch configuration. Finally, to be ready for the actual adaptation process, the implementation of the native `css` function provided by jQuery[13] has to be temporarily substituted by a different approach. This is necessary to be able to undo changes to the style of HTML elements, which is required if the device orientation is changed while viewing the webpage and the adaptation catalogue needs to be re-applied for the changed context.
In particular, the new temporary function changes the style of an HTML element by appending a new CSS declaration to the end of the element's `style` attribute along with an `!important` flag to make sure that it is not overwritten by an already present competing declaration. For example, calling `$('div').css('font-size', '12pt')` would result in:

```
<div style="[oldDeclarations];font-size:12pt !important">
```

However, this guarantees that a rule from the adaptation catalogue can still overwrite a common adaptation which has been applied before by simply appending another declaration, even if it has the same CSS attribute, e.g.:

```
<div style="[oldDeclarations];font-size:12pt !important;font-size:14pt
    !important">
```

In this way, changes to the style of an element can be stored in terms of a simple text string and the style of the corresponding element can be reset by deleting this string from the `style` attribute. In case of the above example, this would mean deleting everything after `[oldDeclarations]`. Moreover, whenever the style of an element is changed using the new temporary function and the jQuery selector contains the ID of a webpage component, such as `$('#thisIsAnID > a')`, that component is removed from `W3T.unusedComponents`, along with all of its descendants which also have an ID, because these might inherit the style from the parent component and must therefore be considered *used* as well. It has to be noted that—while jQuery's `css` function is prepared so that changes can be automatically undone—if a

[13] `http://api.jquery.com/css/`

developer makes use of other functions to alter the layout of a webpage, they have to manually ensure that it is reset to its original state upon orientation changes. This can be realised by incorporating checks for portrait and landscape mode into the adaptation catalogue and to add appropriate `else` statements to conditional rules.
The adaptation catalogue itself, which resides on the server-side in terms of a JavaScript file (see Appendix A.2 for an example), is fetched as a text string via an asynchronous HTTP request and cached in a variable. Subsequently, the adaptation catalogue string is passed to the `eval` function of JavaScript and therefore executed while accessing the context-aware tracking data made available through a global function. In particular, it is possible to use the following context and tracking data functions and properties from within the adaptation catalogue:

- `W3T.isSmartphone()`, to check whether the user's device is a smartphone.
- `W3T.isTablet()`, to check whether the user's device is a tablet PC.
- `W3T.isPortrait()`, to check whether the user's device is currently in portrait mode.
- `W3T.isLandscape()`, to check whether the user's device is currently in landscape mode.
- `W3T.ignore(selector)`, to exclude a component (specified by its ID in terms of a jQuery selector) from default behaviour without changing its style.
- `W3T.getTrackingData(selector).x`, to get the tracking data for a potentially critical component (specified by its ID in terms of a jQuery selector), with `x` being either `avg_zoom`, `viewport_count`, `missed_links`, `hit_links` or `missed_links_ratio`, according to the metrics introduced in Section 3.1.4. If no data for the requested component exist, standard values are returned, such as 1 for `avg_zoom` or 0 for `viewport_count`.

If the adaptation engine is configured to support default behaviour in terms of common adaptations, the corresponding rules are applied *before* the adaptation catalogue string is evaluated. This affects all components of the webpage that have an `id` attribute set and are neither a descendant of a different component associated with an ID[14] nor part of the list of components to be excluded from adaptation. In contrast, if the default behaviour is configured to adapt components not used in the adaptation catalogue, the corresponding rules are applied *after* the adaptation catalogue string has been evaluated, affecting all components which are left in `W3T.unusedComponents`. Finally, after having applied the catalogue and possible default behaviour, the `css` function of jQuery is set back to its original implementation.
In case the orientation of the device used to view the adapted webpage changes, all changes made to the styles of webpage elements by the adaptation engine are undone, as already described above. Moreover, `W3T.unusedComponents` is reset and the available tracking data are changed to the subset of the data fetched for the current device context which matches the new orientation. Subsequently, the catalogue (that has been cached and thus does not need to be fetched again) as well as possible default adaptations are re-applied.

[14]This has to be done to ensure that a common adaptation cannot have higher priority than a rule from the adaptation catalogue that is applied to a parent component and shall be inherited by its descendants.

Finally, we want to note that we are aware of more advanced approaches to context-awareness, such as XCML (Nebeling et al., 2012a). However, we decided for a less sophisticated approach since it is sufficient for the purpose of this thesis and does not introduce additional dependencies.

3.2.10 Live Version

A live version of the final W3Touch prototype can be found at:

`http://dev.globis.ethz.ch/w3touch/demo.html`

Please append `?log` to the above URL for the tracking data visualisation, `?analysis` for the visualisation of potentially critical components and `?adaptation` to apply the adaptation catalogue based on the tracking data.

4 Evaluation

The evaluation of W3Touch has been carried out in three steps. First, we conducted semi-structured expert interviews. These were concerned with usability issues related to an exemplary webpage that had to be accessed using different touch devices, and how the identified issues could be addressed. Second, we carried out a user study to evaluate W3Touch on the same exemplary webpage within a crowdsourcing scenario. The recommendations given by the experts were thereby taken into account regarding the design of suitable adaptations. Finally, we compared the adapted webpage created with the help of W3Touch to a existing mobile version of the same webpage. This happened in terms of a second internal user study as well as specific metrics developed for the evaluation of websites in touch contexts.

The example webpage we chose for the evaluation was the Wikipedia article about *Apple Inc.*, as retrieved on October 7, 2011[1]. This decision was made for two reasons. First, the article is optimised for reading on common desktop screens (1024×768 and above) and therefore requires adaptations for convenient use on mobile touch devices. Furthermore, it features

[1] `http://en.wikipedia.org/w/index.php?title=Apple_Inc.&oldid=454378296`

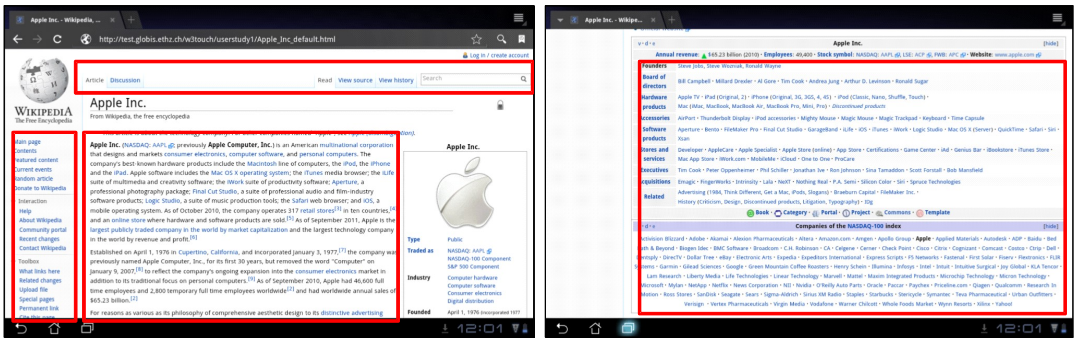
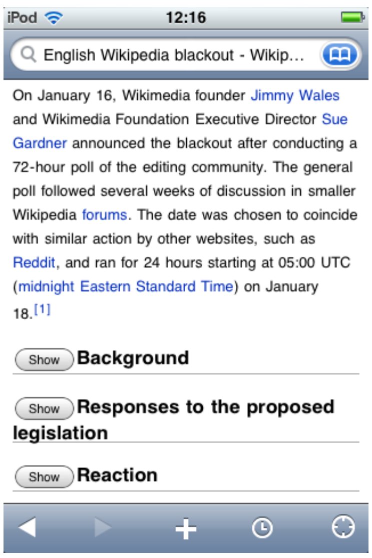
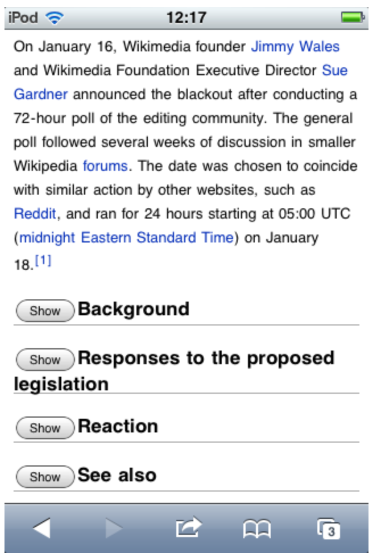


Figure 4.1: Typical webpage components present in Wikipedia articles: header, sidebar navigation, main text and footer.



On January 16, Wikimedia founder Jimmy Wales and Wikimedia Foundation Executive Director Sue Gardner announced the blackout after conducting a 72-hour poll of the editing community. The general poll followed several weeks of discussion in smaller Wikipedia forums. The date was chosen to coincide with similar action by other websites, such as Reddit, and ran for 24 hours starting at 05:00 UTC (midnight Eastern Standard Time) on January 18.[1]

Figure 4.2: The native Wikipedia application for iOS.



On January 16, Wikimedia founder Jimmy Wales and Wikimedia Foundation Executive Director Sue Gardner announced the blackout after conducting a 72-hour poll of the editing community. The general poll followed several weeks of discussion in smaller Wikipedia forums. The date was chosen to coincide with similar action by other websites, such as Reddit, and ran for 24 hours starting at 05:00 UTC (midnight Eastern Standard Time) on January 18.[1]

Figure 4.3: The Wikipedia mobile website.

components which are representative for a wide range of today's websites—a horizontal top navigation, a vertical sidebar, a large amount of continuous text and numerous hyperlinks to related topics contained in a footer (cf. Figure 4.1). Based on our choice of an exemplary website, we start with a review of the differences between the regular and mobile versions of Wikipedia before continuing with the actual evaluation.

4.1 Case Study: Wikipedia

The Wikimedia Foundation offers a native iOS application[2] for offline access to the English version of the Wikipedia, which is an example for device-specific authoring (Bickmore and Schilit, 1997). It is mostly written in HTML5 and JavaScript, put into an Objective-C framework (as required for iOS development[3]), and is therefore very similar to the specific mobile version of the Wikipedia website[4] (cf. Figures 4.2 and 4.3).
The following differences between the implementations of the Wikipedia interface are specific for touch-operated mobile devices:

- Wikipedia Mobile has been rearranged to a single-column layout, as compared to the regular website. That is, the navigation bar on the left (cf. Figure 4.10) has been removed and the according functionality is not available on mobile devices. Moreover, the overview box (typically including images and statistical data) at the right side of the introductory paragraph (cf. Figure 4.4) has been moved to the top of the column.

[2]`http://wikitech.wikimedia.org/view/Mobile_iPhone`
[3]`http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/GS_iPhoneGeneral/_index.html`
[4]`http://en.m.wikipedia.org/`

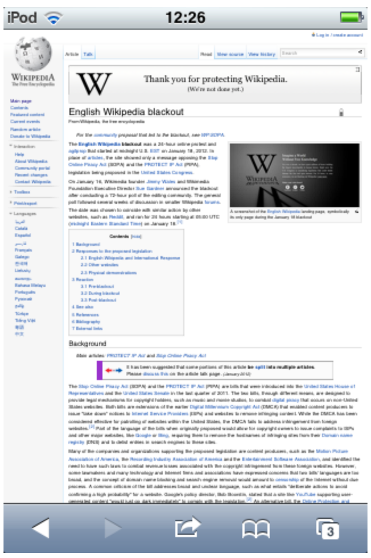
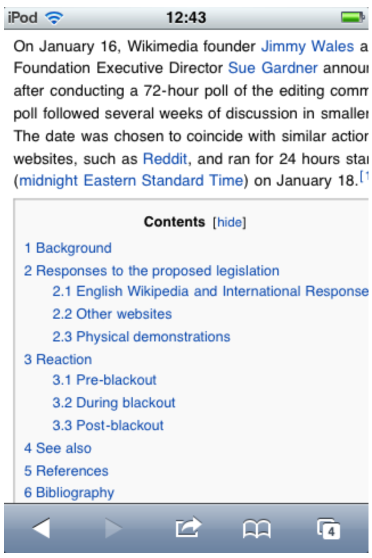

iPod
12:26
Thank you for protecting Wikipedia.
(We're not done yet.)
English Wikipedia blackout
Background

Figure 4.4: The regular Wikipedia website.



On January 16, Wikimedia founder Jimmy Wales a
Foundation Executive Director Sue Gardner annou
after conducting a 72-hour poll of the editing comm
poll followed several weeks of discussion in smalle
The date was chosen to coincide with similar actio
websites, such as Reddit, and ran for 24 hours sta
(midnight Eastern Standard Time) on January 18.[1

Contents [hide]

1 Background
2 Responses to the proposed legislation
2.1 English Wikipedia and International Response
2.2 Other websites
2.3 Physical demonstrations
3 Reaction
3.1 Pre-blackout
3.2 During blackout
3.3 Post-blackout
4 See also
5 References
6 Bibliography

Figure 4.5: The regular Wikipedia website, zoomed in to a readable size.

- In addition to a single-column layout, Wikipedia Mobile makes use of the `viewport` meta tag[5], which ensures that the initial width of the viewport fits the available screen width and the initial scaling is 1.0. This renders readable text with an absolute height of 0.080 in, although the same CSS font size as on the regular website (0.8 em) is used (cf. Figures 4.3, 4.4 and 4.5).

- In the mobile versions, the header of the regular website has been reduced to the included search form (cf. Figures 4.2 and 4.4). That means, it is not possible to edit pages or view article revisions on mobile devices.

- The table of contents, as present on the regular website (cf. Figures 4.4 and 4.5), is not provided in the mobile versions in its original form. Instead, it is incorporated into the article text, i.e. all first-level sections of an article are collapsed (except for the introductory paragraph that is located above the table of contents) and feature a show/-collapse button next to their heading (cf. Figure 4.3). Therefore, the mobile versions yield a table of contents that contains only the first-level headings.

- Wikipedia Mobile features a slightly larger line height (165%; cf. Figure 4.3) at the same font size, as compared to the regular website (150%; cf. Figure 4.5), thus giving hyperlinks more space, although not increasing their touch areas (as line height does not count towards these).

- The footer parts of an article (typically containing cross-references to related articles and article categories) are collapsed in the mobile versions. Moreover, the contained links—originally aligned horizontally and vertically, thus being very densely packed—

[5]`http://developer.apple.com/library/IOs/#documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html`

are presented as bullet point lists, which makes successfully hitting intended links easier.

4.2 Expert Interviews

Six experts have been asked to review the original design of the Wikipedia article with respect to issues originating from its use on touch devices and possible adaptations for potentially fixing those problems. Four of the interviewees were PhD students with research and teaching activities in the fields of web engineering, interaction design and/or usability methods, one had been working on web engineering projects in industry for several years and one was a Bachelor's student in Computer Science who just had completed an internship involving the development of several (multi-)touch applications for both mobile and desktop devices.
The interviews took between 25 and 50 minutes and the interviewees were provided an Apple iPod touch 4G (running iOS 4) and an Apple iPad 1G (running iOS 4) for reviewing the Wikipedia article. The devices were chosen so that the evaluation involved at least two distinct touch contexts, i.e. while an iPod touch features a considerably small screen, the larger iPad has been specifically designed for better reading comfort. We also took into account different device orientations.
The interview transcriptions in bullet point form can be found in Appendix B.

4.2.1 Structure

The interviews were of a semi-structured nature. First, we presented the participant the Wikipedia article on one of the above mentioned devices, whereas the starting device was altered in each interview to counterbalance ordering effects. Next, we asked the interviewee to have a thorough look at all parts of the webpage while thinking aloud. Any potential issues concerning the design of the article or the touch-based interactions should be directly expressed. Moreover, we wanted the participant to also suggest potential fixes for any issues they would find. In a second task, the interviewee was asked to have a closer look at specific components of the article (cf. Figure 4.1), if they had not done so before, and again directly express any potential issues and adequate adjustments while also considering differences between portrait and landscape mode.
After having reviewed the Wikipedia article for a first time, the participant was presented the remaining device and asked to proceed with the same two tasks as already described above.

