Advancing Web-based Dashboards: Providing Contextualised Comparisons in an Air Traffic Discovery Dashboard

Raoul L. Rothfeld

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the
Degree of Master of Science at The University of Glasgow

September 7, 2015

In cooperation with WINGX Advance GmbH

Abstract

Data discovery dashboards, i.e. interfaces that provide business metrics at-a-glance and facilitate visual querying via chart interaction, have established themselves within the business intelligence market. Their emergence as information systems has led to a proliferation of data discovery solutions and, lately, spurred the development of Web-based, open-source discovery tools. These tools allow for the development of online dashboards which provide capabilities for brushing and linking. Other discovery features – such as retaining data context while filtering (i.e. focus+context) – are not yet realised.

An open-source JavaScript library, named CrossCompare.js, has been developed and implemented in an air traffic delay dashboard, which seeks to facilitate contextualised comparisons. Conducted user studies, involving 24 participants, indicate that CrossCompare does reduce completion time and the number of required mouse clicks when performing information seeking dashboard tasks that involve comparisons. The comparison feature did not, however, benefit the completion of non-comparison tasks.

Further results affirm high usability, pleasing aesthetics, and engaging interactivity of the developed air traffic dashboard – rendering Web-based, open-source discovery dashboard development feasible using current Web technologies. Suggestions for future research conclude the thesis.

Keywords: Discovery Dashboard, Open-Source, Interactive Visualisation, Focus+Context

Acknowledgements

Contents

List of Tables

List of Figures

List of Abbreviations

AJAX	Asynchronous JavaScript and XML
API	Application Program Interface
BI	Business Intelligence
BSD	Family of permissive free software licenses
CSS	Cascading Style Sheets
CSV	Comma-separated Values
FIFO	First In, First Out
GPL	General Public License
HCI	Human Computer Interaction
HTML	HyperText Markup Language
IT	Information Technology
JS	JavaScript
KPI	Key Performance Indicator
M	Mean
MIT	Free software license originating from the Massachusetts Institute of Technology
MoSCoW	Prioritization technique (Must-, Should-, Could-, and Won't-Have)
NoSQL	Reference to non-relational databases
SD	Standard Deviation
SQL	Structured Query Language
SUS	System Usability Scale
W3C	World Wide Web Consortium
WINGX	WINGX Advance GmbH

Chapter 1

Introduction

The importance of having "the right information, at the right time, to make the right business decisions" [1] is more important than ever in a global business environment, where economic cycles shorten and competitiveness increases. Technological progress allowed for the introduction of business intelligence (BI) as a data-driven decision support system, which is continuously being advanced as the availability and variety of data increases. As part of this development, dashboards – graphical data representations commonly consisting of charts, visual key performance indicators (KPI), and/or tables – are becoming increasingly popular "for giving users better access to crucial information in a way that doesn't overwhelm them" [2].

The technology surrounding and enabling the creation of dashboards is ever-evolving with novel tools, providers, and features contributing to the fast changing dashboard market. During the past decade, so-called data discovery providers have established themselves beside traditional BI vendors, offering enterprise and end-user solutions for visually exploring data through dashboards. Recently however, community efforts have emerged that seek to provide dashboard functionalities via open-sourced programming libraries. It has yet to be seen, however, whether or not these libraries provide full discovery functionality and can, thus, be utilised to realise an effective and marketable dashboard solution.

1.1 Problem Statement

WINGX Advance GmbH, a Hamburg-based aviation business intelligence provider (hereinafter referred to as WINGX), seeks to explore novel approaches of providing insight from data to its customers. One possibility is data exploration via an interactive, Web-based discovery dashboard.

Existing dashboard and data discovery tools provide capabilities to visually and interactively explore given data, yet require desktop installations and/or server-side deployments including commercial licences. Web-based, open-source dashboard frameworks and libraries facilitate the development of standalone dashboards, yet often lack core data discovery functionalities, such as data filtering and contextualised comparisons. While linked-chart dashboards are currently feasible, presented information loses its context when data is being

focused (i.e. filtered). Despite the increasing number of open-source libraries that provide partial data discovery and/or dashboard features, no coherent Web-based, open-source data discovery dashboard has been developed so far.

Spurring and advancing the development of open-source data discovery dashboards and features will allow a broader public to benefit from data discovery's principle of enabling non-specialists to gain insights from data – rather than relying on data scientists to prepare and present analyses. Furthermore, advancing today's Web-based, open-source dashboard capabilities correlates with the increasing 'webification' [3] of traditional desktop applications – allowing for a more versatile and flexible utilisation of data discovery.

1.2 Research Objective

In view of the lack of Web-based discovery dashboards, missing discovery functionalities (i.e. context-retaining comparisons), and the growing interest in using dashboards within commercial information services; the project aims at meeting the following objectives:

- Review of contemporary dashboard usage and design practices.

- Identification of the limitations of current Web-based discovery technology.

- Development of a JavaScript library facilitating contextualised comparisons.

- Development of an air traffic dashboard, utilising the introduced comparison functionality.

- Documentation of the development requirements, design artefacts, and implementation details.

Additionally, the outcome of this project shall reveal challenges in the development and usage of Web-based data discovery dashboards and provide a foundation for future research.

1.3 Scope and Limitations

A review of dashboards as information systems and the emergence of data discovery dashboards is followed by an examination of current dashboard development and design literature. Upon documenting gathered requirements for this dashboard project, the implementation of a novel dashboard feature – which facilitates contextualised comparisons – is being described. Afterwards, this comparison functionality is being implemented in an air traffic delay dashboard and both – the comparison feature and the overall dashboard – evaluated by conducting user studies. A discussion of the findings, the project's outcomes, and possibilities for future research conclude this thesis.

Chapter 2

Background Review

"Although dashboards seem to have caught on as a management tool", as Dr. Ogan Yig-itbasioglu – lecturer at the Queensland University of Technology – explains, "the scientific literature has failed to keep pace with the developments" [4]. Despite companies showing great interest in dashboards and dashboard technology, research has yet to fully explore the properties and possible enhancements of this novel branch of business intelligence.

In the following chapter, the emergence of dashboards, their diverse definitions, and development aspects shall be reviewed as to lay the foundation for this project. Further, current data discovery providers shall be examined as the increasing interest in dashboards "is [..] evident from the proliferation of dashboard solution providers in the market" [4], which stands in contrast to the "dearth of research on dashboards" [4].

2.1   Emergence of Dashboards

Business intelligence systems, systems that "combine data gathering, data storage, and knowledge management with analytical tools to present complex internal and competitive information to planners and decision makers" [5], arose from an idea by Hans Peter Luhn [6] as early as 1958. His article, 'A Business Intelligence System', describes an arrangement for the automated "retrieval and dissemination" [6] of information abstracted from documents in order cope with the "ever-increasing rate" [6] in which information is being generated and utilised. Luhn's system lays the foundation for BI with introducing its general idea of an automated process to provide necessary or desired information for specific business activities – yet, without computerized support [7].

BI only started to evolve and thrive with the emergence of "computerized quantitative [modelling]" [7] during the 1960s; initiating the era of Analytics 1.0 – as Thomas Davenport [8], Professor of IT and Management at Babson College, has labelled the first era of BI. With the commence of BI, various business activities where "recorded, aggregated, and analysed" [8] for the first time. Davenport [8] further explains that performing analyses took weeks or months, as BI solutions were still inadequate and computational performance was insufficient. Thus, he adds, first-era business intelligence was limited to "[addressing] only what had happened in the past" [8]. Even though technology advanced and BI tools

matured, it was only post-millennial that business intelligence would experience the next game-changing impulse.

Uprising Internet-based companies, such as Google and Amazon, started to "amass and analyse new kinds of information" [8] and paved the way for Analytics 2.0. Information, gathered from internal and external sources, were combined instead of deriving information solely from data which is "generated purely by a firm's internal transaction system" [8]. This development yielded in highly sophisticated analysts, so-called data scientists, specialised on extracting and processing information from amassed data through complex algorithms.

Currently uprising, however, is what Davenport [8] describes as the era of Analytics 3.0. While the transition has not been marked by a crucial technological or analytical advance so far, it distinguishes itself in the fact that "today [..] not just information firms and online companies", but "every firm in every industry" starts analysing their "increasing amounts of data" [8]. The era of Analytics 3.0 is not only marked by the need for faster and more efficient analyses, yet also by a broader scope of data, increased importance of data understanding, and more than ever: the inclusion of the acquired information into business decisions and processes. For that, however, the process of gathering insight must shift from data scientists to a broader public, facilitating non-specialists to quickly make sense of available data – so-called self-service.

According to Eckerson, providing business users with self-service access to information is at the core of BI, with dashboards "represent[ing] the latest incarnation of BI, building on years of innovation to deliver an interface that conforms to the way a majority of users want to consume information" [9, p. 32]. Moreover, BI users' visual expectations have changed with "[u]se of the Internet for business, e-mail, social media, and other activities" [2]; elaborate Bremser and Wagner, professors of accountancy and information systems at Villanova University. Lisa Pappas and Lisa Whitman from the SAS Institute, agree when they explain that "[t]echnical capability and choices are ever-expanding, as are expectations of business data consumers who want the information they need, when they need it, in an easy-to-perceive format, wherever they are" [10].

"Dashboard graphics can enhance ease of use and provide for instant recognition of important changes in key performance metrics" [2], explain Bremser and Wagner, and it is the combination of ever-increasing amounts of data and the need for information systems to "[amplify] cognition and [capitalize] on human perceptual capabilities" [4] from which the development of dashboards stems. We are "[r]iding the technology wave", explain Pappas and Whitman [10], "we are awash in data. Attempts to stem the tide, or at least to manage its flow, have led to a proliferation of dashboards". Stephen Few, on the other hand, mentions the 2001 Enron scandal as causing "heads to turn in recognition of dashboards as much more than your everyday fledgling technology" [11, p. 15]. He bases the rise of dashboards to the aftermath of the scandal which required companies to "demonstrate their ability to closely monitor what was going on in their midst" [11, p. 15]. This lead management to seek ways to "more easily and efficiently keep an eye on performance" [11, p. 15] – hence, the introduction of dashboards.

Currently, dashboards are commonly understood as company-internal performance management and decision support systems [12, 9]. However, dashboards – as Bremser and Wagner propose [2] – should find their way into information and consulting service offerings in the near future. It is this very progression of dashboards into a broader application that this

project aims at facilitating; as to enable companies – like WINGX Advance GmbH – to offer novel information services.

2.2   Dashboard Definition

The term dashboard is still a rudimentary outline for an accumulation of graphical data representations and has yet to be clearly defined: "there is not a clear definition of dashboards", explain Yigitbasioglu and Velcu [4], "neither given by software vendors nor by academics". Initially, Yigitbasioglu and Velcu provide a rather broad description of a dashboard in that it "can be regarded as a data driven decision support system, which provides information in a particular format to the decision maker" [4]. According to Few, a dashboard is "a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance" [11, p. 26]. Few's definition corresponds with that of Yigitbasioglu and Velcu [4] and Bremser [2] – yet, expands it by the limitation on the dashboard's presentation.

However, Yigitbasioglu and Velcu acknowledge the progression of dashboard technology and conclude that, by now, there are different types of dashboards with some being "more simple and static by nature" [4], while others provide a wider set of features – such as interactive drill-down capabilities. It is this differentiation of dashboards that have led to the introduction of the term data discovery to summarize the novel capability of letting users visually and interactively query data, e.g. via brushing and linking (see [13, 14]).

Interactive brushing, i.e. enabling a dashboard user to "interactively mark up interesting data subsets directly in the views" [15], allows dashboard users to visually filter presented data besides interactively retrieving supplementary information (e.g. via hovering the cursor over a data item). According to Hauser, brushing is "very useful to intuitively select interesting subsets in a large and complex dataset" [15], yet "[i]n many cases, [...] combinations of several brushes in different views are necessary" [15] to obtain the desired data subset. This, fortunately, can be achieved by view linking. Linked views (i.e. charts) update instantly "[o]nce a specific data subset is marked with a brush in one view" [15], which maintains 'visual consistency', explains Hauser. He adds, however, that views are often confined "to the selected data subset only" [15] upon brushing, which can cause additional mental load as "it becomes more difficult for the user to stay oriented and to keep the visualized data subset in mental relation with the rest of the data" [15]. This issue can be remedied by utilising the so-called focus+context approach in which unselected data is being "included in a reduced visual form" [15] – instead of being omitted. This enables users to "visually relate selected data to their context" [15]. Brushing, linking, and focus+context combined allow for "powerful information drill-down" [16] and enable novel dashboards to redefine interactivity. In an attempt to emphasize this enhanced interactivity and refine their purpose, Yigitbasioglu and Velcu [4] fittingly conclude:

> In view of the recent developments in their design, their purpose and the concept itself, a more accurate definition of a dashboard might be that of a visual and interactive performance management tool that displays on a single screen the most important information to achieve [...] individual and/or organizational objectives.

In the context of this project, the usage of the term dashboard will follow Yigitbasioglu and Velcu's novel interpretation, in which a dashboard includes interactive features such as brushing and linking – resulting in a data discovery dashboard.

2.3 Dashboard Tools and Service Providers

Gartner, an American information technology research company and renowned for their market reviews, notes that the business intelligence and analytics market is "undergoing a fundamental shift" [17], as new products and competitors emerge, while the customers' needs continuously change. Further, Gartner states that "[n]ow, a wider range of business users are demanding access to interactive styles of analysis and insights [...], without requiring them to have IT or data science skills" [17], supporting the rise of data discovery.

The emergence of data discovery has revealed the trend towards ease-of-use tools and the consequential differentiation within the BI market. It is only since 2014 that the – previously niche – data discovery vendors are now included in Gartner's main report, the 'Magic Quadrant for Business Intelligence and Analytics Platform', which illustrates their new-found market significance and the importance of self-service BI. Further, Gartner [17] lists interactive exploration and analytic dashboards as critical capabilities for their assessment of current BI solutions – benchmarking those tools on their ability to provide effective data discovery dashboards.

This lead traditional business intelligence vendors to try "very hard to meet the needs of the current market" [17] and spurred the development of data discovery/dashboard features. Yet, their "offerings have been pale imitations of the successful data discovery specialists" [17]. Back in 2006, Stephen Few [11, p. 15] already mentioned the BI providers' hurried adaptation to the increasing interest in data discovery and dashboard technology:

> Most BI vendors that hadn't already started offering a dashboard product soon began to do so, sometimes by cleverly changing the name of an existing product, sometimes by quickly purchasing the rights to an existing product from a smaller vendor, and sometimes by cobbling together pieces of products that already existed.

Despite data discovery still being in its "early stages" [18], today's business intelligence market offers a wide range of data discovery/dashboard tools and providers to choose from [19, 2, 17]. Gartner's Magic Quadrant (see Figure 2.1) depicts the current strategic positions of various BI vendors and illustrates the vast amount – but also spread – of BI providers. While this market segment is dominated by emerging, specialised providers, such as Tableau; all leading, conventional BI vendors have released data visualization modules, with examples including IBM Cognos Insight, Microsoft Power View, SAS Visual Analytics, and SAP Crystal Dashboard [20, 2].

It has to be noted that due to a lack of academic BI vendor analyses, market reviews of IT research companies have to be consulted, which may be biased, as many of those companies hold business relations to BI vendors. Yet, various reports agree on the same data discovery tools to be the best-known and most widespread: Qlik and Tableau [17, 21].

Figure 2.1: Gartner's Magic Quadrant for BI and analytics platforms

Qlik was founded in 1993 and was "one of the first large business intelligence and data visualization software companies on the market" [22] and is the "market leader in data discovery" [17]. However, it is Tableau – founded in 2003 – that is currently the "gold standard" [17] for self-service dashboards and has significantly influenced today's customer expectations in regards to interaction, presentation, and intuitiveness. Still, the dashboard market continuously experiences differentiation as start-ups, such as Datameer (Hadoop big data visualisation) or Chartio (easy-to-use, cloud-based dashboards), try to establish themselves by providing a high degree of specialisation.

Even though most data discovery providers offer multiple deployment options, such as Software as a service, desktop applications, or Web servers; most are bound to licensing and subscription models. Merely Qlik's most recent tool, Qlik Sense Desktop, is free for private use with limited features. Further, the available deployment options for data discovery tools allow for intra-firm dashboard development – not, however, for dashboards as an information service to a third-party company or customer (c.f. WINGX Advance GmbH's case). For such cases, dashboards and their underlying technology have to be custom-made. Yet, this may change as dashboard components are emerging as out-of-the-box libraries and frameworks for Web development.

2.4 Dashboard Components for Web Development

As mentioned in Section 2.2, the term dashboard is generally used with varying interpretations. In current Web development, however, dashboard is understood as the presentation and arrangement of multiple widgets, varying in size and type (e.g. charts, images, text), rather than that of data discovery. There are numerous frameworks that provide dashboard features (see Table 2.1). Yet, most libraries focus on arranging, adding, removing, and resizing tiles. These placeholders are then to be filled by Web developers with individual content (i.e. widgets). While some of the listed dashboard frameworks (e.g. RazorFlow) do provide integrated charting and rudimentary drill-down capabilities, none can be considered a framework for data discovery dashboards (i.e. providing brushing and linking).

Despite the – current – unavailability of a full-stack discovery dashboard framework, some libraries do facilitate their development. Charting, the fundamental element of any discovery dashboard, is being provided by countless libraries – so many in fact, that Table 2.2 is merely an excerpt of JavaScript libraries that provide charting with interactive elements (e.g. mouse-over events). The restriction on JavaScript stems from its dominating market share for client-side languages used for Web pages (90% as of August 2015) [23].

To be emphasised, first of all, are VanCharts, ZingChart, and D3.js. In contrast to most other pure charting libraries, VanCharts and ZingChart also offer integrated chart arrangement capabilities and support functions upon selecting a data item within a chart – enabling

Table 2.1: List of dashboard/widget-handling frameworks

Name	Characteristic(s)	License
angular-dashboard-framework	Open-source, module for Angular.js	MIT
Atlasboard	Open-source, module for Node.js	Apache
Dashing	Open-source, Ruby gem	MIT
FnordMetric	Open-source, for charting using SQL	GPL
Gridster.js	Open-source, jQuery plugin	MIT
jDash	For Asp.Net users	Commercial
jSlate	Provided as service	Free
Pyxley	Open-source, for Python users	MIT
RazorFlow	Open-source, provides rudimentary drill-down	Commercial
Reportr	Open-source, built using Node.js	Apache
Shiny	Open-source, for R users	GPL

the development of basic drill-downs. However, brushing and linking are not supported which would lead to dashboards consisting of interactive – yet, isolated – charts. D3.js, on the other hand, stands out as it provides a thorough basis for all sorts of interactive data visualizations and, thus, quickly gained in popularity after its release in 2011 [24, 25]. By now, D3.js is used to power visualizations across industries (for example, Datameer also uses D3.js [26]) and has already brought about numerous instruction and guidance books [27, 28, 29]. It is also D3.js' versatility that allowed for Web-based charts providing brushing and linking [24] and focus+context capabilities.

