Advancing-web driven Dashboards: Development of a marketable, comparison-enabling Dashboard

Raoul 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

July 29, 2015

In cooperation with WINGX Advance GmbH

Abstract

abstract goes here
Keywords: Data Discovery, Dashboard, Open-Source, Data Visualisation

Education Use Consent

Permission is hereby given for this project to be shown to other University of Glasgow students and to be distributed in an electronic format.

Acknowledgements

I am deeply grateful to Dr. Yashar Moshfeghi whose enormous support and insightful comments were invaluable during the course of this study and who, both, challenged and encouraged me continuously throughout the past year. I am also in debt to Christoph Kohler, XXX, and XXX; who provided me with invaluable insights and warm encouragements.
Thank you very much for your sustained support. + SFC funding

Acronyms

BI Business Intelligence
HCI Human Computer Interaction
KPI Key Performance Indicator

Contents

Chapter 1

Introduction

project meant to show marketable dashboard via browser... (every Client might habe different resolution... Literatur geht von einer spezifizierten Auflsung aus...)

"Technical 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." [11]

In a global business environment, where economic cycles shorten and competitiveness increases, the importance of having "the right information, at the right time, to make the right business decisions" [6] is more important than ever. Technological advances allowed for the introduction of business intelligence (BI) as a data-driven decision support system, which is continuously 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 doesnt overwhelm them" [2].

Yet, 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 a dashboard. Recently however, community efforts have emerged that aim to provide dashboard functionalities via open-sourced programming libraries. It has yet to be seen however, whether or not these libraries can be utilised to realise an effective and marketable data discovery solution.

1.1 Problem Statement

While data discovery tools provide the capabilities of visually and interactively exploring given data, they require desktop installations and/or server-side deployments including commercial licences. Web-based, open-source dashboard frameworks and libraries facilitate

the independent development of dashboards, yet lack fundamental data discovery functionalities. A coherent web-based, open-source data discovery framework has yet to be developed as an increasing number of open-source libraries provide partial data discovery functionality or allow for the development of such.

Spurring and advancing the development of open-source data discovery frameworks 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 data discovery capabilities correlates with the increasing 'webification' [1] of – traditional – desktop applications, allowing for more a versatile and flexible utilisation of data discovery.

1.2 Research Objective

This project shall elaborate on the emergence and design of dashboards, elucidate data discovery and its implications on human computer interaction (HCI), document the implementation of a web-based data discovery platform, and conclude with the proposal and evaluation of a visual, ad hoc data comparison feature. In particular, the project aims at meeting the following objectives:

- Identification of current dashboard solutions (software suites as well as web-based, open-source components).
- Review of contemporary dashboard usage and design practices.
- Development of an exemplary interactive, web-based data discovery dashboard using open-source resources.
- Development and evaluation of an ad hoc, graphical comparison component.
- 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

The evolution of business intelligence through to data discovery, the properties of data discovery, and existing tools shall be introduced. While – both – commercial desktop applications and open-source libraries are being presented, an in-depth discussion of potential deployment costs is deliberately excluded.

A review of HCI aspects regarding data discovery dashboards is followed by documentation of the requirements, design, use case, and utilised frameworks of the data discovery dashboard and its ad hoc comparison feature. Upon describing the exemplary implementation

and its development challenges, follows a two-fold evaluation with a description of provided test cases and an analysis of conducted user experiments. A review of the project with regards to the above-mentioned research objectives, together with a roadmap for further development, and suggestions for future research will conclude this thesis.

Chapter 2

Background Review

Dashboards are expected to improve decision making by amplifying cognition and capitalizing on human perceptual capabilities. Hence, interest in dashboards has increased recently, which is also evident from the proliferation of dashboard solution providers in the market. [14]

'Given the dearth of research on dashboards' [14]

'Although dashboards seem to have caught on as a management tool, the scientific literature has failed to keep pace with the developments. While textbooks (e.g. Few, 2006; Rasmussen et al., 2009)and articles in business press (e.g. Miller and Cioffi, 2004; Kawamoto and Mathers, 2007) on dashboards abound, only a handful of studies can be found in academic journals, providing little guidance for practitioners (Pauwels et al., 2009) and researchers.' [14]

–¿ Ich leg keinen Fokus auf: DAShBOARD IN ORGaNIZATIONAL readiness [3] oder deployment [3], also change management und adoption im job, effectiveness (vgl. zb [12]), sondern eher Dashboard entstehung, entwicklung und HCI aspects

2.1 Emergence of Dashboard Information Systems

"BI emerged as a distinct discipline in the early 1990s as a way to provide end users with better access to information for decision making. The initial goal was to give users self - service access to information so they did not have to rely on the IT department to create custom reports. By the early 1990s, BI consisted of two data warehousing and query and reporting tools. Companies began building data warehouses as a way to offload queries from operational systems. Data warehouses became analytical playgrounds that let users query all the data they wanted without bogging down the performance of operational systems. At the time, users needed to know SQL, a database query language, to query the data warehouse. Prescient vendors began shipping query and reporting tools that hid SQL behind a point - and - click Windows interface. Vendors converted these desktop query and reporting tools to the Web in the late 1990s and bundled them with other types of analytical tools to create BI suites or BI platforms. Modern BI. Today, BI is an umbrella

term that encompasses a raft of data warehousing and data integration technologies as well as query, reporting, and analysis tools that fulfill the promise of giving business users self - service access to information. Performance dashboards represent the latest incarnation of BI, building on years of innovation to deliver an inter- face that conforms to the way a majority of users want to consume information." [3, p. 32]

"XmdvTool system" [9]

"The use of dashboards to monitor strategy implementation at various organizational levels is an accelerating trend. Use of the Internet for business, e-mail, social media, and other activities has changed visual expectations. Dashboard graphics can enhance ease of use and provide for instant recognition of important changes in key performance metrics. Professional services firms should consider strategic dashboards for practice management as well as consulting services offerings." [2]

"Historically, the idea of digital dashboards follows the work in the 1970s with the study of decision support systems. In the late 1990s with the surge of the web, digital dashboards as we know them today began appearing. Many systems were home built as the emphasis on efficiency became a passion. Today, digital dashboard technology is available "out-of-the-box" with many software providers on the scene." [10]

"Whereas some solutions come with full features, i.e. interactive drill down capabilities, scenario (what-if) analysis, built- in automated alerts, customization options, etc., others are more simple and static by nature. Dashboards have been well received and interest in them is growing. For example, Negash and Gray (2008) regard themas one of themost useful analysis tools in BI." [14]

dashboard only as decision support systems WITHIN companies[12] [3] ... –¿ more and more dashboards for general information services

"During the 1990s, data warehousing, online analytical processing (OLAP), and eventually business intelligence worked as partners to tame the wild onslaught of the information age. The emphasis during those years was on collecting, correcting, integrating, storing, and accessing information in ways that sought to guarantee its accuracy, timeliness, and usefulness. From the early days of data warehousing on into the early years of this new millennium, the effort has largely focused on the technologies, and to a lesser degree the methodologies, needed to make information available and useful. The direct beneficiaries so far have mostly been folks who are highly proficient in the use of computers and able to use the available tools to navigate through large, often complex databases. What also emerged in the early 1990s, but didn't become popular until late in that decade, was a new approach to management that involved the identification and use of key performance indicators (KPIs), introduced by Robert S. Kaplan and David P. Norton as the Balanced Scorecard. The advances in data warehousing and its technology partners set the stage for this new interest in management through the use of metrics and not just financial metrics that still dominates the business landscape today. Business Performance Management (BPM), as it is now commonly known, has become an international preoccupation. The infrastructure built by data warehousing and the like, as well as the interest of BPM in metrics that can be monitored easily, together tilled and fertilized the soil in which the hibernating seeds of EIStype displays were once again able to grow. What really caused heads to turn in recognition of dashboards as much more than your everyday fledgling technology,

however, was the Enron scandal in 2001. The aftermath put new pressure on corporations to demonstrate their ability to closely monitor what was going on in their midst and to thereby assure shareholders that they were in control. This increased accountability, combined with the concurrent economic downturn, sent Chief Information Officers (CIOs) on a mission to find anything that could help managers at all levels more easily and efficiently keep an eye on performance. 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. The marketplace soon offered a vast array of dashboard software from which to choose." [4, p. 15]

"The architecture of performance dashboards have followed the trajectory of software architectures in general, from mainframe computing to client/server computing to Web - based architectures. Today, rich Internet applications (RIAs) are gaining in popularity because they support dazzling multimedia and visual effects, boosting the appeal and usability of performance dashboards. RIAs, such as Adobe Flash, enable Web - based applications to exhibit the richness and interactivity of desktop applications" [3, p. 251]

