



Intuitive Information Visualisation for the MindXpres Presentation Tool

Proefschrift ingediend met het oog op het behalen van de titel
Master of Science in de Ingenieurswetenschappen: Computerwetenschappen, door:

Yves Baeten

Promotor: Prof. Dr. Beat Signer
Begeleider: Reinout Roels

AUGUSTUS 2015





Vrije Universiteit Brussel

FACULTY OF SCIENCE AND BIO-ENGINEERING SCIENCES
Department of Computer Science
Web & Information Systems Engineering Laboratory

Intuitive Information Visualisation for the MindXpres Presentation Tool

Graduation thesis submitted in partial fulfilment of the requirements for the degree of
Master of Science in Applied Sciences and Engineering: Computer Science, by:

Yves Baeten

Promoter: Prof. Dr. Beat Signer

Advisor: Reinout Roels

AUGUST 2015



*“There is nothing better than a picture for making you
think of questions you had forgotten to ask (even mentally).”*

— JOHN W. TUKEY (1985)

Samenvatting

Informatie visualisaties worden vaak gebruikt tijdens presentaties. Dit voor het presenteren van gegevens op een duidelijke en overzichtelijke manier, en om het publiek te helpen om onmiddellijk inzicht te krijgen over de boodschap die de spreker probeert over te brengen. Om verdere verbetering aan te brengen in de dialoog tussen de spreker en zijn publiek, introduceert dit proefschrift een plug-in met interacties op interactieve informatie visualisaties voor de MindXpres presentatie tool.

Door middel van een literatuurstudie en wat kritisch denkwerk, identificeren we hoe interacties op informatie visualisaties verbeteringen, in het vermogen van de spreker om zijn publiek te boeien en te reageren op vragen over gepresenteerde visualisaties, kunnen aanbrengen. Met de identificatie van interactie en transitie technieken, geleend uit het domein van interactieve en verhalende visualisaties, zal de spreker de mogelijkheid hebben een visualisatie dynamisch aan te passen tijdens een presentatie. Hierdoor krijgt de spreker de gelegenheid om onvoorbereide visualisaties, aangepast aan de noden van zijn publiek, te presenteren.

Abstract

Information visualisations are often used during presentations for presenting data in a clear and uncluttered way, and to help an audience get an immediate insight of what the speaker is communicating about. To further enhance the dialogue between the speaker and his audience, this thesis proposes a plug-in with interactions on interactive information visualisations for the MindXpres presentation tool.

Through literature study and critical thinking, we identify how interactions on information visualisations can enhance the speaker's ability to captivate his audience and react to questions regarding presented visualisations. With the identification of interaction and transition techniques, borrowed from the domains of interactive and narrative visualisations, the speaker will be able to dynamically adapt a visualisation during his presentation. Thus giving the speaker an opportunity to present unprepared visualisations on demand, suited for the needs of the audience.

Acknowledgements

I would like to thank,
my promoter Prof. Dr Beat Signer,
my advisor Reinout Roels,
for the constructive
criticism, guidance, and help,
provided in my research.

Also,
the people who stood close to me,
during this process,
giving me the moral support,
and once in a while,
a shout of encouragement.

Contents

Introduction	1
1 Information Visualisation	5
1.1 A Definition for Information Visualisation	5
1.2 Data Versus Information	7
1.3 The History of Information Visualisation	7
1.4 Visualisation Types	14
2 Interactive Visualisation	17
2.1 Interaction	17
2.2 Interaction Techniques	19
2.2.1 Select and Highlight	20
2.2.2 Brush and Link	20
2.2.3 Reorder	21
2.2.4 Query	22
2.2.5 Filter	22
2.2.6 Zoom	22
2.2.7 Abstract	23
2.3 Transitions	24
2.3.1 View Transformation	24
2.3.2 Substrate Transformation	24
2.3.3 Filter	24
2.3.4 Order	24
2.3.5 Time-step	24
2.3.6 Visualisation Change	25
2.3.7 Data Schema Change	25
2.4 Tasks	25
3 Narrative Visualisation	27
3.1 Story Definition and Model	28
3.2 Storytelling in Visualisation	29
3.3 Visual Narratives	30
3.3.1 Genres	30
3.3.2 Visual Narrative Tactics	31
3.3.3 Narrative Structure Tactics	31

