Visualization of Complex Data —
The Case Study of FUSAMI

João Carlos Almeida Antunes

Master in Multimedia of the University of Porto

Supervisor: Luis Teixeira (PhD)

Co-Supervisor: Klaus Schaefers (MSc)

July, 2013
Visualization of Complex Data — The Case Study of FUSAMI

João Carlos Almeida Antunes

Master in Multimedia of the University of Porto

Approved in oral examination by the committee:

Chair: Eurico Carrapatoso (PhD)
External Examiner: Cristina Sá (PhD)
Supervisor: Luís Teixeira (PhD)
Abstract

With the proliferation of access to the internet, humankind produces and consumes more information than at any other time in its history. The type of data produced ranges from simple pieces of text, to images, video, temporal or geospatial information, among others.

Visualizations try to answer this issue, allowing the understanding and the exploration of large datasets. A good visualization goes beyond the mere static display of information, becoming also the interface with which the user interacts to navigate the data and answering his questions.

The Web Browser has also become the primary window of a user to produce and consume data, being present in every desktop or mobile device with access to the Internet. Mobile Internet usage has surged in the past few years and continues to do so and in the short term will surpass desktop usage.

But no longer previously widely used graphic technologies to build visualizations, such as Flash or Java, can be used to produce visualizations that can be viewed in every device, as those technologies are not ubiquitous in mobile devices. In spite of this unavailability the emergence of HTML5, and other graphic technologies, supported in almost every recent browser, provides an alternative for building visualizations.

The analysis of data can be helpful in many fields, especially analytics and in this thesis we aim to present a case study, FUSAMI, in the area of analytics of usage of mobile phone applications on how they may be visualized to build better applications, by taking into account the real interaction of the user.
Resumo

Com a proliferação do acesso à internet, a Humanidade produz e consome mais informação do que em qualquer outra altura na sua história. Os tipos de dados produzidos vão desde simples pedaços de texto, a imagens, vídeos, informação temporal ou geoespacial, entre outros tipos.

Visualizações procuram ser resposta a este tema de abundância de dados, permitindo extrair conhecimento e exploração de grandes bancos de dados. Uma boa visualização vai além de meros gráficos estáticos de informação, tornando-se também o interface pelo qual o utilizador interage para navegar os dados e procurar respostas às suas questões.

O Browser Web tornou-se também a janela principal do utilizador para consumir e produzir dados, estando presente em cada dispositivo desktop ou mobile com acesso à Internet. O uso de Internet móvel tem aumentado nos últimos anos exponencialmente e espera-se que a breve termo ultrapasse outros tipos de acesso.

Mas já não existem disponíveis para todas as plataformas os recursos que previamente eram utilizados para construir visualizações, como as tecnologias Flash ou Java, deixando de sendo tecnologias ubíquas. Apesar desta falta, o surgimento de HTML5 e a disponibilidade de outras tecnologias gráficas, suportadas por qualquer browser recente, desktop ou móvel, permitem uma alternativa para construir visualizações que se está a tornar o padrão.

A análise de dados pode ser útil em bastantes campos, especialmente em analisar dados de utilização e monitorização de uso, e esta tese apresentará um caso de estudo, FUSAMI, uma plataforma de recolha e análise de dados de aplicações móveis que tem sido desenvolvida para developers e profissionais de HCI construírem aplicações mais usáveis, tomando em consideração a interação real com as mesmas, por parte de utilizadores finais.
Acknowledgements

To Luis Teixeira for the guidance and patience. To Klaus Schaefers, for giving me the opportunity to work on the FUSAMI project and his support.

To Luis Sarmento, Eduarda Mendes Rodrigues and Paula Alexandra Silva for the starting advice. To Carlos Oliveira for his availability.

To Fraunhofer AICOS for allowing me to work in one of their projects that intersects with my interests in both technology and design. To the colleagues and supervisors, many of whom have become friends — to them I can't thank enough.

For my family and few great friends, who have stood by me through all these years.

Thank you.
# Table of Contents

1. **Introduction** ..................................................................................................................1
   1.1. Objectives .............................................................................................................1
   1.2. Project FUSAMI .................................................................................................2
   1.3. Methodology ........................................................................................................2
   1.4. Dissertation Structure ..........................................................................................3

2. **Data Visualization** .......................................................................................................5
   2.1. Introduction ...........................................................................................................5
   2.2. Data Sources .......................................................................................................6
   2.3. Design and User Experience ..............................................................................7
   2.4. Traditional Tools .................................................................................................7
   2.5. State of the Art ....................................................................................................8
      2.5.1. Data Journalism ...............................................................................................8
      2.5.2. Art ..................................................................................................................11
      2.5.3. Social Media and Real Time visualizations ..................................................13
      2.5.4. Quantified Self Movement ...........................................................................13
      2.5.5. Analytics .......................................................................................................14
   2.6. Overview ...............................................................................................................15