4.2.2 Findings

In the first instance, the experts stated that—independent of the device used—the main component of interest is the actual article text. In contrast, header, sidebar navigation and especially footer only take minor roles when interacting with the webpage. One interviewee explicitly stated that they "never click on [the footer]." However, caused by the length of the article, the experts in general also stated that vertical scrolling is a major issue when it comes to interacting with the webpage, particularly in terms of gaining a good overview of the article.

iPod touch

The general consensus of the interviewees was that the webpage—as initially displayed on an iPod touch—is ”just not useful in any way” without zooming since the text is rendered unreadable due to the fact that the article is zoomed out to fit the small display. Furthermore, all but two experts pointed out that the amount of zooming necessary to interact with the webpage posed a considerable problem to them since this made it difficult to get a good overview of the page, also due to the increased amount of horizontal scrolling. This means that the user has neither a good overview when the article is zoomed out nor when the zoom level is adjusted to render text readable. Propositions for adjustment were therefore mainly concerned with increasing the font size of the article text.
Moreover, the experts stated that hyperlinks were generally too small. All but one expert pointed this out in particular regarding the sidebar navigation and footer. Since within these components links are very densely packed and a finger covers several of them even if zoomed in to a readable size, one interviewee particularly stressed that in any case, a developer has to “make sure that all the interactive things you can touch have a reasonable size.” The general recommendation for adaptation therefore was to give hyperlinks more space in order to increase their touch areas and avoid erroneously tapped links.
When reviewing the header component, the interviewees generally saw least problems there. This was due to the small number of links, which moreover feature a larger touch area based on their box-like appearance. Also, the horizontal alignment counters the chance of erroneously tapping an undesired link.
Concerning differences in interaction between portrait and landscape mode, two of the experts explicitly stated that the same adaptations should apply for both orientations since the use of landscape mode does not significantly better the user experience.
Finally, more advanced suggestions for improvement included rearranging the webpage to a single-column layout—similar to the Mobile Wikipedia—in which all the content is aligned vertically, thus making horizontal scrolling unnecessary. In particular, when realising such a solution, the vertical sidebar navigation would have to be transformed into a horizontal bar and moved to the top of the webpage. Furthermore, another popular proposition was to dynamically adjust the line width of the article text to the current size of the browser viewport, as is already done by some browsers provided on certain mobile devices. Also, three experts recommended to make use of pagination features while in contrast, one expert explicitly stated that it would be convenient if the user would have to only scroll in the vertical directions with one thumb while reading the article text.

iPad

The general consensus concerning the iPad was that the view of the webpage, as initially displayed, is almost optimal due to the larger amount of screen real estate available. However, while there is less need for zooming to be able to interact with the article, the differences between portrait and landscape mode were found to be more significant as compared to the iPod touch. In particular, two of the interviewees pointed out that the sidebar navigation could be moved to the top of the page (as already proposed for the iPod touch) in portrait mode to give the main text more horizontal space, whereas it should stay in place in landscape mode to avoid too long lines of text. This also underpins one expert’s general statement that on

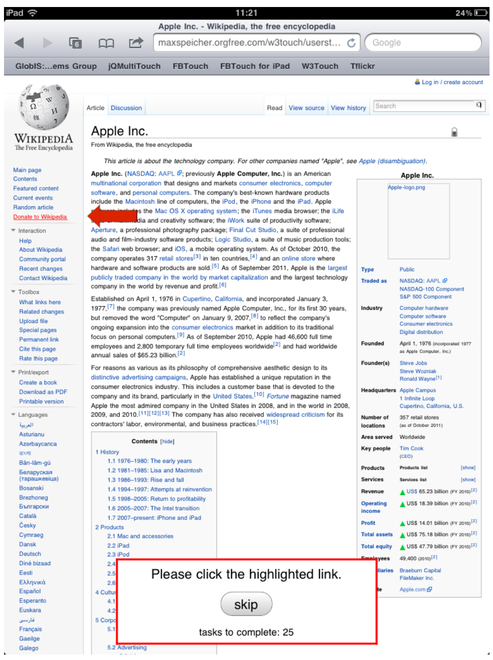
the iPad, "there's more room for improvement" simply because a developer is given more possibilities for adaptations if there is more screen space available.
Moreover, unlike on the iPod touch, font sizes were found to be good in landscape mode while they were generally considered only "a little small" in portrait mode. However, in contrast, most experts still pointed out that hyperlinks are comparably difficult to hit without having zoomed in before. This accounts especially for densely packed links, as e.g. present in the sidebar navigation and footer. Therefore, more spacing around the affected hyperlinks was suggested regarding all components of the webpage. In accordance to what was already reported for the iPod touch, the header component was again found to be the "best" part of the article and would need least adjustments.
More advanced suggestions for improvement included removing less important parts of the navigation and to let the sidebar navigation float to reduce scrolling efforts in vertical directions. One interviewee suggested dynamic sliders, so that a user would be able to adjust font sizes according to their choice. As for the iPod touch, pagination was mentioned again as a possible adjustment, which would help to reduce scrolling efforts.

Summary

To summarise our findings, the interviewed experts generally suggested that the Wikipedia article would need adjustments in terms of larger font sizes and more spacing around densely packed links, in order to better support access with touch devices. These propositions apply more to smaller devices, on which it is desirable to counter excessive zooming, that was found to be a major issue. Furthermore, while the suggested adaptations in general should be applied to all components in equal terms (except for the header), the experts proposed to rather focus on the main article text, as this is the main component of interest from the users' perspective.

4.3 User Study #1: Crowdsourced Adaptation of an Example Webpage

To evaluate W3Touch, we conducted an asynchronous remote usability study with two goals in mind. First, we wanted to collect touch-specific user activity data for the aforementioned Wikipedia article in a real crowdsourcing scenario to investigate which issues would be identified by our tool. Second, we wanted to define suitable adaptations based on the gathered data, while also taking into account the recommendations gained from the expert interviews, and investigate whether these could help improve the user experience. However, designing such a study posed certain challenges in the technical as well as the conceptual dimension. It was necessary to define a set of specific tasks that, while guiding a participant through the study, still guaranteed that the obtained results could be generalised to the overall touch interaction. Moreover, since we wanted to let the layout of the Wikipedia article evolve through user emancipation based on the collected interaction data, but still be able to control the study as much as possible, we decided to split the study into two distinct phases. During the first phase, it was intended to collect data necessary for adaptation while in the second phase, an adequate adaptation catalogue was applied and we again collected tracking data based on the same set of tasks for comparison.

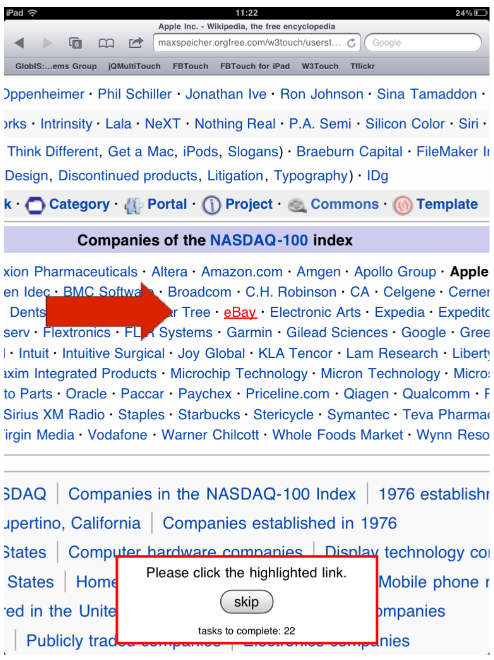

Apple Inc. - Wikipedia, the free encyclopedia
maxspeicher.orgfree.com/w3touch/userst...
Apple Inc.
Contents

Please click the highlighted link.

skip

tasks to complete: 25

Figure 4.6: *Click link* task (navigation).


Apple Inc. - Wikipedia, the free encyclopedia
maxspeicher.orgfree.com/w3touch/userst...
Companies of the NASDAQ-100 index

Please click the highlighted link.

skip

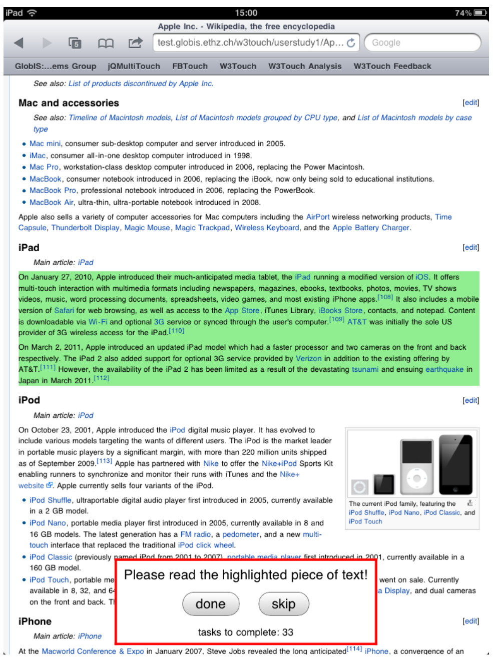
tasks to complete: 22

Figure 4.7: *Click link* task (footer).

We prepared a copy of the Wikipedia article on our server and enhanced it with W3Touch. Since participants were allowed to take part in the study using their own touch devices, we had to ensure as much as possible that our tools worked on a wide range of common devices and mobile browsers. Therefore, if particular hardware or software could not be supported (e.g. Firefox Mobile), we checked for this on an initial qualification page before redirecting the participant to our copy of the article.

4.3.1 Method

A total of 84 participants took part in the user study that were recruited via internal mailing lists and a number of social media services including *Facebook*, *Google+*, *LinkedIn*, *Twitter* and *Xing* among others. Each participant was presented the same set of 33 tasks that were randomised to counterbalance ordering effects. For each of the tasks, the webpage was automatically scrolled to the required position, which ensured that participants only had to focus on the actual interaction that was necessary to successfully complete the task. Moreover, corresponding instructions were displayed in a floating box, fix-positioned at the bottom centre of the viewport. We made sure that this instruction box always covered the same amount of screen space, independent of the current zoom level. However, it has to be noted that the box appeared slightly larger in landscape mode due to the different aspect ratio and less viewport height.
Participants were presented three different types of tasks, which are described below. Every user action (e.g. missing or hitting a link) was recorded along with the current zoom level, device orientation and time elapsed since the current task had started. For cases in which a user was not able to complete a task, they were given the possibility to skip it by clicking a button in the instruction box, which was logged accordingly.


iPad 15:00 74%

Apple Inc. - Wikipedia, the free encyclopedia

test.globis.ethz.ch/w3touch/userstudy1/Ap… Google

GlobIS:...ems Group jQMultiTouch FBTouch W3Touch W3Touch Analysis W3Touch Feedback

See also: List of products discontinued by Apple Inc.

Mac and accessories [edit]

See also: Timeline of Macintosh models, List of Macintosh models grouped by CPU type, and List of Macintosh models by case type

- Mac mini, consumer sub-desktop computer and server introduced in 2005.
- iMac, consumer all-in-one desktop computer introduced in 1998.
- Mac Pro, workstation-class desktop computer introduced in 2006, replacing the Power Macintosh.
- MacBook, consumer notebook introduced in 2006, replacing the iBook, now only being sold to educational institutions.
- MacBook Pro, professional notebook introduced in 2006, replacing the PowerBook.
- MacBook Air, ultra-thin, ultra-portable notebook introduced in 2008.

Apple also sells a variety of computer accessories for Mac computers including the AirPort wireless networking products, Time Capsule, Thunderbolt Display, Magic Mouse, Magic Trackpad, Wireless Keyboard, and the Apple Battery Charger.

iPad [edit]

Main article: iPad

On January 27, 2010, Apple introduced their much-anticipated media tablet, the iPad running a modified version of iOS. It offers multi-touch interaction with multimedia formats including newspapers, magazines, ebooks, textbooks, photos, movies, TV shows videos, music, word processing documents, spreadsheets, video games, and most existing iPhone apps.[108] It also includes a mobile version of Safari for web browsing, as well as access to the App Store, iTunes Library, iBooks Store, contacts, and notepad. Content is downloadable via Wi-Fi and optional 3G service or synced through the user's computer.[109] AT&T was initially the sole US provider of 3G wireless access for the iPad.[110]

On March 2, 2011, Apple introduced an updated iPad model which had a faster processor and two cameras on the front and back respectively. The iPad 2 also added support for optional 3G service provided by Verizon in addition to the existing offering by AT&T.[111] However, the availability of the iPad 2 has been limited as a result of the devastating tsunami and ensuing earthquake in Japan in March 2011.[112]

iPod [edit]

Main article: iPod

On October 23, 2001, Apple introduced the iPod digital music player. It has evolved to include various models targeting the wants of different users. The iPod is the market leader in portable music players by a significant margin, with more than 220 million units shipped as of September 2009.[113] Apple has partnered with Nike to offer the Nike+iPod Sports Kit enabling runners to synchronize and monitor their runs with iTunes and the Nike+ website. Apple currently sells four variants of the iPod.

The current iPod family, featuring the iPod Shuffle, iPod Nano, iPod Classic, and iPod Touch

- iPod Shuffle, ultraportable digital audio player first introduced in 2005, currently available in a 2 GB model.
- iPod Nano, portable media player first introduced in 2005, currently available in 8 and 16 GB models. The latest generation has a FM radio, a pedometer, and a new multi-touch interface that replaced the traditional iPod click wheel.
- iPod Classic (previously named iPod from 2001 to 2007), portable media player first introduced in 2001, currently available in a 160 GB model.
- iPod Touch, portable me went on sale. Currently available in 8, 32, and 6 a Display, and dual cameras on the front and back. T

iPhone [edit]

Main article: iPhone

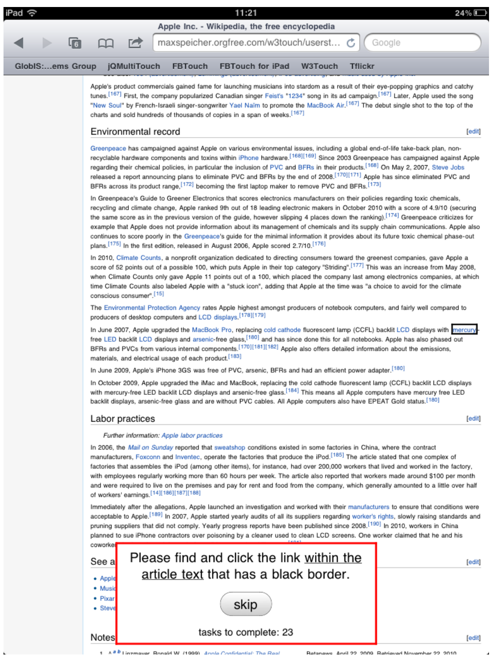
At the Macworld Conference & Expo in January 2007, Steve Jobs revealed the long anticipated[114] iPhone, a convergence of an

Please read the highlighted piece of text!

done skip

tasks to complete: 33

Figure 4.8: *Reading* task.


iPad 11:21 24%

Apple Inc. - Wikipedia, the free encyclopedia

maxspeicher.orgfree.com/w3touch/userst… Google

GlobIS:...ems Group jQMultiTouch FBTouch FBTouch for iPad W3Touch Tflickr

Apple's product commercials gained fame for launching musicians into stardom as a result of their eye-popping graphics and catchy tunes.[167] First, the company popularized Canadian singer Feist's "1234" song in its ad campaign.[167] Later, Apple used the song "New Soul" by French-Israeli singer-songwriter Yael Naïm to promote the MacBook Air.[167] The debut single shot to the top of the charts and sold hundreds of thousands of copies in a span of weeks.[167]

Environmental record [edit]

Greenpeace has campaigned against Apple on various environmental issues, including a global end-of-life take-back plan, non-recyclable hardware components and toxins within iPhone hardware.[168][169] Since 2003 Greenpeace has campaigned against Apple regarding their chemical policies, in particular the inclusion of PVC and BFRs in their products.[168] On May 2, 2007, Steve Jobs released a report announcing plans to eliminate PVC and BFRs by the end of 2008.[170][171] Apple has since eliminated PVC and BFRs across its product range,[172] becoming the first laptop maker to remove PVC and BFRs.[173]

In Greenpeace's Guide to Greener Electronics that scores electronics manufacturers on their policies regarding toxic chemicals, recycling and climate change, Apple ranked 9th out of 18 leading electronic makers in October 2010 with a score of 4.9/10 (securing the same score as in the previous version of the guide, however slipping 4 places down the ranking).[174] Greenpeace criticizes for example that Apple does not provide information about its management of chemicals and its supply chain communications. Apple also continues to score poorly in the Greenpeace's guide for the minimal information it provides about its future toxic chemical phase-out plans.[175] In the first edition, released in August 2006, Apple scored 2.7/10.[176]

In 2010, Climate Counts, a nonprofit organization dedicated to directing consumers toward the greenest companies, gave Apple a score of 52 points out of a possible 100, which puts Apple in their top category "Striding".[177] This was an increase from May 2008, when Climate Counts only gave Apple 11 points out of a 100, which placed the company last among electronics companies, at which time Climate Counts also labeled Apple with a "stuck icon", adding that Apple at the time was "a choice to avoid for the climate conscious consumer".[15]

The Environmental Protection Agency rates Apple highest amongst producers of notebook computers, and fairly well compared to producers of desktop computers and LCD displays.[178][179]

In June 2007, Apple upgraded the MacBook Pro, replacing cold cathode fluorescent lamp (CCFL) backlit LCD displays with mercury-free LED backlit LCD displays and arsenic-free glass,[180] and has since done this for all notebooks. Apple has also phased out BFRs and PVCs from various internal components.[170][181][182] Apple also offers detailed information about the emissions, materials, and electrical usage of each product.[183]

In June 2009, Apple's iPhone 3GS was free of PVC, arsenic, BFRs and had an efficient power adapter.[180]

In October 2009, Apple upgraded the iMac and MacBook, replacing the cold cathode fluorescent lamp (CCFL) backlit LCD displays with mercury-free LED backlit LCD displays and arsenic-free glass.[184] This means all Apple computers have mercury free LED backlit displays, arsenic-free glass and are without PVC cables. All Apple computers also have EPEAT Gold status.[180]

Labor practices [edit]

Further information: Apple labor practices

In 2006, the *Mail on Sunday* reported that sweatshop conditions existed in some factories in China, where the contract manufacturers, Foxconn and Inventec, operate the factories that produce the iPod.[185] The article stated that one complex of factories that assembles the iPod (among other items), for instance, had over 200,000 workers that lived and worked in the factory, with employees regularly working more than 60 hours per week. The article also reported that workers made around $100 per month and were required to live on the premises and pay for rent and food from the company, which generally amounted to a little over half of workers' earnings.[14][186][187][188]

Immediately after the allegations, Apple launched an investigation and worked with their manufacturers to ensure that conditions were acceptable to Apple.[189] In 2007, Apple started yearly audits of all its suppliers regarding worker's rights, slowly raising standards and pruning suppliers that did not comply. Yearly progress reports have been published since 2008.[190] In 2010, workers in China planned to sue iPhone contractors over poisoning by a cleaner used to clean LCD screens. One worker claimed that he and his coworke

See a [edit]

- Apple
- Music
- Pixar
- Steve

Notes [edit]

Please find and click the link within the article text that has a black border.

skip

tasks to complete: 23

Figure 4.9: *Find link* task.