CONTENTS

3.3.4	Author-Driven and Reader-Driven	32
3.4	Storytelling Scenarios	33
4	Visualisation and Presenting	35
4.1	Gapminder	36
4.2	Current Implementations	37
4.2.1	PowerPoint	37
4.2.2	Keynote	39
4.2.3	OpenOffice Impress	40
4.2.4	Alternative Presentation Software	40
4.2.5	Workarounds	41
4.3	The Issue	42
4.4	The Next Step	43
5	MindXpres	45
5.1	The RSL Model	46
5.1.1	The User Model	47
5.1.2	Layers	47
5.1.3	Structural Links	48
5.1.4	Conclusion	48
5.2	Fundamental components	48
5.2.1	XML Authoring language	49
5.2.2	Compiler	50
5.3	Plug-in Mechanism	51
5.3.1	Components	51
5.3.2	Containers	52
5.3.3	Structures	52
5.3.4	Implementing Plug-ins	52
5.4	Conclusion	53
6	Implementation	55
6.1	Goals	55
6.2	Architecture	56
6.2.1	Underlying Libraries	56
6.2.2	Initialisation	57
6.2.3	The Core	63
6.2.4	Interactions	64
6.2.5	API	65
6.2.6	Interface	66
6.2.7	Basic Settings	66
6.3	Integration in MindXpres	67
7	Use Case	69
7.1	The scenario	69

CONTENTS

7.2	Chart Set-up	70
7.3	Narration	70
7.4	Conclusion	74
8	Conclusion and Future Work	75
8.1	Contributions	75
8.2	Future Work	76
8.2.1	User Study	76
8.2.2	Visualisation types	76
8.2.3	Linking visualisations	77
8.2.4	Chart traversal	77
A	Tables and images	79

Introduction

With more than 30 million presentations created every single day, and 5 exabytes of new information produced every year, one can assume even a small percentage of these presentations will be covering some set of data.

When we look at the field of information visualisation, a lot of the emphasis is set on exploring, discovering, and analysing data. However, in the latest years narrative visualisations are also gaining ground in importance. Tying facts together into a story is one of the most effective ways of presenting them and making a point. Although the research community is only still scratching the surface in this domain, it has already taken the first important steps.

This can also be said for presentation tools. For almost 30 years after the introduction of digital presentation tools, very little has changed in the way we present our ideas to an audience. Even though these tools are used on a daily basis, and in various ways ranging from education to business, to making a presentation in your leisure time. The core idea still remains very much the same. Nevertheless, after Tufte's critiques on PowerPoint, more and more researchers have been evaluating these presentation tools. This resulted in the appearance of new tools, stepping aside from the linear structure and presenter-orient features, introducing an enhancing engagement, and liberating interesting conversations from the boredom of one-way presentations by embracing an audience-driven approach.

It is at this point we must return to the field of information visualisations. As mentioned above, there is some research being done about narrative visualisations, e.g. classifications for patterns and approaches, evaluations of animations, effects of enhancement on memory, etc. However, most of the research deals with visualisations for new media presentations, viewed by many people independently and without the ability to interact with the author or ask questions. Because of their nature and given their message, they seek a balance between an author-driven and reader-driven approach.

INTRODUCTION

This is where we make the connection between new presenting approaches for presentation tools and narrative visualisations. When a narrative visualisation fully embraces a reader-driven approach and is accompanied, physically, by a narrator who then guides the reader through the visualisation. We are, in other words, describing the situation of a live presentation. The balance between author- and reader-driven approach for the visualisation, is then mediated by the narrator. The reader, now an audience-member, is being told a story about the data. Thus covering the author-driven aspect for the visualisation. Parallel to the reader-driven approach, the audience is able to ask questions at any point. To which the narrator may answer accordingly.

The problem with presentation tools, is that they do not allow us to incorporate interactive visualisations. We must therefore be satisfied with standard static visualisations or rely on other applications. Thus leaving one visual area, the presentation, and entering another, a visualisation tool, interrupting the flow of the presentation.