"Riding the technology wave, we are awash in data. Attempts to stem the tide, or at least to manage its flow, have led to a proliferation of dashboards. With data dashboards, organizations consolidate important data in a single place, typically accessed via web browser. Dashboard contents may be tables, graphics, or visual key performance indicators (KPIs). While dashboards prolif- erate, displaying actionable data to support decisions, they are often developed by technical professionals inexperienced in human-computer interaction design. Research abounds on visual perception, but typically this is in the context of in-" [11]

"The main reason for data visualization is the limitation of human beings to absorb the large amount of information. The volumes of data are overwhelming and the human visual systems and brain are not equipped to work with the data in this form [2]. Using data visualization, we allow much faster processing of the data and the ability to see the patterns in the data. On the other hand, data" [10]

2.2 What is a Dashboard

"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." [4, p. 26]

"A dashboard can be regarded as a data driven decision support system,which provides information in a particular format to the decision maker." [14]

"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 one or several individual and/or organizational objectives, allowing the user to identify, explore, and communicate problem areas that need corrective action. This definition

emphasizes their interactivity and refines its purpose." [14]

"Dashboards became popular after the Enron scandal in 2001 (Few, 2006) but there is not a clear definition of dashboards, neither given by software vendors nor by academics. The dashboard vendors define dashboards fromthe perspective of characteristics that their products have. Researchers talk about different types of applications of the dashboard concept and different stages in their development (Pauwels et al., 2009)." [14]

Dashboard: 'Essentially, a dashboard focuses on a goal or objective, and it displays the most relevant information on a digital screen. An effective dashboard utilizes visu- alization techniques and cues to engage a user in the information processing experience. The visual techniques may include the use of colors, dials, buttons, graphs, and the positioning of information on the screen.' [2]

usage: 'Professional services firms can use strategic dashboards for practice management, as well as with- in their consulting services offerings.' [2]

Clarification of the meaning and characteristics of a "dashboard" in the context of this project.

Even though data discovery is an already applied BI method, the term itself has not yet been universally defined. The term data discovery, or often referred to as business discovery, visual discovery, exploratory analytics, or investigative analytics; has yet to be understood as a rudimentary outline for how companies want to interact with their data in regards to presentation and utilization. 26 Currently, data discovery means 'different things to different people' 27 depending on the context in which the term is being utilised. In regards to the scope and limitation of this thesis, data discovery is being defined as follows within this thesis:

> 'Data discovery is a business intelligence architecture which allows its users to conduct near real-time analysis of data, through an interactive and intuitive interface.' (OWN DEFINITION based on 28)

+ distinction between DD and Dashboard

type of dashboards: "Operational dashboards monitor operational processes, events, and activities as they occur (every minute, hour, or day). 2. Tactical dashboards measure and analyze the performance of depart- mental activities, processes, and goals. 3. Strategic dashboards track progress toward achieving strategic objectives in a top - down fashion (e.g., a Balanced Scorecard). All"[3, p. 101]

zu tactical: "Visual Analysis. Increasingly, analysis is being done with visual analysis tools that enable analysts to visually interact with charts and tables at the speed of thought and quickly spot outliers. The tools compress and store data in memory, providing subsecond response times for any action taken against the data (e.g., fi ltering, drilling, calculating, sorting, and ranking). Visually, analysts point and click to interact with charts, apply fi lters, and change views. For instance, analysts can use their mouse to lasso data points in a scatter plot to create a new group. This action also automatically fi lters other charts on the page so users can quickly ascertain the" [3, p. 113]

2.3 Dashboard Tools and Service Providers

Comparison of existing dashboard providers; in regards to type (software, web service, framework), features, and architecture. E.g. Qlik/QlikView, Tableau, SAP Hana, ...

Look at the position of dashboard solutions within the BI market, via Gartner's Magic Quadrant.

"A wide range of dashboard development tools are available. Dashboard capabilities have increasingly appeared in product offerings from major vendors, such as SAPs BusinessObjects/Crystal Dashboard Design (formerly called Xcelsius) software and Oracles Business Intelligence Suite. These integrated vendor programs pro- vide tools for designing dashboards that can access and display information from the companys data warehouse. Some tools, such as Crystal Dashboard Design, became popular because they made it easier for end users to develop their own dashboards if they were already integrated with Microsoft Excel. They are also powerful because they can be expanded to support the develop- ment of enterprise-wide dashboard initia- tives. The programs available from major vendors have higher costs that must be jus- tified by improved ease of use, data inte- gration, and quality of the visualization fea- tures. Another issue is the availability of IT department support and policies." [2]

excel can dashboard: "Microsoft Excel is an increasingly attrac- tive tool for developing dashboards. It is a very cost-effective solution for many companies because their personnel may already have Excel skills. Many" [2] but not data discovery

2.4 Open-source Dashboard Components

Existing dashboard frameworks, open for general and commercial use. Reasons for why this project aims at developing one.

currently lacking: ability to quickly add/remove/compare dimensions, adding data itself is possible

2.5 Dashboard Development

"Strategy. Does your organization have a clear, coherent strategy with well - defi ned goals, objectives, and measures? Sponsorship. Is there a high - level executive who strongly believes in the project and is willing to spend time evangelizing and nurturing the project? Urgent need. Does the organization have a demonstrated need for the system? How much is it suffering from an inability to track and measure performance? Buy - in. How willing are mid - level managers to support the project? Will the open sharing of performance results threaten their positions and their hold on power? Scope. Does the group have suffi cient scope so that the implementa- tion can be adapted by other groups in the organization? Team. Does the group have business and technical people with proper skills and experience to deliver a successful project? Culture. Does the group already have

a culture of measurement and make decisions by fact instead of intuition? Alignment. How aligned are the business and technical teams? Do they have a good working relationship and trust one another? Data. Do data exist to populate the measures? How clean, valid, and complete are the data? Infrastructure. Does the group have a solid technical infrastructure that generates the required data and delivers it to users in a format that is easy to monitor and analyze?" [3, p. 55]

"Stage I: Selecting the Key Metrics Ambler Stage II: Populating the Dashboard With Data Using Stage III: Establishing Relationships Between the Dashboard Items Stage Stage IV: Forecasting and Scenarios Stage Stage V: Connecting to Financial Consequences Stage"[12]

dashboard purpose: "consistency, monitor, plan, communicate" [12]

"There are three key questions: 1. Who is my audience? 2. What value will the dashboard add? 3. What type of dashboard am I creating?" [7]

Information Discrimination [7] –¿ only take important information –¿ "choosing the perfect metric" [7] page 10

1. Form: In what format is the dashboard delivered? 2. Structure: How is the dashboard laid out to help users understand the big picture? –¿ plus importance of structure*1 3. Design principles: What are the fundamental objectives that will guide your design decisions? 4. Functionality: –¿ see page 24 [7] + conclusion of [14]: "We recommend that dashboardscomewithsome level of flexibility, i.e. allowing users to switch between alternative presentation formats."

The dashboard landscape now encompasses everything from mouse-driven traditional desktop monitors to smaller-screened laptops with touch pad navigation. The real estate constraints of tablet computers and smartphones are offset with gestures for direct manipulation of data visuals.[11]

*1"Dashboard content must be organized in a way that reflects the nature of the information and that supports efficient and meaningful monitoring. Information cannot be placed just anywhere on the dashboard, nor can sections of the display be sized simply to fit the available space. Items that relate to one another should usually be positioned close to one another. Important items should often appear larger, thus more visually prominent, than less important items. Items that ought to be scanned in a particular order ought to be arranged in a manner that supports that sequence of visual attention. (Pervasive Hurdles to Effective Dashboard Design, Visual Business Intelligence Newsletter, January 2007) [7]

"Customizable Build in flexibility to allow the dashboard to become relevant for different users. The most common way to allow users to customize the dashboard is by defining the scope of the data using filters. There is more that can be done: Does the dashboard let the user save a view of the data that theyve configured? Does it offer easy ways to tag or highlight things that are important to them?" [7]

"The audience and the data should guide your design. It is important to know who will be the users of the dashboard you are designing and what their goals are, so you understand which category of dashboard you will be designing. After obtaining this information through user interviews or requirements, you" [11] –¿ i proxied

"Brushing and Linking Brushing is an interactive selection process that is often, but not always, com- bined with linking, a process for communicating the selected data to other views of the data set. There are many possibilities to visualize multi-dimensional data, each with their own strengths and weaknesses. The idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of individual techniques. Scatterplots of different projections, for example, may be combined by coloring and linking subsets of points in all projections. In a sim- ilar fashion, linking and brushing can be applied to visualizations generated by all visualization techniques described above. As a result, the brushed points are highlighted in all visualizations, making it possible to detect dependencies and correlations. Interactive changes made in one visualization are automatically re- flected in the other visualizations. Note that connecting multiple visualizations through interactive linking and brushing provides more information than con- sidering the component visualizations independently." [8]

categorizing dashboards: role, type of data, data domain, type of measures, span of data, update frequency, interactivity, mechanism of display, portal functionality [4, pp. 30-31]