3. **The Web Browser** ...................................................................................................17
   3.1. The Web Browser .................................................................................................17
   3.2. The Emergence of Plugins .................................................................................17
   3.3. HTML5 .................................................................................................................18
   3.4. Native Browser Graphic Capabilities ...................................................................18
      3.4.1. HTML/CSS .....................................................................................................19
      3.4.2. SVG .................................................................................................................19
      3.4.3. Canvas .............................................................................................................19
      3.4.4. WebGL ..........................................................................................................19
   3.5. JavaScript Evolution .............................................................................................20
   3.6. Visualization Frameworks ....................................................................................20
   3.7. Overview ...............................................................................................................23

4. **FUSAMI: A Case Study** ............................................................................................25
   4.1. Introduction to FUSAMI .......................................................................................25
   4.2. Interviews ............................................................................................................29
   4.3. Wireframes ...........................................................................................................30
   4.4. Implementation .....................................................................................................33
   4.5. Overview ..............................................................................................................33
5. **Conclusions** ..................................................................................................................35
   5.1. Discussion ...........................................................................................................35
   5.2. Future Work ........................................................................................................35
6. **References** ...............................................................................................................37
7. **Appendix A** ..............................................................................................................41
Index of Figures

Figure 1 - The seven stages of visualizing data 5
Figure 2 - Screenshot of the EveryBlock homepage 9
Figure 3. Screenshot of a Project Cascade Visualization 10
Figure 4. Screenshot of Twitómetro 11
Figure 5. Screenshot of WeFeelFine 12
Figure 6. Screenshot of a MTV: Live Twitter Visualization, from 2009 13
Figure 7. Screenshot of a Google Analytics sample Homepage. 14
Figure 8. Screenshot of a heat map visualization 15
Figure 9. Screenshot of the project FUSAMI homepage 25
Figure 10. Screenshot of FUSAMI’s visualization options 26
Figure 11. An example of FUSAMI’s heat maps on a sample application 27
Figure 12. An example of FUSAMI’s sequential patterns mode on a sample application 28
Figure 13. Screenshots reorganization, based on user focus, wireframe 30
Figure 14. Results filter wireframe 31
Figure 15. Query browser wireframe 32
Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AICOS</td>
<td>Assistive Information and Communication Solutions</td>
</tr>
<tr>
<td>FUSAMI</td>
<td>Fraunhofer Usage Mining</td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>SVG</td>
<td>Scalable Vector Graphics</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>WebGL</td>
<td>Web Graphics Library</td>
</tr>
<tr>
<td>D3</td>
<td>Data-Driven Documents</td>
</tr>
<tr>
<td>RIA</td>
<td>Rich Internet Application</td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
</tbody>
</table>
1. Introduction

The motivation for this dissertation comes from the author personally having a design and technology background and on his current position at Fraunhofer AICOS works in both of these fields. A great data visualization combines both to produce unique interfaces to explore information, so there is a need to have knowledge in both fields to be able to produce great work in this area.

Fraunhofer AICOS main work area is in the field of developing solutions for older adults. There is an emphasis on using smartphones and remote solutions to improve older adults lives and connect them more to their families and the world. Also this provides a way to capture a lot of data for analysis, such as patterns of daily lives, medical data and other types, which all have to be visualized so that information can be comprehended easily and patterns easily detected. This knowledge can be reused by the institute and also the community in the information visualization area.

Data Visualization has evolved greatly in the last few years in several fronts. A greater deal of data is being produced, more than we can consume or analyze with resourcing to complex data mining and visualizations to fully comprehend the information, see actionable patterns and find the information that most interests us. This in either in industrial settings or personal ones, such as social networks and personal gadgets. Also we are no longer living in a world where the predominant interface of a person with interactive information is a desktop personal computer, where we access and create information, but now a range of interconnected of personal devices, such as smartphones and tablets are being more used, so there is a lot of research to be done in this area on how to interact and display information.

1.1. Objectives

One of the objectives of this dissertation is to research the field of visualization of complex data, providing technological and design overviews of the state of the art in the field, which can be used as a guide in the development of visualizations in other project, with an extreme focus on the web browser as the deployment target, which in this day and age, with access to the internet on a variety of devices worldwide, is the preferable platform to create and consume information and visualizations on.

Also we aim to incorporate the knowledge gained by applying it on a Fraunhofer AICOS’ FUSAMI project, described in the next section.
1.2. Project FUSAMI

This research is expected to contribute to the development of an application in development at Fraunhofer AICOS, called FUSAMI, which stands for Fraunhofer Usage Mining (Fraunhofer Portugal 2013).

FUSAMI has several modules, but the main one is to provide an advanced analytics platform, of real-time usage data, on smartphones. With this information, developers and HCI (Human Computer Interaction) can leverage it to develop more user friendly applications.