We believe the audience-driven approach for presentations should be pushed into every aspect of the software. Hence, also for visualising data. When this is done correctly, there is no need to switch to other software packages during a presentation. Likewise, it will furthermore enhance the conversation between the presenter and his audience. In addition, it will also resolve some of the issues experienced in current presentation tools with static visualisations, e.g. backup slides with extra visualisations.

In this thesis, we explore the distance covered from information visualisations to narrative visualisations and we take a look at how current presentation tools implement ways to visualise data.

In chapter 1, we start by defining what information visualisation is and look at the history to gain insight into the importance of visualising data.

Chapter 2 will cover interactive visualisations. Here we will discuss the significance of interactions for the formation of a mental model and identify the different interaction and transition techniques. We will also examine the tasks related to information visualisation to support our statement for the need of interactive visualisations in presentation tools.

It is in chapter 3 that we tie information visualisation and presenting together by discussing narrative visualisations. By taking a look at how storytelling is used in visualisation we deepen our understanding in how they also can contribute to presentations.

Chapter 5 introduces us to the MindXpres presentation tool. We explain the information model that is used as an underlying base and describe the plug-in architecture. This plug-in architecture is important to support us in the development of interactive visualisations for presenting.

Chapter 6 describes the actual implementation. We'll go over some of the parts of the architecture and discuss how charts in general are built.

A use case is presented in chapter 7. We set the scene for an example and walk through the presenting process of a dataset, discussing our improvements over existing tools along the way.

Finally, we conclude this thesis with chapter 8 where we discuss our contributions and future work.

By taking a first step in connecting interactive visualisations and presentation tools, we aim to provide a better experience for both the presenter and the audience. We hope that our proposed solution improves the issues we have found and increases the quality of knowledge transfer regarding data.

1

Information Visualisation

In this chapter, we will explore what information visualisation is and how it became to be as we know it now. The structure used in this chapter is one commonly found in books, dissertations, research papers, etc. regarding information visualisation. We therefore do not present any new ideas, however merely cite an overview of established concepts and conclusions. Hereby, we hope to present the reader with an introduction into the field of information visualisation and provide an understanding into the importance for the use of visualisations to represent data.

1.1 A Definition for Information Visualisation

Today's world is being saturated with digital systems that collect gigantic quantities of data, such as cash registers, telephone and computer networks, traffic cameras, and various private and governmental computer systems. According to Lyman & Varian, this resulted in the production of approximately 5 exabytes of new information in 2002 [62]. With a world population of 6.3 billion at the time, this led to 800 megabytes of recorded information produced per person. The equivalent for such an amount of data per person would be 9 meters (30 feet) high were it to be printed on paper¹. Efficient techniques and tools to gain even the smallest insight into a much lesser set of data are evidently required to gain even the smallest insights in a reasonable amount of time.

¹For a more comprehensible idea about sizes in data, table A-1 in appendix A, page 80, provides a comparison of information units with recognisable real life examples.

CHAPTER 1. INFORMATION VISUALISATION

Based on the application focus, the field of data visualisation can be divided into two overlapping areas: Scientific visualisation and information visualisation. The former focuses primarily on physical data referring to, e.g., the earth, the human body, the molecule, DNA, or an aeroplane to which the information can be mapped. While the latter relates to abstract, non-physical data such as text, hierarchies, and statistical information [52, 64, 120].

The above-mentioned classifications are not mutually exclusive and have fuzzy boundaries. For instance, scientific visualisation often involves visualising the multidimensional attribute space of a physical object; this overlaps with information visualisation, which delivers patterns embedded in large-scale information collections [120].

According to Card, Mackinlay and Shneidermann, the only difference in the definitions of the two fields is in the emphasis on the word “*abstract*” when we are talking about information visualisation [16]:

Information visualisation is the use of computer-supported interactive visual representations of ***abstract*** data to amplify cognition.

The distinction between scientific visualisation and information visualisation could be called quite artificial [52], and it has been questioned whether the differentiation is really necessary [86]. Both visualisation communities use a separate but equal approach [75]. Munzner [67] notes that “the sub-field names grew out of an accident of history and have some slightly unfortunate connotations when juxtaposed: information visualisation isn’t unscientific, and scientific visualisation isn’t informative.” The information visualisation and scientific visualisation communities have evolved separately and, for the large part, do not interact [75]. Johnson [52] suggests it might be time to break down the barriers that currently exist between these two communities and work together to solve important problems.