Further, numerous charting libraries were released building on D3.js, which provide pre-configured charts or functions. C3.js, for example, provides all basic chart types (such as bar-, line, and pie-charts) in an easy-to-use manner while requiring only minimal programming or D3.js knowledge. Yet, C3.js – as most other charting libraries – supports neither brushing nor linking. Crossfilter, a JavaScript library for exploring multidimensional data [30, 31], was released in 2012 under Apache license and facilitates coordinated views. Crossfilter initiated the development of dc.js, a charting library combining Crossfilter and D3.js to allow brushing and linking across multiple charts. Still, filtered data context can only be retained within singular views with dc.js. No currently available library facilitates brushing, linking, and focus+context across multiple or all graphs of a Web-based dashboard.

Nonetheless, current libraries and frameworks can be utilised to build a coherent discovery dashboard – using solely free and open-source resources. The development of such a discovery dashboard is intended to be a major research outcome of this project – as stated in Section 1.2. Further, the development of the exemplary dashboard shall reveal possibilities for advancing existing libraries and spur the development of dashboards with data discovery capabilities.

Table 2.2: List of interactive JavaScript charting libraries

Name	Characteristic(s)	License
AnyChart		Commercial
C3.js	Open-source, built using D3.js	MIT
Chart.js	Open-source	MIT
D3.js	Open-source; highly customizable visualization library	BSD
dc.js	Open-source, provides brushing and linking via native Crossfilter support, built using D3.js	Apache
FusionChart		Commercial
gRaphaël	Built using Raphaël	MIT
Highcharts	Open-source	Commercial
JavaScript InfoVis Toolkit	Open-source	MIT
NVD3.js	Open-source, built using D3.js	Apache
Plotly.js		Commercial
Project EON	Open-source, for real-time data, built using C3.js	MIT
Sencha Ext JS	Includes dashboard framework	Commercial
VanCharts	Includes dashboard framework, chart builder, and basic drill-down capability	Commercial
ZingChart	Includes dashboard and basic drill-down capability	Commercial
ZoomCharts		Commercial

2.5 Dashboard Development

As dashboards arose from company-internal performance management systems, many authors emphasise organisational aspects of dashboard development – such as organisational readiness, change management, and employee adoption [12, 9]. The following review of dashboard development, however, will focus on technological and HCI aspects, as this project aims to explore the technical feasibility of Web-based discovery dashboards – rather than their company-internal deployment. Guidance for performance dashboard development within an organisational context is given by Wayne Eckerson [9] with his comprehensive book, 'Performance Dashboards: Measuring, Monitoring, and Managing Your Business'.

Nevertheless, it is important to recognise the different roles that dashboards can fulfil in organisations, as each type has its unique implications on dashboard requirements and usage. Commonly, three dashboard types are being distinguished: (1) operational, (2) analytical/tactical, and (3) strategic [9, 10, 11, 32]. "Operational dashboards", explains Eckerson [9, p. 101], "monitor operational processes, events, and activities as they occur (every minute, hour, or day)" and often provide alerting features as "to notify users about exception conditions as they happen" [9, p. 107]. Whereas analytical/tactical dashboards "tend to emphasis[e] analysis more than monitoring or management" [9, p. 17] and provide visual analysis capabilities – allowing analysts to "visually interact with charts and tables" [9, p. 113]. Strategic types "track progress toward achieving [..] objectives in a top-down fashion" [9, p. 101] and are intended to be used "to communicate strategy and review performance" [9, p. 18] – enabling effective management.

Eckerson [9, p. 101] adds, though, that these options are not mutually exclusive: "[a] majority of organizations have all three", he elaborates and claims that "no organization is truly effective without all three", as "the three [dashboard types] are complementary". It is, however, the analytical dashboard type that comes closest to the understanding of a discovery dashboard with its capabilities for visual analysis. Their emphasis on analytical use "doesn't mean that they don't support monitoring or reporting but, comparatively, there is more analysis conducted in tactical dashboards than the other two types" [9, p. 111]. "Most of the analysis", concludes Eckerson [9, p. 111], "is based on time series or categorical comparisons with various options to drill up and down hierarchies and across dimensions and attributes" – defining the main purpose of tactical dashboards and aligning them with the requirements for data discovery.

In an effort to support dashboard development and facilitate requirements elicitation, some authors provide lists of common dashboard characteristics and the identified options. While various authors provide differing options (c.f. [33] and [11, Table 2-1]), their categorisations can be used to clarify and communicate a dashboard's purpose (see Table 2.3 as presented by Juice [33], a US-based data analysis and visualisation provider). It has to be noted though, that the tactical dashboard type was omitted as an available role option in Table 2.3 – instead, the characteristic "level of detail" [33] was introduced. Besides, Eckerson notes that, in reality, most dashboards combine properties of multiple categories and do not cleanly fit within specific boundaries [9, p. 121].

Table 2.3: Dashboard characteristics and available options

Characteristic	Options			
Scope	**Broad**: Displaying information about the entire organization		**Specific**: Focusing on a specific function, process, product, etc.	
Business role	**Strategic**: Provides a high-level, broad, and long-term view of performance		**Operational**: Provides a focused, near-term, and tactical view of performance	
Time horizon	**Historical**: Looking backwards to track trends	**Snapshot**: [..] performance at a single point in time	**Real-time**: Monitoring activity as it happens	**Predictive**: Using past performance to predict [...]
Customisation	**One-size-fits-all**: Presented as a single view for all users		**Customizable**: Functionality to let users create a view that reflects their needs	
Level of detail	**High**: Presenting only the most critical top-level numbers		**Drill-able**: Providing the ability to drill down to detailed numbers [...]	
Point of view	**Prescriptive**: The dashboard explicitly tells the user what the data means [...]		**Exploratory**: User has latitude to interpret the results as they see fit	

While a dashboard's business role (i.e. type) is an – if not the – major dashboard characteristic, customisability has to be emphasised as well. "Build in flexibility to allow the dashboard to become relevant for different users", advises Juice [33]. Yigitbasioglu and Velcu agree and "recommend that dashboards come with some level of flexibility, i.e. allowing users

to switch between alternative presentation formats" [4]. However, customisability can be delivered in various ways, e.g. by filtering data, saving configured views, or the capability to highlight and arrange information [33]. The importance of customisation stems from its ability to facilitate "information discrimination" [33], i.e. excluding or hiding data that is irrelevant to the user. Customisability lets each user decide what information to emphasise and can, hence, increase system effectiveness and user satisfaction [34].

With regards to the accelerating popularity of mobile devices [35], Pappas and Whitman add that "[t]he dashboard landscape now encompasses everything from mouse-driven traditional desktop monitors to smaller-screened laptops with touch pad navigation" [10]. This development requires dashboards to support a wide range of presentation sizes, formats, and controls; and has significant influence on dashboards' architecture and design.

2.5.1 Dashboard Architecture

"The architecture of performance dashboards", explains Eckerson [9, p. 251], "have followed the trajectory of software architectures in general, from mainframe computing to client/server computing to Web-based architectures". More important than choosing a specific technology is understanding how and where the user interface, application logic, and data processing are being handled [9, p. 251]. Hence, Eckerson implies to pursue a Model-View-Controller architecture (c.f. [36]), which "has been widely embraced as an approach for developing Web-based applications" [37] and has shown to increase maintainability and re-usability by resulting in less coupled, more cohesive architectures [37].

With the emergence of Web applications, processing shifted from desktop machines (i.e. fat clients) to application servers where desktop machines only render the HTML-based representation within a Web browser (i.e. thin clients) [9, p. 252]. According to Eckerson [9, p. 252], this has the advantage of avoiding client-side processing loads and local software installation. Further benefits are the possibility of a centralised administration and increased security [9, p. 252]. However, "the downside of HTML-based thin clients is lack of performance and functionality" [9, p. 252], Eckerson notes; as client-server communication is prone to latency and, thus, introduces a delay between user action and system response. On this basis, Web-based architectures experience a 'thickening' as to "take advantage of the processing power of desktop computers and make Web-based applications more interactive and dynamic" [9, p. 252] – resulting in, so-called rich Internet applications.

Rich Internet applications can be realised using various Web-based technologies, such as Java applets or ActiveX controls, embedded scripting languages (e.g. JavaScript, TypeScript), or multimedia frameworks (e.g. Adobe Flash, Microsoft Silverlight). As Java applets and ActiveX controls represent "mini-applications that run inside a Web browser and execute within a virtual machine" [9, pp. 252], they more closely resemble desktop applications than Web pages. Further, these software framework are infamous for raising security concerns [9, p. 253] and are gradually reaching the end of their useful life, as HTML5 replaces them in many of their functions.

Embedding scripting languages inside HTML pages, on the other hand, is a more lightweight approach [9, p. 253]. Via scripts, a developer can modify the downloaded HTML page or "[retrieve] new content from the server [...] without interfering with the display and

behaviour of the page" [9, p. 253] – so-called asynchronous JavaScript and XML (AJAX). Currently, Web developers' most prevalent choice, to achieve dynamic and interactive Web pages, is using scripting languages – in particular, JavaScript [23]. Yet, Eckerson [9, p. 253] notes that using scripting languages also has its disadvantages, as one has to ensure cross-browser compatibility and may potentially experience performance and reliability issues.

"Another popular approach", Eckerson [9, p. 253] adds, "is to use multimedia development platforms, such as Adobe Flash". These multimedia frameworks, however, require the user to "download a Web browser plugin [...], which remains permanently installed on their machine" [9, p. 253]. Further, Eckerson [9, pp. 253] puts forth the increasing browser support for multimedia plugins, their browser-independence, and offline capabilities as advantages. However, his book was published in 2010 and Web technology has changed remarkably since then. HTML5 has rendered most plugins obsolete by providing native multimedia capabilities and was officially recommended by the World Wide Web Consortium (W3C) in 2014 [38]. While Eckerson [9, p. 253] hints at HTML5, he could not foresee the shift towards "plugin-free browsing" [39] as plugins – executing outside of the browser – require browser-independent updating and are often prone to vulnerabilities [39]. Thus, the combination of HTML5 and JavaScript has established itself as the current standard for interactive Web development, with "no end in sight to the rise of JavaScript" [40] – especially, with its "proliferation of open source tools and libraries" [40].

In contrast to commercial data discovery products, which make use of complex, multi-layered architecture and deployment structures (see for example Qlik [41] and Tableau [42] architectures); a purely Web- and open-source-based discovery dashboard will have its architecture somewhat defined by the available and utilised libraries. Crossfilter [30], the earlier-mentioned JavaScript library facilitating in-browser cross-dimensional data analysis, for example, was developed to execute on client-side – burdening the local Web browser with all of a dashboard's calculations. However, there are a few exploratory attempts at utilising Crossfilter on server-side (i.e. on a Web server) with the aim of providing better support for massive datasets, which cannot be processed within the memory limits of a Web browser [43, 44]. Besides influencing the requirements on client-side hardware, the decision between client- and server-side processing is based on the trade-off between the introduction of a delay (due to necessary client-server communication) and decreased processing time (as a server is expected to provide more processing power).

Regardless of a particular architecture, all dashboards are subject to common HCI rules and principles and are, thus, supposed to adhere to existing guidance for the front-end design of dashboards. It is Eckerson, who concludes that "[d]ashboard architects need to consider [...] various Web technologies [...] to deliver an attractive, interactive, and high-performance user interface" [9, p. 254].

2.5.2 Dashboard Design

By definition, dashboards greatly rely on visualisations. The interface – the look and feel of a dashboard – can determine whether it succeeds or fails, according to Eckerson [9]. "Information presentation is a balancing act" [33] between conveying a lot of information and not overwhelming the user; capturing users' attention and not distracting them; and between making it feel intuitive while still offering a wide range of features. To tackle the

challenge of developing a well-designed dashboard, one can subdivide the topic into four major parts: (1) form, (2) structure, (3) design principles, and (4) functionality [33].

Form

Technically, a dashboard's form – i.e. the method of delivering a dashboard – can range from Microsoft Excel sheets and online apps to static screens or even paper-based presentations [33]. The decision for or against a certain form should be made based on a dashboard's requirements, such as timeliness, mobility, detail, or interactivity [33]. This project, however, will focus exclusively on the delivery of an online application – which, is described as being the most versatile of all dashboard forms [33].

While there are differing opinions (see [33]), most authors suggest confining a dashboard to a single screen [10, 11, 9]. "A dashboard is meant to be viewed at-a-glance", explain Lisa Pappas and Lisa Whitman [10], "[...] without having to scroll or navigate to multiple pages". They reason that "[t]his allows for processing the information with minimal effort" [10]. Wayne Eckerson agrees and also states that "[u]sers should not have to scroll down or across a screen to view critical data" [9, p. 230]. He admits, though, that presenting all relevant data within such constraints is the "first and toughest goal of a dashboard designer" [9, p. 230]. Stephen Few [11] puts forth the limitations of our short-term memories as a reason for ensuring that a dashboard stays confined to a single screen. Few [11] further elaborates:

> One of the great benefits of a dashboard as a medium of communication is the simultaneity of vision that it offers: the ability to see everything that you need at once. This enables comparisons that lead to insights, those "Aha!" experiences that might not occur in any other way. Clearly, exceeding the boundaries of a single screen negates this benefit.

To achieve single-page – yet, highly informative – dashboards, developers must make use of space-efficient visualisations [10], rigorously exclude non-essential information [33], and utilise dashboard structure to convey meaning [9, p. 234].

Structure

"Dashboard content must be organized in a way that reflects the nature of the information and that supports efficient and meaningful monitoring", explains Stephen Few [45], when he emphasizes to arrange, position, and size dashboard elements according to their importance. As users pay particular attention to the top left quadrant of a dashboard, while the bottom right quadrant receives least attention; developers must place the most essential dashboard element prominently within the top left part of the screen [2, 10, 11, 33, 9, p. 234].

Besides positioning, it is also the visual grouping of elements that intuitively conveys relations between dashboard items or indicates a flow of expected actions [33, 9, p. 234]. Grouping can be achieved by intelligent use of whitespace and adhering to a pre-defined grid layout [33] (see Figure 2.2 for an example). Such grid systems, i.e. columns of equal

13

width, ensure consistent alignment of elements and "brings a coherence and order to the page that puts users at ease" [33].

White-space, an important aspect in interface design and often overlooked [33], can be used to group or separate dashboard items without requiring any additional graphical elements (such as background images or borders) [33, 2, 9, p. 237]. Additionally, by using spacing to indicate inter-element relations, one "[creates] places for the eye to 'rest' so that the non-white space has more impact" [33]. Without spacing, visual prioritisation would be lost, as elements would seamlessly blend into each other [33]. One can, for example, observe the use of whitespace between the columns of the illustrated and highlighted grid layout as shown in Figure 2.2.

Figure 2.2: Grid layout of New York Times' Web site (highlights added)

Design Principles

Furthermore, a set of design principles has proven itself to be beneficial with regards to user perception and is, thus, often referred to within dashboard design literature: the so-called 'Gestalt' principles by Moore and Fitz [46]. 'Gestalt', the German word for pattern, describes the psychology behind "the way our minds perceive wholes out of incomplete elements" [4] and offers insight that can directly be applied to dashboard design [11]. Among these principles – and of particular interest for dashboard design – are proximity, similarity, closure, continuity, symmetry, and figure & ground [46, 4, 11, 2, 9, p. 237]:

Proximity All else being equal, elements that are placed closer together than others are perceived to be more related to each other than to distant elements (see Figure 2.3a).

Similarity Similar elements (via – for example – colour, size, or shape) are perceived to be more related to each other than to dissimilar elements (see Figure 2.3b).

Closure When an element is incomplete or a space is not completely enclosed, people tend to see a recognizable pattern by filling in missing information (see Figure 2.3c).

Continuity People tend to perceive intersecting elements as independent, uninterrupted elements (see Figure 2.3d, where the figure is perceived as the intersection of a line and a curve – yet, could also be the touching point of two mirrored elements).

Symmetry Symmetrical elements are perceived to more related to each other than unsymmetrical elements and centre around a focus point (see Figure 2.3e).

Figure & ground The eye differentiates an element (i.e. figure) form its surrounding area (i.e. background). Balancing figure and ground improves the clarity of a perceived image (see Figure 2.3f).

In order to maintain a pleasing and clear dashboard appearance, a designer should minimise decorations and not misuse or overuse colour [11, 33, 10]. "Variations in chart color

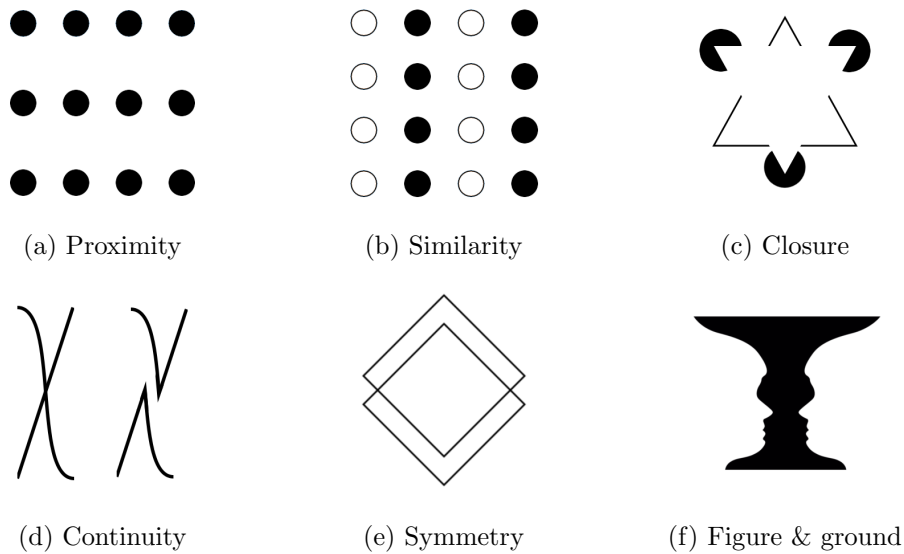|		
(a) Proximity	(b) Similarity	(c) Closure
(d) Continuity	(e) Symmetry	(f) Figure & ground

Figure 2.3: Illustrations of 'Gestalt' principles

that do not encode a meaning can be another source of distraction or distortion", explains Bremser [2]. Edward Tufte [47] has introduced the concept of reducing chart junk and, thus, increasing the so-called data-ink ratio. Numerous authors [4, 33, 11] refer to this concept and emphasise its importance for dashboard development, as a high data-ink ratio maximises attention paid to the actual, presented information. Borders, background colours, or grid lines; for example, should be removed when they are non-essential [11]. Pappas and Whitman [10] go further and describe round visualisation (such as pie charts, speedometers, or dials) as inefficient choices, as they require a lot of space for conveying relatively little information.