Click Link Task

A particular text hyperlink, highlighted in red and further marked with an arrow pointing at it in a distance of 40 pixels, had to be clicked by the participant (cf. Figures 4.6 and 4.7). These measures were taken to not bother the participant with having to find the link prior to clicking it. However, we intentionally did not involve borders in highlighting the link in order to not reveal the bounding box of the touch area, which would have biased the user. All hyperlinks contained in the article were deactivated, except for internal links and the links used to show or collapse panels of links in the sidebar navigation (such as "Toolbox" or "Languages"). This made sure that one would not be redirected to webpages other than the Wikipedia article prepared for the study. Every time the participant failed to hit the intended link, this was recorded in the log by W3Touch, also taking into account whether a different link or the respective link underlay (cf. Section 3.1.1) was tapped. A total of 25 *click link* tasks that had been carefully selected to provoke these issues were distributed over the different components of the article page.

Reading Task

A particular section of the article text that the participant was asked to read was highlighted with a light green background colour (cf. Figure 4.8). Again, the webpage was automatically scrolled to the required position prior to starting the actual task and measuring the time needed for reading. The floating instruction box contained a "Done" button for this kind of task, that the participant had to hit once they had finished reading. Two pieces of text of similar length had to be read during the study, which were the section "iPad" (172 words) and the first two paragraphs of the section "Corporate Affairs" (188 words).

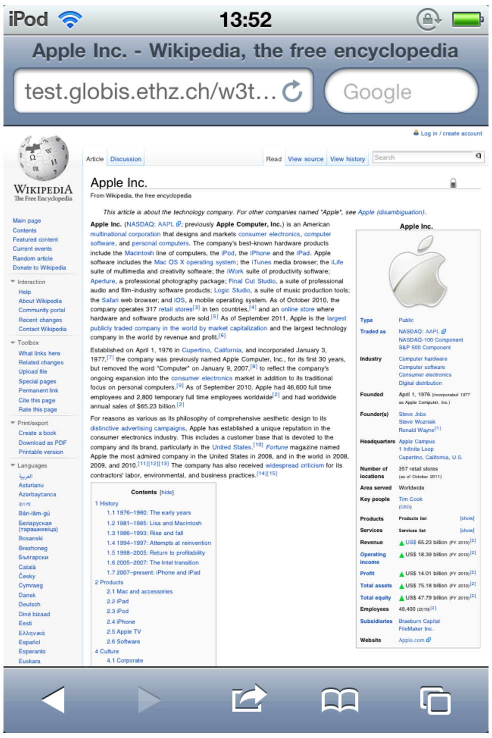
Find Link Task

The participant was asked to find a particular link within the article text that was bordered in black in order to make the highlighting not too obvious (cf. Figure 4.9). For this, the webpage was scrolled to the very top prior to starting the task and measuring the time until the link was found and tapped. Although not typical for interaction with Wikipedia articles, we included this type of task as a control task since we expected the article page to increase in length based on the adaptations that would be applied during the second phase of the study. Therefore, we wanted to measure the impact on user experience resulting from higher scrolling effort and a worse website overview. Four of these tasks were included in the study, with links selected from the top, middle and bottom of the main article text (“NeXT” in the section “1994–1997: Attempts at reinvention”, “iPod Shuffle” in the section “2007–present: iPhone and iPad”, “FaceTime” in the section “iPhone” and “Mercury” in the section “Environmental Record”).

The participants who completed all of the 33 tasks were presented a post-study questionnaire that asked them to rate statements concerned with the efficiency and easiness involved in solving the different types of tasks based on a 5-point Likert scale (5 = *strongly agree*, 1 = *strongly disagree*). Furthermore, in terms of background information specific for the study, we wanted to know which touch devices participants used to complete the study, how often they use these devices in general and for web browsing specifically and how they rate their expertise in the fields of web design, web development and web usability based on a 4-point scale (3 = *expert knowledge*, 0 = *no knowledge*). The complete post-study questionnaire can be found in Appendix C.1.
In case a participant aborted the user study before completing all of the 33 tasks and proceeding to the questionnaire, they still contributed valid data for the subset of tasks they completed. Therefore, these data are as well considered in the following analysis of results.

4.3.2 Results

Out of the 84 participants overall, 39 took part in the study during the first phase and 45 took part during the second phase. 50 completed the post-study questionnaire, 42 were male and 8 were female at a median age of 25. According to the background information given in the questionnaire, participants used their touch devices *several times a day* for both general purposes as well as for web browsing. Moreover, they were knowledgeable in web design and development (median = 2) while having passing knowledge in web usability (median = 1). A total of 64 participants used a smartphone to take part in the study, including iPhone, HTC Desire, Motorola Defy, Samsung Galaxy S and Nokia N9. The remaining 20 participants took part using a tablet PC, including iPad, Lenovo ThinkPad, Motorola Xoom and Archos 70. The results have been grouped according to these two high-level classes of devices, further divided by device orientation (portrait/landscape), which makes a total of four different use contexts to be considered. In this way, we take into account general trends in mobile device usage while not applying a too fine-grained differentiation and thus obtaining reasonable amounts of data for all considered contexts. We note that this is however not a technical limitation since the underlying context engine could distinguish at a much finer level (Nebeling and Norrie, 2011).
In the second phase of the study, the applied adaptations were mainly based on the zoom levels collected during phase 1 (cf. Appendix A.2). In fact, the Wikipedia article was adjusted on a



Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox
What links here
Related changes
Upload file
Special pages
Permanent link
Cite this page
Rate this page

Print/export
Create a book
Download as PDF
Printable version

Languages
العربية
Asturianu
Azərbaycanca
বাংলা
Bân-lâm-gú
Беларуская (тарашкевіца)
Bosanski
Brezhoneg
Български
Català
Česky
Cymraeg
Dansk
Deutsch
Diné bizaad
Eesti
Ελληνικά
Español
Esperanto
Euskara

Apple Inc.

From Wikipedia, the free encyclopedia

This article is about the technology company. For other companies named "Apple", see Apple (disambiguation).

Apple Inc. (NASDAQ: AAPL; previously **Apple Computer, Inc.**) is an American multinational corporation that designs and markets consumer electronics, computer software, and personal computers. The company's best-known hardware products include the Macintosh line of computers, the iPod, the iPhone and the iPad. Apple software includes the Mac OS X operating system; the iTunes media browser; the iLife suite of multimedia and creativity software; the iWork suite of productivity software; Aperture, a professional photography package; Final Cut Studio, a suite of professional audio and film-industry software products; Logic Studio, a suite of music production tools; the Safari web browser; and iOS, a mobile operating system. As of October 2010, the company operates 317 retail stores[3] in ten countries,[4] and an online store where hardware and software products are sold.[5] As of September 2011, Apple is the largest publicly traded company in the world by market capitalization and the largest technology company in the world by revenue and profit.[6]

Established on April 1, 1976 in Cupertino, California, and incorporated January 3, 1977,[7] the company was previously named Apple Computer, Inc., for its first 30 years, but removed the word "Computer" on January 9, 2007,[8] to reflect the company's ongoing expansion into the consumer electronics market in addition to its traditional focus on personal computers.[9] As of September 2010, Apple had 46,600 full time employees and 2,800 temporary full time employees worldwide[2] and had worldwide annual sales of $65.23 billion.[2]

For reasons as various as its philosophy of comprehensive aesthetic design to its distinctive advertising campaigns, Apple has established a unique reputation in the consumer electronics industry. This includes a customer base that is devoted to the company and its brand, particularly in the United States.[10] *Fortune* magazine named Apple the most admired company in the United States in 2008, and in the world in 2008, 2009, and 2010.[11][12][13] The company has also received widespread criticism for its contractors' labor, environmental, and business practices.[14][15]

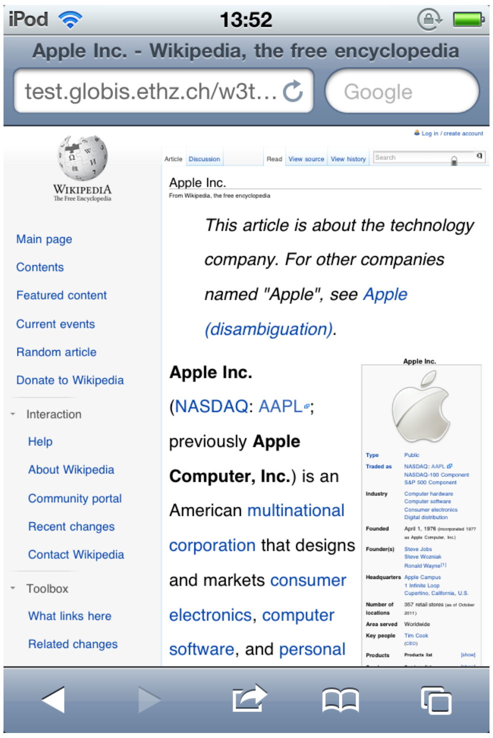
Contents [hide]

1 History
1.1 1976–1980: The early years
1.2 1981–1985: Lisa and Macintosh
1.3 1986–1993: Rise and fall
1.4 1994–1997: Attempts at reinvention
1.5 1998–2005: Return to profitability
1.6 2005–2007: The Intel transition
1.7 2007–present: iPhone and iPad
2 Products
2.1 Mac and accessories
2.2 iPad
2.3 iPod
2.4 iPhone
2.5 Apple TV
2.6 Software
4 Culture
4.1 Corporate

Apple Inc.

Type	Public
Traded as	NASDAQ: AAPL NASDAQ-100 Component S&P 500 Component
Industry	Computer hardware Computer software Consumer electronics Digital distribution
Founded	April 1, 1976 (incorporated 1977 as Apple Computer, Inc.)
Founder(s)	Steve Jobs Steve Wozniak Ronald Wayne[1]
Headquarters	Apple Campus 1 Infinite Loop Cupertino, California, U.S.
Number of locations	357 retail stores (as of October 2011)
Area served	Worldwide
Key people	Tim Cook (CEO)
Products	Products list [show]
Services	Services list [show]
Revenue	▲ US$ 65.23 billion (FY 2010)[2]
Operating income	▲ US$ 18.39 billion (FY 2010)[2]
Profit	▲ US$ 14.01 billion (FY 2010)[2]
Total assets	▲ US$ 75.18 billion (FY 2010)[2]
Total equity	▲ US$ 47.79 billion (FY 2010)[2]
Employees	49,400 (2010)[2]
Subsidiaries	Braeburn Capital FileMaker Inc.
Website	Apple.com

Figure 4.10: The regular version of the Wikipedia article on an iPod touch.



Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox
What links here
Related changes

Apple Inc.

From Wikipedia, the free encyclopedia

This article is about the technology company. For other companies named "Apple", see Apple (disambiguation).

Apple Inc. (NASDAQ: AAPL; previously **Apple Computer, Inc.**) is an American multinational corporation that designs and markets consumer electronics, computer software, and personal

Apple Inc.

Type	Public
Traded as	NASDAQ: AAPL NASDAQ-100 Component S&P 500 Component
Industry	Computer hardware Computer software Consumer electronics Digital distribution
Founded	April 1, 1976 (incorporated 1977 as Apple Computer, Inc.)
Founder(s)	Steve Jobs Steve Wozniak Ronald Wayne[1]
Headquarters	Apple Campus 1 Infinite Loop Cupertino, California, U.S.
Number of locations	357 retail stores (as of October 2011)
Area served	Worldwide
Key people	Tim Cook (CEO)
Products	Products list [show]

Figure 4.11: The adapted version of the Wikipedia article on the same device.

per-component basis taking into account the zoom factors to scale the font sizes accordingly (cf. Figures 4.10 and 4.11). However, the resulting font sizes for sidebar navigation and footer component were weighted at 67% only since the interviewed experts considered the article text to be the main component of interest. Also, the navigation bar would otherwise have taken too much horizontal space, particularly on small-screen devices such as smartphones. In addition to a larger touch area through bigger font sizes, we set the line height of hyperlinks to 2 em (or 133% of the standard value) for additional spacing, as is also done in the mobile version of the Wikipedia, for which many of the interviewed experts noted that it contains good adaptations for touch.

The header component is excluded from the results presented below because it was not adapted for the second phase of the study. This is based on the fact that its layout involved fixed-size images and would therefore be difficult to adjust using only features of W3Touch. However, this is still in line with the expert suggestions, where the header was considered to be the "best" part of the regular webpage and thus requires least adaptations.

Zoom Factors and Missed Links

In Figure 4.12, we present the average zoom levels collected for the main article text during phase 1 of the study. As can be seen, the text was zoomed by an average factor of 2.73 on smartphones if used in portrait mode, and by an average factor of 1.95 if used in landscape mode. We detected comparably lower zoom levels on tablet PCs, where participants zoomed by an average factor of 1.78 in portrait mode and by an average factor of 1.05 in landscape

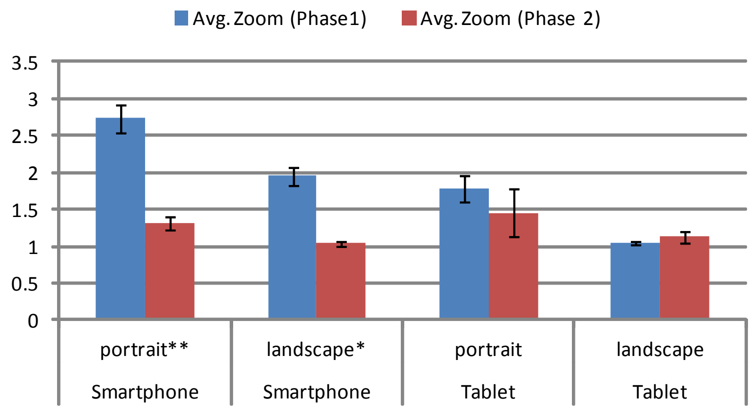


Figure 4.12: Average zoom levels for the article text. (* $p < 0.01$; ** $p < 0.001$)

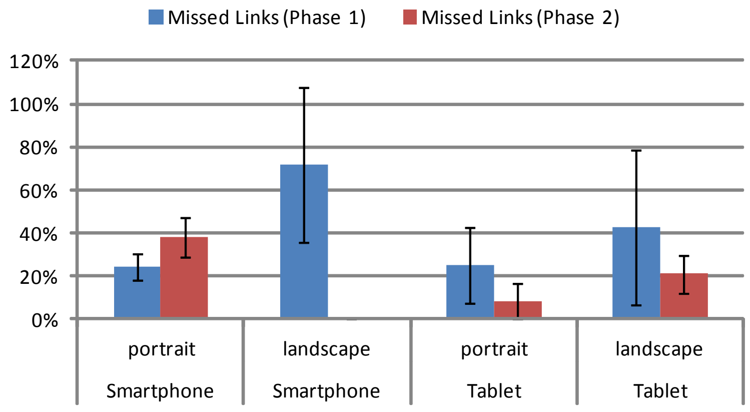


Figure 4.13: Missed links per *click link* task for the article text.

mode. Regarding missed links per task in the article text, we recorded fairly high rates for all four contexts (cf. Figure 4.13). The intended target was missed in 25 out of 102 cases (25%) on smartphones held in portrait mode while even missing in 20 out of 28 (71%) cases in landscape mode. Similar results account for tablet PCs, where the link to be hit was missed in 25% (5 out of 20) of the cases in portrait mode and in 43% (12 out of 28) of the cases in landscape mode. It can be seen that the ratio of missed links tends to be higher in landscape mode, although it yields better readability at the same zoom level, as compared to portrait mode. An explanation for this could be that participants zoomed less (and therefore probably not enough) when being in landscape mode, thus raising the chance of missing links.
In general, the results concerning average zoom levels in phase 1 of the study are very similar for all components. In contrast, the results concerning missed link differ considerably between components and also feature a greater statistical dispersion. Since these are omitted here, the interested reader might refer to Appendix C.2 for the complete results.
When comparing the tracking data collected in the first phase with that collected during the second phase, the average zoom levels used for viewing the article text went down significantly on smartphones, independent of the orientation (cf. Figure 4.12). This shows a positive effect of the font size enlargement. In general, we found significant differences on smartphones for sidebar navigation and footer component as well. While we could not detect significant results when comparing the average values recorded on tablet PCs—probably due to the smaller number of participants and the fact that the crowdsourced adaptations were least drastic—most of the differences suggest that the average zoom factors might decrease when using the adapted version of the article. When reviewing the differences between missed link ratios, no significant results can be found (cf. Figure 4.13), except for the footer component, where smartphones in portrait mode performed significantly worse. While there were also some better missed link ratios recorded during phase 2—especially in the sidebar navigation and article text—which might indicate that some links can be hit more precisely, this also shows that the adaptations applied to hyperlinks were not enough and might need, e.g. additional padding to further increase touch areas.