"It is not necessary to fi gure out which type of dashboard you want to build before beginning a project. In reality, many dashboards don t fi t cleanly within the boundaries described in this chapter. Rather, the purpose of the framework is to help you understand the various purposes for which performance dashboards are built and the range of functionality that they can exhibit." [3, p. 121]

extensive section on metrics: [3]

2.6 Dashboard Architecture

" Thin Clients. The advent of the Web shifted computing architectures from fat clients (i.e., desktop machines that handled the graphical inter- face and logic) to thin clients in which most of the client and server processing occurred on back - end servers, not desktop machines. In a thin client architecture, a Web server renders the HTML, which is displayed on the desktop machine via a Web browser, while application and data pro- cessing occurred on one or more application servers. Thin client processing has its benefi ts. Since all processing occurs on the server, users don t need to purchase an expensive desktop machine to run the application or install any software on their computer. All administration is handled centrally, saving the company time and money. In addition, corporate fi rewalls generally don t block HTML from passing through, unlike Java applets or ActiveX controls, which are applications that can run independently on the client machine and thus pose a security threat. However, the downside of HTML - based thin clients is lack of perfor- mance and functionality. All user inputs are sent over the network to the server, which processes the request, renders the new screen, and pushes the resulting HTML code to the browser. This round - trip processing causes an unsettling delay, especially when users want to do something simple, such as change a background color or sort a table. Rich" [3, p. 252] –¿ i hab nen mitteldicken client

"Rich Internet Applications Given the limitations of thin HTML clients, software vendors have begun to thicken Web clients to take advantage of the processing power of desktop

computers and make Web - based applications more interactive and dynamic. Java Applets/Active X Controls. Java applets and ActiveX controls are mini - applications that run inside a Web browser and execute within a virtual machine, or sandbox. Actions execute as fast as compiled code, making them an easy way to re - create full - featured applications on the Web. However, as mentioned earlier, they raise security concerns, and many information technology (IT) administrators prevent users from downloading such controls through corporate fi rewalls, limiting their pervasiveness. DHTML and AJAX. A lighter - weight approach is to embed a scripting language inside HTML pages, such as JavaScript, that executes functions in the browser. Dynamic HTML (DHTML) uses scripting to animate a downloaded HTML page. For example, DHTML often is used to animate drop - down boxes, radio buttons, mouse - overs, and tickers as well as capture user inputs via forms. AJAX (asynchronous JavaScript and XML) takes this one step further and retrieves new content from the server in the background without interfering with the display and behavior of the page. Basically, AJAX enables users to add new data to the dashboard without having to reload the entire page. It can also be used to prefetch data, such as the next page of results. However, DHTML and AJAX have some signifi cant drawbacks. DHTML doesn t always work the same way with all browsers, creating a mainte- nance headache. And AJAX falters if users disable JavaScript in their browsers. Also, performance, reliability, and error handling can be prob- lematic with AJAX since it uses scripts instead of a programming language, and browsers don t make good use of memory. As a result, some veteran business intelligence (BI) developers claim AJAX isn t suitable for advanced BI applications. Multimedia Plug - ins. Another popular approach is to use multimedia development platforms, such as Adobe Flash, Java applets, Microsoft Silverlight, and Mozilla Scalar Vector Graphics (SVG), which add animation and movies to Web pages. To use these multimedia applications, users download a Web browser plug - in (e.g., Adobe Flash Player to use Adobe Flash), which remains permanently installed on their machine and serves as a runtime engine and sandbox for the applications. These multimedia platforms also offer a programming model and scripting language that enables developers to create interactive Web - based applications, much as AJAX does. Multimedia Plug - ins. Another popular approach is to use multimedia development platforms, such as Adobe Flash, Java applets, Microsoft Silverlight, and Mozilla Scalar Vector Graphics (SVG), which add animation and movies to Web pages. To use these multimedia applications, users download a Web browser plug - in (e.g., Adobe Flash Player to use Adobe Flash), which remains permanently installed on their machine and serves as a runtime engine and sandbox for the applications. These multimedia platforms also offer a programming model and scripting language that enables developers to create interactive Web - based applications, much as AJAX does. Currently, almost all browsers support the Adobe Flash plug - in, while about three - quarters support Java applets and two - thirds support Microsoft Silverlight. The next major version of HTML (HTML 5) should provide native support for SVG, reducing the need for users to download external multi- media plug - ins. Google Chrome currently supports SVG, and Microsoft has announced that Internet Explorer 9 will support HTML 5 with native SVG support. So, in the future, most Web browsers will provide native, multimedia capabilities. Compared to Java scripting, the so - called RIA platforms provide stunning graphics and animation for displaying quantitative information, which make user interfaces very appealing to business users. For instance, 1 - 800 CONTACTS, profi led in Chapter 7 , now uses Microsoft Silverlight to build its dashboards. (See Exhibits 7.2 and 7.3 .) In addition, since these applica- tions run in their own container (i.e., the plug - in), they don t have to be adapted to work with different browsers, making them easier to maintain. Most important, they load both visualizations and data simultaneously in a

single fi le rather than dishing up dozens or hundreds of pages. Although this makes the initial load slower than a comparable DHTML or AJAX application, performance thereafter is exceptionally fast since data required to display all components on a page resides locally. It also means that users can run the applications when disconnected from the Internet, providing greater fl exibility. Downsides. Like any technology, RIAs have some downsides. For instance, Flash can t leverage keyboard navigation options and its animations are inaccessible to sight - or mobility - impaired customers. In addition, Flash plug - ins aren t interoperable. Users running an older version of a plug - in won t be able to view any part of the application until they upgrade the plug - in. Finally, Flash dashboard products typically offer a limited number of visualizations and can t incorporate new ones easily. However, the biggest limitation of RIAs is scalability. When the size of the Flash fi le becomes too large, it takes too long to load within a Web browser, which frustrates users who don t like to wait. Ideally, Flash fi les should be less than 1 megabyte (MB), but fi les that are more than 10MB are tolerable. Obviously, these fi le sizes limit the size of the applications that developers can build with the technology. Multitier Flash Applications. As RIA applications rise in popularity in the business world, developers are devising new ways to circumvent the load bottleneck. For example, some BI vendors have extended Flash from a static, browser - based, desktop application to a dynamic, multitier application that fetches data on demand as users drill into a chart or switch page views. This dynamic data retrieval reduces the amount of data that the application needs to download at start - up, improving load performance signifi cantly. —¿ Rules of Thumb Mixing Elements. Dashboard architects need to consider how to mix the various Web technologies just described to deliver an attractive, interactive, and high - performance user interface." [3, pp. 252]

"Crossfilter, which is a fast browser side in-memory filtering mechanism across multiple dimensions and measures. One of the major limitations of using Crossfilter is that it keeps data in-memory on client-side in a browser. " [13] –¿ smartfilter could be better as client is restricted via browser in memory

Alternative: crossfilter auf server-side [5]

2.7 Dashboard Design

The use of Gestalt principles to improve perception by Moore and Fitz (1993) –¿ "psychology was born in reaction to atomism at the end of the 19th century with the view of things as more than the sum of their parts. The Gestalt psychologists were intrigued by the way our minds perceive wholes out of incomplete elements (Behrens, 1984; Mullet and Sano, 1995). Among the Gestalt principles that dashboards use are proximity, similarity, continuity, figure-ground, symmetry, and the closure of objects (Moore and Fitz, 1993)." [14]

FIND SOURCES THAT SAY: Difficult to compare in brushing dashboards –¿ high-end dashboards can be modified by user to built in comparisons, but that requires user knowledge of dimensions and data structure, column names –¿ we propose smth simpler. also a way to create a snapshot

types of dashboards: [4] and [11] –¿ i'm analytical

CONTRA ALL ON ONE PAGE: "The conventional view has been that dashboards need to be constrained to a single page; we believe dashboards can come in many forms. A short e-mail can serve as a dashboard if it works for the recipients. Likewise, a wall-mounted 55 plasma TV showing an animated presentation has the potential to be an effective dashboard." [7]

PRO ONE PAGE: "A dashboard is meant to be viewed at-a-glance, so once the visuals have been selected, they must be arranged in a display that can be viewed all at once, such as a computer screen, without having to scroll or navigate to multiple pages. Information is effectively integrated, risk is quickly noticed, and decisions are most easily obtained when information is displayed using visuals that are arranged together so they can be seen simultaneously on one screen. This allows for processing the information with minimal effort [4]" [11]