Implied by the definition above, information visualisation is always computer-supported and interactive. Should we then just forget about print format? How would it categorise all the classic information visualisations? Cartographers, working with paper and ink, were the developers for some of the very influential ideas in information visualisation [86].

Both views of information visualisation are quite common. Card et al. [16] regard information visualisation as a computer-related activity only. Siirtola [86] states that the adoption of a wider view should be considered, as do Spence [91] and Ware [111].

The following definition was presented in the foreword of the first IEEE Information Visualization Symposium by Gershon and Eick (1995). This might still be the best characterisation for the field.

Information visualisation is a process of transforming data and information that are not inherently spatial, into a visual form allowing the user to observe and understand the information.

This definition covers the time before computers, notes the extension to consider abstract data, and declares a goal of obtaining insight into data [86]. “The purpose of visualisation is to gain insight, not pictures” [16]. “In general, the notion of insight is broadly defined, including unexpected discoveries, a deepened understanding, a new way of thinking, eureka-like experiences, and other intellectual breakthroughs” [18].

1.2 Data Versus Information

When looking at the dissimilarity of data and information, a subtle distinction can be noticed. Generally, data is considered to refer to facts, collected together for analysis or reference. Information on the other hand, is the interpretation of data [12]. In other words, data is a “resource” for information, resulting the transformation to be subject- and context-dependent. [86]. Making one’s data, another’s information.

Depending on the context and purpose, a difference between the two can be irrelevant. Hence, adoption of these concepts can be used interchangeably. Due to the nature of this thesis, we therefore, likewise, do not differentiate between the two.

1.3 The History of Information Visualisation

Pinpointing the very first time when information visualisation techniques were applied is impossible. However, information visualisation has a quite long history. The first visualisation was likely created by a caveman, drawing something in the dirt or on the wall of his cave.

Through recorded history, there are some well-documented applications of information visualisation approaches that have withstood the test time. Though, the concept of information visualisation was introduced quite recently.

As a research area, information visualisation was derived from several sources: the statistical data graphics, human-computer interaction, psychology, artificial intelligence, scientific visualisation, and computer graphics communities were all influential. The roots of information visualisation reach deep into the history of data graphics. And more recently, have stretched far into computing and human-computer interaction. Following is a list of some selected milestones in these areas [17, 18, 32, 34, 35, 86, 120].

CHAPTER 1. INFORMATION VISUALISATION

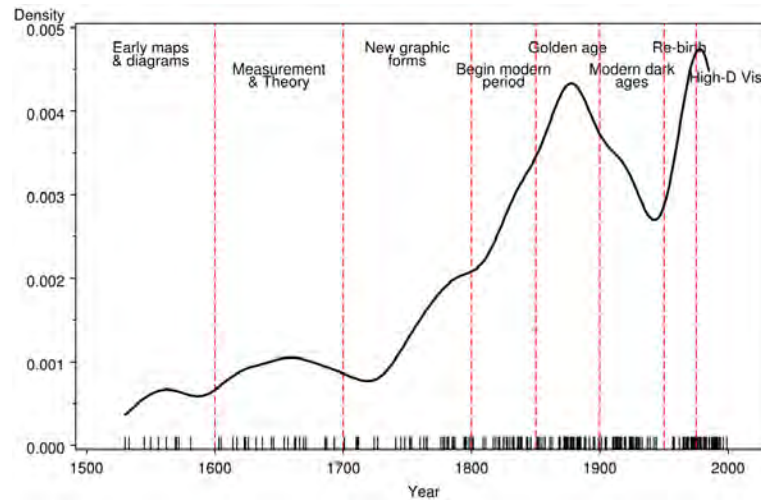


Figure 1-1: The time distribution of events considered milestones in the history of data visualisation, shown by a rug plot and density estimate. (Source: Friendly [34])

Pre-1800: Pioneers

The earliest seeds of visualisation arose in geometric diagrams, in tables of the positions of stars and other celestial bodies, and in the making of maps to aid in navigation and exploration. The idea of coordinates was used by ancient Egyptian surveyors in laying out towns. Earthly and heavenly positions were located by something comparable to latitude and longitude, at least by 200 BC. The map projection of a spherical earth into latitude and longitude by Claudius Ptolemy² in Alexandria would serve as reference standards until the 14th century.