It is already normal to do usability tests in the field with real users, normally for a couple of minutes, and by observing them, but nothing is as good as live field tests of the applications in the hands of the end users. FUSAMI receives usage data in realtime, analyses it and provides screenshot based views for the developers to see how their work is being used.

This thesis aims to contribute to this project, improving the current visualizations and conceiving new ones, which will be described in more detail on Chapter 4.

As it is a industry project there was a delay in the start of the work for this dissertation which was not fully done, but it is described in the future work section and it is expected to be completed in the coming months.

1.3. Methodology

The development of this dissertation followed the following structure.

First, a research of the state of the art was conducted, focusing on the technological and interaction design aspects of visualizations, to learn and evaluate how to best build them targeting web browsers

To apply the knowledge gained a case study was selected, the Fraunhofer AICOS analytics application FUSAMI. When the thesis started the application was already under testing stages, with some visualizations of the data it analyzed already preliminary built so interviews and usage observation were conducted with the target audience of the product - developers and design professionals - in order to evaluate them. This follows an User-Centered Design methodology in the approach of the development, by involving the users early on in the process and having their input taken into account (Yvonne Rogers 2007).

Wireframes were then produced and again evaluated with the users.
1.4. Dissertation Structure

The dissertation is organized in 5 chapters.

This first chapter introduced the context of this dissertation, outlining the motivation, objectives, the working project and methodology.

The second chapter provides an introduction to the field of Data Visualization, its increased importance nowadays, to its tools and to some state of the art visualizations in different fields.

The third chapter tracks the history of the web browser, since the inception of the internet, and how it has evolved its capabilities to be more relevant in a multi-platform world and new types of data required new technologies to emerge that help tremendously the work on the field visualizing data.

The fourth chapter presents a case study of Fraunhofer Portugal project, FUSAMI, where we take what was researched in the previous chapters and apply in several components of the project to improve it.

Finally, the fifth chapter presents and overview of the results of this research, reporting contributions and future work to be developed.
2. Data Visualization

By complex data, or information, we refer to data that goes beyond mere numbers or names that can be charted, resourcing to an Excel spreadsheet, to data that can have multiple types of values or objects (such as photos, geolocation, time, social graph objects, and so on) that require an interactive visualization to explore it and a mere static infographics can’t convene all the information.

In its book, ‘Visual Complexity’ (Lima 2011), the author and researcher Manuel Lima, states that “Our ability to generate information now far exceeds our capacity to understand it. Finding patterns and making meaningful connections inside complex data networks has emerged as one of the biggest challenges of the twenty-first century”.

A good visualization becomes the tool that enables the exploration of sets of data and enables the user to choose its own path in the interaction with the information.

2.1. Introduction

Visualization as a sub-field of science, statistics, and graphics has only been recognized as its own entity since the mid-to late-80s (Fry 2008). Originally it was used to communicate more effectively via info graphics in newspapers or representations in science, but it has increasingly become an important communication method to convey information, and with the emergence of digital technology and the internet, a way to explore information accessible to more people.

As represented in Figure 1, the production of visualizations require a wide range of skills, from programming, to mathematics and design, different in each of the stages:

1. **Acquire** - the process of obtaining the data, that can be from resources on files to APIs in the internet
2. **Parse** - when the data has been gathered, it needs to be provided structure and made sure that is consistent.

3. **Filter** - removing all but the data of interest.

4. **Mine** - the application of methods from statistics or data mining, as a way to discern patterns or place the data in mathematical context.

5. **Represent** - determination of a simple representation, whether the data takes one of many shapes such as a bar graph, list or tree.

6. **Refine** - improvements to the basic representation to make it clearer and more visually engaging for the user.

7. **Interact** - the addition of methods for manipulating the data or controlling what features are visible.

For this dissertation the main stages explored were **represent, refine and interact**, with a focus on the technology and design process behind the creation of data visualizations for the web.

### 2.2. Data Sources

Our information production as increased greatly but also the information that one has available to access to has increased, with companies and institutions becoming slowly more open, providing access to sets of data.

Internet companies started to see the value of opening APIs (Application Programming Interface) to its data, to enable the building of applications by other developers, creating ecosystems that are mutually beneficial to the company and the developer. Example success cases of this approach are companies like Twitter and Facebook, which allow the use of their APIs has building blocks for other applications, which include visualizations.

Also institutions, such as local or national governments, started to see transparency as a good practice and supplying sets of data to the public, inviting them to explore it. Good examples are the websites from the projects DATA.GOV (Government 2013) in the United States of America and DATA.GOV. UK, from the United Kingdom, which provide a large quantity of datasets.

In Portugal there is the example of the websites dados.gov (Admnistrativa 2010), still in beta and with a limited number of datasets. Another approach is taken by PORDATA (Santos 2009), created in 2009, which acquired data directly from institutions and provides visualizations of it, but without supplying the underlying data they are built on.
The easier access to information and datasets creates an opportunity for more growth in the development of data visualizations - a field in constant evolution, fertile for research and development of new tools of dissemination and consumption of information by a mass audience.