Reading

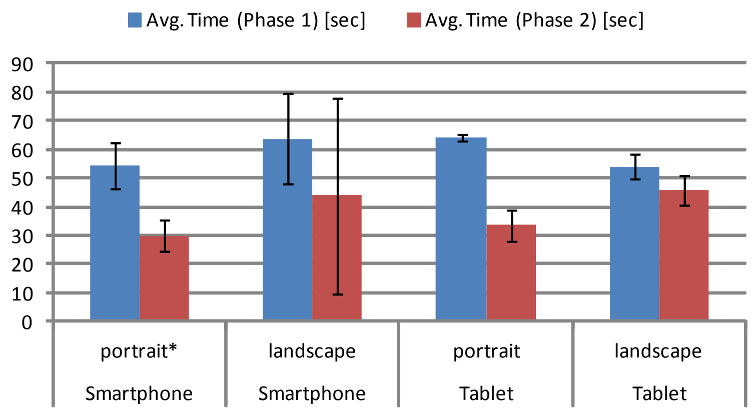


Figure 4.14: Average reading times for the “iPad” text section. (* $p < 0.05$)

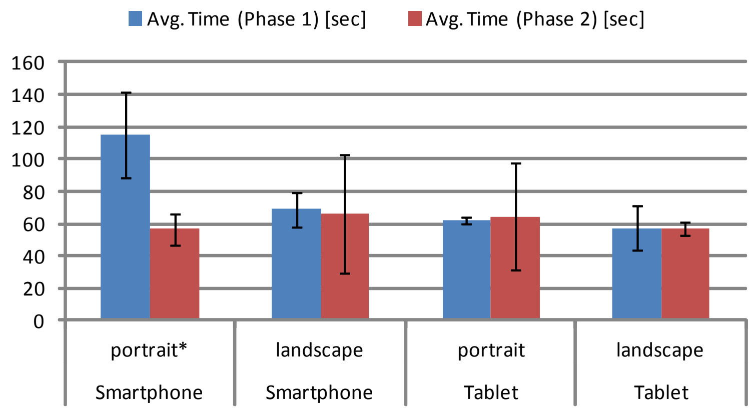


Figure 4.15: Average reading times for the "Corporate Affairs" text section. (* $p < 0.01$)

Figure 4.14 shows the average reading times collected for the "iPad" section of the Wikipedia article. As can be seen, during the first phase, participants needed about 54 seconds on smartphones in portrait and tablet PCs in landscape mode while the average times for smartphones in landscape and tablet PCs in portrait mode were about 64 seconds. In phase 2, these times dropped to 30 seconds (smartphone/portrait), 44 seconds (smartphones/landscape), 34 seconds (tablet/portrait) and 46 seconds (tablet/landscape), whereas only the improvement on smartphones in portrait mode was statistically significant. However, these results indicate that the reading performance was better in the second phase of the study due to the larger initial font size that resulted in less zooming and horizontal scrolling, especially concerning the small screens of smartphones.
The data collected for the second piece of text to be read (cf. Figure 4.15) are considerably different from those described above, although both sections were almost equal in length. In particular, the average times on smartphones in portrait mode—although showing a significant improvement—are about twice as high, as compared to the "iPad" section. Moreover, although the average times in phase 1 are similar between the two text sections for the remaining three contexts, times collected in the second phase did not decrease as much regarding the "Corporate Affairs" text. Reasons for this might include outliers in combination with small sample sizes (e.g. $N = 3$ for smartphone/landscape with a range of $[13.869; 135.879]$), as is also indicated by the high statistical dispersions. However, while we cannot assume significant improvements on tablet PCs—where also the interviewed experts found the default font sizes to be already comparably good—our results still suggest that reading performance can be improved particularly on small-screen devices such as smartphones with our crowdsourced version of the Wikipedia article.
Concerning the information we collected about device orientations, it could be seen that participants preferred portrait mode on smartphones during the first phase, except for the reading tasks. In contrast, participants primarily used portrait mode for reading in phase 2, which might indicate that font sizes were already of a good enough size without switching to landscape mode. Regarding tablet PCs, the majority of participants preferred landscape mode for all tasks during both phases.

Finding Links

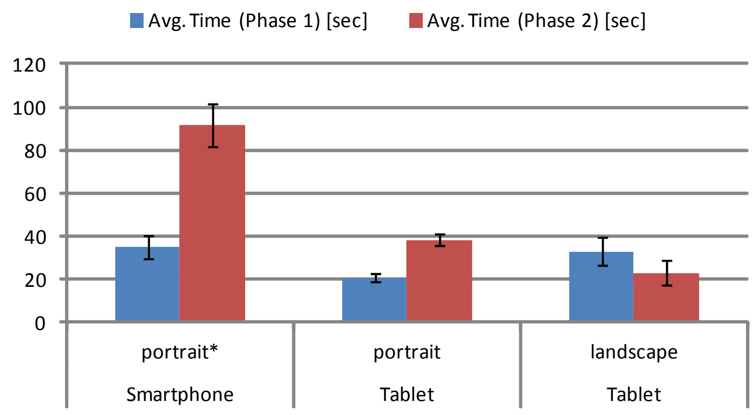


Figure 4.16: Average times needed for finding the link “FaceTime”. (* $p < 0.001$)

Figure 4.16 shows results for average times needed to find a particular link within the main text in terms of the link “FaceTime”, that is roughly located in the middle of the article. In particular, the average time increased for tablet PCs and even significantly for smartphones, both in portrait mode, while decreasing for tablet PCs in landscape mode. The smartphone/landscape context is missing since it was not used in the second phase of the study. These results are very similar to those of the remaining links to be found, whereas the difference is also significant for the link from the top part of the article (“NeXT”) on smartphones in portrait mode. Interestingly, the data for tablet PCs in landscape mode suggest improvements, with one difference being significant (“NeXT”). This shows that, while the vertical scrolling effort increased as expected on smartphones and tablet PCs in portrait mode, the crowdsourced adaptations were least drastic on tablet PCs in landscape mode, which was also the context considered to be best in the expert interviews. The complete data for the *find link* tasks can be found in Appendix C.2.

Perceived User Experience

While above we aimed at measuring user experience based on several metrics, participants were also asked in the post-study questionnaire to rate the easiness and efficiency of each type of task (cf. Appendix C.1). Considering smartphone users (cf. Figure 4.17), we could not find differences between the two phases regarding the *click link* tasks and only a tendency that participants endorsed the adaptations concerning the *reading* tasks. However, the perceived easiness and efficiency of the *find link* tasks decreased significantly, which underpins our findings above regarding the increased efforts in vertical scrolling.
Concerning tablet PC users (cf. Figure 4.18), there is no clear trend regarding the *click link* tasks since the perceived easiness improved while the perceived efficiency decreased slightly. Also, there is not much of a difference between the two phases of the study regarding the *reading* tasks, which might be due to the fact that ratings were already very good during phase 1. In contrast to smartphones, the perceived user experience concerning the *find link* tasks did not decrease significantly.

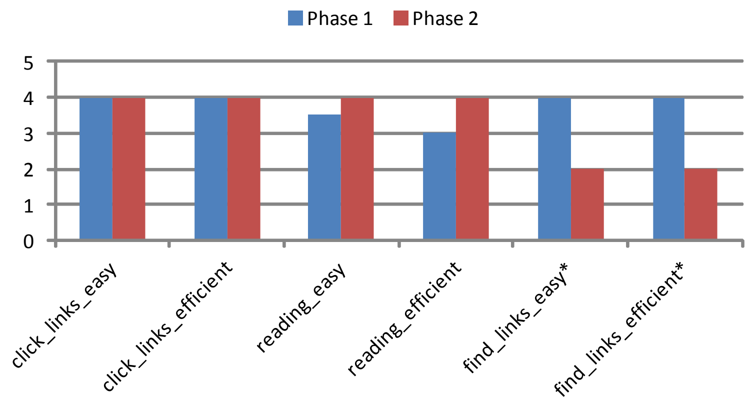


Figure 4.17: Perceived user experience of smartphone users. (* $p < 0.01$)

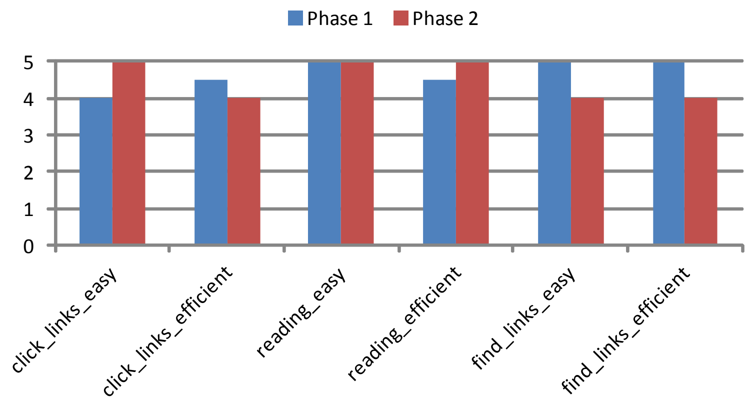


Figure 4.18: Perceived user experience of tablet PC users.

In general, the findings drawn from perceived user experience support the results of our quantitative measurements. By this we mean that the touch areas of links were not enlarged enough to yield significant improvements while participants tend to experience better reading comfort due to larger font sizes, less zooming and less horizontal scrolling. Moreover, increased scrolling effort caused by the adaptations affects the user experience particularly on smartphones. Finally, participants show the tendency to perform better on tablet PCs, where also the differences between the two phases of the study were smaller due to less drastic adaptations as compared to smartphones.

4.4 User Study #2: Comparison to Existing Mobile Version

The goal of the second user study was to compare the regular and adapted Wikipedia articles from the first study to the existing mobile version of the same article that is optimised for smartphones. This happened in terms of measures based on particular tasks described below. The main consideration behind this comparison was that a crowdsourced adaptation involving W3Touch requires comparably low effort by the developer, whereas a version specifically designed for mobile touch devices includes a much higher amount of manual work.

4.4.1 Method

The usability tests were carried out in terms of a follow-up lab study for which 13 participants were recruited. They were provided an iPod touch 4G (running iOS 4) for completing the study. We chose three sections of text that a participant had to find and read, each one using a different version of the Wikipedia article. The order of versions was randomised and counterbalanced to avoid ordering effects. Also, which piece of text had to be found and read using which version was randomised and counterbalanced to ensure evenly distributed conditions. Furthermore, we prepared a particular text section in which participants had to sequentially click all the links contained, whereas the current link to be clicked was highlighted and underlined in red. This particular section had to be found and processed using each of the versions of the article, resulting in a total of two sections to be found per version, of which one was prepared to click links and one had to be read. The order in which the two sections had to be processed was randomised and counterbalanced to avoid ordering effects. Unlike in the first user study, we did not take into account hyperlinks from the sidebar navigation and footer component of the desktop version since these were not contained in the mobile version of the Wikipedia article.
The chosen text sections were taken from the first, second, third and fourth quarter of the main article text (“1981–1985: Lisa and Macintosh”, “2007–present: iPhone and iPad”, “Culture” and “Environmental Record”). From the times needed to find these, we computed, for each version of the article, one average time needed for finding a section based on all involved pieces of text. The section “1981–1985: Lisa and Macintosh” was prepared for clicking links while the remaining three had to be read. Therefore, the sections were adjusted to be similar in length (489, 498 and 490 words respectively) and equal in terms of contents across all three article versions.
Concerning each text section to be read, participants were told to read carefully since they had to answer five questions afterwards (to be found in Appendix D.1). It was not necessary to remember answers by heart, but rather, participants had to point at the corresponding sentence

within the text and read the correct answer aloud. This decision was made since people who had preliminary knowledge about Apple products could have known some answers even without reading the text, but in this way were required to pay equal attention to reading and answering as participants without the same knowledge. We randomised the order of the questions to ensure that participants had to scroll through the text and could not answer in a linear manner. The times needed for reading and answering were recorded as a measure regarding user experience. However, we did not measure the time needed for clicking links since the different versions of the article yielded different latencies for detecting a click, which made corresponding measures incomparable. Instead, for each clicked or missed link[6], we recorded the zoom level as a measure regarding user experience. Finally, independent of particular tasks, we recorded the device orientation with which a task was accomplished.
Participants had to fill out a questionnaire prior to the study that asked for demographic information as well as background information specific for the user study. In particular, we asked for providing information on how often participants use their touch device(s) in general and specifically for web browsing. Moreover, we wanted to know whether they were familiar with the regular and/or mobile version of the Wikipedia and how they would rate their knowledge concerning web design, web development and usability/HCI based on 4-point scales (4 = *expert*, 1 = *no knowledge*). Additionally, after having finished the tasks for a version of the article, participants had to fill out a post-task questionnaire in which they rated statements about the perceived user experience as well as interaction with the webpage in general, based on 5-point Likert scales. In particular, we asked how efficient and easy the tasks were to solve[7] and whether in general they had a good webpage overview (1 = *strongly disagree*, 5 = *strongly agree*). Furthermore, we asked about the effects of the involved amounts of zooming and scrolling (1 = *strongly agree*, 5 = *strongly disagree*). The complete questionnaires can be found in Appendix D.2.

4.4.2 Results

Of the 13 participants, 9 were male and 4 were female at an average age of 29 years. In general, they used their touch device(s) *several times a day,* and *several times a week* for web browsing. 12 of the participants were familiar with the regular version of Wikipedia and 5 were familiar with the mobile version; one did not provide feedback on these questions. Participants were *knowledgeable* in web design and development and had *passing knowledge* in usability/HCI.

Finding Sections

We found significant differences between the three versions of the article when trying to find a particular section within the text (cf. Figure 4.19). In particular, the difference between desktop and crowdsourced version results from the larger amount of vertical scrolling necessary in the latter one, due to the scaled font sizes. Contrary to the desktop and crowdsourced versions, the mobile version of Wikipedia employs a different approach concerning the table of contents, in which only the first-level headings are shown and the corresponding sections

[6]Based on hyperlink underlays (cf. Section 3.1.1).

[7]Except for answering questions since easiness and efficiency of answering a question does not only rely on the layout of the corresponding text, but also on the individual participant's ability to remember certain text passages.

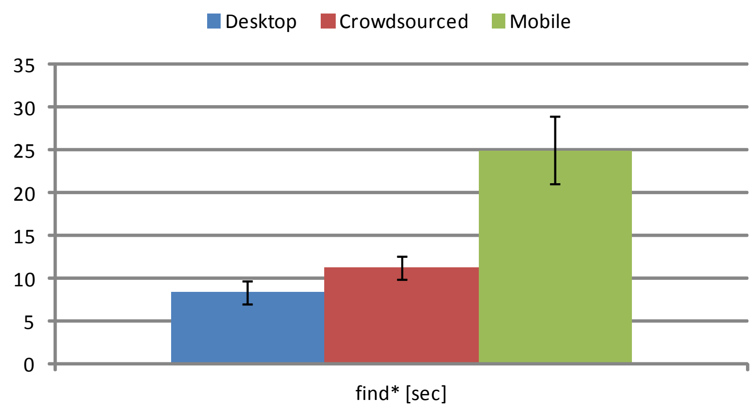


Figure 4.19: Average times needed for finding a particular section within the article. (* see Appendix D.3 for pair-wise significances)

and subsections can then be shown/collapsed in-place if wanted. We took into account that the different tables of contents would most probably affect the results for finding particular sections. However, it is not possible to remove the table of contents from the mobile version where it is tightly coupled with the actual text. Since we also did not want to confuse users familiar with the Mobile Wikipedia by initially extending all collapsed sections, we decided not to alter any version of the article.

Reading Text & Answering Questions

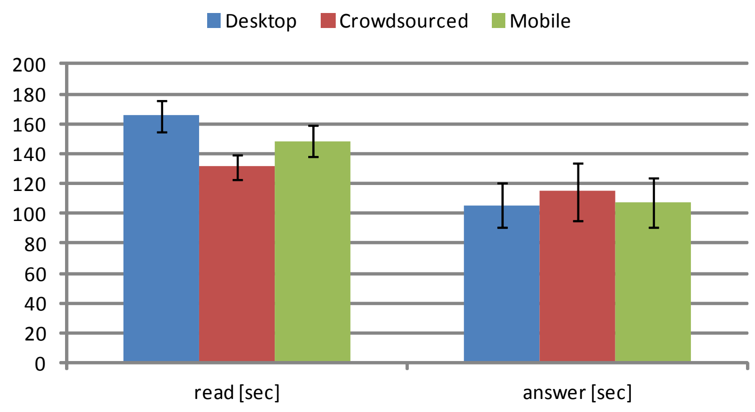


Figure 4.20: Average times needed for reading a particular section and answering 5 text-related questions.

The results concerning text sections that had to be read by the participants did not yield significant differences between the different versions of the article (cf. Figure 4.20). However, they still suggest that reading requires most time when using the regular Wikipedia since zooming in to a readable font size leads to horizontal scrolling, which is not the case for the crowdsourced and mobile versions. Furthermore, no clear tendencies can be extracted from the times needed to answer text-related questions.

Clicking Links

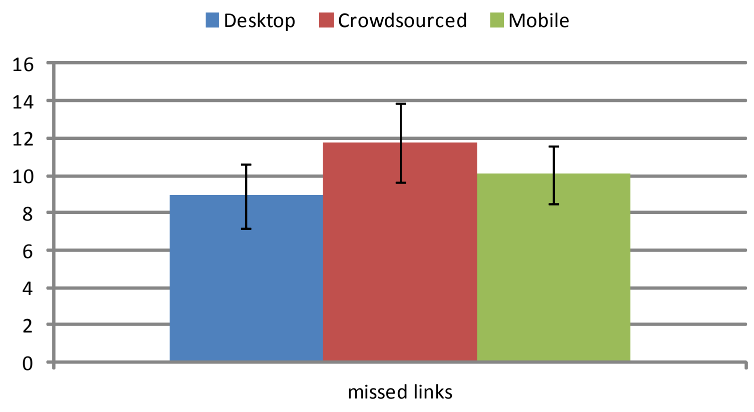


Figure 4.21: Average numbers of missed links.