3.1. Exceeding the Boundaries of a Single Screen [4, pp. 39]

Display Information on a Single Screen [3, p. 230]

common mistakes: "Displaying Excessive Detail or Precision, Choosing Inappropriate Display Media, Encoding Quantitative Data Inaccurately, Cluttering the Display with Useless Decoration, Misusing or Overusing Color,

"Webpage and dashboard designers who rely on gestalt principles focus on the over- all look, rather than on the details. The fol- lowing are important gestalt and design principles: n Similarity. Similar things (e.g., color, size, shape) are perceived to be more relat- ed than dissimilar things. n Proximity. Things that are close to each other are perceived to be more related than things spaced farther apart. n Closure. When looking at an arrangement of individual elements, people tend to see a recognizable pattern (i.e., incomplete shapes are perceived in a complete manner). n Continuity. Objects arranged on a line or on a curve are perceived to be related. n Past experience. People tend to group together elements in a way that reflects past experience, either as individuals or as a group. n Focal point. A point of interest or emphasis will capture and will tend to hold the viewers attention. The space between design elements is called white space, or negative space. There might be a temptation to completely fill the dashboard screen with data and charts, which is the positive space, but this usu- ally overloads the user. White space is used to organize elements and improve the visual experience. It can also help achieve professionalism and sophistication in a dashboard design; this can be thought of as providing an attractive frame for the dashboard design elements that will draw the users attention to the content. Dashboard" [2]

"One technique is positioning. Because the upper left and center areas on the the dashboard are subject to the greatest visu- al focus by users, this is the place for the most important performance measurement information. The lower right area generally attracts the least attention, but visual tech- niques, such as color intensity and line width, can be used to counteract this." [2]

"In selecting charts, the third dimension of depth on charts should be avoided. Gridlines in bar graphs are a distraction, unless the user is expected to read values directly from a graph. Variations in chart color that do not encode a mean- ing can be another source of distraction or distortion. Dashboard" [2] ... bubble chart reduced from 3 dimension (avg

delay, avg distance, and #flights to removed avg distance)

"It is also useful to understand how your audience is accustomed to viewing data and with what visuals they are already familiar in order to choose visuals on your dashboard that will be easy for them to interpret." [11]

"Information presentation is a balancing actHow do you convey a lot of information without making it feel overwhelming? How do you capture attention without distracting your audience? How do you make information feel simple yet profound?" [7]

people scan page from top left to bottom right [7] –¿ page 30 also [4, 97]

GRIDS and whitespace [7] –¿ page 31

"Visuals that are related to each other should be close to each other, with white space around the group. The users should be able to easily find important information. Since people read left to right, put the most important information in the top-left corner, or the information that a user will want to look at first. If an object is in the center of the display, it will be noticed first, so if anything is placed directly in the center of the display, it should be important [5]. Fig." [11]

1. Reduce chart-junk and increase data-to-ink ratio + LABELS + no smoothing on lines [7] also [4, pp. 84]

High data to ink ratio to maximize attention paid to important information –¿ Tufte (2006)

list features for advanced dashboards: [7] page 48

Less efficient data visualizations include pie charts, speedometers, and dials. [11]

"It is best to focus on the data and avoid any visual distractions. Background images should especially be avoided, since reading text laid over them is difficult. Motion attracts attention, so avoid using extraneous or looping animations, such as tickers. Using too many colors, or colors that are too bright, is also distracting [6]. Since some users may be color blind, color alone should not be used to convey meaning. Instead, combine color with intensity or border thickness. Use labels to show values. If colors are used to show comparisons in data, their shades should vary. Printing" [11]

"[11]The ability to expand or collapse visuals can be useful when many visuals crowd a small display area, al- lowing users to focus on the information most relevant to them. Interactions like data brushing, or interactive highlighting, can also help a user to focus their attention on important data. Data brushing is a technique where, as the user changes data selection in one view, corresponding linked data in one or more other views is highlighted [7]. For example, they can click on a row in a table or move a slider along a line chart (as shown in Figure 5) to reveal a bar graph showing detailed information for the date or category selected. In this way, the user can choose what is displayed on their dashboard, so that space is utilized in a way that is most beneficial to them."

¿—-¿ comparison: "According to cognitive fit theory, tasks that require comparisons (spatial tasks) are better supported with graphs. Since a dashboards' primary purpose is to

display and compare KPI's, the use of graphs (including dials) by default seems to be warranted. However, users should have the option to switch to an alternative display format, i.e. to tabular format, in case this is preferred. For different user backgrounds and personality types (e.g. an accountant with high analytical skills), a similar strategy as above could be implemented." [14]

GESTALT USED BY FEW FOR GRAPHS, not dashboard as a whole "Back in 1912, the Gestalt School of Psychology began its fruitful efforts to understand how we perceive pattern, form, and organization in what we see. The German term "gestalt" simply means "pattern." These researchers recognized that we organize what we see in particular ways in an effort to make sense of it. Their work resulted in a collection of Gestalt principles of perception that reveal those visual characteristics that incline us to group objects together. These principles still stand today as accurate and useful descriptions of visual perception, and they offer several useful insights that we can apply directly in our dashboard designs to intentionally tie data together, separate data, or make some data stand out as distinct from the rest. We'll examine the following six principles: ? Proximity ? Closure ? Similarity ? Continuity ? Enclosure ? Connection" [4, p. 74]

organizing information on a dashbaord: "? Organize groups according to business functions, entities, and use. ? Colocate items that belong to the same group. ? Delineate groups using the least visible means. ? Support meaningful comparisons. ? Discourage meaningless comparisons." [4, p. 139] –¿ redo main page, and maybe remove comparison buttons on bad charts

more graphical and more tabular approaches (e.g. [3, p. 238]) –¿ down to user preference. I will focus on graphical, as I want to show brushing

5.4. Research on New Dashboard Design Features –¿ 'Prototypes and proof of concepts are needed to see how these new features can add value to the users and whether these concepts are feasible.' [14]

Chapter 3

Requirements and Design

verzicht auf uml weil JS keine classes wie java

+download data

FORM: web app; pre-decided, yet not pad as per [7] page 15

structure: "A good dashboard structure requires a deep understanding of how the system you are measuring works. There are many ways to break something down into manageable parts. For example, the performance of a (American) football game can be deconstructed in many ways: 1) by offense,defense, and special teams; 2) by down and distance; 3) by time period; 4) by drive; 5) by running vs. passing plays. A dashboard built around each of these organizing principles would tell a different story."[7] –¿ I broke it down into airport, airline, ...

strategic dashboard –¿ "The most widely recognized use of digital dashboards is that of the executive dashboard. Its purpose is to communicate to management the organizations perform- ance relative to corporate objectives. Its nature invites comparative data, contrasting current with past performance or current to target levels." [11] –¿ The strategic dashboard allows for a quick overview of an organizations health, so to speak; assisting with executive decisions such as the formation of long-term goals. The strategic dashboard, therefore, doesnt require real-time data: what is going on right now is not important, what is pressing is what has been going on. [3] –¿ requirement of real time data not necessary, but possible

ICH MACH ANALYSTICAL: "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." [11] ¡—¿ SEE TABLE 1!!!! –¿ my dashboard is not operational bc uts bot a source to look up whether ones flight is on time, it's for timeliness performance and traffic over time

berleitung: The starting point for a dashboard project is to clearly define the projects goal. Because a dashboards focus is on a goal or task, the design must be goal- and user-oriented. [2] –¿ use of my dashboard: traffic volume and delay dashboard

"Determining the intended recipient, we can surmise the level of cur- rency required and the types of measures or categories relevant for the role." [11]

phasen des dashboard baus? : "Define the dashboard objective. n Define the dashboard metrics. n Seek user input. n Build the initial dashboard and test. n Publish the dashboard and monitor its use" [2]

3.1 Requirements & Users

data filtered down b/c "choosing the perfect metric" [7] page 10 so that my dashboard only provides actionable, common interpretable, ... info

type of dashboard: "Scope Business role Broad: Displaying information about the entire organization Specific: Focusing on a specific function, process, product, etc. 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: Showing performance at a single point in time Real-time: Monitoring activity as it happens Predictive: Using past performance to predict future performance Customization 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 Point of view" [7] page 7

Specification list (must-have, should-have, could-have, would-like-to-have), identification of potential end-users (including user needs matrix/use case diagram). Towards an open-source framework.