2.3. Design and User Experience

Technology is important and serves as the backbone of building an interactive data visualization, but if the design and the user experience is not aesthetically appealing, easy to use and, most important, the information we try to convene easily assimilated then the visualization fails its goal of informing the user.

A visualization involves many fields in the design stage, such as visual design, animation, good aesthetics, interaction design, physics and others.

This differentiates from infographics, which usually involves more graphic design than any other field, merely convening static visual representations of information, previously chosen and providing its own conclusion to a user. A visualization doesn’t provide its own conclusion and offers the user the possibility to navigate the information so this requires the involvement of other fields of design and computing, in order to provide a good visualization.

2.4. Traditional Tools

Traditional data analysis tools such as Excel or others that are static, not interactive, or merely a set of presets, are not useful for this modern age or data can be of multiple types and not merely strings of numbers or letters.

To really provide good visualizations of complex data it is necessary to design interactive experiences having in mind the user needs, by following user-centered design methodologies, and using technologies that make the visualization accessible in a myriad of devices and also future-proof - an HTML document from 20 years ago, with a table chart, is still readable by today’s browsers for example.

So this thesis, as previously stated, as a focus on the web.
2.5. State of the Art

Data visualizations are used in many different areas, ranging from either topics and complexity. We focus on a few of the more popular areas and particular ones, accessible on the web, where more information is being produced, innovation is happening more and best practices are being developed and implemented, selected from the book and companion website Visual Complexity (Lima 2011) - a renowned resource of visualizations by the portuguese researcher Manuel Lima.

2.5.1. Data Journalism

Journalism is in a transitional period. Looking for new sources of revenue, new approaches are being experimented. Not just to automate several tedious process, find patterns that might become news in this social media world we live in today, but also to captivate the reader.

EveryBlock

In 2005 a journalist, Adrian Holovaty, created a mashup of the recently made available Google Maps with crime reports from the Chicago Police, locating crimes in the map (Wikipedia). It caught national attention in the US and was a glimpse of how newspapers could leverage often disregarded statistical data to present news online to the reader.

Later Holovaty when on to create the Python framework Django and received a grant from the Knight Foundation, focused on enabling technological advances in journalism to create EveryBlock.
The main concept behind EveryBlock was to analyze data provided by public institutions, such as Municipals and Police, and display it in a geographic context. Types of data included: crimes, city permits, fires, public events and others. While this data may seem uninterested to a more wider audience, it has relevance for the people that are located nearby. This is interesting as a new way of doing automated local journalism, relevant to the reader.

EveryBlock was bought by MSNBC, but, despite popular in neighborhood communities, it wasn’t scalable and it was closed down in 2013 (Wikipedia). Due to receiving a journalism grant for the development, the source code is open source (Everyblock 2009).

Project Cascade - New York Times

The renowned newspaper New York Times has a Research and Development department dedicated to build new ways of visualizing news (Times). One of the most interesting projects to emerge recently is Project Cascade (Times).
Figure 3. Screenshot of a Project Cascade Visualization

Project Cascade provides a 3D overview of how New York Times stories are shared on Twitter. This visualization enables to see how individual stories propagate on Twitter over time, which users re-tweet the stories and those who have more influence in the story propagation.

**Twitómetro**

Twitómetro is a good example of work done at FEUP in collaboration with SAPO (LABS 2011). This application monitored Twitter’s public feed of portuguese tweets mentions of each party leader, in the 2011 elections. and analyzed daily the sentiments of the Portuguese people toward each leader.

This visualization used the Raphaël visualization framework, which is analyzed in Chapter 3.
2.5.2. Art

We Feel Fine

We Feel Fine (Harris 2007) is among the most popular and first data visualizations to gather awareness to the medium. Contrary to the other examples showcased it is built on Java, relying that the browser has the required Java Plugin installed to experience it.

The website collects data from blogs, photo sharing sites and social networks and extracts phrases searching for certain feelings. Then it connects them with photos or sentences, creating an emotional connection to the user that visits the website.
It is perceived has a good and pioneering example of what a good data visualization must be: aesthetically appealing, creating a connection with the user (in this case emotional), inviting him to explore more of the data on the website.

**Chrome Experiments**

After Google launched its web browser Chrome it wanted to display its state of the art graphic capabilities so it created a website to showcase them: [http://www.chromeexperiments.com/](http://www.chromeexperiments.com/)

The goal was to display what is possible to achieve graphically in the browser using the latest open web technologies (such as HTML5, Canvas, SVG and WebGL - described in the next chapter of the dissertation). Not all the content on this website showcase is visualizations, but some are and demonstrates the power of using just the browser’s native capabilities.
2.5.3. Social Media and Real Time visualizations

MTV: Live Twitter Visualizations

Figure 6. Screenshot of a MTV: Live Twitter Visualization, from 2009

Stamen (http://stamen.com), a design and technological studio in San Francisco developers visualizations for clients such as Twitter and MTV. Their live Twitter visualizations for MTV Awards (Stamen 2009) ceremonies are a good example of what is possible to achieve when access is provided to live streams of data, and how it is possible to make sense of it.