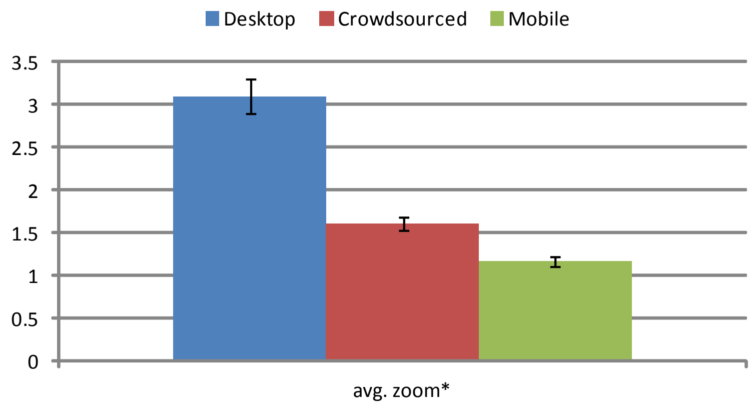


Figure 4.22: Average zoom levels for clicking links. (* see Appendix D.3 for pair-wise significances)

When reviewing the average zoom levels collected during clicking links, significant differences between all versions of the article can be found (cf. Figure 4.22). In combination with the average amounts of missed links, which yield no significant differences (cf. Figure 4.21), this shows that for each version, users zoomed in as much as necessary to be able to easily hit all the links. Therefore, the finding to be drawn from this is, that it is most difficult to hit links in the desktop version (although the measured amount of missed links was the smallest) while it is easiest in the mobile version, with the crowdsourced version laying in between.

Device Orientation

Regarding the device orientations used for completing the different tasks (cf. Table 4.1), we found that portrait mode was preferred for finding particular sections within the article. While reading took place mostly in landscape mode in the desktop version, the preference switched to portrait mode when using the crowdsourced or mobile version, which can be explained by the initially better readable font sizes. The same tendency accounts for answering questions. Finally, in terms of clicking links, the orientations used were almost equal in the desktop version with a slight preference for portrait mode regarding the crowdsourced and mobile versions. In general, landscape mode was least popular in the mobile version.

Table 4.1: Numbers of times the two device orientations were used for completing a task, separated by article version.

#	Desktop		Crowdsourced		Mobile	
	portrait	landscape	portrait	landscape	portrait	landscape
find	22	4	18	8	22	4
read	2	11	8	5	11	2
answer	5	8	8	5	11	2
click links	7	6	8	5	9	4

Perceived User Experience

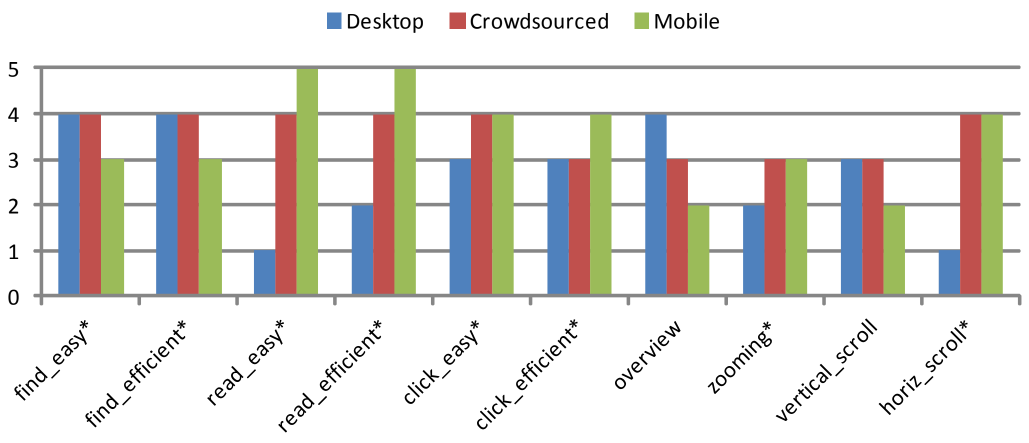


Figure 4.23: Perceived user experience based on post-task questionnaires. (* see Appendix D.3 for pair-wise significances)

Based on the post-task questionnaires filled out by the participants, Figure 4.23 shows the perceived user experience in terms of easiness and efficiency of the tasks to be solved (except for answering questions) as well as other aspects important for interaction, including website overview and the amount of zooming. In particular, participants gave lower ratings for finding a particular section of text in the mobile version, which can be explained by the different table of contents that provides less details. When it comes to reading, the crowdsourced and mobile versions have been rated significantly higher than the desktop versions, which shows that participants appreciated that no horizontal scrolling was required. However, there was no statistical difference between the two higher-ranked versions in this respect. Regarding clicking links, participants found the mobile version to be significantly better as compared to the other versions. This might be due to the lower amount of zooming involved, as shown by the measured zoom factors. Concerning website overview and amount of vertical scrolling, we could not find significant differences between all of the versions with medians ranging from 2 to 4, which suggests that neither version is optimal in these respects. Finally, participants in general stated that the crowdsourced and mobile versions affected their efficiency way less by zooming and horizontal scrolling. This stands in contrast to the desktop version, which got median ratings of 2 and 1 respectively. The ratings given for the reading task, where the crowdsourced and mobile versions perform significantly better as well, clearly support this finding.

Overall, the second study indicated that the adaptations applied in the crowdsourced version can compete with the existing mobile version in several aspects, which is very positive given the fact that they involve considerably less effort for the developer. First of all, participants found it (from a statistical perspective) equally good to read a text on both the crowdsourced and mobile version, because both initially render text at a readable size without the necessity of horizontal scrolling. This is also supported by the measured reading times, which show no significant differences. While overview and amount of vertical scrolling can not be considered to be optimal in both the crowdsourced and mobile versions, participants found that the crowdsourced version yields the same advantages in terms of less zooming and horizontal scrolling, as compared to the mobile version. A negative point regarding the adapted article is clearly the performance in clicking links, which is underpinned by the measured zoom factors necessary for a successful hit as well as the questionnaire feedback. The Mobile Wikipedia was found to be clearly better in this respect.
When furthermore comparing the results of the two user studies, it is possible to find several similarities. First, the second study underpinned our findings that the adaptations applied to the crowdsourced version of the article result in less zooming and horizontal scrolling efforts, which is indicated by both the measured zoom factors and the perceived user experience. Moreover, although the measured times from the second study do not show significant differences, the questionnaire feedback tells that reading is to be considered better in the crowdsourced version of the article, as compared to the desktop version. This seems to be also in line with the findings from the first study. In terms of clicking links, both studies showed that the applied adaptations could not lead to significant improvements. Finally, it has to be noted that no task equivalent to the *find link* task from the first study was included in the second one since finding an entire section of text is less difficult (due to its size and headline) than finding one particular inline link. Therefore, we cannot compare results in this regard.

4.5 jQMetrics4touch

To develop metrics for the evaluation of webpages on touch devices, we built on jQMetrics, that implement a set of metrics originally designed by Nebeling et al. (2011) to assess text-intensive webpages, such as news sites, in large-screen contexts. However, since large screens pose considerably different requirements to developers, in contrast to mobile touch devices, we had to perform a series of adjustments to be able to obtain reasonable results. Moreover, we compute the metrics for a "snapshot" of a webpage, i.e. the current content of the mobile browser viewport, rather than for the entire content of a page, which is reasonable given the limited processing power of mobile devices.

In the first instance, individual metrics that were specific for large screens (e.g. by focusing on "wasted" screen real estate) and therefore would not contribute to an assessment on mobile touch devices, were deactivated. The affected metrics were *content-window ratio, wide text ratio* and *visible links ratio*. In particular, the visible links ratio was removed since on touch devices, it is more important that links displayed in the current viewport are easily hittable rather than being able to view as many links as possible where the user has no chance to precisely tap on a desired target. This consideration is in line with the results of the expert interviews and first user study. Moreover, we deactivated the *media-content ratio* since the webpages to be assessed were Wikipedia articles and we therefore focused on aspects of text rather than images or videos.

The metrics to be left activated were *document-window ratio, small text ratio* and *visible text ratio*, whereas we modified the document-window ratio to only consider the horizontal dimension, thus effectively yielding the *horizontal scroll factor*, which is equal to the zoom level. We decided to ignore the vertical dimension here, because the vertical scroll factor can be seen as a possible approximation for the amount of content currently contained in the browser viewport. However, since we intended the assessment of a text-intensive webpage, this measure is already logically contained in the visible text ratio and can therefore be left out. Moreover, we modified the small text ratio to be based on the text currently displayed in the viewport rather than being based on the entire text of the document. Therefore, if all of the displayed text can be considered to be not too small, the value should be 1.

The above set of metrics was complemented by two new metrics specifically designed for mobile touch devices. First, we added a *small links ratio* in analogy to the small text ratio to cater for the novel input modalities involving less precise fingers instead of mouse cursors. Second, we introduced *overall quality*, that is an overall value based on the other four metrics and should make it possible to compare user experience approximations for different snapshots of webpages using a single value. In particular, the overall quality is determined by the following formula (whereas the small links ratio is only considered if links are contained in the viewport; otherwise also the divisor is changed to 5):

$$\frac{\text{visible text ratio}_N + 2 \cdot \text{small text ratio}_N + 2 \cdot \text{horizontal scroll factor}_N + \text{small links ratio}_N}{6}$$

A subscript N thereby indicates that the respective value has been normalised to a value between 0 and 1, if not already the case. For example, the normalised value of the horizontal scroll factor is computed by dividing 1 by the original value, thus obtaining a normalised value that converges to 0 for large scroll factors. The small text ratio is weighted twice, because we consider it more important to have readable text than to have a large amount of text that the user cannot read. This is in accordance with the findings of the interviewed experts who stated that the initial view of the regular Wikipedia is completely useless on an iPod touch, although it displays a comparably large amount of text. On the other hand, if a text is zoomed in too much and not too small, but more "useless" than at a reasonable zoom

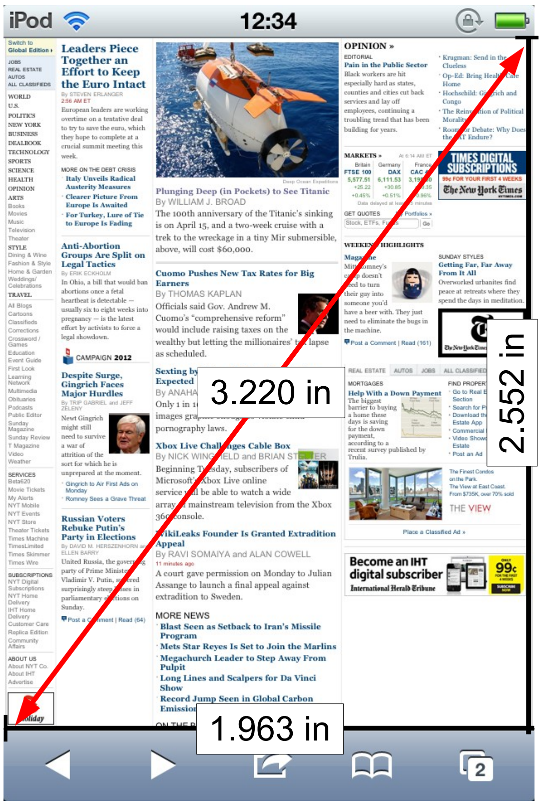
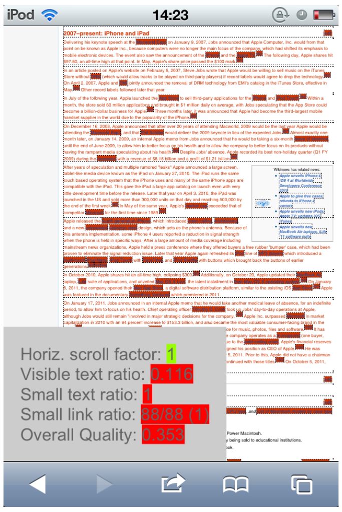


Figure 4.24: Browser viewport diagonal on an iPod touch.

level, this is counterbalanced by the horizontal scroll factor which gets worse the more a user zooms in. Therefore, the horizontal scroll factor is weighted twice as well.

4.5.1 Implementation

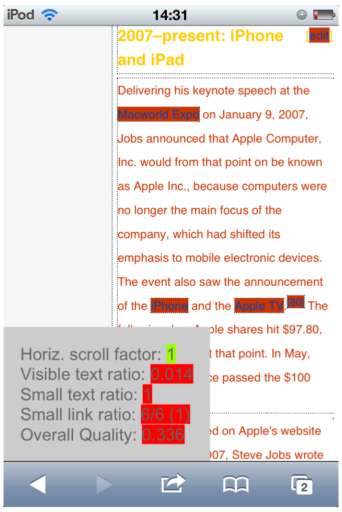
In order to determine what a *small link* or *small text* is, appropriate thresholds needed to be defined beforehand. Concerning links, we oriented at the findings of Holz and Baudisch (2011), who state that the average offset error resulting from touch input is 4 mm, or 0.157 in. Therefore, as a first simple heuristic, we consider links with an absolute height of less than 0.157 in to be *small*.
Concerning text, we asked the participants of the second user study (described in Section 4.4) to adjust the zoom level of a sample page showing a large amount of text on an iPod touch 4G. The goal was to adjust it in such a way that they would say the size of the characters is optimal for reading in terms of being neither too small nor too large. The threshold resulting from this was a character height of 0.115 in (with a standard deviation of 0.026 in). Therefore, again as a first simple heuristic, we consider text with an absolute character height of less than 0.115 in to be *small*.
When computing the absolute heights of links or characters in terms of inches for comparison with the respective thresholds, it is not possible to rely on the given resolution of a device, which is, e.g. 163 pixels per inch for an iPod touch 2G. Rather, the viewport of a mobile browser effectively displays different amounts of pixels, depending on the current zoom level.



2007–present: iPhone and iPad

Horiz. scroll factor: 1
Visible text ratio: 0.116
Small text ratio: 1
Small link ratio: 88/88 (1)
Overall Quality: 0.353

(a) Desktop version.



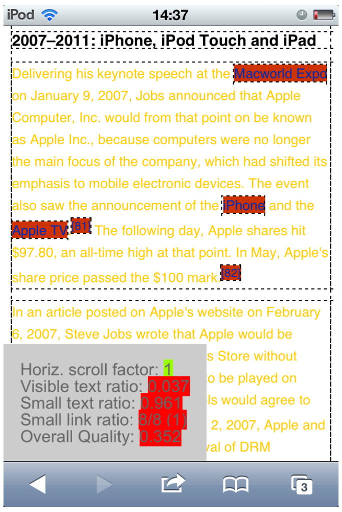
2007–present: iPhone and iPad edit

Delivering his keynote speech at the Macworld Expo on January 9, 2007, Jobs announced that Apple Computer, Inc. would from that point on be known as Apple Inc., because computers were no longer the main focus of the company, which had shifted its emphasis to mobile electronic devices. The event also saw the announcement of the iPhone and the Apple TV.[80] The following day, Apple shares hit $97.80, an all-time high at that point. In May, Apple's share price passed the $100

In an article posted on Apple's website on February 6, 2007, Steve Jobs wrote

Horiz. scroll factor: 1
Visible text ratio: 0.014
Small text ratio: 1
Small link ratio: 6/6 (1)
Overall Quality: 0.336

(b) Crowdsourced version.



2007–2011: iPhone, iPod Touch and iPad

Delivering his keynote speech at the Macworld Expo on January 9, 2007, Jobs announced that Apple Computer, Inc. would from that point on be known as Apple Inc., because computers were no longer the main focus of the company, which had shifted its emphasis to mobile electronic devices. The event also saw the announcement of the iPhone and the Apple TV.[81] The following day, Apple shares hit $97.80, an all-time high at that point. In May, Apple's share price passed the $100 mark.[82]

In an article posted on Apple's website on February 6, 2007, Steve Jobs wrote that Apple would be

Horiz. scroll factor: 1
Visible text ratio: 0.037
Small text ratio: 0.961
Small link ratio: 8/8 (1)
Overall Quality: 0.352

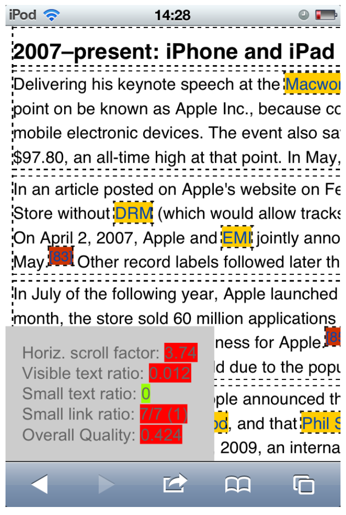
(c) Mobile version.

Figure 4.25: Comparison between quality of initial views.

For example, when displaying a website designed for desktop screens that has a fixed width of 800 px, Mobile Safari would initially scale the page to fit the width of the small screen, thus squeezing 800×1040 pixels into the browser viewport in portrait mode. Then, when zooming in to read a piece of text, these values could change to, e.g. 320×416 pixels that are effectively displayed. Obviously, the absolute height of a character depends on this variable resolution of the browser viewport, which can be obtained using `window.innerWidth` and `window.innerHeight` in Mobile Safari. Moreover, it is necessary to take into account the browser viewport diagonal, which is 3.220 in on an iPod touch in portrait mode (cf. Figure 4.24) and 3.337 in in landscape mode, to calculate the effectively displayed pixels per inch within the viewport (which might be different from 163 pixels per inch) and compute absolute heights of links and characters based on this value.