With regards to novel dashboard design features, Yigitbasioglu and Velcu [4] urge researchers to develop and examine prototypes and proof of concepts, as to produce experimental dashboard elements and identify their potential value.

Functionality

While each dashboard will differ in the features that it is expected or required to provide, there are some functionalities that "should be considered for any dashboard" [33] while other features can "differentiate your dashboard and provide exceptional user control and value" [33]. The capability to drill-down (i.e. retrieving additional detail) is considered one of a dashboard's basic features and can be provided in numerous ways (such as links, drop-down menus, tabs, or brushing and linking). Filters, either global (i.e. affecting all views) or local (i.e. affecting one view), provide the user with the ability to "define the scope of the data in the dashboard" [33], thus, reducing presented information to what the user is most interest in. Facilitating comparisons is a fundamental functionality of any dashboard and should be provided by, for example, allowing for multi-line or stacked charts [33]. Depending of the type of dashboard, alerts can be indispensable (e.g. for real-time monitoring dashboards) and should "[h]ighlight information based on pre-defined criteria" [33]. Lastly, as for basic functionality, dashboards should provide means to export

information (e.g. as a CSV file) and create print-outs of the dashboards – in the case of a Web-based dashboard, the latter is usually automatically provided by the Web browser.

More advanced functionality that can be included are automated text-based summaries, the ability to customize the layout of a dashboard, creating user annotations, saving filter states, or offering more advanced charting options (such as tree-maps, tag clouds, or scatter-plots) [33]. Eckerson, however, advises caution as to avoid "[exposing] too much functionality too quickly" [9, p. 244]. "Users", he [9, p. 244] further elaborates, "can be so distracted by icons and overwhelmed by options that they stop using the tool". Instead of trying to cram as much functionality into a dashboard as possible, developers should focus on the required essentials.

Chapter 3

System Requirements

This project was initiated in collaboration with WINGX Advance GmbH, as to evaluate the feasibility of developing a Web-based dashboard for commercial use (i.e. offering the dashboard as a service to customers). Requirements elicitation was conducted by gathering possible use cases from Christoph Kohler, founder and managing director of WINGX Advance GmbH, and by combining the industry experience of Mr. Kohler and the author.

As dashboards have the intrinsic aim of empowering users "by giving them self-service access to information" [9, p. 8], it is important to specify the target audience of any dashboard before continuing with defining requirements. WINGX pursues the effort of offering aviation information dashboards to industry partners (such as airport, airline, and traffic analysts) and private customers, who are interested in obtaining aviation market reviews or retrieve information regarding specific flight events. This lays the foundation for the to-be-developed dashboard.

3.1 Requirements

The gathered requirements were priorities according to the so-called MoSCoW technique, as described by Qiao Ma [48]. By assigning all requirements to one of four groups of differing importance, i.e. must-have, should-have, could-have, and won't-have; one can make use of the descriptive categorisations of this prioritisation method. It is, however, the least important classification, won't-have, that is often referred to as would-like-to-have or as the wish-list – clarifying its role as a group of features that would be liked to be seen, yet are agreed to be out of scope of the current development phase. It has to be noted that, while the more important requirements were defined by WINGX, others have been derived after reviewing relevant dashboard literature and may not have been specified by WINGX.

(1) **Must-have Requirements**

    (a) **Air traffic analyses.** The main requirement for this project is the context of an aviation analyses dashboard as set by WINGX. While WINGX itself is currently focused on BI regarding business aviation, i.e. private – in contrast to commercial

– air traffic; the dashboard can provide more general aviation insight, as it serves as a proof-of-concept for marketable, Web-based dashboards. Developers require a "deep understanding of how the system you are measuring works" [33], as to define the parts, metrics, and dimensions on which a potential dashboard operates – stressing the importance domain knowledge. Previous experience in, and documents of, air traffic analyses show airlines, airports, and flights over time as the main dimensions for any air traffic analysis. The dashboard must be structured primarily around these dimensions.

(b) **Web-Based technologies.** As WINGX seeks to provide a potential dashboard as a service, it is required for the dashboard to be based on Web technology – in contrast to most commercial data discovery vendors, which require desktop installations. Developing a Web-based service requires cross-browser compatibility and compliance with current Web standards. Further, the dashboard must be optimised for Web access, e.g. minimising file sizes to reduce loading times.

(c) **Static display of KPIs.** The minimum set of functionality was defined by WINGX as being the static display of KPIs, i.e. calculating metrics (such as averages) from retrieved data and displaying them on the dashboard in numeric format. While being simple, static KPI displays remain fundamental elements in most dashboards (c.f. [10]).

(d) **Interactive charts.** As dashboards typically provide more graphical and more interactive representations of KPIs than static displays, WINGX also asked for charts which display air traffic data over time. The interactivity in those charts has been defined as providing basic mouse-over functionality (such as highlighting data and displaying additional information).

(e) **Comparison capability.** Discussions with WINGX have proven the importance of being able to compare various data subsets for air traffic analyses. While some analytical tasks may only require a narrowed view on a specific subset, the majority of tasks require comparisons and the ability to see specific subsets in context with other subsets or the overall market. The capability to compare data within a dashboard, however, can be realised in numerous ways and has not been specified in further detail by WINGX.

(f) **Database access.** Large datasets are primarily stored within databases. WINGX also stores and modifies its data, primarily, via MySQL and MongoDB databases. In accordance with Bremser, who states that dashboards "should be able to access a company's database in a timely fashion" [2], this project's dashboard is required to be able to access and retrieve the latest tables or data subsets from either of the two, mentioned database types. With regards to the current rise of Big Data and NoSQL [49], it was decided that the dashboard must provide the capability to connect to a MongoDB database.

(g) **Single-page presentation.** It has been decided to confine the dashboard to fit on a single screen – without the necessity to scroll or navigate multiple pages. This requirement was added after evaluating the first dashboard prototype and complies with existing dashboard design literature (c.f. Section 2.5.2 on form). As the dashboard is primarily intended to be used on laptops and desktop machines, the dashboard must be fully visible using the – currently – most common screen resolutions of 1366×768 pixels [50].

(2) **Should-have Requirements**

(a) **Visual querying.** Besides mouse-over information, the dashboard should provide chart brushing and linking to enable users to filter presented data and, thus, query for specific data subsets with all dashboard views being updated in near real-time. This requirement refers to the definition of a discovery dashboard (see Section 2.2) and corresponds with providing discovery functionality similar to commercial data discovery tools.

(b) **Focus+context capability.** Visual querying partly provides the required comparison features (see Requirement 1e) via brushing and linking. This, however, will lead to unselected data being hidden, which makes the selected data lose context (see Section 2.2). "Part of the problem", states Few [11], "is that we can hold only a few chunks of information at a time in short-term memory"; with Eckerson [9, pp. 241] agreeing that comparisons should be made easy. The requirement for comparison capabilities and brushing and linking, hence, yield an additional, derived requirement for providing focus+context capabilities – to facilitate comparisons without losing context.

(c) **Mobile accessibility.** While designing dashboards that work on desktop platform as well as mobile devices is challenging [2], this project's dashboard should provide mobile accessibility. Functionality may be reduced to better suit the interaction possibilities and screen size of a smart-phones and tablets.

(d) **License for commercial use.** The dashboard should be built using resources and libraries which are provided under licenses free for commercial use (such as the MIT License) as to facilitate the development and provision of dashboard services by WINGX.

(e) **Intra-dashboard information.** As dashboard users might have varying experience with dashboard technology [2], it is important to provide guidance and explanations via a dashboard to its users. Therefore, this project's dashboard should provide additional information and help within the dashboard.

(f) **Data export.** The dashboard should provide means of retrieving a local copy of the presented data. As mentioned in Section 2.5.2, allowing users to export data is one of a dashboard's basic functions.

(g) **Pleasing aesthetics.** Besides functionality, dashboard acceptance also relies on a dashboard's pleasing aesthetics [9, 11, 33]. Thus, the dashboard should generally be aesthetically pleasing. This requirement is kept general on purpose, as the look and feel of a dashboard is perceived differently by every user and is dependent on the user's personal preferences.

(3) **Could-have Requirements**

(a) **Purely open-source.** Open-source tools and libraries allow for more flexibility in developing a product or platform as the developer can access and modify each component's source code. The dashboard, thus, could be built using only open-sourced resources.

(b) **Big Data capabilities.** The dashboard could provide capabilities to allow the processing and display of huge datasets, enabling users to visually explore those amounts of data.

(c) **Real-time data capabilities.** The dashboard could provide capabilities to allow for near real-time data and instant view updates. Frequent data updates, with the dashboard's views being redrawn instantly, allows for the dashboard to be utilised for monitoring purposes.

(d) **Interface customisation.** The dashboard could provide widget handling capabilities (such as moving, adding, or removing dashboard elements), allowing the dashboard user to customise the interface.

(e) **User accounts.** The dashboard could provide user account management, as to allow interface customisation to be linked to accounts and, thus, saved across sessions. Account management could also allow for different levels of detail, varying set of features, or the ability to save various dashboard states during sessions.

(4) **Won't-have Requirements**

(a) **Interactive route-map.** An interactive world map, on which flight routes are displayed according to the routes' great-circle distances (see [51] for more information) and the number of flights, would provide a highly vivid illustration of air traffic volumes and flows. While similar services exist (e.g. Great Circle Mapper), realisation of this feature is not deemed manageable within this project's scope and time limitations.

(b) **Data source customisation.** Most data discovery tools provide users with the ability – or even require users to – add and analyse their own data. While the development of this feature is feasible for pure dashboard service providers, WINGX aims at providing insight into their data – not requiring users to provide their own. In later development, a similar feature might be realised as to enable users to supplement presented information.

(c) **Text-based summaries.** Automated, text-based summaries of the latest, for example, air traffic developments would provide the dashboard user with alternative ways of gaining insights and benefiting from the dashboard service. Utilising and developing algorithms to create effective summaries from KPIs – and possibly news reports – requires considerable cross-domain knowledge and would yield a project on its own.

3.2 Dashboard Properties and Type

Following the dashboard categorisations shown in Table 2.3 and based on the listed requirements, the dashboard's characteristics and type can be specified, as to give guidance for the dashboard's development and allow for more effective communication between developers and WINGX.

Scope. Even though the dashboard must allow for air traffic analyses and, thus, has a reduced scope around a specific topic (i.e. air traffic), the dashboard's scope can still be described as being broad. The dashboard is intended to provide a general market overview, instead of focusing on a specific aspect of aviation or a particular company.

Time horizon. While the dashboard is mainly intended to analyse trends, i.e. perform historical analyses; users can apply time frame filters and, thus, obtain an air traffic snapshot at a specific point in time. Furthermore, the under could-have listed real-time data

capabilities could extend the dashboard's time horizon classification with real-time monitoring.

Customisation. As customisation features are assigned relatively low priority, the dashboard is mainly designed to be a one-size-fits-all solution. It could be classified as customisable only in case time and resources allow for the implementation of widget handling and user account management.

Level of detail. With filters, brushing, and linked views; the dashboard is built to be drillable, facilitating moving from obtaining an overview to selecting and analysing singular data points.

Point of view. This project aims at providing a dashboard following the advances of data discovery, with the principle of self-service, exploratory analyses – leaving users freedom in obtaining and interpreting results.

Dashboard Type. In facilitating long-term performance, while simultaneously allowing for monitoring activities; this project's dashboard combines features of strategic as well as operational dashboards. With its emphasis on enabling analyses, it would best be classified as being an analytical dashboard; following Pappas and Whitman's [10] description:

> Analytical dashboards share attributes of both strategic and operational dashboards. Like the strategic dashboards, the timeframes may be wider. Like the operational dashboards, drill-down and visual exploration are essential for discovering patterns and trends in the data.

It has to be noted though that each of these classifications are intended for guidance and are not excluding features or facets that might be better described by other categorisations.

Chapter 4

Design & Implementation

During early development, it became apparent that current Web-based, open-source technology is insufficient to satisfactory cover some of the must- and should-have requirements – in particular the capability to compare (Requirement 1e), while retaining data context (Requirement 2b). The inability of current libraries to facilitate brushing, linking, and focus+context across multiple views within a Web-based dashboard, as mentioned in Section 2.4, combined with the requirement of providing context during comparisons; yielded in the decision to extend this dashboard project with the development and application of a novel JavaScript library – facilitating contextualised comparisons.

4.1 Development Methodology

This project's development was, therefore, structured around comparison feature and dashboard development rather than – the more common – versioning of one unified product. Nonetheless, this project's development plan is loosely based on the Scrum methodology [52] with the concept of incremental and iterative development and was split into four, two-week sprints (i.e. phases): (1) project and Web server setup, (2) comparison feature development, (3) dashboard development, and (4) application testing. In accordance with the Scrum methodology, all sprints were preceded by a review of the requirements and concluded with an examination of the achieved development progress. Application testing, however, was handled separately, as it included debugging, interface adjustments, as well as preparations for the dashboard's following user evaluation – instead of further product development.

4.2 CrossCompare.js

As mentioned, the need for comparisons, while retaining data context, has led to the development of CrossCompare.js: a JavaScript library enabling contextualised comparisons on data served by Crossfilter (hence the name) through dc.js charts (see Table 2.2). As most Web development libraries are written in JavaScript – so are, both, Crossfilter and

dc.js – CrossCompare would follow suit as to facilitate its development and later usage. JavaScript's rise and prevalence within Web development [40] is further reason for its selection and is, therefore, expected to enhance the longevity of this project.

Figure 4.1 illustrates a rudimentary implementation of dc.js, based on sample data served through Crossfilter, which allows for brushing and linking. Figure 4.1a shows two charts, a row chart and a bar chart, without any filters applied to the sample data, whereas Figure 4.1b depicts the same charts – yet, with a visual filter (i.e. brushing) applied to the bar chart. The row chart has automatically been updated to only represent the filtered data subset (i.e. linking). The filtered (i.e. focused) subset remains in context to the full sample data only within the bar chart via colour coding. The data dimension illustrated by the row chart, however, has no indication of its relation to the unselected, hidden data. By updating linked views upon applying a filter and – effectively – hiding unselected data, the remaining information subset loses its context to the full dataset or, alternatively, to other filtered states.
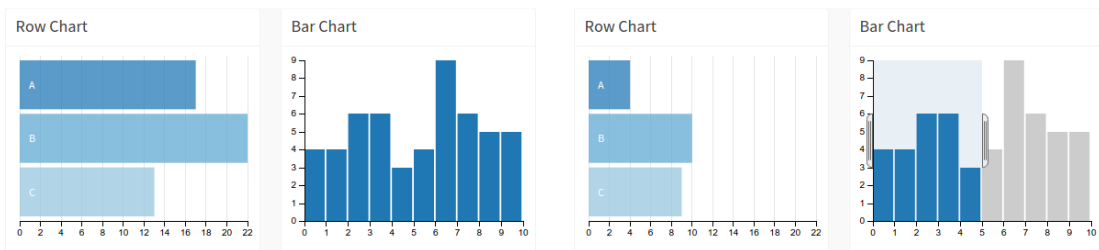
(a) Full data presented in both charts, as no filters are applied.

(b) Data subset presented in row chart, as filter is applied to bar chart.

Figure 4.1: Brushing and linking using dc.js and Crossfilter

Thus, conducting comparisons between different dashboard states requires the user to either remember, write down, or quickly switch between the values of the to-be-compared states; as in an actual dashboard, both states would not be shown side-by-side (in contrast to Figure 4.1). Further, the example shown in Figure 4.1 has static axes (i.e. axes are not automatically rescaled to maximize the display of the remaining data subset after filtering). Frequently, however, linked charts in dashboards provide this auto-scaling feature, which further hampers the user's ability to compare varying filtered states by switching between them, as the proportions within charts are being changed from one state to another. The purpose of CrossCompare is, hence, facilitating comparisons by providing context across various filtered dashboard states.

After considering various preliminary designs for an implementation of CrossCompare (such as introducing additional colour coding or combining chart via drag and drop), it has been decided to develop CrossCompare so that it is able to cache (i.e. save) different dashboard states and create a separate comparison chart, showing all cached dashboard states in relation to each other, upon request. Despite the comparison chart itself, CrossCompare is a functional – instead of a charting – library; leaving developers unrestricted in their dashboard design choices. The figures presented in Appendix A illustrate an implementation of CrossCompare to the charts shown in Figure 4.1. CrossCompare solves the mentioned issue, of comparing the unfiltered and the filtered state of Figures 4.1a and 4.1b, by introducing the ability to cache both states and, subsequently, rendering a comparison chart.

4.2.1 Architecture & Functioning

The comparison functionality, as shown in Figure A.1, can be added to any dc.js-based dashboard by including the CrossCompare library, i.e. including the 'crosscompare.js' file as a script, within the dashboard's HTML template. CrossCompare depends primarily on dc.js (for charts as the data source) and C3.js (for rendering the comparison chart), while jQuery is used throughout CrossCompare to modify HTML elements (e.g. display status texts, place the comparison chart, or listen to mouse events).