2.5.4. Quantified Self Movement

People are more and more starting to track their habits, with products such as the Fitbit (http://fitbit.com), Nike+ (http://nikeplus.nike.com/plus/), Jawbone Up (https://jawbone.com/up), and also through mobile applications such as Foursquare (https://foursquare.com/) or the Moves application (http://www.moves-app.com/) - where the visualization of information plays a important role in the user adoption of the application.

Facebook even hired one of the pioneers in the field of data visualization, Nicholas Feltron (http://feltron.com), known for his yearly infographics of his life (Wikipedia 2013) and his web application Daytum (http://daytum.com), in order to redesign Facebook’s homepage - introducing the concept of personal data to more than a billion people.

Many devices with sensors are appearing in the market and the data produced will have to be kept and visualized in some way. People’s interest for using this devices may range from...
keeping personal memories, storing medical information and more, even uses that we can’t imagine yet. This is a trend that we will see more and more in the future and where visualizations will play a key role first in the adoption and afterwards in the analysis of the data collected.

2.5.5. Analytics

Google Analytics

One of the tools most commonly used for gathering analytics, on the web, is Google Analytics (previously Urchin, later bought by Google and renamed). It allows a website owner to analyze the incoming traffic to its website, and it’s free to use (Google).

![Figure 7. Screenshot of a Google Analytics sample Homepage.](image)

Crazy Egg

While Google Analytics is more focused on traditional visitor’s statistics, such as type of browser or location, there are other analytics applications more focused on analysis of the user interaction — on which the case study FUSAMI focuses.

The application Crazy Egg (Egg) is an example of this type of application, tracking where a user interacts with the mouse on a web page.
2.6. Overview

In this chapter an overview of the state of the art of visualizations was presented, focusing on specific fields, but there is a vast amount of visualizations fields and examples that could be provided, such as the scientific, so this is only a mere sample of the work being produced today on and for the web.

FUSAMI is in the field of analytics, which we will present in further detail in the 4th chapter.
3. The Web Browser

In this chapter we will discuss the web browser and its evolution.

Initially developed as an interface to the World Wide Web, with support for simple text formatting, it has evolved to one of the preferred environment to develop applications and even businesses on.

3.1. The Web Browser

When the WWW (World Wide Web) was created in 1990 the Web Browser became the application to access and navigate it.

Different browsers exist with some in the past having been discontinued to the decreased market share, such as Netscape Navigator, which later became Firefox. The most popular browsers nowadays are Chrome, Firefox and Internet Explorer. Taken into account the recent surge in mobile internet Safari is also one of the main ones (W3Counter 2013).

HTML was in the past limited and each browser vendor had different implementation of web standards which made browser compatibility and accurate display of designs complicated. Also it was limited to text and images, and so plugins emerge to fill this gaps, such as playing videos, listening to music, more dynamic websites, advanced applications and games, for example (Keith 2010).

3.2. The Emergence of Plugins

When building interactive visualizations one had to resource to these browser plugins, to enable more graphic advanced capabilities and interaction. Here follows a description of the most popular and common ones.

Java

Java is a programming language, created by Sun Microsystems in 1995. Used primarily to develop desktop or server-side applications, it can also be used to develop software that can be used in a browser via Java Applets. But there are issues to this approach: besides not being native it is slow to load, processor intensive and not operating system native looking.
**Flash**

Developed by Macromedia (later bought by Adobe) the Flash ecosystem was the most popular one among designers to build interactive experiences on the web, so it also became the ideal development environment to use for visualizations.

Almost every browser, until a few years ago, shipped with the Flash plugin already integrated which made it the most popular one for rich interactive experiences, applications and video playback.

With the emergence of devices such as smartphones and tablets, without proper support (or even any support in the case of Apple devices) for the Flash plugin, combined with the web standards evolution of supporting graphic and interaction only previously possible with Flash in the browsers of every device, it has been loosing market share and popularity among developers.

**Silverlight**

Silverlight was Microsoft’s attempt to counter Flash’s popularity and have a product in area, but didn’t achieved wide spread usage as it required being installed by the end user in most browsers and even Microsoft is now investing more in HTML5.

### 3.3. HTML5

HTML5, in a strict technical meaning, refers to the addition of new tags and functionalities to version 5 of the HyperText Markup Language, but it widely used to refer a whole set of support for new CSS standards and JavaScript capabilities that make the the browser approach the status of a desktop application, with capabilities for a more closer interaction with the host operating system and user.

Web Standards are nowadays more followed and rapidly implemented by browser vendors, so browser differences aren’t so noticeable as before and the new native graphic capabilities bring new possibilities to the field of data visualization.