- Airport spezifisches dashboard - User sind dann firmen/privatkunden, welche sich ein bild ber die situation eines airports schaffen wollen - Karte mit strecken -¿ select strecke –¿ mehr infos - Statische KPIs + Interactive charts (+ komplet dashboard verlinkung (qlik))

user problem: Some users might have a lot of experience with this technology, whereas others might be relative novices. [2]

should have DB access: Dashboards can use static or real-time data. Thus, the dashboard should be able to access a companys database in a time- ly fashion. The static data could come from periodic reports (daily, weekly, monthly) or Microsoft Excel spreadsheet models. [2]

mobile is problematic: "In todays environment, developing a mobile dashboard for a smart-phone or tablet appears to be very attractive; how- ever, the technology for nonstatic dash-boards is expensive, so users should con- sider some important cost-benefit issues. Mobile devices present a whole new set of design issues and opportunities. In most cases, it is not a good practice to simply export a dashboard from a desktop plat- form to a mobile device, and it is impor- tant to know what the target device will be when creating a dashboard." [2]

3.2 Application Architecture

System Architecture (n-tier architecture using existing frameworks/components (Web-server: Node.js and Express.js, DB: MySQL and MongoDB, ...)) with Sequence/UML Diagrams.

+ js libraries diagramm

3.3 Site Layout & URL Mapping

Structure of page navigation of the complete web-service.

3.4 Front-end Design

Wireframes and design mock-ups.

problem for user: what interaction with what graph –¿ different cursors, normal mouse = nothing, hand = select, crosshair = brush

show grid structure of my app (columns faded over page) + flow from row to row, and increasing lvl of detail as you go down (Gradual reveal) [7]

3.5 Development Methodology

Scrum methodology (Development period of 7/14 days) with the following artefacts: Product Backlog, Sprint Backlog, Burndown Chart.

Chapter 4

Implementation

ATM when caching, i put the data into the queue labeled with info, some info is redundant, and especially, is queue the data separately, but could combine it directly, instead of combining it only when render — NO it is more efficient/responsive to just grab it and only rework and combine it when rendering, imagine someone presses caching often and everytime it has to merge (not efficient) only merge and work when actual render happens, could be, that the cache is discarded, then all the merging was for nothing, if we merge within the queue

(project meant to show marketable dashboard via browser...) every Client might habe different resolution... Literatur geht von einer spezifizierten Auflsung aus...

Separate cc.js ONLY vs cc.js in example (e.g. cc.js only does not use sliding via popupoverlay; example does)

beim legende erstellen fr cc.js -¿ problem wie man die benennt, da ja jeder graph nen anderen filter haben kann und dann msste man alle (oder zumindest die, die sich verndert haben) hin drucken

B/C all on one page –¿ small –¿ new ways to compare and look into detail of your data –¿ our solution: crosscompare as overlay/etc

in my example, the selection option for airline and airport are hard coded (as i don't want to have a secnod lookup for airline/airport codes and their names), but it could easily be automated with: code:

```
$.each(group.top(Infinity), function (i, item) {
  $('#airlineSelect').append($('<option>', {
    value: item.key,
    text : item.key
  }));
});
```

removed ordering function even though was implemented: NO I REWORKED IT, to work for the first cache in queue, then all others are ordered according to the first's order. this problem occurs as c3.load relies on the order of provided data, rather than identifiers such

as the key —– OR the sort function is wrong, it switches values for given keys? –no, but c3.load reshuffles if data comes in in new order WITHOUT reshuffling existing bars

```
function sort(key, asc) {
  cache = cache.sort(function(a, b) {
    if (asc) return (a[key] > b[key]) ? 1 : ((a[key] < b[key]) ? -1 : 0);
    else return (b[key] > a[key]) ? 1 : ((b[key] < a[key]) ? -1 : 0);
  });
}

if (first)
  if (order != 'default') {
    if (order == 'asc') sort([item.id], true);
    else sort([item.id], false);
};
```

4.1 JavaScript Coding

HAVEN'T done JS before, so had to learn..., also learn JADE

don't forget to minify!

DC js row chart does not render correctly when all rows have negative values ¿ see weekday delay chart

PROBLEM: read out what filters currently apply

4.2 Database Request Handling

4.3 Page and Data Representation

dc.js rowchart does not render correctly if all values below 0 (weekday chart)

4.4 Open-source Adjustments

Chapter 5

Evaluation

Using the use case (airport dashboard) for technical and user evaluation.

5.1 Application Testing

Technical evaluation: test cases, SQL inject testing, code review.

5.2 User Evaluation

User experiments with qualitative data analysis (e.g. interviews).

compare drag and drop with overlay with appearing box (atm)

5.2.1 Research Methodology

5.2.2 Evaluation Results

5.2.3 Discussion

WEAKNESS of my dashboard –¿ my data has no targets/benchmarks (when is smth good, bad, ...) b/c of general data

Chapter 6

Conclusion

6.1 Project Outcome

6.2 Further Research and Work

dc.js needs composite charts add labels to compare graph about what filters where used multiple data sources (compare to actual data discovery tools) ATM: all data is actually sent to user, confidentiality?

.elasticY(true) NOT WORKING on scatter plot for dc js crosscompare: support for stacked dc.js graphs

c3 bar graphs haben probleme mit einer variablen bar-width, weil nachladen das ganze kompliziert macht. –¿ todo in future, options.bar = width: ratio: 0.2 ; responsive machen

BIG problems with scatter (scatter is new in dc js –¿ works differently then all the others –¿ special handling required) AND naming the different data snapshots (read filters from dc is ok, but crossfilter doesn't allow that, as I use a mixture of dc graphs and crossfilter direct (airport, airline), difficult! only dc graphs allow the extraction of active filters...)

Bibliography

[1] 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.

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

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

[4] Stephen Few. Information Dashboard Design the Effective Visual Communication of Data, 2006.

[5] Ganesh Iyer. Big Data visualizations using crossfilter and dc.js, 2014.

[6] CGI Group Inc. Next Generation Business Intelligence: Seven steps to improved business intelligence through data discovery, 2011.

[7] Inc. Juice. A Guide to Creating Dashboards People Love to Use. *Design*, (November):1–11, 2009.

[8] Daniel Keim and Matthew Ward. Visual Data Mining Techniques. *Techniques*, 2002.

[9] Allen R. Martin and Matthew O. Ward. High Dimensional Brushing for Interactive Exploration of Multivariate Data. page 271, October 1995.

[10] Wan Maseri, Binti Wan, Abdullah Embong, Jasni Mohd Zain, and Software Engineering. 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.

[11] 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.

[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.

[13] Darshit Shah. SmartFilter Not in-memory server-side Crossfilter for BigData, 2015.

[14] 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.