The oldest preserved table was created in the 2nd century in Egypt to organize astronomical information as a tool for navigation. A table is primarily a textual representation of data, but it uses the visual attributes of alignment, white space, and at times rules (vertical or horizontal lines) to arrange data into columns and rows. Tables, along with graphs and diagrams, all fall into the class of data representations called charts. Although tables are predominantly textual, their visual arrangement of data into columns and rows was a powerful first step toward later developments, which shifted the balance from textual and visual representations of data.

²Claudius Ptolemy (*c. 100–†c. 178) was a hugely important geographer and astronomer working in Ancient Rome. His work informed mapmakers on the size of the Earth, and the co-ordinates for the positions of all the places and features indicated on the map. A map based on Ptolemy’s book *Geographia* can be found in appendix A on page 82.

In the Middle Ages, the coordinate plane was used as a field of operation for the study of curved lines. In 1637, Descartes³ based the idea of analytic geometry on coordinates on a plane, resulting in a system now known as the *Cartesian coordinate system*.

While this method of representing quantitative data was originally invented for performing a type of mathematics based on a system of coordinates, and not for presenting. Later, however, scientists soon began to display empirical data by graphing it.

In 1786, William Playfair⁴, widely considered the inventor of most of the graphical forms generally used today, developed several data representation systems. First the line graph and bar chart, later the pie chart and circle graph also known as “*Playfair’s circles*”.

1800–1850: Beginnings of modern graphics

With the fertilization provided by the previous innovations of design and technique, the first half of the 19th century witnessed explosive growth in statistical graphics and thematic mapping at a rate which would not be equalled until modern times.

In statistical graphics, all of the modern forms of data display were invented or further developed: bar and pie charts, histograms, line graphs and time-series plots, contour plots, scatter plots, and so forth. In Thematic cartography, mapping progressed from single maps to comprehensive atlases, depicting data on a wide variety of topics; economic, social, moral, medical, physical, etc., and introduced a wide range of novel forms of symbolism. During this period graphical analysis of natural and physical phenomena; lines of magnetism, weather, tides, etc., began to appear regularly in scientific publications as well.

In October 1831, the first case of Asiatic cholera occurred in Great Britain, and over 52 000 people died in the epidemic that came to pass over the next 18 months or so. Subsequent cholera epidemics in 1848–1849 and 1853–1854 produced similarly large death tolls, but the water-born cause of the disease was unknown until 1855, when Dr. John Snow⁵ produced his famous dot map (Figure 1-2) showing deaths due to cholera clustered around the Broad Street pump in London.

³René Descartes (*1596–†1650) was a French philosopher, mathematician and writer who spent most of his life in the Dutch Republic. Much subsequent Western philosophy is a response to his writings, which are studied closely to this day.

⁴William Playfair (*1759–†1823) was a Scottish engineer and political economist, the founder of graphical methods of statistics. One of Playfair’s most well-known charts can be found in appendix A on page 82.

⁵John Snow (*1813–†1858) was an English physician and a leader in the adoption of anaesthesia and medical hygiene. Considered as one of the fathers of modern epidemiology.