### 3.4. Native Browser Graphic Capabilities

Despite many of the technologies described not being recent developments in technologies, some even date back to the 1990s, their browser support was never ubiquitous and APIs consistent. In the last few years we have seen standardization of these graphic capabilities natively in almost every browser on the market, such as Firefox, Safari, Chrome and even Internet Explorer (2013), which has lacked behind in the past when it comes to the implementation of web standards.Today, recent browsers, either desktop or mobile, have
implemented support for these new technologies, with a continued effort by browser makers to support web standards. This wider adoption by browsers allows to finally this technology being used in new and creative ways by developers.

One other advantage of using natively the browser capabilities and not plugins is that it makes the source code and data available to be indexed by search engines, improving the discoverability and dissemination of work, which wasn’t possible with plugins in the past.

### 3.4.1. HTML/CSS

Though it might seem odd the inclusion of HTML/CSS here, as they are ‘old’ technologies, but recent additions to CSS are worth a mention as they are better suited for simple visualizations, instead of more computer intensive and complicated graphic counterparts.

CSS3 now has support for features such has 3D transforms, animation, filters, shadows, reflection and more, that when combined with a JavaScript framework controlling them can be a powerful visualization structure, while remaining simple and in a syntax already familiar to most front-end developers.

### 3.4.2. SVG

Scalable Vector Graphics (SVG) is a vector image format for 2D graphics. Though an open standard since 1999 only in the last few years has gained wide browser support (Wikipedia 2013). Before this wider adoption on the web it was mainly used in desktop graphic software, as an interchangeable format, based on an XML based format files.

It enables the browser to render vector graphics without the resource to any plugins, being Flash the most common tool used for this in the past.

### 3.4.3. Canvas

HTML5 introduced the new Canvas element (WHATWG 2013), which allows to draw in 2D on an element and is scriptable, via JavaScript. Initially developed by Apple for their own rendering engine it was adopted by other browsers. The API allows the drawing of common graphic elements, such as lines, rectangles, dots and transformations.

Canvas can also be used in a 3D context resourcing to WebGL.

### 3.4.4. WebGL

WebGL is an implementation of the OpenGL ES 2.0 low-level 3D Graphics API (Khronos), which can be used on the new HTML5 Canvas element previously discussed. It’s royalty free and brings 3D support to the Web. It is already supported by the main browsers, though not all have it activated, sometimes being necessary for the user to activate the functionality in the preferences.
3.5. JavaScript Evolution

JavaScript was developed at Netscape in the 1990s (Keith 2010) as a browser programming language that enabled more user and server interaction with HTML documents.

It was initially dismissed and only later adopted by Microsoft’s Internet Explorer and it only reached wider use in 2005 with the popularity of what is now called AJAX - Asynchronous JavaScript and XML, a technique that provided the building of more dynamic applications in the browser.

Besides JavaScript being used in web front-end development it has also become popular to be used in server-side applications, this way the developers only have to program in one language for either the backend and front-end of the application.

But most importantly it is through JavaScript provided APIs that the browser manages to display technologies such as WebGL and Canvas, and allows developers and designers to build visualizations with them and frameworks are built in JavaScript for easier development.

3.6. Visualization Frameworks

When deciding to use a technology, Miguel Rios, leader of the Visual Analysis & Insights team at Twitter, advises developers to ask themselves the following questions (Rios 2013):

• How many elements are in the visualization?
• How complex are the elements?
• Is it going to be interactive?
• Will it have animations or transformations?
• Do You need to support legacy browsers?
• Are there related open source examples or frameworks?

The last question, according to Miguel Rios, is the most important one, because working natively with the graphic technologies is not a good idea due to development time spent, bugs in older browsers, or current browsers different implementations of an API can delay substantially a project. A previously well developed framework, especially by a web developer community, is very helpful to get started more quickly, which is important when working with data — to have the ability to quickly prototype and iterate. Also there is no need to rebuild functionality that a framework provides for us.
Next we present the currently most common used frameworks for visualizations on the web:

**D3 — Data-Driven Documents**

As we saw in the early stages of the World Wide Web the main goal was to share knowledge, resourcing to HTML. D3 *(Bostock 2013)*, an abbreviation for Data-Driven Documents and also sometimes referred to as D3.js, has also this goal in mind, using native browser technologies.

It writes data in browser native formats, such as HTML, SVG and CSS, and then resourcing to JavaScript provides interactivity to be built upon that information.

Built from the ground up to build visualizations, this framework provides a complete toolset that deals with requirements such as design, selection interaction with the browser DOM, handling and transforming data, interaction and animation and provides extensibility via modules.

**Processing**

Processing *(Ben Fry 2013)* is probably the most well known programming language targeted to the visual arts. It had its origin at MIT and it is built upon Java.

Initially the deployment environment, that is, the way to distribute works built on it, were Java Applets - either for the desktop or browser plugin.