Appendix A

Appendix

A.1 Code listing: crosscompare.js

```
1
2  /*!
3   *  Font Awesome 4.3.0 by @davegandy - http://fontawesome.io - @fontawesome
4   *  License - http://fontawesome.io/license (Font: SIL OFL 1.1, CSS: MIT
        License)
5   */
6
7
8  var crosscompare = {
9    // Setting variables (with default values)
10   height: 200,
11   width: 'auto',
12   padding: { top: 20, right: 5, bottom: 10, left: 25 },
13   anchor: '#crosscompare',
14   dateFormat: '%d/%m/%Y',
15   overwrite: false,
16   xGrid: false,
17   yGrid: false,
18   flash: true,
19   legends: [],
20
21   // Operational variables
22   chart: {},
23   charts: {},
24   queue: []
25 };
26
27 crosscompare.setHeight = function(height) {
28   if (height >= 0)
29     this.height = height;
30   return this;
31 };
32
33 crosscompare.setWidth = function(width) {
34   if (width >= 0 || width == 'auto')
35     this.width = width;
36   return this;
37 };
```

```
38
39 crosscompare.setPadding = function(top, right, bottom, left) {
40   if (typeof top === 'number' && typeof right === 'number' &&
41     typeof bottom === 'number' && typeof left === 'number')
42     this.padding = { top: top, right: right, bottom: bottom, left: left};
43   return this;
44 }
45
46 crosscompare.setAnchor = function(anchor) {
47   if (typeof anchor !== 'undefined' && anchor.length > 0)
48     this.anchor = anchor;
49   return this;
50 };
51
52 crosscompare.setDateFormat = function(format) {
53   if (typeof format !== 'undefined' && format.length > 0)
54     this.dateFormat = format;
55   return this;
56 };
57
58 crosscompare.setOverwrite = function(overwrite) {
59   if (typeof overwrite === 'boolean')
60     this.overwrite = overwrite;
61   return this;
62 }
63
64 crosscompare.setGrid = function(xGrid, yGrid) {
65   if (typeof xGrid === 'boolean' && typeof yGrid === 'boolean') {
66     this.xGrid = xGrid;
67     this.yGrid = yGrid;
68   }
69   return this;
70 }
71
72 crosscompare.setFlash = function(active) {
73   if (typeof active === 'boolean')
74     this.flash = active;
75   return this;
76 }
77
78 crosscompare.addLegend = function(legend, title) {
79   if (typeof legend !== 'undefined')
80     this.legends.push({ 'legend': legend, 'title': (title ? title : '') });
81   return this;
82 }
83
84 crosscompare.add = function(chart, options) {
85
86   // options: type, value, anchor, order, yLabel, xLabel
87   var allOptions = {
88     'chart': chart,
89     'type': 'line',
90     'ratio': 0.5,
91     'value': 'default',
92     'anchor': chart.anchor() + '-cross',
93     'order': 'default',
94     'yLabel': '',
95     'xLabel': ''
96   };
97
```

```
98    if(typeof options !== 'undefined')
99      $.each(options, function(key, value) {
100       allOptions[key] = value;
101     });
102
103   var anchor = allOptions.anchor;
104
105   this.charts[anchor] = allOptions;
106
107   // render crosscompare chart if anchor clicked
108   $(anchor).on('click', function() {
109     // flashing animation (10 ms -> 700 ms)
110     if (crosscompare.flash)
111       $(chart.anchor()).fadeTo(10, 0.3).fadeTo(700, 1.0);
112
113     var text = 'You have cached ';
114     if ($.isEmptyObject(crosscompare.chart)) // nothing so far
115       text += '1 state.';
116     else if (crosscompare.chart.source == chart) // added same chart
117       text += (crosscompare.queue.length + 1) + ' states.';
118     else { // added smth else
119       crosscompare.clear();
120       text = 'Overwritten with new state.'
121     }
122
123     // fill active as to signal that a crosscompare has been created
124     crosscompare.chart.source = chart;
125
126     $(crosscompare.anchor + '-info').text(text);
127
128     crosscompare.cache(anchor);
129
130   });
131
132   return this;
133 };
134
135 crosscompare.cache = function(anchor) {
136
137   var legend = '';
138
139   if (this.legends.length == 0) // no legend specified
140     legend += (this.queue.length + 1);
141   else {
142     $.each(this.legends, function(i, item) {
143
144       if (typeof item.legend === 'string') { // string --> legend via html
              selection
145         if (item.title != '') legend += item.title + ':';
146         legend += $(item.legend).val() + ' ';
147       } else { // chart --> legend via filter
148
149         var filters = item.legend.filters(),
150             format = d3.time.format(crosscompare.dateFormat)
151
152         if (typeof filters !== 'undefined' && filters.length > 0) {
153
154           if (item.title != '')
155             legend += item.title + ':';
156
```

```
157           if (filters[0].constructor === Array) {
158
159             if (filters[0][0] instanceof Date)
160               legend += format(filters[0][0]) + ' - ' + format(filters
                    [0][1]);
161             else
162               legend += filters[0][0] + ' - ' + filters[0][1];
163           } else
164             legend += filters + ' ';
165         }
166       }
167     });
168
169     // Prevent empty legend in case legends have been provided but no
          filters
170     if (legend == '')
171       legend = 'ALL ';
172
173     if (!this.overwrite)
174       $.each(crosscompare.queue, function(key, value) {
175         if (value.id == legend)
176           legend += '(' + (crosscompare.queue.length + 1) + ')';
177       });
178   }
179
180   var chart = this.charts[anchor].chart;
181   var value = this.charts[anchor].value;
182
183   // cache the charts underlying data (deep copy)
184   var cache = $.extend(true, [], chart.group().all());
185
186   // retrieve data filters and filter cache
187   var filters = chart.filters();
188
189   // have to go reverse, as deleting elements from array prob in JS
190   var i = cache.length;
191   while (i--) {
192
193     // set chosen value if value group
194     if (value != 'default')
195       cache[i].value = cache[i].value[value];
196
197     if (typeof filters !== 'undefined' && filters.length > 0) {
198
199       if (filters[0].constructor === Array) { // number range as filter
200         if (cache[i].key < filters[0][0] || cache[i].key > filters[0][1]) {
201           cache[i][legend] = null; // cannot delete, else c3 hover is broken
202           delete cache[i].value;
203         }
204       } else { // filters are actual elements
205         if (filters.indexOf(cache[i].key) == -1) {
206           cache.splice(i, 1); // delete instead of nulling elements, so it
                doesn't show up --> categ
207           continue; // skip rest of iteration (below) as entry is gone
208         }
209       }
210     }
211
212     // rename as per cat, so c3 can stack values
213     cache[i][legend] = cache[i].value;
```

```
214     delete cache[i].value;
215   };
216
217   this.queue.push({ 'anchor': anchor, 'id': legend, 'data': cache});
218 };
219
220 crosscompare.reset = function() {
221   var chart = crosscompare.chart.rendered;
222   if (typeof chart !== 'undefined')
223     chart.destroy();
224
225   crosscompare.clear();
226 };
227
228 crosscompare.clear = function () {
229   $(crosscompare.anchor + '-info').text('Cache cleared.');
230   this.chart = {};
231   this.queue = [];
232 }
233
234 crosscompare.render = function() {
235
236   // http://stackoverflow.com/questions/881510/sorting-json-by-values
237   function sort(array, key, asc) {
238     array = array.sort(function(a, b) {
239       if (asc) return (a[key] > b[key]) ? 1 : ((a[key] < b[key]) ? -1 : 0);
240       else return (b[key] > a[key]) ? 1 : ((b[key] < a[key]) ? -1 : 0);
241     });
242   }
243
244   if (this.queue.length > 0) { // if queued data exists
245
246     // retrieve chart type
247     var anchor = this.queue[0].anchor,
248         type = this.charts[anchor].type,
249         order = this.charts[anchor].order,
250         yLabel = this.charts[anchor].yLabel,
251         xLabel = this.charts[anchor].xLabel;
252
253     var globalMin, globalMax;
254
255     var orderBy = this.queue[0].id, // always sort by first dimension
256         n = this.queue[0].data[0].key,
257         isDate = n instanceof Date,
258         isNumber = !isNaN(parseFloat(n)) && isFinite(n);
259
260     var testo = [], typeso = [];
261
262     $.each(crosscompare.queue, function(index, item) {
263
264       var cache = item.data;
265
266       typeso.push(item.id);
267
268       var min, max;
269       if (isDate || isNumber) {
270         // find min
271         for (i = 0; i < cache.length; i++) {
272           if (cache[i][item.id] != null) {
273             min = cache[i].key;
```

```
274             break;
275           }
276         };
277
278         // find max
279         for (i = cache.length -1; i >= 0; i--) {
280           if (cache[i][item.id] != null) {
281             max = cache[i].key;
282             break;
283           }
284         };
285
286         // if not in first run
287         if (index != 0) {
288           min = Math.min(min, globalMin);
289           max = Math.max(max, globalMax);
290         }
291
292         globalMin = min;
293         globalMax = max;
294       }
295
296       // combine all caches
297       $.extend(true, testo, cache);
298     });
299
300     if (order != 'default') {
301       if (order == 'asc')
302         sort(testo, orderBy, true);
303       else
304         sort(testo, orderBy, false);
305     }
306
307     var options = {
308       bindto: crosscompare.anchor,
309       size: { height: crosscompare.height },
310       padding: crosscompare.padding,
311       data: { json: testo, keys: { x: 'key', value: typeso } }, // CHANGE
              ALL TO TYPESO
312       zoom: { enabled: true },
313       color: { pattern: d3.scale.category10().range() }
314     };
315
316     if (crosscompare.legends.length == 0)
317       options.legend = { show: false };
318
319     if (crosscompare.width != 'auto')
320       options.size.width = crosscompare.width;
321
322     if (isDate || isNumber) {
323
324       options.axis = { x: {
325         tick: { fit: false },
326         min: Number(globalMin),
327         max: Number(globalMax)
328       } };
329
330       if (isDate) { // this style otherwise overwriting upper
331         options.axis.x.type = 'timeseries';
332         options.axis.x.tick.format = crosscompare.dateFormat;
```

```
333       }
334     } else { // Category -> overwrites previous axis settings above
335       options.axis = { x: { type: 'category' } };
336     }
337
338     if (type != 'line')
339       options.data.type = type;
340
341     if (type == 'bar')
342       options.bar = { width: { ratio: this.charts[anchor].ratio } };
343
344     options.grid = {
345       x: { show: crosscompare.xGrid },
346       y: { show: crosscompare.yGrid }
347     }
348
349     if (yLabel != '')
350       options.axis.y = { label: yLabel };
351
352     if (xLabel != '')
353       options.axis.x.label = xLabel; // other format than y axis, otherwise
              overwrite of x
354
355     // make available later (see below)
356     crosscompare.chart.rendered = c3.generate(options);
357
358   } else $(crosscompare.anchor + '-info').text('Nothing cached.');
359 };
360
361 // Node.js export
362 if (typeof exports !== 'undefined'){ module.exports = crosscompare };
```