The main components within CrossCompare, as depicted by Figure 4.2, are the chart and legend registers, the queue of cached states, and the comparison chart. Charts, built using dc.js, can be registered with CrossCompare for later comparisons by using the crosscompare.add() function. This registration serves three purposes: (1) it allows the developer to specify comparison chart options that may be different for each data source chart (such as axes labels or sorting); (2) it makes the added dc.js chart internals visible to CrossCompare, as – upon caching – the chart's data and filters have to be retrieved; and (3) it creates an event listener which triggers the added chart's caching upon activating an associated caching link.
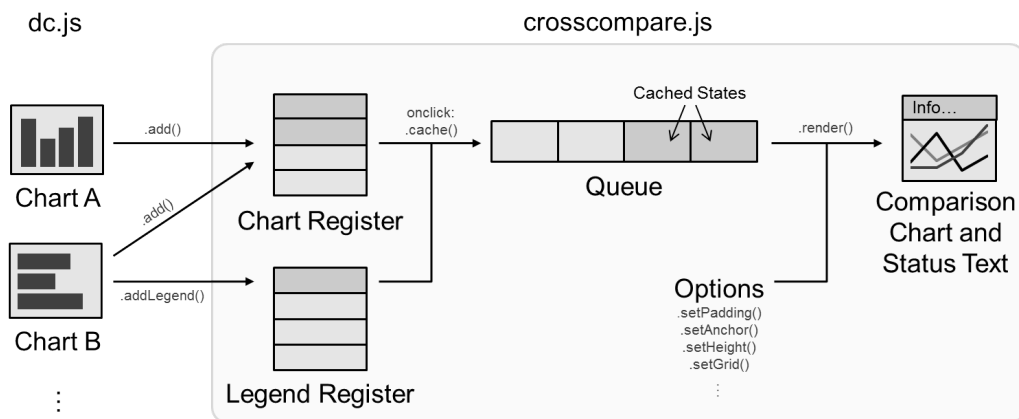

Figure 4.2: Illustration of the internal functional flow of CrossCompare.js

Further, dc.js charts can also be registered as providing naming for cached dashboard states, as users should be able to tell apart different cached states and identify which filters were applied for each cached state. This is being achieved by showing a legend within the comparison chart, identifying the displayed cached states. Upon caching a dc.js chart, CrossCompare retrieves the filters of all charts within the legend register and combines those to form a descriptive title for the cached state – listing the filters that were applied at the time of caching. It is also possible to add a dc.js chart only to the legend register, as there might be charts that are used only for brushing (i.e. setting filters) and not for retrieving actual information. The reverse is possible as well: charts that are used to retrieve information, yet do not offer filtering capabilities, can be added only to the chart register to enable caching and will not have an influence on the naming of cached states. The functionality of the legend register can be observed in Figure A.1c, where the compared states are labelled 'ALL' (as the first cached state had no filters applied) and '0 - 5' (as the second state was filtered by the bar chart to a subset from zero to five).

Upon activating a caching link, the associated chart's data is being retrieved, labelled according to registered legends, and stored in an internal queue. The queue is a list of cached dashboard states, which – afterwards – can be retrieved in order of insertion (so-called first in first out (FIFO)) to populate the to-be-rendered comparison chart – displaying all cached states. There is no maximum number of cached states or registered charts/legends set by CrossCompare. The order of caching dashboard states affects the comparison chart outcome, if the cached dc.js chart has been registered with the option to be ordered (ascending or descending). In that case, the comparison chart will always be ordered according to the first cached state (c.f. the sorted categories 'A', 'B', and 'C' in the comparison chart shown in Figure A.1c).

CrossCompare benefits from the versatile capabilities of providing in-chart interaction by utilising C3.js as the charting library for creating the comparison chart. The comparison chart, thus, provides capabilities to retrieve additional information via mouse-over, to zoom in and out, to seamlessly change the chart's type (e.g. from a bar chart to an area chart), and to hide or highlight certain cached states via the legend. Besides the comparison itself, CrossCompare also outputs status information after each interaction with CrossCompare (c.f. the changing, upper left status text in Figures A.1a, A.1b, and A.1c).

4.2.2 Documentation

CrossCompare.js is an open-source project that will allow interested developers to advance and modify CrossCompare after project submission. It is, thus, of particular importance to provide extensive documentation, as to ensure the project's maintainability and re-usability. Hence, CrossCompare provides three major forms of documentation: (1) in-code comments, (2) a usage tutorial, and (3) an application program interface (API).

In-code comments. All logical sequences within CrossCompare's source file, 'crosscompare.js', have been annotated by descriptive comments. Furthermore, each function provides an explanatory description in JSDoc [53] format. JSDoc is the JavaScript equivalent of the more well-known Java documentation format, JavaDoc [54]. The source file itself also provides a file header detailing the file's CrossCompare version, the author, the applicable license (MIT), and a link to CrossCompare's online repository.

Tutorial. A tutorial, providing code examples and describing the necessary steps to implement CrossCompare, has been written as to guide potential developers. More than just describing CrossCompare's setup, the tutorial explains the process of creating a HTML file, adding data and setting up Crossfilter, rendering dc.js charts, and – finally – providing the functionality of comparing the created dc.js charts through CrossCompare. It is also the tutorial that provides a functioning version of the charts shown in Figure A.1.

API reference. All functions, their parameters, return types, and possible options are listed and detailed within the API reference. Code samples for each function illustrate their application and example options (e.g. on how to customize the comparison chart's appearance).

4.2.3 Application Testing

Besides continuous beta testing throughout development, 47 unit test cases have been written using Mocha, a JavaScript test framework, to examine CrossCompare's functionality. These unit tests can be re-run after making changes to CrossCompare's source file as to ensure conformity with expected behaviour. While Mocha does provide the possibility to test front-end functionality via an in-browser test suite, CrossCompare's front-end altering functions have not been included in the unit tests as CrossCompare relies on jQuery and C3.js for its charting – both of which are well-tested and maintained libraries.

Further, CrossCompare and all its functions have been run on multiple operating systems using various browsers and can be described as being cross-browser compatible. Tests have been performed on Linux (Ubuntu) with the browsers Chromium 43, Midori 0.4.3, and QupZilla 1.6.0; on Windows 7 with Chrome 43, Firefox 38, Opera 29, and Internet Explorer 9; and on OS X with Safari 5.1.

4.2.4 CrossCompare Limitations

Currently, CrossCompare supports caching from dc.js's row-, pie-, bar-, and line-charts. While dc.js' latest build supports additional charting options (such as scatterplots), CrossCompare does not yet support their usage as data sources for comparisons. Similarly, dc.js offers rudimentary support for stacked charts – consisting of multiple singular charts – which, as a whole, are not supported to be used with CrossCompare.

Furthermore, CrossCompare relies on the JavaScript library jQuery for some of its functionality – especially for HTML modifications. This dependency, however, could be removed by replacing all jQuery functions with more complex – yet, independent – code.

4.3 Air Traffic Delay Dashboard

After the introduction of CrossCompare, an air traffic dashboard was developed in accordance with WINGX's requirements and the objectives of this project – identifying the feasibility and marketability of offering Web-based dashboards as an information service. For the realisation of an air traffic dashboard, public air traffic delay data has been retrieved from the American Statistical Association [55], which comprises flight information (such as origin and destination, date and time, airline code, or delays) for flights within the Unites States from 1987 to 2008.

4.3.1 System Architecture

Despite the availability of numerous Web application frameworks, provided in various programming languages; Node.js was chosen for this dashboard project, as it is a light-weight – yet, versatile – framework. Further, utilising the same language on client- as well as server-side allows for more flexibility in changing an application's structure (i.e. shift workload

from client to server or vice versa) [40]. Crossfilter – a client-side JavaScript library used for the dashboard's data processing – could, thus, also be implemented as a server-side service.

As per requirement, MongoDB has been used to store and serve the air traffic data. For the realisation of this dashboard project, static data has been used which proved to render the use of the database unnecessary as the dashboard always requested the same flight subset. Therefore, the required subset was exported to a static CSV file which could be accessed by Crossfilter directly – accelerating the dashboard's development and facilitating its deployment without MongoDB. Nevertheless, all required settings, functions, and connections to be able to use a MongoDB database, instead of static files, have been kept and commented within the Web server's code.
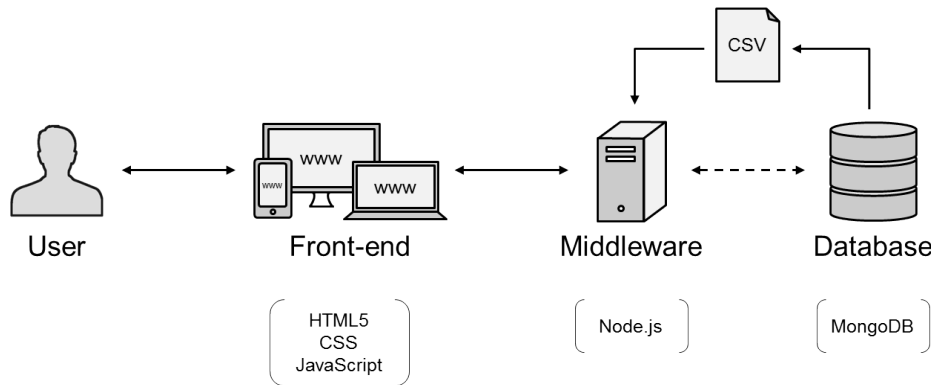

Figure 4.3: System architecture of the air traffic delay dashboard

Figure 4.3 illustrates the system architecture of the air traffic delay dashboard with the extracted static data file. The utilised front-end technologies (i.e. HTML5, CSS, and JavaScript) do not require any browser plugins (such as Adobe Flash) to run and display the dashboard – resulting in the dashboard being independent of the chosen end-user device.

4.3.2 Dashboard Resources & Architecture

Various libraries and frameworks have been utilised to be able to assemble the air traffic dashboard. Table 4.1 provides an overview of all resources that have been employed during the dashboard's development – including a brief description of their functionality within the dashboard. Further, all components are open-source and distributed under free licenses that allow their usage for commercial purposes (such as MIT and Apache), as per Requirement 2d.

The core dashboard functionality, however, is defined in the 'example.js' file in which the majority of the mentioned resources are being combined to allow for the dashboard to function. The file is named 'example.js' as the air traffic delay dashboard is embedded within the CrossCompare Web site to demonstrate its capabilities. Just as 'crosscompare.js', 'example.js' is also commented using the JSDoc notation and also provides a file header. Due to its sequential structure, 'example.js' is grouped into five parts: (1) global settings and variables, (2) Crossfilter data processing, (3) dc.js charting, (4) CrossCompare configuration, and (5) tooltip activation.

Table 4.1: List of utilised libraries and resources

Name	Usage	License
AdminLTE	Dashboard front-end framework	MIT
Bootstrap	Provides auto-scalable front-end	MIT
C3.js	Rendering of comparison chart	MIT
CrossCompare.js	Provides contextualised comparisons	MIT
D3.js	Basis of all charting functionalities	BSD
dc.js	Rendering of charts with brushing and linking	Apache
FastClick	Increases responsiveness on mobile devices	MIT
jQuery	Basis of most HTML-modifying functions	MIT
jQuery Popup Overlay	Control of information and comparison overlays	MIT
Node.js	Web server	MIT
Crossfilter	Client-side data processing	Apache

While the first section provides quick access to all dc.js charts and constants (e.g. chart heights), it is the second part, Crossfilter, where the dashboard's logical control commences. Initially, the data source CSV file is being retrieved from the Node.js server. Alternatively, an API request could be made instead, which – combined with a corresponding Node.js route for the created request – would return the requested data from a MongoDB database. The data rows are then processed into dimensions (e.g. date, airport, and airline) which are then being grouped in order to derive totals, averages, or counts.

The third section configures the previously defined dc.js charts, populates them with the Crossfilter dimensions and groups, and defines the charts' visual appearances. Following their configuration, all charts are being rendered by a distinct function which ensures the responsiveness of charts upon screen size/display changes. Having Crossfilter and dc.js deployed, CrossCompare is being introduced by adding all comparable and filterable dc.js charts to CrossCompare's chart and legend register. Further, CrossCompare's controls are being made available for control via HTML links and buttons by using jQuery functions. Similarly, jQuery functions are used to enable message popups, which are intended to provide additional information on each dc.js chart.

4.3.3 User Interface

The dashboard's user interface experienced multiple redesigns as requirements changed and the limitation of a single-screen presentation was added. The realisation of an interface that would comply with this constraint proved to be very challenging, as each end-user device might have different screen resolutions. This is of particular importance with regards to mobile accessibility of the dashboard (see Requirement 2c), as adhering to a single-screen display on mobile devices is currently infeasible. Utilising Bootstrap (see Table 4.1), however, allowed for the dashboard to be displayed on any device – be it desktop, tablet, or mobile. The dashboard's components are, thus, scaled and arranged automatically according to the device's screen size. However, automatic scaling is only provided for the elements' widths, not heights; which can lead to unnecessary whitespace below the charts on large screens or very high resolution displays.

The dashboard elements, as shown in Figure 4.4, were arranged according to their importance and laid out in a grid format (c.f. Section 2.5.2 on dashboard structure). Below a thin header with the two main filtering options for airlines and airport, KPIs for number of flights and average delay, and an overall filter reset button; follows, prominently, the area chart indicating the number of flight movements over time in the upper left quadrant. The area chart is accompanied by a slim bar chart underneath, which shows the number of flights grouped per day – providing a broader overview. The upper right quadrant shows a scatterplot, where each dot represents a flight according to their arrival or departure time and the corresponding delay in minutes. While allowing for interaction, the charts within the upper half are, primarily, intended for visual analysis – rather than brushing.
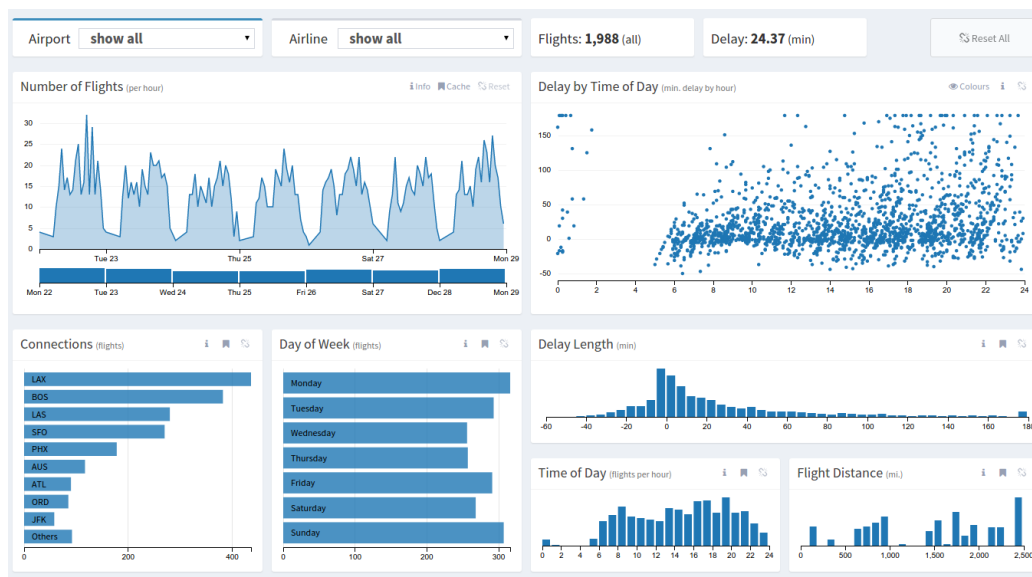

Figure 4.4: Default air traffic delay dashboard user interface

The lower half, however, providing multiple bar and row charts, is specifically designed for brushing (i.e. filtering) purposes. The appearance of these charts have, thus, been reduced even further as to increase the mentioned data-ink ratio (c.f. Section 2.5.2 on design principles). Possible filters can be applied to, for example, weekdays, delay lengths, or flight distance. If screen size allows, a minimised table will be displayed below the dashboard, which can be expanded to show a list of the latest, filtered flights. Further, this table provides the user with the ability to download all dashboard data in a CSV format for self-contained analyses.

Each chart provides interactive functionality. However, these differ between the various chart types. For that reason, every chart provides an information tooltip, which can be opened by pressing the i-button above a chart which guide users on a chart's content and interactive features. Further, the mouse cursor changes its appearance when hovering over the various charts, indicating how to interact with a chart. The standard cursor icon denotes the ability to hover over a data point for additional information, a hand icon suggests pressing a mouse button to make a selection, and a crosshair cursor denotes the ability to click and drag for applying a filter via brushing. Furthermore, the scatterplot provides additional guidance via the above eye button, which dyes the scatterplot's data points (i.e. flights) according to the severity of their delay. The, otherwise, uniform blue colouring allows for a calm and distraction-free dashboard appearance.

In this line, CrossCompare has been implemented as an unobtrusive overlay, due to the single-page limitations and the resulting scarcity of charting space. Via caching/bookmark buttons above most charts, the chart's current data can be cached for later comparisons. Upon pressing one of these caching buttons, a semi-transparent container will be made visible – displaying CrossCompare's status information and controls. it is the combination of the – relatively simple – charts for filtering and the ability to freely compare any filtered state that facilitates the dashboard's analytical capabilities.

4.3.4   Dashboard Limitations

With more time and resources, the dashboard's features could be further refined and extended. For example, the current customisability lies within applying filters, yet could also include the rearrangement or removal of dashboard elements (e.g. by implementing widget-handling framework as listed in Table 2.1).

Unfortunately, the dataset had to be reduced considerably to form a subset with which the functionality and responsiveness of the dashboard could be guaranteed – regardless of end-user device. The remaining dataset comprises 1,988 flights by three, selected airlines and airports between the 22nd and 28th December 2008 – which is a mere fraction of the originally, available dataset. A potential performance benefit by using Crossfilter on server-side could not be examined within the scope of this project.

4.4   Project Deployment

The project's code base, accompanying this paper, can be used to deploy a local version of the CrossCompare site with the air traffic delay dashboard. After ensuring that the local machine has Node.js installed, open a terminal and navigate to the CrossCompare-js folder. The terminal command 'npm start' will run the application, which is then reachable via opening 'http://localhost:3000/' in a Web browser. The Mocha tests can be re-run via executing './node_modules/.bin/mocha'. The experiment setup is accessible via 'http://localhost:3000/experiment'.

As to reduce loading times, all larger JavaScript files have been minimised (i.e. compressed), which removes all comments and annotations. Thus, the main JavaScript files for this project, 'crosscompare.js' and 'example.js', can be found uncompressed in the project's root directory.

Additionally, all project code can be found in the Github online repository at 'https://github.com/RRothfeld/CrossCompare-js'. The online repository also provides a README file with further deployment options. A live version of the project can also be found at 'http://rrothfeld.github.io/CrossCompare-js/'. This hosted version, however, does not offer experimental functions, which were used to evaluate CrossCompare's and the dashboard's usability.

Chapter 5

User Evaluation

In order to investigate our research objectives of providing contextualised comparisons within a dashboard information system, two user studies have been conducted. Both surveys involved the participants performing information seeking tasks and concluded with an overall dashboard evaluation using the System Usability Scale (SUS) – a robust, and reliable evaluation tool for the subjective measure of a system's usability [56, 57].

5.1 Research Question

In particular, the conducted experiments are intended to allow for an examination of a potential reduction in task completion time and mouse clicks required to retrieve desired information from the dashboard. The following hypotheses have, thus, been formulated:

(1) Task completion time:
  H_0 : The mean task completion time for dashboard usage *with* and *without* contextualised comparisons, on a desktop machine, is the same.
  H_1 : The mean task completion time for dashboard usage *with* and *without* contextualised comparisons, on a desktop machine, is not the same.