John Resig, author of the popular jQuery JavaScript library, started to develop a port of Processing to JavaScript, in 2008, enabling it to run natively, and faster, on the browser, without resourcing to plugins.

The last version of Processing 2, released this year, taking advantage of previous mentioned port, incorporates a JavaScript mode which enables the use of this language, and several modules, natively on the browser without requiring a Java plugin. This is welcomed feature, has it now allows work to be displayed in devices such as smartphones and tablets, that don’t support Java in the browser or in the system.

**Raphaël**

Raphaël is a JavaScript Library that simplifies working with vector graphics on the web *(Baranovskiy 2013)*. It is cross-platform, supporting older versions of Internet Explorer by enabling the render of vectors in VML (Vector Markup Language), a format developed by Microsoft and only supported by Internet Explorer.
**PhiloGL**

PhiloGL is a WebGL framework targeted to visualizations, creative coding and also games (Labs 2013). Instead of working with WebGL directly, this framework eases the development of WebGL targeted applications.

**three.js**

three.js is a 3D Library, which via its JavaScript API, supports the drawing in a 3D space, resourcing to SVG, WebGL or Canvas, depending on the browser native support and support (three.js 2013). It is lightweight and optimized not to be processing intensive. Its development is mainly sponsored by Google.

**Others**

There exist other frameworks, JavaScript ones, that are used to provide features for general web development that also provide some graphic capabilities and building blocks for visualizations. Popular ones such as jQuery, MooTools and Dojo (on which FUSAMI is built upon), but their strength and development goal is not to build visualizations, so they have many limitations and require more extensive coding to achieve the same capabilities as a visualization targeted framework, when that is possible, which in more complex visualizations they don’t have the capability.

The Dojo JavaScript framework (Foundation 2013) was initially chosen as the developers were familiarized with it, from previous projects and it provided the necessary complements to build the initial stages of the application front-end. It also provided basic graphics and charting capabilities as an extension, in an already familiar structure - which was useful to speed up development. But building custom visualizations with these components was time-consuming and hard, not performing well sometimes, taking up too much computing resources and time to render. So it was necessary to research JavaScript frameworks that were build specifically to handle interactive visualizations.
3.7. Overview

Some small experiments were made with each one to learn them, and the best choice and the one that has more active development is D3 (Bostock 2013), with support for a variety of graphic technologies, good documentation, implements best practices for code writing, debugging and performance, having already support for a wide range of common visualization use cases.

As we could see in this chapter the technological advances of the last few years, in multiple fronts, such as software and hardware, have changed the landscape of what is possible to build. We now have better data sources, better technologies and more types of devices to consume visualizations.
4. **FUSAMI: A Case Study**

4.1. **Introduction to FUSAMI**

FUSAMI is a tool being developed at Fraunhofer AICOS to aid in the design of smartphone applications, providing better insights to the usage by testing them during the development stage. Traditional HCI methods such as interviews and usage observation can be subjective and not obtain all the information that could be useful when testing an application.

So a tool that can provide information from real user interaction, applying state of the art analytical algorithms, drawing attention to possible design issues is a valuable one. Providing indications of what works or not in a smartphone interface.

![Figure 9. Screenshot of the project FUSAMI homepage (AICOS 2013)](image-url)
FUSAMI can be installed in Android and iOS applications and monitors the usage of the user, constantly sending it back to a server where it is analyzed and can be accessed by the application developers, providing several visualization options.
Figure 11. An example of FUSAMI’s heat maps on a sample application

To be more accessible, and a feature that differentiates itself from other tools in the field, is that it is screenshot based and overlays visualization information on top of them, or between the different application screens.
One of the most useful features of FUSAMI is recording the information of where a user clicks in an application screen and showing the screen he is going to. This provides developers a way to see more easily navigation patterns that work and those that don’t, where the user has difficulty. With this actionable information the design of an application can become more user-friendly.
4.2. Interviews

To find out how FUSAMI could be improved, and which visualizations to build, interviews and observations were conducted on-site at Fraunhofer, with 1 HCI designer and 2 Android developers. As FUSAMI is so far an internal Fraunhofer project, only used by its employees no interviews were conducted with people outside the research center.

An interview script can be found in Appendix A of this document, and a summary of the findings will now be presented.

The first questions was if they commonly used analytics programs in the developed applications and the answer was for all the 3 interviewees yes, but it was only used after building them and not during the development and testing with users. A tool of the kind of FUSAMI, to be used during development and user testing, that would enable them to get more actionable feedback than mere interviews with users was considered to be a valuable resource to them.

Two of the three interviewees had already tried FUSAMI, but found it too cumbersome too use. For example, they stated that the first thing that interested them more was to see the most parts of an application and it was complicated if the application had lots of screens. They also referred that FUSAMI was a little bit confusing to navigate and understand some of the statistics, especially for the HCI designer.