A.2 Code listing: example.js

```
1  // INFO
2  /*!
3   *  Font Awesome 4.3.0 by @davegandy - http://fontawesome.io - @fontawesome
4   *  License - http://fontawesome.io/license (Font: SIL OFL 1.1, CSS: MIT
       License)
5   */
6
7
8
9  // DC VERSION 2.1.0-dev
10 //------------------------------General / DC
       ----------------------------------------
11 // Define charts
12 var totalAverageDelay = dc.numberDisplay('#delay'),
13     flightsTable = dc.dataTable('#flightsTable'),
14     flightDelay = dc.scatterPlot('#flightDelay'),
15     movementsChart = dc.lineChart('#movementsChart'),
16     movementsTimeChart = dc.barChart('#movementsTimeChart'),
17     airportsChart = dc.rowChart('#airportsChart'),
18     weekdayChart = dc.rowChart('#weekdayChart'),
19     todChart = dc.barChart('#todChart'),
20     delayChart = dc.barChart('#delayChart'),
21     distanceChart = dc.barChart('#distanceChart');
22
23 // http://colorbrewer2.org/
```

```
var colorRange = ['rgb(165,0,38)','rgb(215,48,39)','rgb(244,109,67)','rgb(253,174,97)','rgb(254,224,139)','rgb(217,239,139)','rgb(166,217,106)','rgb(102,189,99)','rgb(26,152,80)','rgb(0,104,55)'];

// Date and number formats
var dateInFormat = d3.time.format('%d-%m-%Y %H:%M'),
    dateOutFormat = d3.time.format('%d/%m/%Y %H:%M'),
    numberFormat = d3.format('0,000'),
    precisionFormat = d3.format('.2f');

//--------------------------------Crossfilter----------------------------------------

// Load data from csv file
//d3.csv('/data/flightsDec08.csv', function(data) {
d3.csv('/data/example.csv', function(data) {

  // Parse dates and times from .csv
  data.forEach(function (d) {
    d.DateTime = dateInFormat.parse(d.DateTime);
  });

  // Set up crossfilter
  var flights = crossfilter(data),
      all = flights.groupAll();

  // Define dimensions
  var airport = flights.dimension(function(d) { return d.Airport; }),
      date = flights.dimension(function(d) { return d.DateTime; }),
      scndAirport = flights.dimension(function(d) { return d.Airport2; }),
      scatter = flights.dimension(function(d) { return [d.DateTime.getHours() + Math.floor(d.DateTime.getMinutes()/5)*5/60, (Math.max(-60, Math.min(179, d.Delay)))]; }),
      weekday = flights.dimension(function(d) {
        // Make sunday last (let week begin with Monday)
        var adjustedNum = (d.DateTime.getDay() == 0) ? 7 : d.DateTime.getDay() ;
        return '' + adjustedNum + ' ' + d3.time.format('%A')(d.DateTime);
      }),
      hour = flights.dimension(function(d) { return d.DateTime.getHours(); }),
      delay = flights.dimension(function(d) { return Math.max(-60, Math.min(179, d.Delay)); }),
      distance = flights.dimension(function(d) { return Math.min(d.Distance, 2499); }),
      airline = flights.dimension(function(d) { return d.Airline; });

  // Define groups (reduce to counts)
  byDate = date.group(d3.time.day),
  byDateHour = date.group(d3.time.hour),
  byHour = hour.group(),
  byScndAirport = scndAirport.group(),
  byScatter = scatter.group(),
  byDelay = delay.group(function(d) { return Math.floor(d / 5) * 5; }),
  byDistance = distance.group(function(d) { return Math.floor(d / 100) * 100; }),
  byWeekday = weekday.group().reduce(
    function(p, v) { ++p.n; p.sumDelay += Number(v.Delay);
      p.avgDelay = p.n ? p.sumDelay / p.n : 0; return p; },
    function(p, v) { --p.n; p.sumDelay -= Number(v.Delay);
```

```
74        p.avgDelay = p.n ? p.sumDelay / p.n : 0; return p; },
75      function() { return { n: 0, sumDelay: 0, avgDelay: 0 }; }
76    ),
77    averageDelay = flights.groupAll().reduce(
78      function(p, v) { ++p.n; p.sumDelay += Number(v.Delay); return p; },
79      function(p, v) { --p.n; p.sumDelay -= Number(v.Delay); return p; },
80      function() { return { n: 0, sumDelay: 0 }; }
81    );
82
83    //---------------------------------DC
          ----------------------------------------------------
84
85    // second row height
86    var heightTall = 300,
87        heightShort = 107;
88
89    // Date range
90    var minDate = d3.time.day(date.bottom(1)[0].DateTime),
91        lastDay = d3.time.day(date.top(1)[0].DateTime),
92        maxDate = lastDay.setDate(lastDay.getDate() + 1);
93
94    // Non-graph data representation
95    dc.dataCount('#flights')
96    .dimension(flights)
97    .group(all)
98    .html({ some:'Total Flights: <strong>%filter-count</strong><small>/%total-
          count</small>',
99      all:'Total Flights: <strong>%filter-count</strong><small> (all)</small>'
            });
100
101   dc.dataCount('#resetAll')
102   .dimension(flights)
103   .group(all)
104   .html({ some: '<a class=\'btn btn-default btn-block\'><i class=\'fa fa-
          chain-broken\'></i> Reset All</a>',
105     all:'<a class=\'btn btn-block btn-default disabled\'><i class=\'fa fa-
            chain-broken\'></i> Reset All</a>' });
106
107   totalAverageDelay
108   .group(averageDelay)
109   .formatNumber(precisionFormat)
110   .valueAccessor(function(d) { return d.n ? d.sumDelay / d.n : 0; });
111
112   // Define charts properties
113   movementsChart
114   .clipPadding(10)
115   .renderArea(true)
116   .height(250)
117   .margins({top: 5, right: 30, bottom: 20, left: 25})
118   .dimension(date)
119   .group(byDateHour)
120   .mouseZoomable(true)
121   .elasticY(true)
122   .x(d3.time.scale().domain([minDate, maxDate]))
123   .renderHorizontalGridLines(true)
124   .rangeChart(movementsTimeChart)
125   .brushOn(false)
126   .xAxis().ticks(4);
127   movementsChart.yAxis().ticks(6);
128
```

```
129   movementsTimeChart
130   .height(36)
131   .margins({top: 0, right: 30, bottom: 17, left: 25})
132   .dimension(date)
133   .group(byDate)
134   .elasticY(true)
135   .x(d3.time.scale().domain([minDate, maxDate]))
136   .xUnits(d3.time.days)
137   .round(d3.time.day.round)
138   .xAxis().ticks(d3.time.days);
139
140   flightDelay
141   .height(291)
142   .margins({top: 5, right: 30, bottom: 20, left: 25})
143   .clipPadding(10)
144   .dimension(scatter)
145   .group(byScatter)
146   .symbolSize(4)
147   .y(d3.scale.linear().domain([-60, 185]))
148   .x(d3.scale.linear().domain([0, 24]))
149   .renderHorizontalGridLines(true);
150   flightDelay.yAxis().ticks(6);
151
152   airportsChart
153   .height(heightTall)
154   .margins({top: 0, right: 25, bottom: 17, left: 5})
155   .dimension(scndAirport)
156   .group(byScndAirport)
157   .rowsCap(9)
158   .elasticX(true)
159   .ordering(function(d) { return -d.value; })
160   .xAxis().ticks(3);
161
162   weekdayChart
163   .height(heightTall)
164   .margins({top: 0, right: 25, bottom: 17, left: 5})
165   .dimension(weekday)
166   .group(byWeekday)
167   .valueAccessor(function(d) { return d.value.avgDelay; })
168   .elasticX(true)
169   .label(function (d) { return d.key.split(' ')[1]; })
170   .xAxis().ticks(3);
171
172   todChart
173   .height(heightShort)
174   .margins({top: 0, right: 25, bottom: 17, left: 5})
175   .dimension(hour)
176   .group(byHour)
177   .elasticY(true)
178   .x(d3.scale.linear().domain([0, 24]))
179   .round(dc.round.floor);
180
181   delayChart
182   .height(heightShort)
183   .margins({top: 0, right: 25, bottom: 17, left: 10})
184   .dimension(delay)
185   .group(byDelay)
186   .elasticY(true)
187   .x(d3.scale.linear().domain([-60, 180]))
188   .xUnits(function(){ return 48; }) // ( 180 + 60 ) / 10
```

```
  .round(function(n) { return Math.round(n / 5) * 5; });

  distanceChart
  .height(heightShort)
  .margins({top: 0, right: 30, bottom: 17, left: 5})
  .dimension(distance)
  .group(byDistance)
  .elasticY(true)
  .x(d3.scale.linear().domain([0, 2500]))
  .xUnits(function(){ return 25; }) // ( 2500 + 0 ) / 100
  .round(function(n) { return Math.round(n / 100) * 100; })
  .xAxis().ticks(6);

  // Format info labels
  movementsChart.title(function(p) {
    return 'Date: ' + dateOutFormat(p.key) + '\n'
    + 'Number of Flights: ' + numberFormat(p.value);
  });

  airportsChart.title(function(p) {
    return p.key + '\n'
    + 'Number of Flights: ' + numberFormat(p.value);
  });

  weekdayChart.title(function(p) {
    return p.key.split(' ')[1] + '\n'
    + 'Mean Delay: ' + precisionFormat(p.value.avgDelay) + ' minutes\n'
    + 'Number of Flights: ' + numberFormat(p.value.n);
  });

  resetableCharts = [
    movementsTimeChart,
    flightDelay,
    airportsChart,
    weekdayChart,
    todChart,
    delayChart,
    distanceChart
  ];

  // Add reset button handling
  resetableCharts.forEach(function(chart) {
    chart.on('preRedraw', function(chart, filter) { checkReset(chart); });

    $('#reset' + chart.anchorName()).click(function() {
      chart.filterAll();
      dc.redrawAll();
    });
  });

  function checkReset(chart) {
      var name = chart.anchorName();

      if ($('#' + name + ' > .reset').css('display') != 'none') {
        $('#reset' + name).removeClass('disabled');
      } else {
        $('#reset' + name).addClass('disabled'); }
  };

  flightsTable
```

```
  .size(20)
  .dimension(airline)
  .group(function (d) { return d3.time.format('%d %B %Y')(d.DateTime) })
  .columns([
    {
      label: 'Time',
      format: function (d) { return d3.time.format('%H:%M')(d.DateTime) }
    },
    {
      label: 'Delay',
      format: function (d) {
        if (d.Delay > 0) return '<span style=\'color:' + colorRange[0] + ';\'>+' + d.Delay + ' min</span>';
        else return '<span style=\'color:' + colorRange[colorRange.length - 1] + ';\'>' + d.Delay + ' min</span>';
      }
    },
    {
      label: 'Origin',
      format: function (d) {
        if (d.Inbound == 1) return d.Airport2;
        else return d.Airport;
      }
    },
    {
      label: 'Destination',
      format: function (d) {
        if (d.Inbound != 1) return d.Airport2;
        else return d.Airport;
      }
    },
    {
      label: 'Airline',
      format: function (d) { return d.Airline; }
    },
    {
      label: 'Distance',
      format: function (d) { return d3.format('0,000')(d.Distance) + ' miles'; }
    },
    {
      label: 'Type',
      format: function (d) {
        if (d.Inbound == 1) return 'Arrival';
        else return 'Departure';
      }
    },
  ])
  .sortBy(function(d) { return -d.DateTime; }) // minus as newest is top
  .on('postRender', function(chart) { // Correct group ordering of table
    var table = $('#' + chart.anchorName());
    var tbodies = table.children('tbody');
    table.append(tbodies.get().reverse());
  })
  .on('postRedraw', function(chart) { // Correct group ordering of table
    var table = $('#' + chart.anchorName());
    var tbodies = table.children('tbody');
    table.append(tbodies.get().reverse());
  });