(2) Mouse clicks:
  H_0 : The mean number of mouse clicks for dashboard usage *with* and *without* contextualised comparisons, on a desktop machine, is the same.
  H_1 : The mean number of mouse clicks for dashboard usage *with* and *without* contextualised comparisons, on a desktop machine, is not the same.

Further, the evaluation is also meant to assess the overall perceived usability of the dashboard system – providing a comprehensive foundation for discussing the project's outcomes.

5.2   Experimental Methodology

After initial pilot experiments, a first user evaluation was performed. The tasks of the first study, however, proved to be unsuited for examining the potential benefit of CrossCompare. Thus, a second study was conducted as to allow for a closer examination of contextualised comparisons. While both evaluations have a very similar structure and use the same entry and exit questionnaires, their tasks differ considerably and are, thus, being examined separately.

5.2.1   Experiment Design

Both studies used a within subject design, with the independent variables being task difficulty and the availability of CrossCompare. Task difficulty was controlled by the number of required actions (i.e. filters to be applied to the dashboard) before the desired information could be retrieved. Further, by highlighting or hiding CrossCompare's implementation within the air traffic dashboard, the users were forced to – or kept from – using the comparison feature. The dependent variables are the qualitative (gathered through the accompanying questionnaires) and quantitative (gathered through system interaction logging) data.

5.2.2   Tasks

The experiment tasks aimed at simulating a user seeking specific air traffic related information by using the dashboard's functionalities. While both studies have tasks that vary in terms of their difficulty, it is the difference in requiring the participant to compare data points with each other, which sets the tasks of both studies apart. The tasks of the first study did not require any comparisons across multiple dashboard states (i.e. non-comparison tasks). All tasks of the second survey, on the other hand, are specifically formulated as to require the participants to contrast various filter states (i.e. comparison tasks).

Non-Comparison Tasks

Within the first user evaluation, each participant was asked to answer four different tasks. The tasks were split with two being classified as 'easy' and the remaining two as 'difficult'. Further, for two of the tasks, CrossCompare was enabled; while this feature was otherwise hidden. The order of the tasks and whether or not the comparison functionality was enabled was previously randomized and structured in the style of the Graeco-Latin square design [58]. The following tasks were presented to the participants within the first study:

(1a) *At what time on December 26th, did each airline have its peak flights per hour?*
This task required the participant to apply one time filter, while cycling through all airlines. This task was set to be 'easy'.

(1b) *What airport provides the most connecting flights before 12:00, with at least 20 minutes delay, from/to BOS, LAS, and GPT?*
This task required the participant to apply one time, one delay, and multiple connections filters; while cycling through all airports. This task was set to be 'difficult'.

(1c) *How many flights were operated per weekday?*
This task required the participant to apply no filters and was, thus, set to be 'easy'.

(1d) *How many American (AA) airline flights were operated during the busiest hour at Chicago (ORD) airport, which had a delay of 0-60 and 60-120 minutes?*
This task required the participant to apply one airline and one airport filter, while switching between two delay filter states. This task was set to be 'difficult'.

Comparison Tasks

For the second study, the nature of the task was shifted to explicitly require the user to perform comparisons between filtered states. After feedback from the participants of the first study, the number of tasks was reduced from four to three – as to shorten the experiment duration. Further, the task difficulty was now described by three categories: (1) 'easy', (2) 'medium, and (3) 'difficult'. Again, the order of the tasks and whether or not CrossCompare would be enabled was previously randomized. The following tasks were presented to the participants within the second study:

(2a) *On Thursday the 25th , what is the difference in number of flights with a delay of less than 20 minutes compared to the number of flights with a delay of 20 or more minutes; at 8, 12, 16, and 20 o'clock?*
This task required the participant to apply one time filter and compare the two different delay filter states at four points in time. This task was set to be 'difficult'.

(2b) *For each day of a week, which airport has the most flights per weekday?*
This task required the participant to cycle through all airports and contrast the weekday values. This task was set to be 'easy'.

(2c) *At New York (JFK) airport, how often does Delta (DL) airlines operate more flights per hour (time of day) than any other airline?*
This task required the participant to apply one airport filter, while cycling through all airlines and contrast the hourly flight values. This task was set to be of 'medium' difficulty.

5.2.3 Procedure

Both studies were conducted in laboratory settings. The participants were seated in front of a standard desktop computer with a mouse, a keyboard, an Internet connection, and a maximized browser window. The browser was preconfigured to display the experiment start page via the CrossCompare-js Web server – which was temporarily hosted. Following an oral introduction to the study, its purpose, and structure; the participants were handed the paper-based questionnaires with entry questions, a training leaflet, the upcoming tasks,

and an exit questionnaire (c.f. Appendices C and D, for the questionnaire templates used for the first (B) and second (C) study).

The participants were given the possibility to ask questions and were notified that they can withdraw from the experiment at any time before signing a Consent Form. After completing the introductory questions, the questionnaire required the participant to notify the researcher of their progress, as to start the training and familiarisation phase under the researcher's guidance. The various dashboard elements, their purpose, and functionalities were explained to the participants, which then had time to try each function and get comfortable in using them.

When the participants were ready, they could progress with the questionnaire and answer each task by using the dashboard, presented via the desktop computer. For each task, the dashboard would automatically highlight the required components to find the correct answers. Upon completing a task, the participant would click a 'Continue' button within the dashboard which instructed the participant to continue with the paper-based questionnaire. Upon completing all tasks, the participants were asked to fill in the remaining questionnaire, which provided the participants with the possibility to comment on and rate the overall dashboard and comparison feature. All participants were thanked and given the researcher's email address after completing the questionnaire.

5.2.4 Participants

Overall, 24 experiments were conducted equally distributed between, both, the non-comparison and comparison study (i.e. 12 participants per study). Eight participants were female (33.33%) with the remaining 16 (66.67%) being male. All participants were between 18 and 64 years old, with the majority being between 18 and 24 (62.59%) or between 25 and 34 (25.00%) years old. Many nationalities were represented (such as Finnish, Omani, Greek, or Portuguese), however, the largest nationality group was British (41.67%), followed by German (20.83%).

Most participants were students (83.33%) at the time of the experiments – only few were professionals (16.67%). Thus, half of the participants reported having obtained a Bachelor's degree as their highest, current educational qualification; while others held a Master's degree (20.83%), a PhD (8.33%) or none of these options (20.83%). Further, most participants' field of work or study was computing (66.67%), with other participants working in or studying various fields (such as law, medicine, or tourism).

Finally, most participants were very or moderately familiar with reading charts (70.83%), others were extremely familiar (16.67%) or slightly familiar (12.50%). No participant was completely unfamiliar with reading charts. The participants, however, were less familiar with dashboards (moderately familiar (45.83%); slightly familiar and not at all familiar (54.17%)) and were, mainly, not at all familiar with air traffic data (75.00%). Only few participants reported to be slightly familiar (25.00%) with air traffic data.

5.3 Results

While the identical entry and exit questionnaires allow for a combined examination of the overall perceived usability of the air traffic dashboard, the quantitative analysis will be split according to the first and second study, as the differing task experiment structures render them incomparable.
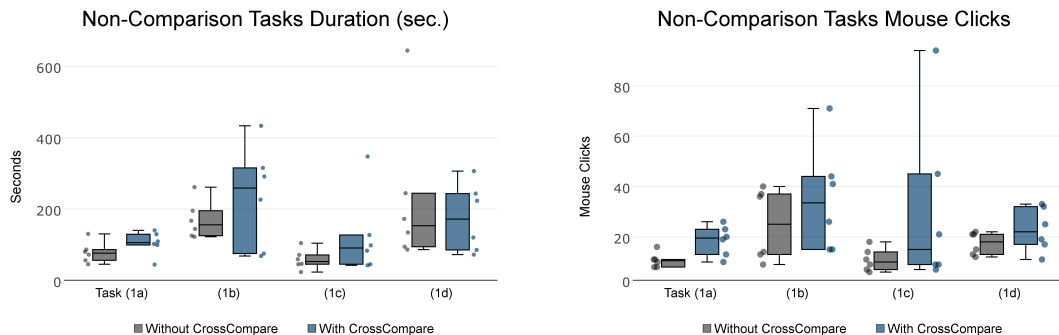
Non-Comparison Tasks Analysis

Each task of the first study – i.e. Tasks (1a), (1b), (1c), and (1d) – was performed 12 times, once by each participant, in equal proportions either *with* or *without* the comparison feature. Each task has, thus, been performed six times *with* and six times *without* CrossCompare.

Across all tasks of the first study, the mean (M) task completion time *without* the comparison feature yielded 133.54 seconds, with a standard deviation (SD) of 125.17 seconds. *With* the comparison feature, the mean task completion time amounted to $M = 159.25$ seconds, with $SD = 110.41$ seconds. The mean number of mouse clicks *without* CrossCompare was $M = 15.92$ ($SD = 9.48$), while it was $M = 26.88$ ($SD = 20.54$) *with* the comparison feature.

As multiple hypothesis are being tested, the p-value has been adjusted to counteract the problem of conducting multiple comparisons. By applying the Bonferroni correction [59], statistical significance is assumed with $p < 0.0125$. Based on the above means and deviations, the two-tailed p-value for task completion time equals 0.45 and, hence, results in no significant outcome ($p > 0.0125$). For the number of mouse clicks, the p-value is also below the threshold of 0.0125 with $p = 0.02$. Therefore, with the results of the first study, neither of the two null hypotheses can be rejected.

While the research focus is the general comparison of using the dashboard *with* and *without* CrossCompare, the differences between separate tasks can support discussing the overall findings. Figure 5.1, hence, illustrates each task's completion times (Figure 5.1a) and mouse clicks (Figure 5.1b).

(a) Completion time per non-comparison task (b) Mouse clicks per non-comparison task

Figure 5.1: Quantitative analysis of non-comparison tasks

Task (1b) *with* using CrossCompare has the highest average task completion time ($M = 234.67s$ ($SD = 143.04s$)), while the lowest mean completion time was recorded for Task (1c) *without* the comparison feature ($M = 58.00s$ ($SD = 27.66s$)). Task (1b) *with* Cross-Compare also required the most mouse clicks on average ($M = 35.33$ ($SD = 21.42$)), while Task (1a) *without* the comparison feature required the least ($M = 10.67$ ($SD = 2.94$)).

Comparison Tasks Analysis

As with the first study, each task of the second study – i.e. Tasks (2a), (2b), and (2c) – was performed 12 times, once by each participant, in equal proportions either *with* or *without* the comparison feature. Each task has, thus, been performed six times *with* and six times *without* CrossCompare.

Across all tasks of the second study, the task completion time *without* the comparison feature yielded $M = 416.39s$ ($SD = 265.17s$). *With* the comparison feature, the results are $M = 156.39s$ ($SD = 107.25s$). The mean number of mouse clicks *without* CrossCompare was $M = 88.89$ ($SD = 54.98$), while it was $M = 23.56$ ($SD = 14.78$) *with* the comparison feature. For, both, task completion time and the number of mouse clicks, the *p*-values are below 0.0005 and, thus, fulfil the requirement for statistical significance of $p < 0.0125$. Hence, for the second study, both null hypotheses can be rejected.

Again, the differences between separate tasks will be presented to allow a more thorough discussion of the findings. Figure 5.2 illustrates each task's completion times (Figure 5.2a) and mouse clicks (Figure 5.2b).

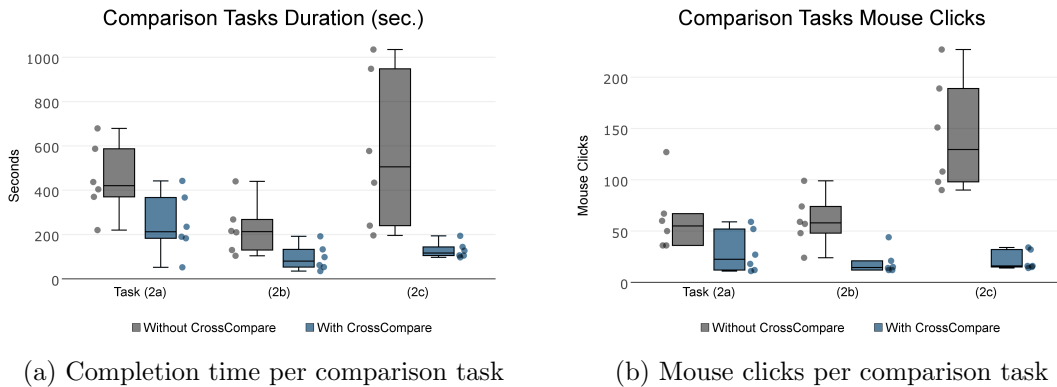(a) Completion time per comparison task   (b) Mouse clicks per comparison task

Figure 5.2: Quantitative analysis of comparison tasks

Task (2c) *without* using CrossCompare has the highest average task completion time ($M = 571.67s$ ($SD = 353.95s$)), while the lowest mean completion time was recorded for Task (2b) *with* the comparison feature ($M = 95.50s$ ($SD = 58.84s$)). Task (2c) *without* CrossCompare also required the most mouse clicks on average ($M = 143.83$ ($SD = 55.30$)), while Task (2b) *with* the comparison feature required the least ($M = 19.67$ ($SD = 12.37$)). All mean task completion times, mouse clicks, and task difficulties of the first and second study are summarised, per task, in Appendix E.

General Analysis

Across both studies, 23 out of 24 (95.83%) participants reported that they preferred using the dashboard *with* the comparison function. Further, all participants found CrossCompare to be somewhat (25.00%) or very (75.00%) helpful; while it was being described as somewhat unintuitive (8.33%), somewhat intuitive (37.50%), and very intuitive (54.17%).

The overall dashboard was described as very (87.50%) or somewhat (12.50%) appealing; leaving the participants very (87.50%) or somewhat (8.33%) satisfied with their experience, while one participant (4.17%) was somewhat dissatisfied. Additionally, the participants were presented the System Usability Scale and were asked to rate the overall dashboard by the scale's predefined categories. A short description of each of the categories can be found in Table 5.1 together with the average response value. A full list of questions can be found in each study template (Appendices C and D). The participants answer each SUS question with values between 1 (strongly disagree) and 5 (strongly agree). These values are then converted to an overall system usability score which ranks a system on a scale from 0 to 100. The air traffic dashboard has received an average SUS rating of $M = 81.15$ ($SD = 12.58$) by its 24 users (i.e. experiment participants).

Table 5.1: Average SUS responses

SUS Category	Avg. Response
Would use frequently	4.38
Unnecessarily complex	1.54
Easy-to-use	4.21
Requires technical support	2.29
Well integrated	4.63
Too much inconsistency	1.21
Easy to learn	4.38
Cumbersome to use	1.67
Confidence during usage	3.71
High learning load	2.13

The questionnaires also invited each participant to comment on the comparison feature and the overall dashboard. Recurring themes in the comments regarding the comparison feature were that it made dashboard usage "easier" and "faster". "[The] ability to cache and then compare filtered data significantly aided the [data's] interpretation", stated one participant with others agreeing that CrossCompare was "ideal for complex querying" and for filtering out "the factors that matter to you". However, multiple participants stated that the comparison feature did not benefit the completion of simple tasks. Regarding the overall dashboard, participants repeatedly commended the dashboard's "clean looks", its responsiveness, the easy-to-navigate layout, and the interactivity provided by the charts. Multiple participants, however, wished for more coherent labelling and the ability to apply a filter to bar charts by clicking, instead of brushing (i.e. click and dragging).

5.4 Discussion

As neither null hypothesis could be rejected when using non-comparison tasks (first study), it has to be assumed that CrossCompare did not reduce completion time or required mouse clicks. If anything, the participants took longer and required more clicks for this kind of task. As, due to randomisation, some of the non-comparison tasks had to be solved using CrossCompare, it seemed that most participants used the dashboard as if they would not have access to the comparison feature and only activated it after they were already certain in

their answers – resulting in additional time and clicks. The participant's opinions, however, are in contrast to the quantitative data of the first study as most participants preferred the dashboard with the comparison feature. It has to be mentioned though, that feature preference cannot be equated with potential usage.

Fortunately, the participants' perceived task difficulty was consistent with the intended difficulty level of each task (i.e. Tasks (1a), (1c), and (2b) being 'easy'; Tasks (1b), (1d), and (2a) being 'difficult'; while Task (2c) could have been somewhat easier as it was also perceived as rather 'difficult' – instead of 'medium' difficulty). The various difficulties, however, did not show any effect on the usage of the dashboard *with* or *without* CrossCompare.

Some participants apparently struggled with particular tasks, especially with Tasks (1b) and (1c) when using the comparison feature, as one task completion has required significantly more time and mouse clicks (see Figures 5.1a and 5.1b). When examining the individual interaction logs, it seems that, for Task (1b), a participant had falsely set filters which hid the required information from the dashboard view – resulting in multiple filter resets and, thus, required more time and clicks. Another participant found the notation on the comparison chart confusing which yielded the completion time outlier and required mouse click maximum for Task (1c). Task (1d), however, with one outlier for task completion time is believed to be an inaccurate measurement, where the participant continued with the questionnaire without pressing the continue button in the experiment dashboard screen, which would have stopped the timing of the task. This is based upon the observation that, despite this long completion time, no participant required substantially more clicks for Task (1d) (see Figure 5.1b).

While set to be 'easy', Task (2c) required the most comparisons (out of all comparison tasks). Some participants were unsure of their retrieved results and, often, repeated steps as to increase their confidence in their answers – resulting in the wide spread of obtained completion time and clicks when performing the task *without* CrossCompare. Two participants, performing Task (2c) *without* the comparison feature described the task as "quite laborious" and even "infuriating".

For comparison tasks, both null hypotheses could be rejected, suggesting that CrossCompare did reduce task completion time and required mouse clicks for those tasks. Further, it is striking that the spread of required times and clicks between the participants was reduced when using the comparison feature. Thus, it is assumed that CrossCompare facilitated its usage with all participants, regardless of task difficulty, their familiarity with the subject, or field of work/study – an observation which correlates with the participants stating that the comparison tool was mostly helpful and intuitive. Many participants directly commended the comparison feature's ability to retain data context while having various filtered states, which – fortunately – is the intention of CrossCompare.

The overall system usability score of 81.15 renders the dashboard as being above industry-average – which Jeff Sauro [57] describes as being 68 (based on more than 500 studies). Further, the dashboard aesthetics haven been very well received with participants enjoying the responsiveness and interactivity of brushing and linking. Still, there is plenty room for improvements. When analysing the SUS categories and their average response values (as shown in Table 5.1), the dashboard could especially improve by requiring less technical support, reducing the initial learning load, and – in particular – provide indications or features to reinforce confidence during dashboard usage.