4.5.2 Results

We have taken two metrics snapshots for each investigated version of the Wikipedia article (desktop, crowdsourced and mobile); one for the initial view without having zoomed the page and one when zoomed in to a piece of text, so that it can be easily read according to the given threshold. Regarding the latter one, we tried to zoom in to a similar absolute character size in all versions.
The evaluation of the initial views (cf. Figure 4.25) shows no considerable differences between the three versions. In particular, the horizontal scroll factors and small links ratios are 1 in all cases and the small text ratios lie above 96% for all snaphots. Therefore, the small variations in overall quality are effectively determined by the amount of visible text, which has the highest value in the desktop version.
The font sizes in the crowdsourced and mobile versions are considered to be too small based on the threshold determined by the participants of the second user study. This is interesting since both versions feature significantly larger font sizes than the desktop version and in particular, the Mobile Wikipedia has been designed to provide readable text without zooming.



2007–present: iPhone and iPad

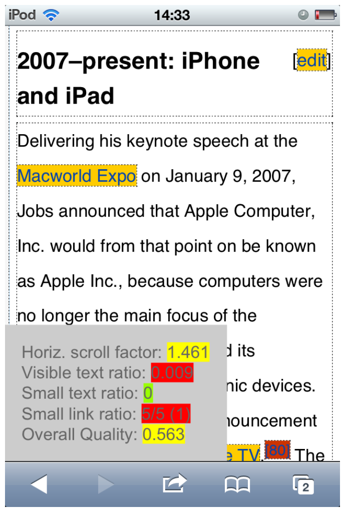
Delivering his keynote speech at the Macwo
point on be known as Apple Inc., because cc
mobile electronic devices. The event also sa
$97.80, an all-time high at that point. In May,

In an article posted on Apple's website on Fe
Store without DRM (which would allow tracks
On April 2, 2007, Apple and EMI jointly anno
May.[83] Other record labels followed later th

In July of the following year, Apple launched
month, the store sold 60 million applications
ness for Apple.[8
d due to the popu
ple announced th
od, and that Phil S
2009, an interna

Horiz. scroll factor: 3.74
Visible text ratio: 0.012
Small text ratio: 0
Small link ratio: 7/7 (1)
Overall Quality: 0.424

(a) Desktop version.

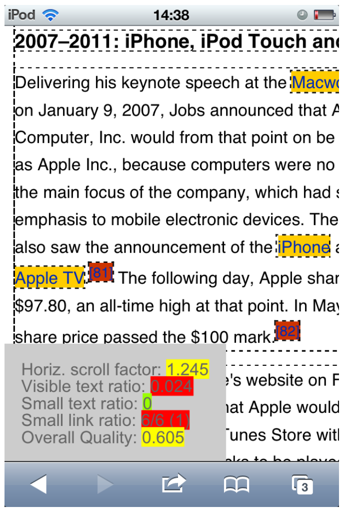


2007–present: iPhone and iPad [edit]

Delivering his keynote speech at the
Macworld Expo on January 9, 2007,
Jobs announced that Apple Computer,
Inc. would from that point on be known
as Apple Inc., because computers were
no longer the main focus of the
d its
nic devices.
nouncement
e TV.[80] The

Horiz. scroll factor: 1.461
Visible text ratio: 0.009
Small text ratio: 0
Small link ratio: 5/5 (1)
Overall Quality: 0.563

(b) Crowdsourced version.



2007–2011: iPhone, iPod Touch an

Delivering his keynote speech at the Macw
on January 9, 2007, Jobs announced that A
Computer, Inc. would from that point on be
as Apple Inc., because computers were no
the main focus of the company, which had s
emphasis to mobile electronic devices. The
also saw the announcement of the iPhone a
Apple TV.[81] The following day, Apple shar
$97.80, an all-time high at that point. In May
share price passed the $100 mark.[82]
's website on F
at Apple would
unes Store witl

Horiz. scroll factor: 1.245
Visible text ratio: 0.024
Small text ratio: 0
Small link ratio: 6/6 (1)
Overall Quality: 0.605

(c) Mobile version.

Figure 4.26: Comparison between quality of zoomed-in views.

However, this shows that the threshold for small text needs further adjustments. In case it would be changed accordingly, the initial view of the mobile version could earn a significantly better rating similar to the one in Figure 4.26(c), given that the small text ratio would be 0% in this case.

Concerning the zoomed-in views of the versions of the Wikipedia article (cf. Figure 4.26), in which the text is readable according to the pre-defined threshold, the rating of the crowdsourced version (56.3%) is slightly worse than that of the mobile version (60.5%). However, both versions are clearly better than the regular Wikipedia (42.4%). In particular, the small links ratios stay at 100% since the threshold for small links is larger than the one for small text. Threfore, the overall quality is effectively determined by horizontal scroll factor and visible text ratio. Clearly, the mobile version earns the best rating since it yields least horizontal scrolling and most visible text (collapsed text does not count towards this value) while the crowdsourced version performs slightly worse in both. The bad rating of the desktop version is mainly due to its extensive horizontal scroll factor.

We have also investigated snapshots of initial and zoomed-in views in landscape mode. However, these are omitted here since the results are almost equal to those described above in all respects.

Finally, we want to note that the metrics have been specifically developed for assessing the different versions of the Wikipedia and their use is still highly *experimental*. Therefore, the above results should be interpreted with caution. In particular, the thresholds for small text and small links, which were meant to be first simple heuristics, need further adjustments. This is underpinned by the fact that there were effectively no differences between the metrics for initial views. However, it would be especially desirable if the Mobile Wikipedia received a higher rating compared to the desktop version since it clearly features better readable text without zooming.

5 Conclusions

In this thesis, we presented W3Touch—a tool to support crowdsourced evaluation and adaptation of non-optimised web interfaces concerning the specifics of mobile touch devices. The basic idea is to extend existing approaches to usability evaluation with more advanced means for tracking high-level context-aware data that is related to touch input. Based on this information about users' activities, our tool tries to infer critical components of a webpage and to inform suitable adaptations for various contexts to improve the user experience. This happens in terms of visualisations to provide information about potentially critical and often used components to the developer as well as automatic adjustments. Regarding the great diversity of today's touch devices, which makes it difficult for web developers to cater for all of these, we have shown that implicit crowdsourcing can help to optimise an existing web interface for many different contexts with typically less manual efforts than required by existing mobile websites. Thereby, responsibilities are split between users—who deliver the relevant data—and developers—who make use of that data through an adaptation catalogue that can contain basic rules or even rearrange a whole web interface.

5.1 Contributions

In a user study that investigated a real crowdsourcing scenario, we have shown that it is possible to realise a self-adjusting webpage for several browsing contexts that yields improvements in user experience—particularly in terms of less zooming and horizontal scrolling—by applying one basic adaptation catalogue. This adaptation catalogue was equivalent to the one shown in Appendix A.2 and already led to good results although containing only relatively simple rules. While an internal lab study underpinned our findings, it also showed that our crowdsourced web interface could already compete with an existing mobile website to a certain extent. This is especially positive regarding the fact that individual mobile websites usually involve expensive and time-consuming manual work to cater for various browsing contexts, as opposed to defining more general rules and letting a non-optimised web interface evolve through user emancipation. In general, W3Touch could help to provide

touch-optimised websites for a wider range of browsing contexts, although they might not be as optimised as specifically developed mobile websites. Moreover, crowdsourcing web interfaces with the help of W3Touch can save a considerable amount of work for developers, especially regarding the fact that the diversity of devices used for web access tends to be steadily increasing.
Finally, we have extended a set of usability metrics to further support developers with assessing their web interfaces on touch devices. The resulting set of metrics can complement W3Touch by delivering additional evaluations of the aspects of a webpage that are of particular interest when it comes to touch input.

5.2 Limitations

Concerning the evaluation of W3Touch, several points have to be noted:

1. In the first user study, participants had to click a "Done" button provided in the instruction box when having finished reading a text. However, in some rare cases, extraordinarily short or long reading times suggest that the respective participant might not have found the button (probably because they did not thoroughly read the instructions) or clicked it instead of the "Skip" button when they did not want to complete the task. Since this influences average values and statistical dispersion and therefore also the quality of our results, we introduced the necessity to answer questions in the second user study as a stimulus to read texts more carefully. Yet, in an asynchronous remote usability test, this poses additional technical challenges and still cannot help to guarantee that participants pay more attention to instructions.

2. The enlargement of hyperlink touch areas by increasing font sizes could not lead to clear improvements in successfully hitting links, as our results do not show significances in this respect. Therefore, additional spacing in terms of CSS padding would have been necessary. However, this poses certain challenges concerning inline links contained in continuous text. In particular, individual horizontal or vertical padding for those links leads to odd occurrences of white space which affect consistent line spacing and aesthetic appearance of text paragraphs. Regarding this, also reading comfort could suffer from larger touch areas.

3. The lack of significant results concerning tablet PCs is caused by two main reasons. First, the number of participants taking part in the first study using these particular devices was comparably small, thus yielding too much statistical dispersion. And second, the Wikipedia article needed least adaptations for the corresponding browsing contexts (i.e. portrait and landscape). Therefore, the differences in user experience between the regular and the crowdsourced version tend to be smaller, as compared to smartphones. This makes a larger amount of data necessary to be able to obtain significant results, if possible at all.

5.3 Future Work

Since W3Touch is currently only implemented in terms of a *proof of concept* and is related to comparably novel fields of research in terms of crowdsourcing and touch input modalities, there are several aspects that require improvement, further investigation or could not be covered in this thesis.

- So far, W3Touch has only been evaluated for text-centric webpages, as represented by an exemplary Wikipedia article, while taking into account only relatively basic adaptations that were, however, adequate for the investigated webpage. Therefore, further investigations are necessary, that involve other types of websites, such as web portals, and also different and more complex adaptations. In particular, future work should take into account more advanced ideas for adjustment, such as rearranging the overall layout to a single column, removing less important items from the navigation bar, or reorganising links based on their popularity. All of these have already been mentioned in the expert interviews.

- The reduced amount of zooming which results from the adaptations applied in the crowdsourced version leads to a considerably larger amount of vertical scrolling. Since excessive scrolling can become a major usability issue, future work could investigate possibilities to mitigate this shortcoming.

- Although W3Touch is able to recognise scrolling gestures, we currently make use of this only in terms of position changes of zoomed-in viewports, which are handled the same way as *zooming gestures*, also concerning visualisations. However, since extensive scrolling in both directions has to be considered a usability issue, it should be investigated how scrolling gestures could be visualised and contribute to the identification of potentially critical components. Moreover, additional tracking data could be provided in the adaptation catalogue that are based on scrolling interaction to answer questions like “How many changes in scrolling direction occurred with a specific component being present in the viewport?”

- Since W3Touch splits responsibilities between users and developers, evaluation should not only happen from the users’ perspective, i.e. “Can users benefit from crowdsourced adaptations in terms of usability?” Therefore, the developers’ perspective could be investigated in terms of exemplary deployments of W3Touch on real-world websites. This would help to assess the feasibility of W3Touch for developers, especially in terms of using the adaptation catalogue.

- As described above, in the current state it is the responsibility of the developer to provide an adaptation catalogue, or at least basic default behaviour defined in the W3Touch configuration. However, it could also be possible to set up a server-side installation of W3Touch on an arbitrary domain and let users enhance webpages with W3Touch through bookmarklets (similar to the approach described by Nebeling and Norrie, 2011). Regarding this, a user could manage their own adaptation catalogues, or public adaptation catalogues could be set up for different webpages, which would be a scenario similar to *Greasemonkey* user scripts[1].

- Currently, the adaptation catalogue provides only a rather limited amount of predefined context information, i.e. smartphone vs. tablet PC and portrait vs. landscape.

[1]`http://userscripts.org/`

However, to provide more flexible and fine-grained contexts, the catalogue could be extended to a fully-fledged domain-specific "language", taking into account the principles of XCML (Nebeling et al., 2012a). This would in particular also affect the context engine, as it delivers the context information that are available from within the catalogue as well as the configuration, that is responsible for default behaviour.

- We have extended a set of usability metrics to be able to assess webpages on touch devices. However, these metrics are still experimental and have only been tested with the different version of the Wikipedia article. Therefore, jQMetrics4touch requires additional fine-tuning, taking into account various types of websites. In particular, this accounts for the thresholds defining small text and small links as well as for the *overall quality* formula.

Bibliography

Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the User's Every Move – User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In *Proc. WWW*, 2006.

Patrick Baudisch, Bongshin Lee, and Libby Hanna. Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. In *Proc. AVI*, 2004a.

Patrick Baudisch, Xing Xie, Chong Wang, and Wei-Ying Ma. Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. In *Proc. UIST*, 2004b.

Michael S. Bernstein, Greg Little, Robert C. Miller, Bjrn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. Soylent: A Word Processor with a Crowd Inside. In *Proc. UIST*, 2010.

Timothy W. Bickmore and Bill N. Schilit. Digestor: Device-independent Access to the World Wide Web. In *Proc. WWW*, 1997.

Nilton Bila, Troy Ronda, Iqbal Mohomed, Khai N. Truong, and Eyal de Lara. PageTailor: Reusable End-User Customization for the Mobile Web. In *Proc. MobiSys*, 2007.

Orkut Buyukkokten, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Power Browser: Efficient Web Browsing for PDAs. In *Proc. CHI*, 2000.

Orkut Buyukkokten, Oliver Kaljuvee, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Efficient Web Browsing on Handheld Devices Using Page and Form Summarization. *TOIS*, 20(1), 2002.

Tonio Carta, Fabio Paternò, and Vagner Figuerêdo de Santana. Web Usability Probe: A Tool for Supporting Remote Usability Evaluation of Web Sites. In *Proc. INTERACT*, 2011.

Andre Charland and Brian LeRoux. Mobile Application Development: Web vs. Native. *CACM*, 54(5), 2011.

Yu Chen, Wei-Ying Ma, and Hong-Jiang Zhang. Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In *Proc. WWW*, 2003.

Krzysztof Z. Gajos, Jacob O.Wobbrock, and Daniel S. Weld. Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In *Proc. CHI*, 2008.

Gen Hattori, Keiichiro Hoashi, Kazunori Matsumoto, and Fumiaki Sugaya. Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information. In *Proc. WWW*, 2007.

Christian Holz and Patrick Baudisch. Understanding Touch. In *Proc. CHI*, 2011.

Jason I. Hong, Jeffrey Heer, Sarah Waterson, and James A. Landay. WebQuilt: A Framework for Capturing and Visualizing the Web Experience. In *Proc. WWW*, 2001.

Jeff Howe. The Rise of Crowdsourcing. *Wired*, 14(6), 2006.

Ben Lafreniere, Andrea Bunt, Matthew Lount, Filip Krynicki, and Michael Terry. Adaptable-GIMP: Designing a Socially-Adaptable Interface. In *Proc. UIST (Poster)*, 2011.

Heidi Lam and Patrick Baudisch. Summary Thumbnails: *Readable* Overviews for Small Screen Web Browsers. In *Proc. CHI*, 2005.

Luis A. Leiva. Restyling Website Design via Touch-based Interactions. In *Proc. MobileHCI*, 2011.

Luis A. Leiva and Enrique Vidal. Assessing Users' Interactions for Clustering Web Documents: A Pragmatic Approach. In *Proc. HT*, 2010.

Michael Nebeling and Moira C. Norrie. Tools and Architectural Support for Crowdsourced Adaptation of Web Interfaces. In *Proc. ICWE*, 2011.

Michael Nebeling and Moira C. Norrie. jQMultiTouch: A Lightweight Toolkit and Rapid Prototyping Framework for Multi-touch/Multi-device Web Interfaces, 2012. Submitted for review.

Michael Nebeling, Fabrice Matulic, and Moira C. Norrie. Metrics for the Evaluation of News Site Content Layout in Large-Screen Contexts. In *Proc. CHI*, 2011.

Michael Nebeling, Michael Grossniklaus, Stefania Leone, and Moira C. Norrie. XCML: Providing Context-Aware Language Extensions for the Specification of Multi-Device Web Applications. *WWW*, pages 1–35, 2012a. DOI: 10.1007/s11280-011-0152-2.

Michael Nebeling, Maximilian Speicher, and Moira C. Norrie. Methods and Tools for Crowdsourced Evaluation and Adaptation of Web Interfaces for Touch, 2012b. In submission.

Jeffrey Nichols, Brad A. Myers, Michael Higgins, Joseph Hughes, Thomas K. Harris, Roni Rosenfeld, and Mathilde Pignol. Generating Remote Control Interfaces for Complex Appliances. In *Proc. UIST*, 2002.

Fabio Paternò, Carmen Santoro, and Lucio Davide Spano. MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *TOCHI*, 16(4), 2009.

Willian Massami Watanabe, Renata Pontin de Mattos Fortes, and Maria da Graça Campos Pimentel. The Link-Offset-Scale Mechanism for Improving the Usability of Touch Screen Displays on the Web. In *Proc. INTERACT*, 2011.