With the developer that hadn’t tried FUSAMI an observation was conducted to see how he used the application. He found it hard to use, not understanding what the features the navigation referred were actually about and didn’t understand the numerical statistics provided. Though he did find the screenshot-based approach to provide information a differentiator form other mobile analytics had and it would be easier to see where users actually clicked and which screens used.

When asked about suggestions for the application they said they would like a more easy to filter interface and also query for data regarding a specific screen or interaction.
4.3. Wireframes

After the conclusion of the interviews we took that feedback into account to improve existent features of FUSAMI and produce new ones, that would make it more adequate and appealing for use by the target users.

Screenshots focus

The first problem to be analyzed was to make FUSAMI ‘less crowded’, when it had lots of screenshots, and enable the user to focus.

Based on the application screenshot selected, those that are not related will fade and have the opacity reduced to 10%, when the user hovers with the mouse. This way the user is more focused and can obtain the information in a faster and clearer way.

Figure 13. Screenshots changes, based on user focus, wireframe
**Filter**

When lots of application screenshots are in view they are usually displayed in a random order, nor organized by any method, so this makes it harder to find and get information.

There should be a filter toggle created, so that a user may select the order, according to the visualization he selected, to organize the screenshots.

![Figure 14. Results filter wireframe](image_url)

<table>
<thead>
<tr>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order by most used</td>
</tr>
<tr>
<td>Order by most clicked</td>
</tr>
</tbody>
</table>

Figure 14. Results filter wireframe
**Query Browser**

A query browser is a new feature, that would enable the users of FUSAMI to select the screens they want to see information of, and also sets of screens that may have related information.

![Query Browser Wireframe](image)

**Figure 15. Query browser wireframe**

The wireframe above shows a selection of a single application screen and screens related to it. But a user can also press the + sign on the interface and select further application screens that are related and see related interactions.

After building these wireframes and comparing them with the FUSAMI application, we asked the previous interviewed users to compare these ideas to the FUSAMI current build. They stated that these changes would be helpful for them to interact the application and more easily find the information that were looking for. Also the new query tool would be of value to them, to search for usage patterns among the developed applications monitored by FUSAMI.
4.4. Implementation

Although FUSAMI is built with the Dojo JavaScript framework (Foundation 2013), that has some capabilities for building visualizations (the more traditional ones), is not really targeted to build custom visualizations, requiring extensive custom coding to rework existing ones.

In order to build upon the interviews and wireframes built the choice was to use the D3 visualization framework.

The backend of FUSAMI has APIs that return JSON data that can be leveraged by the D3 visualization framework, without requiring any extensive rewriting of the code on the backend. As the framework is also lightweight (in size and computation requirements) it integrates well with the Dojo JavaScript framework.

4.5. Overview

With the work done for this case study, we managed to improve the design of the visualizations to make them more accessible to the target audience and elaborate a new feature that will be useful to extract the information the user may want to access, by querying it.
5. Conclusions

5.1. Discussion

With this dissertation we managed to obtain knowledge on the building of visualizations with complex data for the web and apply on a case study to improve the FUSAMI tool. Beyond the research, the interviews, observation and initial design the proposed changes to FUSAMI still need to be implemented in programming and visual design.

As the information volume created each day in the world increases, the subject of Data Visualization starts to become paramount in order to not only analysis of the information but also to communicate effectively. This thesis aimed to provide a starting point in building visualizations which we believe we achieved.

Through our research we analyzed different visualization libraries, examples of best practices, web standards and obtained what we expected to be a reference to be build upon and used at Fraunhofer and by the academic community.

5.2. Future Work

Implementation

There was a delay, by two months, in the approval of the start of the work on the project at Fraunhofer AICOS so only wireframes and mockups of the visualizations prototypes have been produced so far. It is expected in the coming two months to integrate the developed ideas and research with the FUSAMI project, using the technology researched in chapter 3 and implementation described in section 4.4.

A reference for future visualization works

Also the research work on browser graphic capabilities can be placed and updated in a live document, such as a wiki, to further it in the future as there is a constant evolution to browsers specifications, support and framework development, that may be useful in a process of choosing the right tool and approach to visualize different types of data.

Fraunhofer AICOS works with many projects related to sensors and data gathering, for which dashboards with visualizations start to become a necessity and this thesis provides a foundation on which to build reusable components in the present and future for different and complex types of data.
References


dados.gov.pt.


raphaeljs.com/.


d3js.org/.

crazyegg.com.

code.google.com/p/ebcode/.


Appendix A

Interview script for FUSAMI users

1. Have you used any analytics program for your applications?
   If the answer is positive, ask the following questions:
   
   1.1. Did you use it during development (user testing) or after launching?
   
   1.2. Did you measure any variables that contributed to the design or optimization of the interface?
   
   1.3. What sort of usage data would be useful for your work during the development?

2. Have you used FUSAMI?

3. What statistics you were more interested on knowing?

4. Did you find what you were looking for easily?

5. What would you like to see added?