Chapter 6

Conclusion

The increasing interest in dashboards as information systems has transformed the business intelligence landscape. BI vendors and service providers are adapting to this ever-developing branch of visual data analytics which has led to a diversification of the BI market towards visual data discovery. These data discovery solutions, however, require licenses, complex deployments, and are intended for company-internal (i.e. performance management) usage. To facilitate dashboard usage as information services outside a company-focused context, additional research efforts are required.

Thus, contemporary dashboard usage and design practices have been reviewed to lay the foundation for the development of a Web-based discovery dashboard. An examination of current Web-based dashboard and charting solutions has shown that Web-based dashboard development is feasible and can already provide the user with highly interactive views which allow brushing and linking. However, some discovery features were not available. Especially the ability to compare various dashboard states, following the principle of focus+context, was yet to be realised.

This project, hence, resulted in the development of CrossCompare.js, a JavaScript library to facilitate contextualised comparisons, and its implementation in an exemplary air traffic delay dashboard – according to the requirements of WINGX, who seeks to investigate the marketability of Web-based dashboards. This paper accompanies the development process and outcomes of, both, CrossCompare and the air traffic dashboard. Their following evaluation has shown the benefits of contextualised comparisons and proven the viability of Web-based dashboards.

With the release of CrossCompare.js v1.0.0 and the air traffic dashboard, both of which are open-source and built using only freely licensed resources, it is hoped to have advanced and spurred the development of Web-based discovery dashboards. By now, it is fair to assume that Web-based, open-source dashboards are fully feasible for commercial information services and will allow a broader public to take advantage of the manifold benefits of visual analytics.

Future Work and Research

While a discovery dashboard has been developed, it required substantial development effort and the combination of numerous libraries. The introduction of a full-stack, open-source, Web-based dashboard framework could vastly accelerate the usage and adoption of online dashboards. Further, balancing client- and server-side data processing would be an interesting development project as to optimize performance and allow for – potentially – Big Data analytics. CrossCompare will also undergo further development as to support a wider range of chart types and charting libraries.

The popularity and increasing usage of mobile devices also poses challenges to dashboard design, as it renders the common requirement of a single-screen view infeasible. Exploring different variants of the required compromise between screen-size and presenting all information at a glance, might provide valuable insights for mobile dashboard design and dashboard perception in general. Additionally, the differing possibilities for interaction on a mobile device, compared to a desktop machine, could lead to a more intuitive usage experience for all users – regardless of their preferred platform.

Based on the experience gathered from the conducted user studies, further experiments could identify and investigate different types of dashboard tasks and – possibly – examine the effect of various dashboard functionalities (such as contextualised comparisons) on the users' satisfaction and effectiveness. A subject of particular interest and, thus, potential topic of future research projects, are the factors that give users confidence in their dashboard usage. Identifying aspects that – besides aesthetics – reinforce the users' confidence might add a new dimension to dashboard development.

Finally, it is hoped that this project – and possible future research based on it – contribute to the amendment of the "dearth of research on dashboards" [4].

Bibliography

[1] CGI Group Inc. Next Generation Business Intelligence: Seven steps to improved business intelligence through data discovery, 2011. URL: http://www.cgi.com/sites/default/files/white-papers/business-intelligence-white-paper.pdf.

[2] Wayne G. Bremser and William P. Wagner. Devoloping dashboards for performance management. *The CPA Journal*, 83(7):62–67, 2013.

[3] Mária Bieliková, Pavol Návrat, Daniela Chudá, Ivan Polášek, Michal Barla, Jozef Tvarožek, and Michal Tvarožek. Webification of Software Development: General Outline and the Case of Enterprise Application Development. *Global Journal on Technology, Vol 3 (2013): 3rd World Conference on Information Technology (WCIT-2012)*, 03:1157–1162, 2013.

[4] Ogan M. Yigitbasioglu and Oana Velcu. A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, 13(1):41–59, March 2012. URL: http://www.sciencedirect.com/science/article/pii/S1467089511000443, doi:10.1016/j.accinf.2011.08.002.

[5] Solomon Negash. Business intelligence. *The Communications of the Association for Information . . .*, 13, 2004. URL: http://aisel.aisnet.org/cgi/viewcontent.cgi?article=3234&context=cais.

[6] H.P. Luhn. A Business Intelligence System. *IBM Journal of Research and Development*, 2(4):314–319, 1958. doi:10.1147/rd.24.0314.

[7] Seth Grimes. BI at 50 Turns Back to the Future, 2008. URL: http://www.informationweek.com/software/information-management/bi-at-50-turns-back-to-the-future/d/d-id/1073576?page_number=1.

[8] Thomas H. Davenport. Analytics 3.0. *Harcard Business Review*, pages 64–72, December 2013.

[9] Wayne W. Eckerson. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business.* John Wiley & Sons, 2010.

[10] Lisa Pappas and Lisa Whitman. Riding the technology wave: Effective dashboard data visualization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6771 LNCS(PART 1):249–258, 2011. arXiv:9780201398298, doi:10.1007/978-3-642-21793-7_29.

[11] Stephen Few. *Information Dashboard Design - The Effective Visual Communication of Data.* OReilly Media, 2006.

[12] K. Pauwels, T. Ambler, B. H. Clark, P. LaPointe, D. Reibstein, B. Skiera, B. Wierenga, and T. Wiesel. Dashboards as a Service: Why, What, How, and What Research Is Needed? *Journal of Service Research*, 12(2):175–189, 2009. doi:10.1177/1094670509344213.

[13] Richard A Becker, William S Cleveland, and Murray Hill. Brushing Scatterplots. *Technometrisc*, 29(2):127–142, 1987. doi:10.2307/1269768.

[14] Daniel Keim and Matthew Ward. Visual Data Mining Techniques. *Techniques*, 2002. doi:10.1097/ALN.0b013e31825dd7ac.

[15] H Hauser. Interactive Visual Analysis-an Opportunity for Industrial Simulation. *SimVis*, pages 1–6, 2006. URL: http://www.vrvis.at/publications/pdfs/PB-VRVis-2006-014.pdf.

[16] Johannes Kehrer, Peter Filzmoser, and Helwig Hauser. Brushing moments in interactive visual analysis. *Computer Graphics Forum*, 29(3):813–822, 2010. doi:10.1111/j.1467-8659.2009.01697.x.

[17] Rita L. Sallam, Bill Hostmann, Kurt Schlegel, Joao Tapadinhas, Josh Parenteau, and Thomas W. Oestreich. Magic Quadrant for Business Intelligence and Analytics Platforms, 2015. URL: http://www.gartner.com/technology/reprints.do?id=1-2ACLP1P&ct=150220&st=sb.

[18] Thomas H. Davenport. The Rise of Data Discovery, 2008.

[19] Wan Maseri, Wan Mohamad, Embong Abdullah, and Mohamad Zain Jasni. Improve Knowledge Visualization through an Interactive Graph-based Dashboard System with Key Performance Indicator : A Case Study of University Dashboard for Higher. pages 1–7, 2007.

[20] Doug Henschen. How To Choose 'Advanced' Data Visualization Tools, 2012. URL: http://www.informationweek.com/software/information-management/how-to-choose-advanced-data-visualization-tools/d/d-id/1105480?

[21] B Y Wayne Eckerson. Visual Discovery Tools : Market Segmentation and Product Positioning. (March):1–30, 2013.

[22] Andrew Foley. QlikView Vs. Tableau: Software Showdown, 2015. URL: https://www.clearpointstrategy.com/qlikview-vs-tableau/.

[23] Q-Success. Usage Statistics of JavaScript for Websites, August 2015. URL: http://w3techs.com/technologies/details/cp-javascript/all/all.

[24] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D 3 : Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[25] Drew Skau. Why D3.js is So Great for Data Visualization, 2013. URL: http://blog.visual.ly/why-d3-js-is-so-great-for-data-visualization/.

[26] Christophe Viau. Whats behind our Business Infographics Designer? D3.js of course., 2012. URL: http://www.datameer.com/blog/uncategorized/whats-behind-our-business-infographics-designer-d3-js-of-course-2.html.

[27] Scott Murray. *Interactive Data Visualization for the Web*. O'Reilly Media, Inc., 2013. URL: http://chimera.labs.oreilly.com/books/1230000000345.

[28] M Dewar. *Getting Started with D3*. 2012.

[29] Nick Qi Zhu. *Data Visualization with D3.JS Cookbook*. Packt Publishing Ltd., 2013.

[30] Square. Crossfilter - Fast Multidimensional Filtering for Coordinated Views, 2012. URL: http://square.github.io/crossfilter/.

[31] Chris Weaver. Multidimensional visual analysis using cross-filtered views. *VAST'08 - IEEE Symposium on Visual Analytics Science and Technology, Proceedings*, pages 163–170, 2008. doi:10.1109/VAST.2008.4677370.

[32] Grant Stanley. Dashboards : Take a closer look at your data. URL: http://canworksmart.com/three-types-of-dashboards/.

[33] Juice Inc. A Guide to Creating Dashboards People Love to Use, 2009. URL: http://www.cpoc.org/assets/Data/guide_to_dashboard_design1.pdf.

[34] Boon Wan Tan and Tak Wah Lo. The impact of interface customization on the effect of cognitive style on information system success. *Behaviour & Information Technology*, 10(4):297–310, 1991. doi:10.1080/01449299108924291.

[35] Cody W. Hanson. Chapter 2: Mobile Devices in 2011. *Library Technology Reports*, 47(2):11–23, 2011.

[36] John Deacon. Model-view-controller (MVC) Architecture, 2009. URL: https://techsimplified2.com/Uploads/Agendas/October28,2011.pdf.

[37] Nick Heidke, Joline Morrison, Mike Morrison, and Eau Claire. Assessing the Effectiveness of the Model View Controller Architecture for Creating Web Applications. *Science*.

[38] W3C. HTML5 is a W3C Recommendation, October 2014. URL: http://www.w3.org/blog/news/archives/4167.

[39] Julien Sobrier. The move to plugin-free browsers, February 2013. URL: http://research.zscaler.com/2013/02/the-move-to-plugin-free-browsers.html.

[40] Rohan Pearce. The rise and rise of JavaScript, January 2014. URL: http://www.techworld.com.au/article/536950/rise_rise_javascript/.

[41] QlikTech. Qlikview Development and Deployment Architecture, February 2011. URL: http://www.qlikview.com/us//~/media/Files/resource-library/global-us/direct/datasheets/DS-Technical-Brief-Dev-and-Deploy-EN.ashx.

[42] Marc Rueter and Ellie Fields. Tableau for the Enterprise : An Overview for IT, May 2012.

[43] Ganesh Iyer. Big Data visualizations using crossfilter and dc.js, 2014. URL: http://ganeshiyer.net/blog/2014/07/19/big-data-visualization-using-crossfilter-and-dc-js/.

[44] Darshit Shah. SmartFilter Not in-memory server-side Crossfilter for BigData, 2015. URL: http://axiomnext.com/blog/smartfilter-not-inmemory-server-side-crossfilter-for-bigdata/.

[45] Stephen Few. Pervasive Hurdles to Effective Dashboard Design. *Perceptual Edge*, (January):7, 2007.

[46] Patrick Moore Fitz and Chad. Gestalt Theory and Instructional Design, 1993.

[47] Edward R. Tufte. *Beautiful Evidence*. Graphics Press LLC, 2006.

[48] Qiao Ma. The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements : A Systematic Literature Review. (November):1–83, 2009.

[49] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of Big Data on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2014. doi:10.1016/j.is.2014.07.006.

[50] Frederic Lardinois. Move Over 1024768: The Most Popular Screen Resolution On The Web Is Now 1366768, April 2012. URL: http://techcrunch.com/2012/04/11/move-over-1024x768-the-most-popular-screen-resolution-on-the-web-is-now-1366x768/.

[51] Eric W. Weisstein. Great Circle. URL: http://mathworld.wolfram.com/GreatCircle.html.

[52] Eduardo Antonio Cecilio Fernandes. Scrum explained, January 2015. URL: http://www.codeproject.com/Articles/704720/SCRUM-explained.

[53] Jeff Williams, Jannon Frank, Michael Mathews, and Tim Schaub. @use JSDoc, August 2015. URL: http://usejsdoc.org/.

[54] Ashley J S Mills. JavaDoc. Technical report, The University Of Birmingham, Birmingham, 2005.

[55] American Statistical Association. Data expo '09 - Get the data, 2009. URL: http://stat-computing.org/dataexpo/2009/the-data.html.

[56] John Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.

[57] Jeff Sauro. Measuring Usability with the System Usability Scale (SUS), January 2011. URL: http://www.measuringu.com/sus.php.

[58] James V Bradley. Complete Counterbalancing of Immediate Sequential Effects in a Latin Square Design. *Journal of the American Statistical Association*, 53(282):525–528, 1958. doi:10.1080/01621459.1958.10501456.

[59] Olive Jean Dunn. Multiple Comparisons Among Means. *Journal of the American Statistical Association*, 56(293):52–64, 1961. doi:10.2307/2282330.
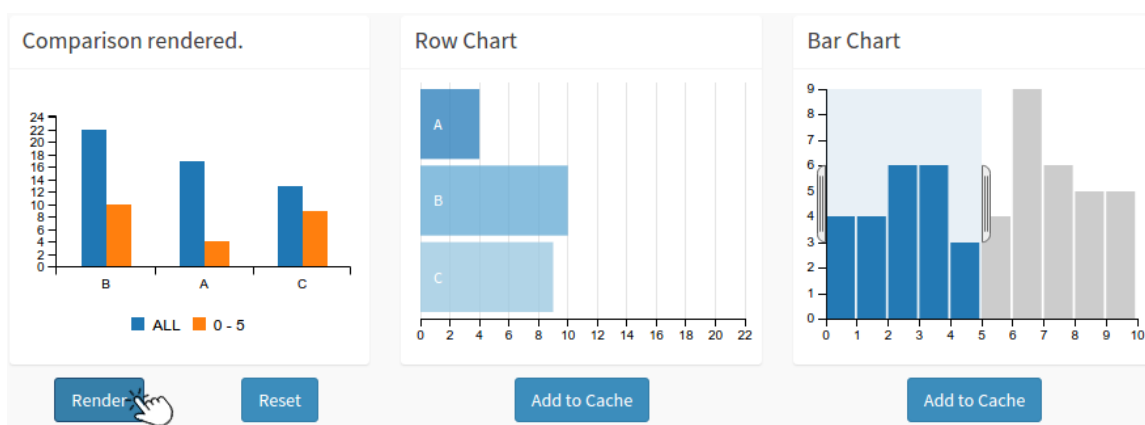
Appendix A

CrossCompare.js Usage Illustration
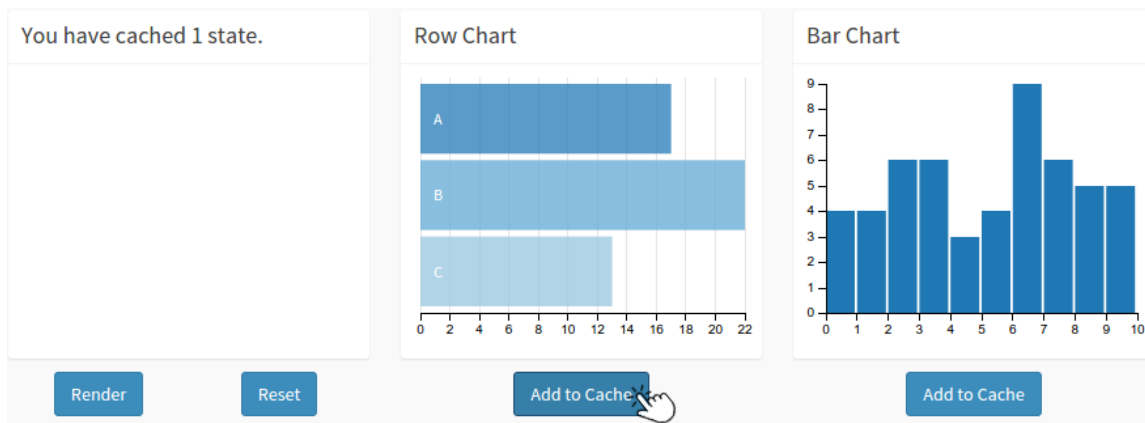
(a) Illustration of adding the unfiltered state of the row chart to CrossCompare's cache (i.e. saving the row chart's current data) upon pressing the 'Add to Cache' button below the to-be-cached chart.
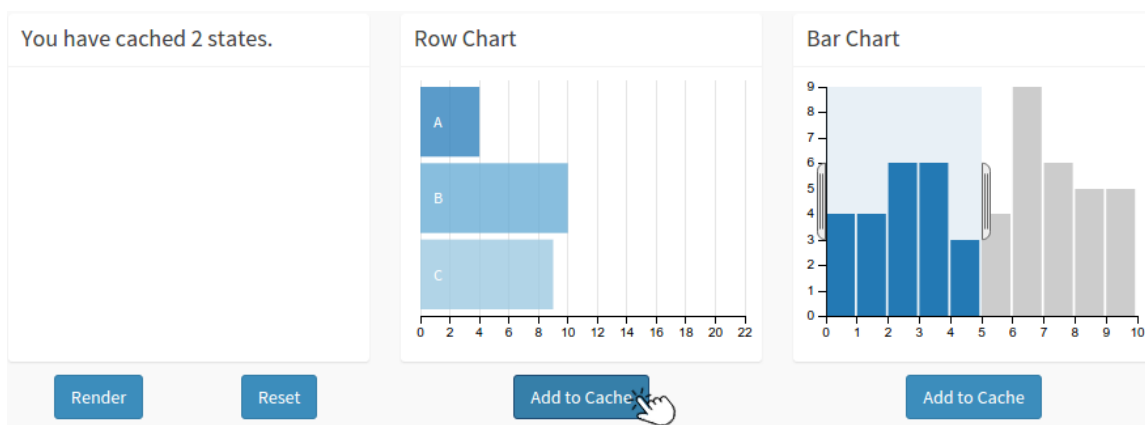
(b) Illustration of adding another – filtered – state of the row chart to CrossCompare's cache (i.e. saving the row chart's filtered data subset) upon re-pressing the 'Add to Cache' button.

(c) Illustration of the comparison chart showing both cached states (i.e. no filters and filtered) in one grouped column chart upon pressing the 'Render' button.