Appendices

A
Source Code

A.1 Example Configuration

```
var W3TConfig = {

  path: 'http://www.yourdomain.com/w3touch',

  excludeFromTracking: ['header', 'footer'],

  excludeFromAdaptation: ['header'],

  useDefaultBehaviour: true,

  useAsCommonAdaptations: false,

  defaultBehaviour: function($, selector) {
    if (W3T.isSmartphone()) {
      $(selector).css('font-size', '150%');
    } else {
      $(selector).css('font-size', '100%');
    }
  }

};
```

A.2 Example Adaptation Catalogue

```
function getCssValue(selector, property) {
  return parseInt($(selector).eq(0).css(property).replace(/px/, ''));
}

var oldFontSize, oldWidth, zoom;


/* Navigation */

oldFontSize = getCssValue('#mw-panel', 'font-size');
oldWidth = getCssValue('#mw-panel', 'width');
zoom = W3T.getTrackingData('#mw-panel').avg_zoom;

// weighting of adaptation
zoom = (zoom * 0.67 >= 1) ? (zoom * 0.67) : 1;

$('#mw-panel').css({
  fontSize: (oldFontSize * zoom) + 'px',
  width: (oldWidth * zoom) + 'px'
});

$('#mw-panel a, #mw-panel h5').css('line-height', '2em');

$('#content, #mw-head-base').css('margin-left',
   (oldWidth * zoom) + 'px'
);

$('#left-navigation').css('left',
   (oldWidth * zoom) + 'px'
);


/* Main Text */

oldFontSize = getCssValue('#main-article', 'font-size');
zoom = W3T.getTrackingData('#main-article').avg_zoom;

$('#main-article').css({
  fontSize: (oldFontSize * zoom) + 'px',
  lineHeight: '2em'
});


/* Footer */

oldFontSize = getCssValue('#collapsibleTable0', 'font-size');
zoom = W3T.getTrackingData('#collapsibleTable0').avg_zoom;

// weighting of adaptation
zoom = (zoom * 0.67 >= 1) ? (zoom * 0.67) : 1;

$('#collapsibleTable0').css('font-size',
   (oldFontSize * zoom) + 'px'
);
```

```
oldFontSize = getCssValue('#collapsibleTable1', 'font-size');
zoom = W3T.getTrackingData('#collapsibleTable1').avg_zoom;

// weighting of adaptation
zoom = (zoom * 0.67 >= 1) ? (zoom * 0.67) : 1;

$('#collapsibleTable1').css('font-size',
  (oldFontSize * zoom) + 'px'
);

$('#collapsibleTable0 a, #collapsibleTable1 a').css('line-height', '2em');
```

B Expert Interviews

Expert #1	iPod touch	iPad
Header	no need for adaptation, zooming is enough	
Navigation	increase size of links	
Article Text	increase size of links	
Footer	no need for adaptation, zooming is enough; not enough space available if links were bigger	increase size of links according to users choice
Hyperlinks		links have proper size, only increase in portrait mode
General	website too small in general	initial zoom level is appropriate; too much scrolling; “There’s more room for improvement on the iPad.”
Advanced Suggestions	pagination would be helpful; collect links separately per paragraph; equalize horizontal and vertical scrolling	pagination; sort links in footer; zoom individual components instead of whole site (e.g., using a slider)

Expert #2	iPod touch	iPad
Header		fine (large touch areas)
Navigation		zooming necessary (difficult to hit exact line); easier if clustered horizontally
Article Text	when zoomed in, text lines don't fit screen (major issue)	difficult to hit links that are close to each other; zoom in only for difficult links, otherwise no "harm" when missing a link
Footer		"forget about hitting a particular link"; zooming necessary & worth the effort b/c densely packed; costly to hit wrong link; more vertical spacing; larger box for each link (e.g., grid)
Hyperlinks	issues similar to iPad	
General	used portrait mode only; not able to do anything at default zoom level; would expect to adapt to mobile version; increase font size or zoom automatically; zooming more of an issue than scrolling b/c less precise; scrolling (active usage) is perceived to be faster than navigating (short delay); difficult for designer to know constraints of platform	used portrait mode only; def. zoom level not optimal, but able to read text; article very long
Advanced Suggestions	create columns of text that fit screen width; initial viewport that shows 1st paragraph + headline; separate gallery for images that repeats relevant parts of text; pagination might be solution for long text, but likes scrolling more	detect zoom gesture on imgs, then adjust resolution (async); back-to-top links

Expert #3	iPod touch	iPad
Header	“could probably stay”; adjust font size to text	
Navigation	adjust font-size to text; maybe add spacing b/c of high link density	readable w/o larger font size; increase spacing b/c of link density
Article Text	adapt to screen width; increase font size for better readability w/o zooming	more spacing around densely packed links (e.g., footnote links)
Footer	issues w/ font size (smaller than in main text) and density of links; adjust font size to text; add padding / margin	density still very high; increase spacing
Hyperlinks	better hittable w/ larger font size; additional spacing only necessary for densely packed links	
General	same adaptations for portrait and landscape; difficult to find things even if you know where they are (excessive zooming necessary); scrolling is possible issue w/ larger font size; Wikipedia mobile leaves out certain features that one would like to use	used landscape mode; able to read text immediately w/o zooming; even footer is readable; issues w/ length of article / scrolling; font size could be kept; certain features should not be hidden from mobile users (“not fair to exclude mobile users from editing Wikipedia”); portrait might need different adaptations than landscape (e.g., move navigation); landscape similar to desktop experience; text has enough space in landscape, otherwise lines would be too long
Advanced Suggestions	place navigation bar on top, maybe w/ floating button for fast access (more space for text); leave out less important parts (e.g., languages) & put them into drop-down menu; collapse parts of text; pagination; use lightbox feature for imgs, show caption only in fullscreen mode	collapse parts of text; pagination; use lightbox feature for imgs; leave out less important parts (e.g., languages) & put them into drop-down menu

Expert #4	iPod touch	iPad
Header	too small, but still the easiest to click on whole page; difficult to read even in landscape mode	no real issue; could be a bit optimized for touch; easier to click than the other links on the page
Navigation	lager font size; spread out links more; "more forgiving for those of us with fat fingers"	should be a bit more spread out; font size a little small
Article Text	have font size large enough to be readable w/o zooming; increase line-spacing; use easy-to-read font	links very close; larger font size, more line-spacing / padding
Footer	finger covers about 20 links; hitting links even difficult w/ maximum zoom; use more space for each link (e.g., grid); larger font; more spacing	small, very close links; "my finger covers about 6 words"; more structured layout (e.g., grid) would be important; give links more space
Hyperlinks	same issues as on iPad, but worse	
General	mostly used landscape mode; portrait: completely unreadable; landscape: slightly better, but still not readable w/o zooming; zooming makes horizontal scrolling necessary (panning in 2 directions is annoying); bigger font size necessary	layout a bit narrow in portrait mode (give text more space); text a little small to read in portrait mode; very long article; most statements apply to portrait & landscape (limited width applies more for portrait)
Advanced Suggestions	simplify layout (use full width for content); rearrange navigation in horizontal way; zoom in and have text adjusted to screen width (no horizontal scrolling necessary); overlay w/ zoomed-in img of higher resolution; collapse navigation panel and expand on demand	overlay w/ zoomed-in img of higher resolution; use navigation bar on top for most important links; difference between portrait & landscape could be solved using media queries

Expert #5	iPod touch	iPad
Header	fine; large touch area	
Navigation	always stays at top (user has to scroll back); links easily visible; easy to hit links with help of zooming; no changes necessary	collapse links are nice
Article Text	too small; even difficult to read in landscape; if zooming, text is outside margins	
Footer	too much information to go through; only rarely used	
Hyperlinks	used to zooming in and then clicking a link w/o problems	no issues in portrait & landscape
General	would never read such a long text on iPod but rather look for keywords only; huge amount of scrolling / zooming involved; page is only fine if you know what you're looking for; would like to remove unnecessary information from page; zooming is not an issue (focus on certain parts of text); less content would give the user a better overview	better readability; would probably not use zooming very often; would use iPad to really read this page; you lose overview when you have to scroll a lot; portrait would need more adaptations than landscape
Advanced Suggestions	collapse paragraphs like in mobile Wikipedia; CTRL+F; fit text to width of viewport; reorganize footer based on viewport; use dropdown menu instead of TOC	floating navigation (less scrolling); CTRL+F; feature that helps to remember where you stopped reading (maybe pagination); adjust text according to viewport in portrait mode; remove parts that you have already read (keep track of what you did)

Expert #6	iPod touch	iPad
Header		portrait: link size too small
Navigation		doesn't use much space in landscape mode; links too small; enlarge touch areas
Article Text	even zoomed-in text is too small if line width fits viewport	portrait: small font-size and links; size of links not as important (focus on text)
Footer		"I never click on that."
Hyperlinks		portrait: too small, zooming necessary
General	used landscape mode (usual way of reading an article); horizontal scrolling is an issue; try to use all the available space; crucial to make sure that text is readable; different versions for portrait & landscape would not make a big change; zooming even less convenient than on iPad	all interactive things must have a reasonable size (avoid zooming); slower performance than dedicated apps; try to use all the available space; scrolling on tablet is convenient (no need for pagination); "You have to be aware of the target device." (would have separate smartphone & tablet app)
Advanced Suggestions	move navigation to top (make use of drop-down menus); fit viewport to width of text; single-column layout; less options in navigation; proxy touch points wouldn't have enough space on an iPod; goal must be interaction w/ one thumb (usual way of holding a smartphone in portrait mode); interaction w/ two thumbs easier on tablet; fast-scrolling feature (e.g., using article preview)	one column, collapsed paragraphs; fewer menu options; proxy touch points for links; full-screen version of imgs (after tapping); menu button (cf. Android) instead of navigation; remove less important items from navigation; landscape: floating left column navigation; portrait: move to top (e.g., using pull-down menus); back-to-top feature (maybe using gestures); menu options when page swiped to left / right (cf. Fennec); article preview for fast scrolling

C User Study #1

C.1 Post-study Questionnaire

Your Feedback

Final step: Please provide feedback in terms of the 10 short questions below!

Use of touch devices

1. How often do you use your touch device(s)?

several times a day	once a day	several times a week	once a week	less than once a week
○	○	○	○	○

2. How often do you browse the Web using your touch device(s)?

several times a day	once a day	several times a week	once a week	less than once a week
○	○	○	○	○

3. Which touch device have you used to complete the user study?

Your feedback

4. Please assess the following statments with respect to *clicking links*.

I found it easy to click the links.

Strongly agree	○	○	○	○	○	Strongly disagree

I felt efficient at clicking the links.

Strongly agree	○	○	○	○	○	Strongly disagree

5. Please assess the following statments with respect to *reading text*.

I found it easy to read the pieces of text.

Strongly agree	○	○	○	○	○	Strongly disagree

I felt efficient at reading the pieces of text.

Strongly agree	○	○	○	○	○	Strongly disagree

---

6. Please assess the following statments with respect to *finding details in images*.

I found it easy to find the details that were necessary to answer the questions.

Strongly agree	○	○	○	○	○	Strongly disagree

I felt efficient at finding the details that were necessary to answer the questions.

Strongly agree	○	○	○	○	○	Strongly disagree

---

7. Please assess the following statments with respect to *finding links within the text*.

I found it easy to find the links within the text.

Strongly agree	○	○	○	○	○	Strongly disagree

I felt efficient at finding the links within the text.

Strongly agree	○	○	○	○	○	Strongly disagree

About you

8. What is your gender?

○ female

○ male

---

9. What is your year of birth?

19

---

10. What is your level of expertise in the following areas?

Design (User Interface Design, Web Design, Graphics Design, etc.)

Expert ○ ○ ○ ○ No Knowledge

Development (Software Engineering, Web Development, etc.)

Expert ○ ○ ○ ○ No Knowledge

Usability and HCI (Usability Engineering, Interaction Design, User Experience Design, etc.)

Expert ○ ○ ○ ○ No Knowledge

Is there anything else you would like to tell us? Please let us know whether you had *difficulties understanding or performing* a task, in case of *general concerns* or if you have *suggestions for improvement*.

Submit your response...

C.2 Results

For determining the significance of zoom level differences (Table C.1), we applied *Wilcoxon signed-rank tests* since we could not assume normal distributions. Moreover, the font sizes in the second phase of the study were based on the actual zoom levels collected in phase 1, which made the compared samples dependent. The samples compared in Tables C.2, C.3 and C.4 are independent, but we could not assume normal distributions as well. Therefore, we conducted preliminary *Kolmogorov–Smirnow tests* to ensure that two samples had the same distribution before applying a *Mann–Whitney U test*. If the distributions were different, we used a *Median test* instead (separately indicated in the tables). The samples in Table C.5 are independent and based on an ordinal scale, wherefore we applied *Mann–Whitney U tests*.
All tests of significance have been performed using *IBM SPSS Statistics 20*. The *a priori* significance levels of all tests were $\alpha = 0.05$. Significant results are highlighted with an asterisk.

Error bars contained in bar charts indicate *standard error* ($\frac{\sigma}{\sqrt{N}}$).

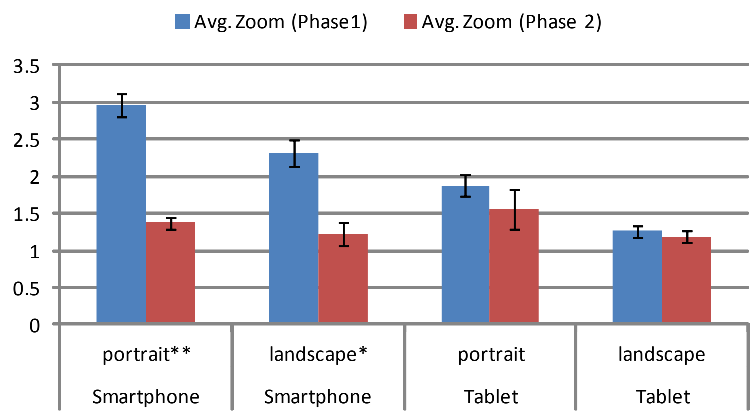


Figure C.1: Average zoom levels for the sidebar navigation. (* $p < 0.05$; ** $p < 0.001$)

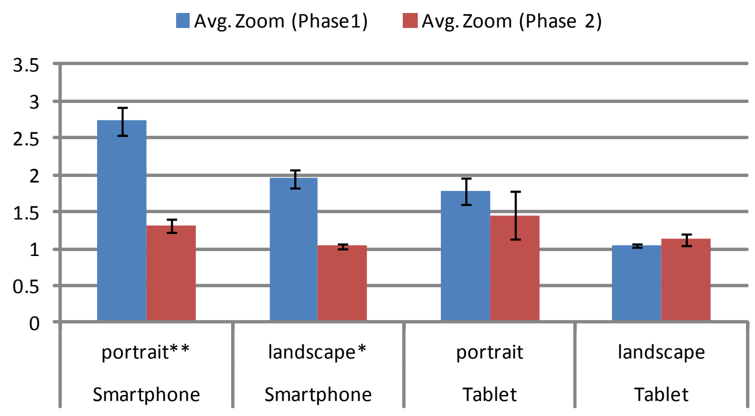


Figure C.2: Average zoom levels for the article text. (* $p < 0.01$; ** $p < 0.001$)

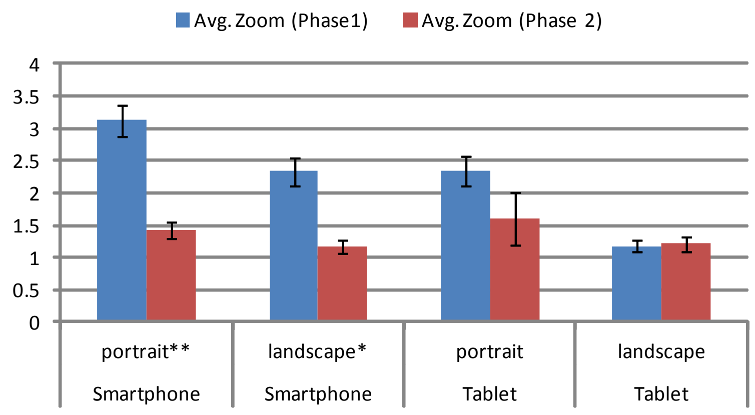


Figure C.3: Average zoom levels for the footer component. (* $p < 0.05$; ** $p < 0.001$)

Table C.1: Significance of differences in *average zoom levels* per component between phase 1 and phase 2 of the study, p values (2-tailed).

p	Smartphone		Tablet	
	portrait	landscape	portrait	landscape
Navigation	0.000*	0.026*	0.509	0.339
Main Text	0.000*	0.007*	0.130	0.203
Footer	0.000*	0.037*	0.260	0.677

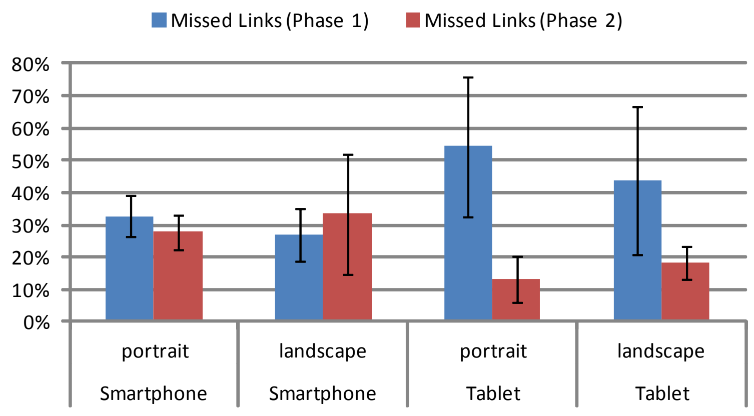


Figure C.4: Missed links per *click link* task for the sidebar navigation.

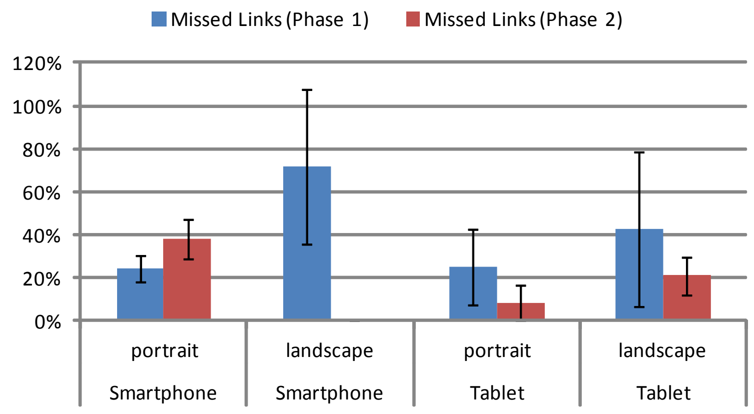


Figure C.5: Missed links per *click link* task for the article text.