```

```
306
307   // Update charts' widths
308   function renderCharts() {
309     // Retrieve available space for charts via DOM
310     var half = $('#width-half').width(),
311         quarter = $('#width-quarter').width();
312
313     // Set chart widths
314     movementsChart.width(half)
315     movementsTimeChart.width(half);
316     flightDelay.width(half);
317     airportsChart.width(quarter);
318     weekdayChart.width(quarter);
319     delayChart.width(half);
320     todChart.width(quarter);
321     distanceChart.width(quarter);
322
323     // Update all charts
324     dc.renderAll();
325
326     // Hide loading icons
327     $('.loading').hide();
328   }
329
330   // Render charts upon page load
331   renderCharts();
332
333   // reset all when page is loaded
334   dc.filterAll();
335
336   // jQuery Events
337   // Reset button
338   $('#resetAll').on('click', function() {
339     dc.filterAll();
340
341     $('#airportSelect').val('ALL');
342     airport.filterAll();
343
344     $('#airlineSelect').val('ALL');
345     airline.filterAll();
346
347     dc.redrawAll();
348   });
349
350   // Render charts upon page resize
351   $(window).resize(function() {
352     // Filters are not re-sizable ...
353     renderCharts(); // Re-render charts
354   });
355
356   // Airport selection menu
357   $('#airportSelect').on('change', function() {
358     if (this.value == 'ALL') airport.filterAll();
359     else airport.filter(this.value);
360
361     dc.redrawAll();
362   });
363
364   // Airline selection menu
365   $('#airlineSelect').on('change', function() {
```

```
366     if (this.value == 'ALL') airline.filterAll();
367     else airline.filter(this.value);
368 
369     dc.redrawAll();
370   });
371 
372   // Colour chooser
373   var colours = false;
374   $('#colourflightDelay').on('click', function() {
375     $(this).find('i').toggle();
376 
377     if (!colours) {
378       flightDelay.colors(colorRange)
379       flightDelay.colorDomain([90, 0]) // switched
380       flightDelay.colorAccessor(function (d) { return d.key[1]; })
381       colours = true;
382     } else {
383       flightDelay.colors(d3.scale.category10().range()[0]); // standard blue
384       colours = false;
385     }
386 
387     flightDelay.redraw();
388   }); // toggle icons
389 
390 //-------------------------------CC
    ----------------------------------------------
391 
392   // Example handling for showing/hiding CrossCompare
393   $('.openCross').on('click', function() {
394     $('#crosscompareInfo').slideDown('fast');
395   });
396 
397   $('.closeCross').on('click', function() {
398     $('#crosscompareInfo').slideUp('fast');
399   });
400 
401   // CrossCompare specific logic
402   crosscompare
403   .setHeight(500)
404   .setDateFormat('%d/%m %Hh')
405   .addLegend('#airportSelect')
406   .addLegend('#airlineSelect')
407   .addLegend(movementsTimeChart)
408   .addLegend(airportsChart, 'Airports')
409   .addLegend(delayChart, 'Delay')
410   .add(movementsChart, { type: 'area', yLabel: 'Flights per Hour' })
411   .add(airportsChart, { type: 'bar', order: 'desc',
412     yLabel: 'Flights', xLabel: 'Connected Airports' })
413   .add(weekdayChart, { type: 'bar', value: 'avgDelay',
414     yLabel: 'Average Delay', xLabel: 'Day of Week' })
415   .add(delayChart, { type: 'bar', ratio: 0.2, yLabel: 'Flights', xLabel: '
        Delay (min)' })
416   .add(todChart, { type: 'bar', ratio: 0.4, yLabel: 'Flights', xLabel: 'Time
        of Day (hour)' })
417   .add(distanceChart, { type: 'bar', ratio: 0.4, yLabel: 'Flights', xLabel:
        'Distance (miles)' });
418 
419   $('.maxCrossCompare_open').on('click', function() {
420     var width = $(window).width() * 0.78;
421     crosscompare.setWidth(width).render();
```

```
422   });
423
424   $('#maxCrossCompare').popup({ transition: '0.2s all 0.1s' });
425
426   $('.resetCrossCompare').on('click', function() { crosscompare.reset(); });
427
428 });
429 //--------------------------------Overlay
      ------------------------------------------------
430 var infoOptions = {
431   type: 'tooltip',
432   vertical: 'top',
433   horizontal: 'left',
434   offsetleft: 75,
435   transition: '0.3s all 0.1s',
436   closeelement: '.info_close'
437 };
438
439 $('#infoMovementsChart').popup(infoOptions, { tooltipanchor: $('.
      infoMovementsChart_open') });
440 $('#infoflightDelay').popup(infoOptions, {  tooltipanchor: $('.
      infoflightDelay_open') });
441 $('#infoairportsChart').popup(infoOptions, {  tooltipanchor: $('.
      infoairportsChart_open') });
442 $('#infoWeekdayChart').popup(infoOptions, { tooltipanchor: $('.
      infoWeekdayChart_open') });
443 $('#infoTodChart').popup(infoOptions, { tooltipanchor: $('.
      infoMovementsChart_open') });
444 $('#infoDelayChart').popup(infoOptions, { tooltipanchor: $('.
      infoMovementsChart_open') });
445 $('#infoDistanceChart').popup(infoOptions, { tooltipanchor: $('.
      infoMovementsChart_open') });
```