Figure A.1: Sample illustration of CrossCompare.js usage

Appendix B

Signed Ethics Checklist Form

School of Computing Science
University of Glasgow

Ethics checklist for 3rd year, 4th year, MSci, MRes, and taught MSc projects

This form is only applicable for projects that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, getting information about how a system could be used, or evaluating a working system.

If no other people have been involved in the collection of information, then you do not need to complete this form.

If your evaluation does not comply with any one or more of the points below, please submit an ethics approval form to the Department Ethics Committee.

If your evaluation does comply with all the points below, please sign this form and submit it with your project.

---

1. Participants were not exposed to any risks greater than those encountered in their normal working life.

   Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback

2. The experimental materials were paper-based, or comprised software running on standard hardware.

   Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and PDAs is considered non-standard.

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.

   If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be published), then signed consent is necessary. A separate consent form should be signed by each participant.
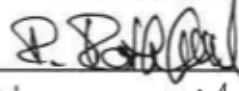
   Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.

4. No incentives were offered to the participants.

   The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.

5. No information about the evaluation or materials was intentionally withheld from the participants.

   Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.

6. No participant was under the age of 16.

   Parental consent is required for participants under the age of 16.

7. No participant has an impairment that may limit their understanding or communication.

   Additional consent is required for participants with impairments.

8. Neither I nor my supervisor is in a position of authority or influence over any of the participants.

   A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.

9. All participants were informed that they could withdraw at any time.

   All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.

10. All participants have been informed of my contact details.

    All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module co-ordinator or supervisor as part of the debriefing.

11. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions.

    The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation.

12. All the data collected from the participants is stored in an anonymous form.

    All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.

Project title *Advancing Web-based Dashboards*

Student's Name *Raoul Rothfeld*

Student's Registration Number *2164502R*

Student's Signature *P. Rothfeld*

Supervisor's Signature *Yashar Moshfeghi*

Date *23/08/2015*

Appendix C

Non-Comparison Study Template

Interactive Dashboard Questionnaire

An air traffic delay dashboard has been developed, as part of a Master of Science project for Software Development at the University of Glasgow. This questionnaire aims at gathering information to improve the presented dashboard and advance web-based dashboard technology.

Research Topic

Dashboards allow non-data scientists to quickly explore and grasp presented information. Yet, dashboard technology often relies on expensive and specialised software. In an effort to facilitate open-source, web-based dashboards, an air traffic delay dashboard has been developed. Further, a comparison feature has been introduced to allow quick comparisons of differing dashboard states.

For this survey, we are exploring the key differences between dashboard usage with and without the comparison feature. In particularly, we are looking at time-efficiency, answering accuracy, and the number of mouse clicks when gaining insights via using the dashboard and its features.

This project is being carried out in compliance with The University of Glasgow's Computer Science Department's Ethics Standards.

Survey Structure

The survey consists of four parts: introductory questions, a training phase, task phase (with four tasks), and conclusive questions:

The **introductory questions** are meant to gather information on you and your existing experience with dashboards.

The **training phase** is meant to familiarise yourself with the dashboard, its features, and the upcoming tasks.

In the **task phase**, you will be presented **four task in random order**. Some of which are more difficult than others and two of which allow you to use an added comparison ability. This comparison feature will be explained to you before the task that requires its usage. You will have a **comparison training** phase before the task that requires usage of the comparison feature. Each task will be followed by some post-task questions.

Conclusive questions are meant to identify your preferences and opinions.

Consent Form

Project Title: **Advancing web-driven Dashboards**

Researcher: **Raoul Rothfeld** (2164502r@student.gla.ac.uk)

Please tick box

1. I confirm I have read and understand the information sheet for the above study and have had the opportunity to ask questions. ☐

2. I understand that my permission is voluntary and that I am free to withdraw at any time, without giving any reason, without my legal rights being affected, and that any data gathered on me will be destroyed. ☐

3. I confirm that I allow the collection and use of any data gathered on me (task completion time and number of mouse clicks), during the above study, for analysis purposes. ☐

4. I acknowledge that all material gathered on me during this study will be permanently destroyed upon the completion of the data analysis. ☐

5. I agree to take part in the above study. ☐

6. I would like to receive a summary sheet of the experimental findings ☐

If you wish a summary, please leave an email address: _____

_____   _____   _____
Name of Participant   Date   Signature

_____   _____   _____
Researcher   Date   Signature

Personal Details

What is you gender?

☐ Female

☐ Male

What is you age?

☐ 18 to 24

☐ 25 to 34

☐ 35 to 44

☐ 45 to 54

☐ 55 to 64

☐ 65 or older

What is you nationality? _____

What is you current occupation? _____

What is your field of work/study? _____

What is the highest educational degree that you hold at the moment?

☐ Bachelor's degree

☐ Master's degree

☐ Doctor of Philosophy (PhD)

☐ None of the above

Existing Experience

How familiar are you with reading charts and graphs (e.g. Excel/Powerpoint)?

- [ ] Extremely familiar
- [ ] Very familiar
- [ ] Moderately familiar
- [ ] Slightly familiar
- [ ] Not at all familiar

How familiar are you with dashboard usage?

- [ ] Extremely familiar
- [ ] Very familiar
- [ ] Moderately familiar
- [ ] Slightly familiar
- [ ] Not at all familiar

How familiar are you with air traffic data?

- [ ] Extremely familiar
- [ ] Very familiar
- [ ] Moderately familiar
- [ ] Slightly familiar
- [ ] Not at all familiar

Training Task

Please notify the researcher that you are ready for the training question. He will guide you through the exemplary task below.

Please enter the questionnaire number (first page) into the pop-up box on your screen.

How many flights were operated with delays of 0-20 and 40-60 minutes?

Flights with 0-20 min. delay: 655 _____

Flights with 40-60 min. delay: _____

Additional information: Hovering the mouse cursor row, line, and area charts will bring up information about the underlying datapoint.

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Training'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Comparison Training Task

Please notify the researcher that you are ready for the comparison training question. He will guide you through the exemplary task below.

How many flights did each airline operate at on December 27, at 08:00?

Number of flights by American (AA):   8

Number of flights by Jetblue (B6):   _____

Number of flights by Delta (DL):   _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Training'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task A0

For this task, you will <u>NOT</u> be able to use the comparison feature.

At what time on December 26, did each airline have its peak flights per hour?

Time of day with max. flights by American (AA):   _____

Time of day with max. flights by Jetblue (B6):   _____

Time of day with max. flights by Delta (DL):   _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task A1

For this task, you <u>MUST</u> use the comparison feature.

At what time on December 26, did each airline have its peak flights per hour?

Time of day with max. flights by American (AA): _____

Time of day with max. flights by Jetblue (B6): _____

Time of day with max. flights by Delta (DL): _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task B0

For this task, you will <u>NOT</u> be able to use the comparison feature.

What airport provides the most connecting flights before 12:00, with at least 20 minutes delay, from/to BOS, LAS, and GPT?

Airport with most flights from/to BOS: _____

Airport with most flights from/to LAS: _____

Airport with most flights from/to GPT: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task B1

For this task, you <u>MUST</u> use the comparison feature.

What airport provides the most connecting flights before 12:00, with at least 20 minutes delay, from/to BOS, LAS, and GPT?

Airport with most flights from/to BOS: _____

Airport with most flights from/to LAS: _____

Airport with most flights from/to GPT: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task C0

For this task, you will <u>NOT</u> be able to use the comparison feature.

How many flights were operated per weekday?

Flights on Monday: _____

Flights on Wednesday: _____

Flights on Saturday: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task C1

For this task, you <u>MUST</u> use the comparison feature.

How many flights were operated per weekday?

Flights on Monday: _____

Flights on Wednesday: _____

Flights on Saturday: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task D0

For this task, you will <u>NOT</u> be able to use the comparison feature.

How many American (AA) airline flights were operated during the busiest hour at Chicago (ORD) airport, which had a delay of 0-60 and 60-120 minutes?

Number of flights during busiest hour with delay of 0-60 min.:   _____

Number of flights during busiest hour with delay of 60-120 min.:   _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task D1

For this task, you <u>MUST</u> use the comparison feature.

How many American (AA) airline flights were operated during the busiest hour at Chicago (ORD) airport, which had a delay of 0-60 and 60-120 minutes?

Number of flights during busiest hour with delay of 0-60 min.:   _____

Number of flights during busiest hour with delay of 60-120 min.:   _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Post-Task Questions (with comparison)

How difficult was the task?

☐ Very simple

☐ Somewhat simple

☐ Somewhat difficult

☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task

☐ I am satisfied with my performance in this task

☐ I think I could have done better, with _____

Did the comparison feature increase the dashboard's usability for this task?

☐ Yes

☐ No

☐ Indifferent

Do you have any other comments or questions regarding this task?

Post-Task Questions (with comparison)

How difficult was the task?

☐ Very simple

☐ Somewhat simple

☐ Somewhat difficult

☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task

☐ I am satisfied with my performance in this task

☐ I think I could have done better, with _____

Did the comparison feature increase the dashboard's usability for this task?

☐ Yes

☐ No

☐ Indifferent

Do you have any other comments or questions regarding this task?

Post-Task Questions (without comparison)

How difficult was the task?

☐ Very simple
☐ Somewhat simple
☐ Somewhat difficult
☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task
☐ I am satisfied with my performance in this task
☐ I think I could have done better, with _____

Would the comparison feature have increased the dashboard's usability for this task?

☐ Yes
☐ No
☐ Indifferent

Do you have any other comments or questions regarding this task?

Post-Task Questions (without comparison)

How difficult was the task?

☐ Very simple
☐ Somewhat simple
☐ Somewhat difficult
☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task
☐ I am satisfied with my performance in this task
☐ I think I could have done better, with _____

Would the comparison feature have increased the dashboard's usability for this task?
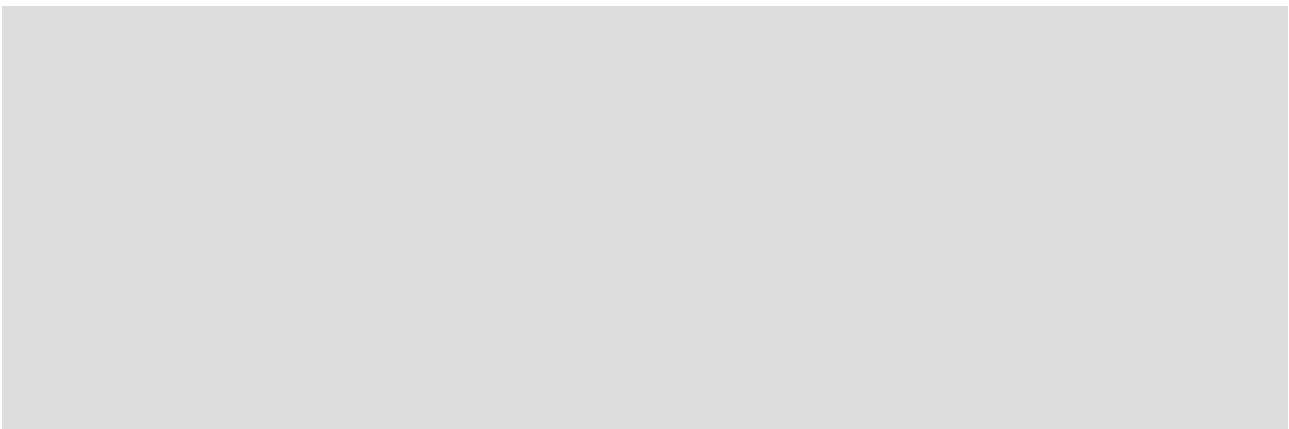
☐ Yes
☐ No
☐ Indifferent

Do you have any other comments or questions regarding this task?

Conclusive Questions

Overall, did you prefer dashboard usage with or without the comparison feature?

☐ With comparison feature

☐ Without comparison feature

Please explain your overall preference?

Overall, how helpful was the comparison feature in completing the tasks?

☐ Very helpful

☐ Somewhat helpful

☐ Not helpful at all

☐ Hindered task completion

Overall, how intuitive was the comparison feature in usage?

☐ Very intuitive

☐ Somewhat intuitive

☐ Somewhat unintuitive

☐ Very unintuitive

Given you were professionally interested in air traffic delay data, please rate the overall dashboard on the following statements:

	Strongly disagree				Strongly agree

I think that I would like to use this system frequently:

1	2	3	4	5

I found the system unnecessarily complex:

1	2	3	4	5

I thought the system was easy to use:

1	2	3	4	5

I think that I would need the support of a technical person to be able to use this system:

1	2	3	4	5

I found the various functions in this system were well integrated:

1	2	3	4	5

I thought there was too much inconsistency in this system:

1	2	3	4	5

I would imagine that most people would learn to use this system very quickly:

1	2	3	4	5

I found the system very cumbersome to use:

1	2	3	4	5

I felt very confident using the system:

1	2	3	4	5

I needed to learn a lot of things before I could get going with this system:

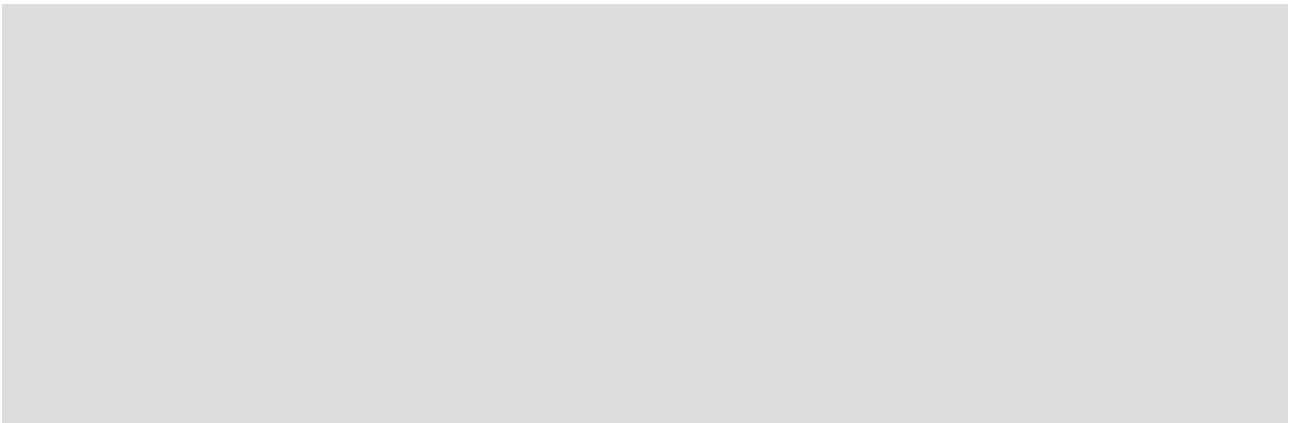|				
1	2	3	4	5

Is the dashboard's design aesthetically appealing?

☐ Very appealing

☐ Somewhat appealing

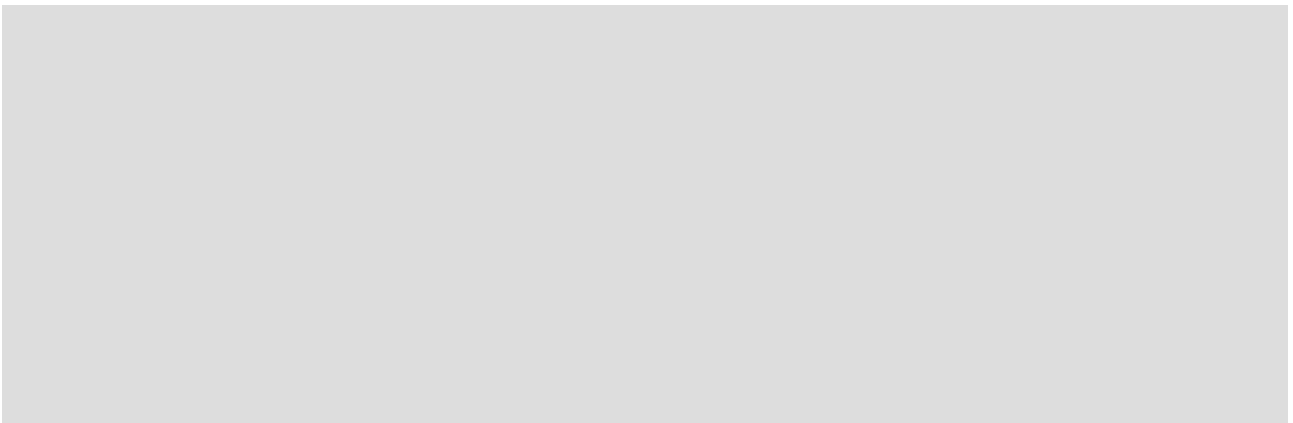☐ Somewhat unappealing

☐ Very unappealing

Overall, are you satisfied with your experience using the dashboard?

☐ Very satisfied

☐ Somewhat satisfied
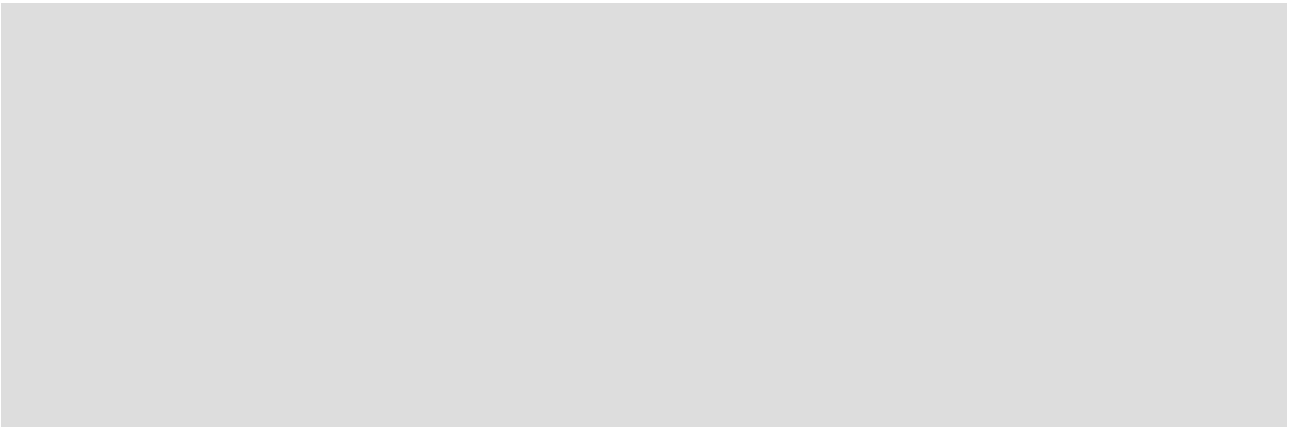
☐ Somewhat dissatisfied

☐ Very dissatisfied

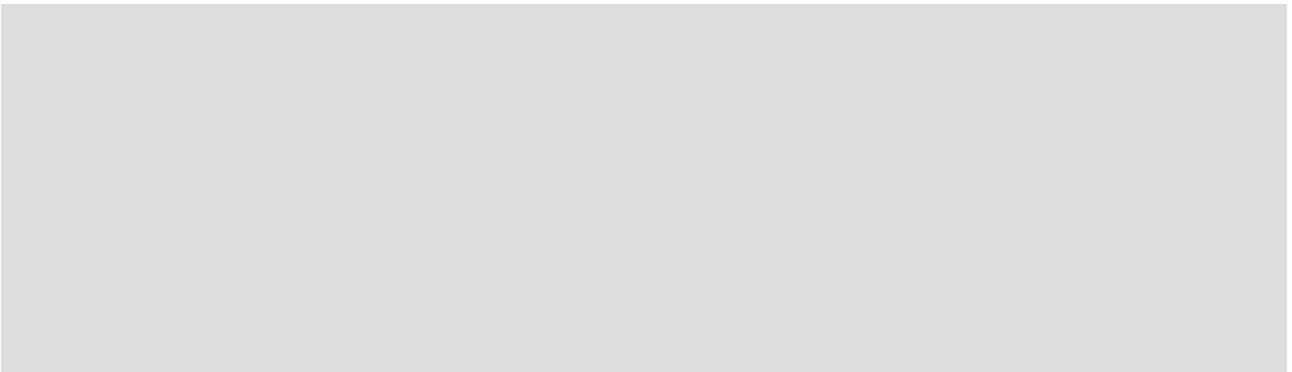**What are the things that you like most about the dashboard?**

What are the things that you would most like to be improved?