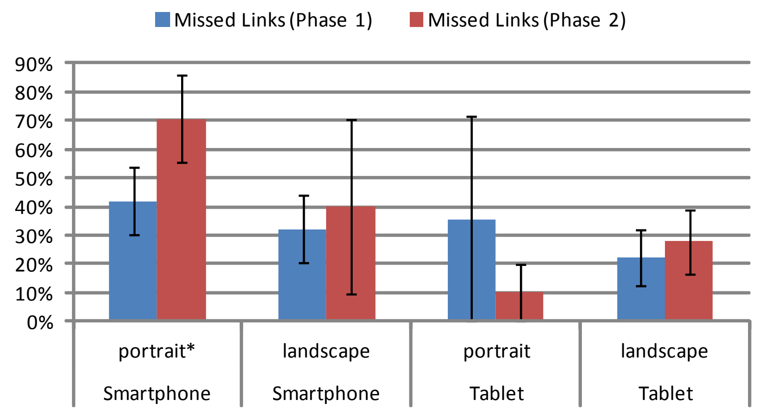


Figure C.6: Missed links per *click link* task for the footer component. (* $p < 0.05$)

Table C.2: Significance of differences in *missed links* per component between phase 1 and phase 2 of the study, p values (2-tailed).

p	Smartphone		Tablet	
	portrait	landscape	portrait	landscape
Navigation	0.731	0.677	0.371	0.861
Main Text	0.279	0.478	0.924	0.691
Footer	0.019*	0.870	0.931	0.895

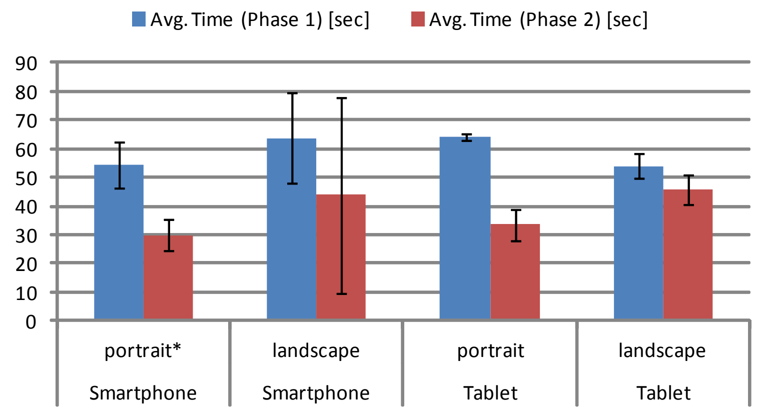


Figure C.7: Average reading times for the “iPad” text section. (* $p < 0.05$)

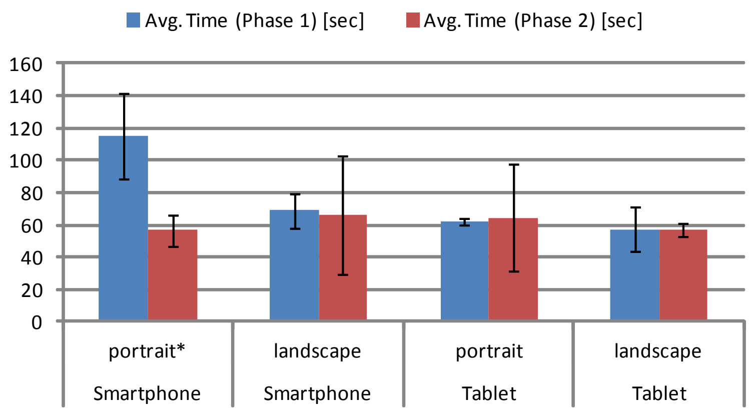


Figure C.8: Average reading times for the "Corporate Affairs" text section. (* $p < 0.01$)

Table C.3: Significance of differences in *reading time* per text between phase 1 and phase 2 of the study, p values (2-tailed).

p	Smartphone		Tablet	
	portrait	landscape	portrait	landscape
"iPad"	0.014*	0.769	0.333	0.318
"Corporate Affairs"	0.004*[a]	0.769	1.000	1.000

[a] p value based on *Median test*

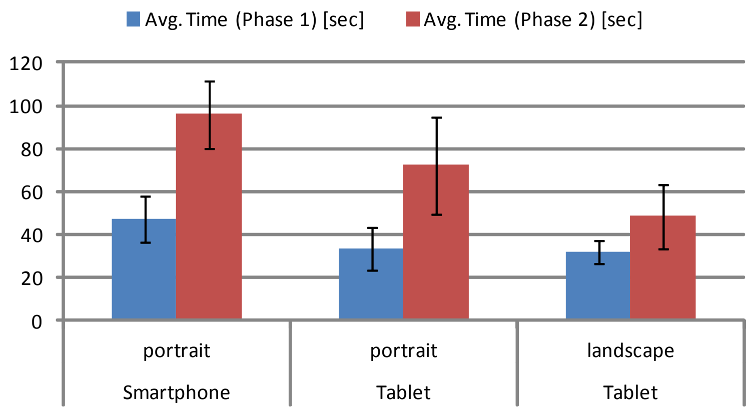


Figure C.9: Average times needed for finding the link "Mercury".

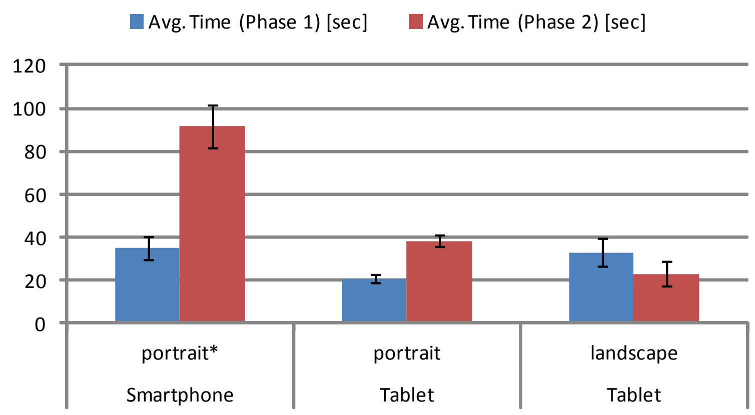


Figure C.10: Average times needed for finding the link “FaceTime”. (* $p < 0.001$)

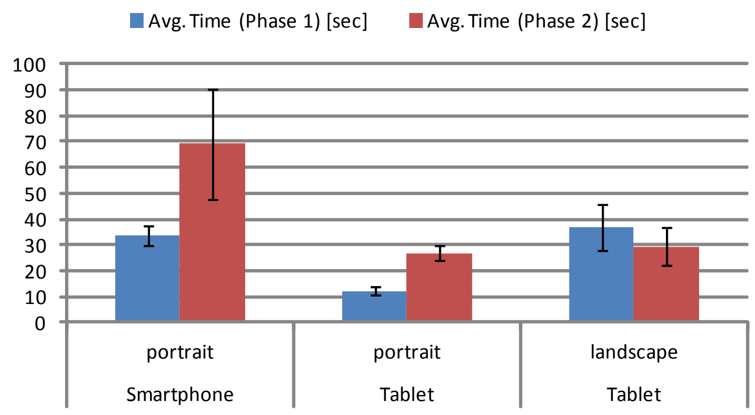


Figure C.11: Average times needed for finding the link “iPod shuffle”.

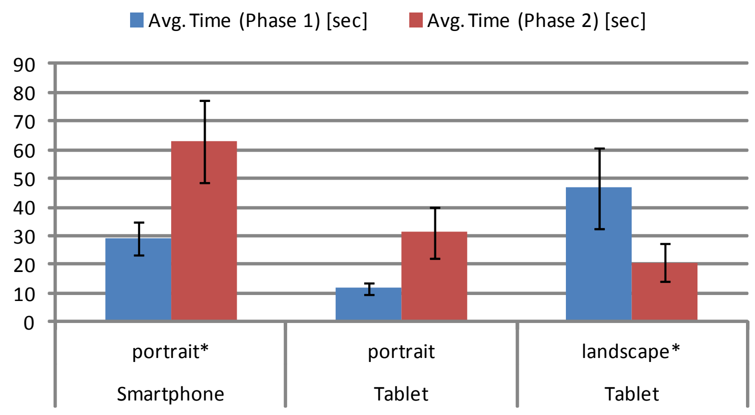


Figure C.12: Average times needed for finding the link “NeXT”. (* $p < 0.05$)

Table C.4: Significance of differences in *time needed for fining a particular link* between phase 1 and phase 2 of the study, p values (2-tailed).

p	Smartphone		Tablet	
	portrait	landscape	portrait	landscape
"Mercury"	0.054^{a}	no data for phase 2	samples too small	0.927
"FaceTime"	0.000^{*a}	no data for phase 2	0.200	0.295
"iPod shuffle"	0.082	no data for phase 2	0.200	0.731
"NeXT"	0.011^{*a}	0.333	0.200	0.035^{*}

[a] p value based on *Median test*

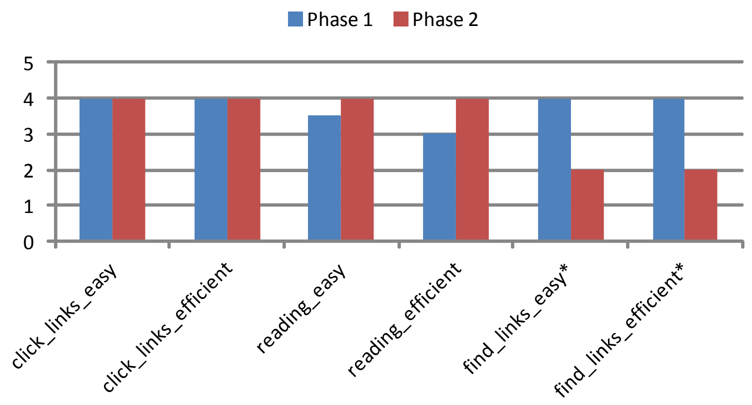


Figure C.13: Perceived user experience of smartphone users. (* $p < 0.01$)

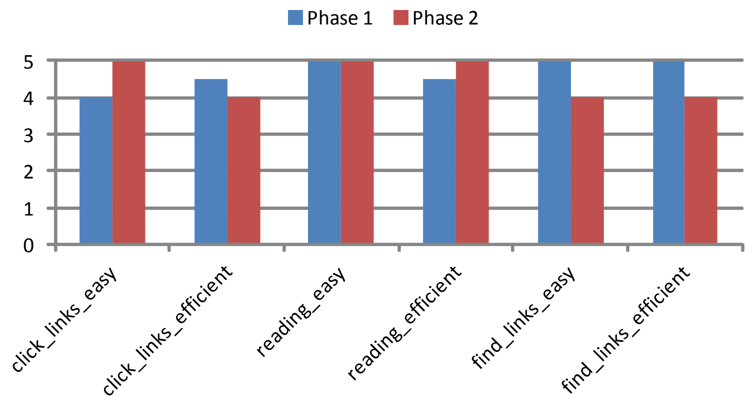


Figure C.14: Perceived user experience of tablet PC users.

Table C.5: Significance of differences in *perceived user experience* (based on the questionnaire) between phase 1 and phase 2 of the study, p values (2-tailed).

p	Smartphone	Tablet
click_links_easy	0.265	0.200
click_links_efficient	0.703	0.673
find_links_easy	0.005*	0.236
find_links_efficient	0.004*	0.114
read_text_easy	0.087	0.606
read_text_efficient	0.250	0.481

D User Study #2

D.1 Text-related Questions

D.1.1 2007–present: iPhone and iPad

1. Because of what did Apple become the third-largest mobile handset supplier in the world?
2. How many iPad units were sold on release day?
3. For how long did Apple attend the Macworld Expo?
4. Which operating system does the iPad run?
5. On which date was the iPhone announced?

D.1.2 Culture

1. What is the annual event for Mac developers?
2. Which other firms have cultural aspects similar to Apple?
3. What is Apple's store in the 5$^{\text{th}}$ avenue in New York City called?
4. What did Apple create to award extraordinary contributions?
5. At which European trade show would Mac users meet?

D.1.3 Environmental Record

1. Against which chemicals has Greenpeace campaigned?
2. Which agency rates Apple highest among producers of notebook computers?

3. Which status do all Apple computers have?
4. What was Apple's score in Greenpeace's Guide to Greener Electronics in October 2010?
5. Since when are MacBook Pro units mercury- and arsenic-free?

D.2 Questionnaires

W3Touch

Userstudy #2

Pre-study Questionnaire

1) How often do you use your touch device(s)?

☐ *Several times a day*	☐ *Once a day*	☐ *Several times a week*	☐ *Once a week*	☐ *Less than once a week*

2) How often do you browse the web using your touch device(s)?

☐ *Several times a day*	☐ *Once a day*	☐ *Several times a week*	☐ *Once a week*	☐ *Less than once a week*

3) Are you familiar with using the regular Wikipedia website?

☐ yes ☐ no

4) Are you familiar with using the Wikipedia iPhone app or the Wikipedia Mobile website?

☐ yes ☐ no

5) What is your level of expertise concerning design (user interface design, web design, graphics design etc.)?

☐ *Expert*	☐ *Knowledgeable*	☐ *Passing knowledge*	☐ *No knowledge*

6) What is your level of expertise concerning development (software engineering, web development etc.)?

☐ *Expert*	☐ *Knowledgeable*	☐ *Passing knowledge*	☐ *No knowledge*

7) What is your level of expertise concerning usability and HCI (usability engineering, interaction design, user experience design etc.)?

☐	☐	☐	☐
Expert	*Knowledgeable*	*Passing knowledge*	*No knowledge*

8) What is your gender? ☐ female ☐ male

9) What is your year of birth? 19_____

W3Touch

Userstudy #2

Post-task Questionnaire

1) It was *easy* to find particular sections within the article.

☐ *Strongly disagree*	☐ *Somewhat disagree*	☐ *Neutral*	☐ *Somewhat agree*	☐ *Strongly agree*

2) Finding particular sections within the article was *efficient*.

☐ *Strongly disagree*	☐ *Somewhat disagree*	☐ *Neutral*	☐ *Somewhat agree*	☐ *Strongly agree*

3) It was *easy* to read the sections of the article.

☐ *Strongly disagree*	☐ *Somewhat disagree*	☐ *Neutral*	☐ *Somewhat agree*	☐ *Strongly agree*

4) Reading the sections of the article was *efficient*.

☐ *Strongly disagree*	☐ *Somewhat disagree*	☐ *Neutral*	☐ *Somewhat agree*	☐ *Strongly agree*

5) It was *easy* to click the links.

☐ *Strongly disagree*	☐ *Somewhat disagree*	☐ *Neutral*	☐ *Somewhat agree*	☐ *Strongly agree*

6) Clicking the links was *efficient*.

☐ *Strongly disagree*	☐ *Somewhat disagree*	☐ *Neutral*	☐ *Somewhat agree*	☐ *Strongly agree*

7) I had a *good overview* of the webpage.

☐	☐	☐	☐	☐
Strongly disagree	*Somewhat disagree*	*Neutral*	*Somewhat agree*	*Strongly agree*

8) The amount of *zooming* involved in solving the tasks affected my efficiency.

☐	☐	☐	☐	☐
Strongly disagree	*Somewhat disagree*	*Neutral*	*Somewhat agree*	*Strongly agree*

9) The amount of *vertical scrolling* (top–down) involved in solving the tasks affected my efficiency.

☐	☐	☐	☐	☐
Strongly disagree	*Somewhat disagree*	*Neutral*	*Somewhat agree*	*Strongly agree*

10) The amount of *horizontal scrolling* (left–right) involved in solving the tasks affected my efficiency.

☐	☐	☐	☐	☐
Strongly disagree	*Somewhat disagree*	*Neutral*	*Somewhat agree*	*Strongly agree*

D.3 Significance of Results

For determining the significance of differences between the three versions of the Wikipedia (D = *Desktop*, C = *Crowdsourced*, M = *Mobile*) in terms of the measured data (Table D.1), we applied *k-samples Median tests* since all compared samples were independent, but we could not assume normal distributions and an explorative data analysis suggested that compared samples did not have the same distribution. If we found no significant results, we give an overall p value for all three samples, otherwise we give the individual pair-wise p values.
Each participant of the study had to fill out the same post-task questionnaire for each of the three versions of the Wikipedia, which means that the given ordinally scaled answers are dependent. Therefore, we applied pair-wise *Wilcoxon signed-ranks tests* to determine the significance of differences (Table D.2).
All tests of significance have been performed using *IBM SPSS Statistics 20*. The *a priori* significance levels of all tests were $\alpha = 0.05$. Significant results are highlighted with an asterisk.

Table D.1: Significance of differences between the three versions of Wikipedia in terms of the measured data, p values (2-tailed).

p	D vs. C	D vs. M	C vs. M
time_find	0.017*	0.000*	0.001*
time_read	0.290		
time_answer	0.488		
missed_links	0.488		
avg_zoom	0.000*	0.000*	0.017*

Table D.2: Significance of differences between the three versions of Wikipedia in terms of questionnaire answers, p values (2-tailed).

p	D vs. C	D vs. M	C vs. M
find_easy	0.305	0.017*	0.022*
find_efficient	0.666	0.024*	0.004*
read_easy	0.020*	0.007*	0.427
read_efficient	0.021*	0.005*	0.155
click_links_easy	0.357	0.011*	0.027*
click_links_efficient	0.337	0.003*	0.065
overview	0.472	0.107	0.204
vertical_scrolling	0.792	0.230	0.248
horizontal_scrolling	0.007*	0.005*	0.398