Do you have any other comments or questions regarding the dashboard?

Do you have any other comments or questions regarding the questionnaire?

Appendix D

Comparison Study Template

Interactive Dashboard Questionnaire

An air traffic delay dashboard has been developed, as part of a Master of Science project for Software Development at the University of Glasgow. This questionnaire aims at gathering information to improve the presented dashboard and advance web-based dashboard technology.

Research Topic

Dashboards allow non-data scientists to quickly explore and grasp presented information. Yet, dashboard technology often relies on expensive and specialised software. In an effort to facilitate open-source, web-based dashboards, an air traffic delay dashboard has been developed. Further, a comparison feature has been introduced to allow quick comparisons of differing dashboard states.

For this survey, we are exploring the key differences between dashboard usage with and without the comparison feature. In particularly, we are looking at time-efficiency, answering accuracy, and the number of mouse clicks when gaining insights via using the dashboard and its features.

This project is being carried out in compliance with The University of Glasgow's Computer Science Department's Ethics Standards.

Survey Structure

The survey consists of four parts and will approximately take **30 minutes** to complete.

The **introductory questions** are meant to gather information on you and your existing experience with dashboards.

The **training phase** is meant to familiarise yourself with the dashboard, its features, and the upcoming tasks.

In the **task phase**, you will be asked to find the solutions to three questions using the dashboard and its features. During the task phase, certain dashboard features will be disabled.

Conclusive questions are meant to identify your preferences and opinions about the dashboard and certain features.

Consent Form

Project Title: **Advancing web-driven Dashboards**

Researcher: **Raoul Rothfeld** (2164502r@student.gla.ac.uk)

Please tick box

1. I confirm I have read and understand the information sheet for the above study and have had the opportunity to ask questions. ☐

2. I understand that my permission is voluntary and that I am free to withdraw at any time, without giving any reason, without my legal rights being affected, and that any data gathered on me will be destroyed. ☐

3. I confirm that I allow the collection and use of any data gathered on me (task completion time and number of mouse clicks), during the above study, for analysis purposes. ☐

4. I acknowledge that all material gathered on me during this study will be permanently destroyed upon the completion of the data analysis. ☐

5. I agree to take part in the above study. ☐

6. I would like to receive a summary sheet of the experimental findings ☐

If you wish a summary, please leave an email address: _____

_____   _____   _____
Name of Participant   Date   Signature

_____   _____   _____
Researcher   Date   Signature

Personal Details

What is your gender?

☐ Female

☐ Male

What is your age?

☐ 18 to 24

☐ 25 to 34

☐ 35 to 44

☐ 45 to 54

☐ 55 to 64

☐ 65 or older

What is your nationality? _____

What is your current occupation? _____

What is your field of work/study? _____

What is the highest educational degree that you hold at the moment?

☐ Bachelor's degree

☐ Master's degree

☐ Doctor of Philosophy (PhD)

☐ None of the above

Existing Experience

How familiar are you with reading charts and graphs (e.g. Excel/Powerpoint)?

☐ Extremely familiar

☐ Very familiar

☐ Moderately familiar

☐ Slightly familiar

☐ Not at all familiar

How familiar are you with dashboard usage?

☐ Extremely familiar

☐ Very familiar

☐ Moderately familiar

☐ Slightly familiar

☐ Not at all familiar

How familiar are you with air traffic data?

☐ Extremely familiar

☐ Very familiar

☐ Moderately familiar

☐ Slightly familiar

☐ Not at all familiar

Training Task

Please notify the researcher that you are ready for the training question. He will guide you through the exemplary task below.

Please enter the questionnaire number (first page) into the pop-up box on your screen.

For flights grouped by delay length, how often does Jetblue (B6) airlines operate more flights than American (AA) airlines?

Number of times where B6 operates more flights by delay than AA: 3

Additional information: Hovering the mouse cursor row, line, and area charts will bring up information about the underlying data-point. Pressing the i-button above a chart brings up more information about the data presented and possibilities for in-chart interaction.

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Training'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task A0

For this task, you will <u>NOT</u> be able to use the comparison feature.

On Thursday the 25th, what is the difference in number of flights with a delay of less than 20 minutes compared to the number of flights with a delay of 20 or more minutes; at 8, 12, 16, and 20 o'clock?

Difference in number of flights with <20 to ≥20 min. delay at 08:00:   _____

Difference in number of flights with <20 to ≥20 min. delay at 12:00:   _____

Difference in number of flights with <20 to ≥20 min. delay at 16:00:   _____

Difference in number of flights with <20 to ≥20 min. delay at 20:00:   _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task A1

For this task, you <u>MUST</u> use the comparison feature.

On Thursday the 25th, what is the difference in number of flights with a delay of less than 20 minutes compared to the number of flights with a delay of 20 or more minutes; at 8, 12, 16, and 20 o'clock?

Difference in number of flights with <20 to ≥20 min. delay at 08:00: _____

Difference in number of flights with <20 to ≥20 min. delay at 12:00: _____

Difference in number of flights with <20 to ≥20 min. delay at 16:00: _____

Difference in number of flights with <20 to ≥20 min. delay at 20:00: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task B0

For this task, you will <u>NOT</u> be able to use the comparison feature.

For each day of a week, which airport has the most flights per weekday?

Airport with the most flights on Monday: _____

Airport with the most flights on Tuesday: _____

Airport with the most flights on Wednesday: _____

Airport with the most flights on Thursday: _____

Airport with the most flights on Friday: _____

Airport with the most flights on Saturday: _____

Airport with the most flights on Sunday: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task B1

For this task, you <u>MUST</u> use the comparison feature.

For each day of a week, which airport has the most flights per weekday?

Airport with the most flights on Monday: _____

Airport with the most flights on Tuesday: _____

Airport with the most flights on Wednesday: _____

Airport with the most flights on Thursday: _____

Airport with the most flights on Friday: _____

Airport with the most flights on Saturday: _____

Airport with the most flights on Sunday: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task C0

For this task, you will <u>NOT</u> be able to use the comparison feature.

At New York (JFK) airport, how often does Delta (DL) airlines operate more flights per hour (time of day) than any other airline?

Number of times where DL operates most flights per hour: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Task C1

For this task, you <u>MUST</u> use the comparison feature.

At New York (JFK) airport, how often does Delta (DL) airlines operate more flights per hour (time of day) than any other airline?

Number of times where DL operates most flights per hour: _____

The necessary dashboard components to answer this question will be highlighted with red background or borders within the dashboard.

Press **'Start Next Task'** on the dashboard screen, when you want to start the training.

Press **'Continue'** in the upper right corner, when you have collected all answers.

Post-Task Questions

How difficult was the task?

☐ Very simple

☐ Somewhat simple

☐ Somewhat difficult

☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task

☐ I am satisfied with my performance in this task

☐ I think I could have done better, with _____

Given you were professionally interested in air traffic delay data, please rate the dashboard for the before task on the following statements:

	Strongly disagree				Strongly agree
I am satisfied with the ease of completing the task:	1	2	3	4	5
I am satisfied with the amount of time it took to complete the task:	1	2	3	4	5
I am satisfied with the support information (labelling, pop-up information) when completing the tasks:	1	2	3	4	5

Do you have any other comments or questions regarding this task?

Post-Task Questions

How difficult was the task?

☐ Very simple

☐ Somewhat simple

☐ Somewhat difficult

☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task

☐ I am satisfied with my performance in this task

☐ I think I could have done better, with _____

Given you were professionally interested in air traffic delay data, please rate the dashboard for the before task on the following statements:

	Strongly disagree				Strongly agree
I am satisfied with the ease of completing the task:	1	2	3	4	5

	Strongly disagree				Strongly agree
I am satisfied with the amount of time it took to complete the task:	1	2	3	4	5

	Strongly disagree				Strongly agree
I am satisfied with the support information (labelling, pop-up information) when completing the tasks:	1	2	3	4	5

Do you have any other comments or questions regarding this task?

Post-Task Questions

How difficult was the task?

☐ Very simple

☐ Somewhat simple

☐ Somewhat difficult

☐ Very difficult

How satisfied were you with the task?

☐ I believe I have succeeded in my performance of this task

☐ I am satisfied with my performance in this task

☐ I think I could have done better, with _____

Given you were professionally interested in air traffic delay data, please rate the dashboard for the before task on the following statements:

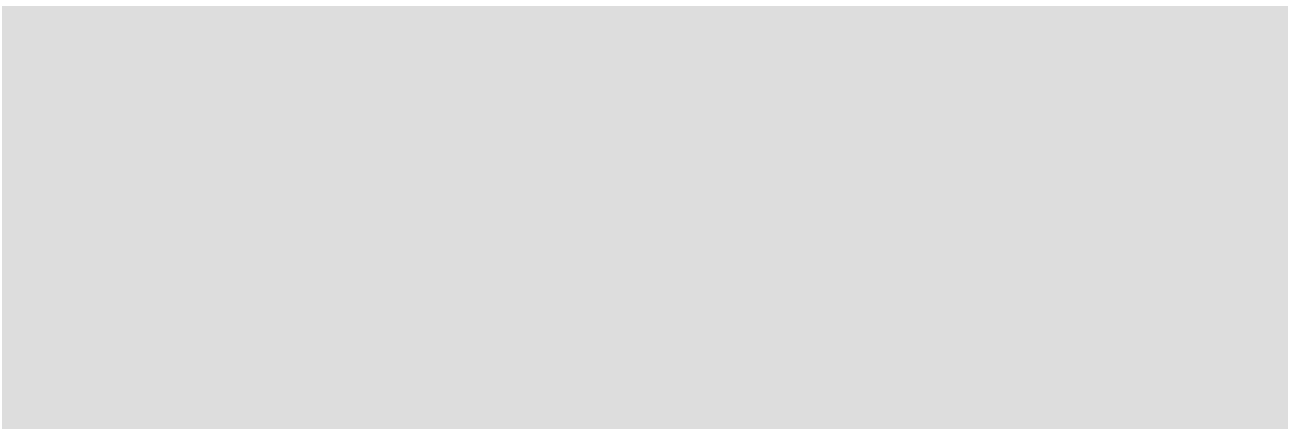|	Strongly disagree				Strongly agree
I am satisfied with the ease of completing the task:	1	2	3	4	5
I am satisfied with the amount of time it took to complete the task:	1	2	3	4	5
I am satisfied with the support information (labelling, pop-up information) when completing the tasks:	1	2	3	4	5

Do you have any other comments or questions regarding this task?

Conclusive Questions

Overall, did you prefer dashboard usage with or without the comparison feature?

☐ With comparison feature

☐ Without comparison feature

Please explain your overall preference?

Overall, how helpful was the comparison feature in completing the tasks?

☐ Very helpful

☐ Somewhat helpful

☐ Not helpful at all

☐ Hindered task completion

Overall, how intuitive was the comparison feature in usage?

☐ Very intuitive

☐ Somewhat intuitive

☐ Somewhat unintuitive

☐ Very unintuitive

Given you were professionally interested in air traffic delay data, please rate the overall dashboard on the following statements:

	Strongly disagree				Strongly agree
I think that I would like to use this system frequently:	1	2	3	4	5

I found the system unnecessarily complex:	1	2	3	4	5

I thought the system was easy to use:	1	2	3	4	5

I think that I would need the support of a technical person to be able to use this system:	1	2	3	4	5

I found the various functions in this system were well integrated:	1	2	3	4	5

I thought there was too much inconsistency in this system:	1	2	3	4	5

I would imagine that most people would learn to use this system very quickly:	1	2	3	4	5

I found the system very cumbersome to use:	1	2	3	4	5

I felt very confident using the system:	1	2	3	4	5

I needed to learn a lot of things before I could get going with this system:	1	2	3	4	5

Is the dashboard's design aesthetically appealing?

☐ Very appealing

☐ Somewhat appealing

☐ Somewhat unappealing
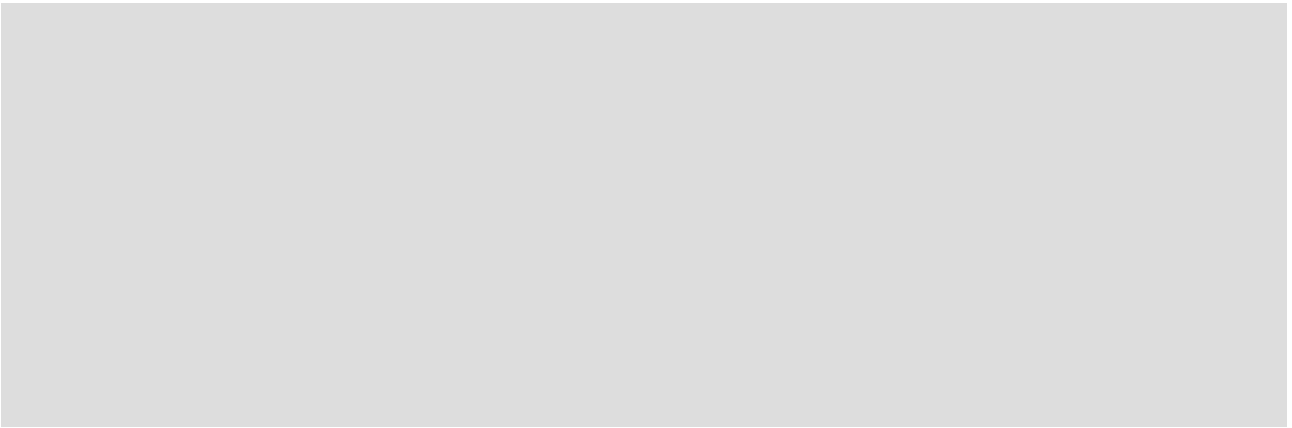
☐ Very unappealing

Overall, are you satisfied with your experience using the dashboard?

☐ Very satisfied

☐ Somewhat satisfied

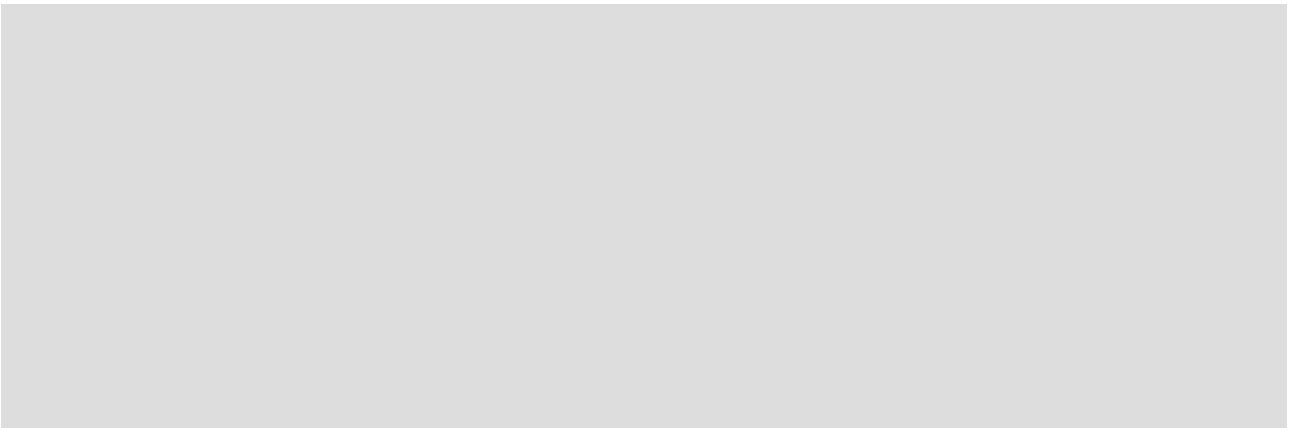☐ Somewhat dissatisfied

☐ Very dissatisfied

What are the things that you like most about the dashboard?

What are the things that you would most like to be improved?

Do you have any other comments or questions regarding the dashboard?

Do you have any other comments or questions regarding the questionnaire?

Appendix E

Quantitative Task Data

Table E.1: Mean completion times and mouse clicks per task ($M(SD)$), including perceived difficulty (0 (easy) to 100 (difficult))

Task	Task completion time		Mouse Clicks		Difficulty
	Without	With	Without	With	
(1a)	78.17 (29.60)	103.83 (33.39)	10.67 (2.94)	18.50 (6.02)	10.42
(1b)	169.00 (52.75)	234.67 (143.04)	24.83 (14.22)	35.33 (21.42)	35.42
(1c)	58.00 (27.66)	123.67 (114.07)	10.83 (4.54)	30.83 (34.08)	16.67
(1d)	229.00 (211.38)	174.83 (95.73)	17.33 (4.50)	22.83 (8.73)	29.17
(2a)	449.50 (162.97)	244.83 (139.94)	62.67 (33.90)	29.83 (20.80)	31.25
(2b)	228.00 (119.99)	95.50 (58.84)	60.17 (25.17)	19.67 (12.37)	12.50
(2c)	571.67 (353.95)	128.83 (36.28)	143.83 (55.30)	21.17 (9.22)	27.08