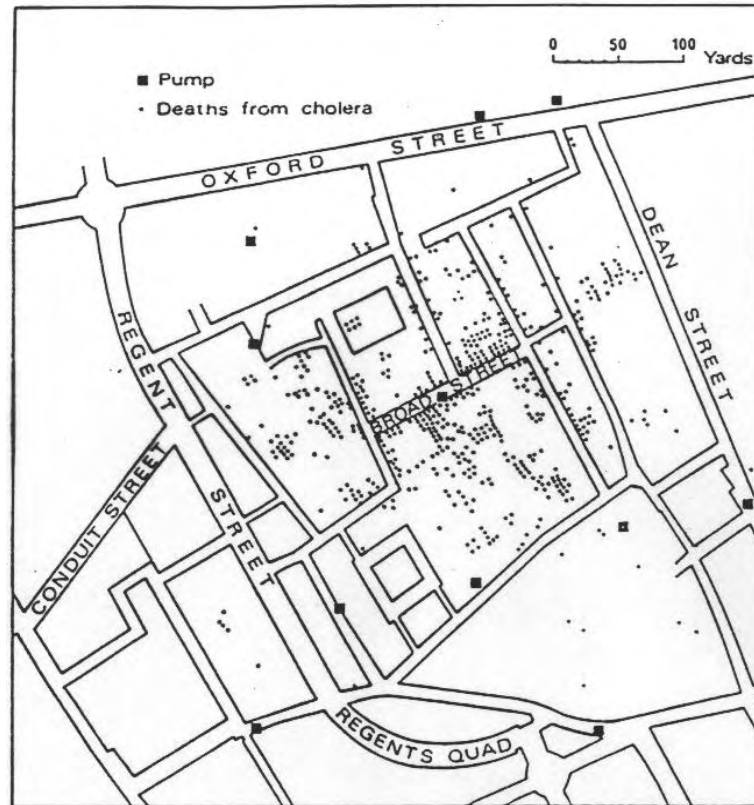


Figure 1-2: The Cholera Map by Dr John Snow shows the geographic distribution of death by Cholera.

By doing so Snow had taken quantitative data and made it visible. By visualising information Snow revealed that all but 10 of the 83 deaths were within close vicinity of the Broad Street pump. He then set about explaining the anomalies.

1850–1900: The Golden Age of statistical graphics

Industrialisation and state statistical offices for social planning throughout Europe fuelled the need to visualise the wealth of numerical data. At the same time, significant advances in both statistical theory and methods were made by Gauss⁶ and Laplace⁷.

⁶Johann Carl Friedrich Gauss (*1777–†1855) was a German mathematician, who contributed significantly to many fields, including number theory, algebra, statistics, analysis, differential geometry, geodesy, geophysics, electrostatics, astronomy and optics.

⁷Pierre-Simon Laplace (*1749–†1827) was a French mathematician and astronomer whose work was pivotal to the development of mathematical astronomy and statistics.

Florence Nightingale⁸ invented polar area charts (1858), known as “Nightingale’s roses” (Figure A-3, p84), to document and visualize the sanitary conditions of the British army.

Minard⁹ (1869) constructed the classic flow visualisation of Napoleon’s ill-fated campaign to conquer Moscow (Figure 1-3). Tufte calls this map “the best graphic ever produced” [102], and Funkhauser gives Minard the designation “the Playfair of France” [36].

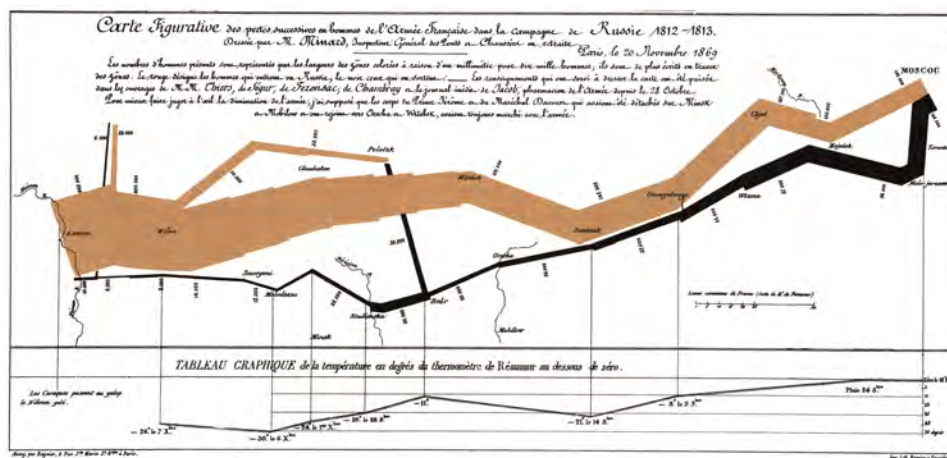


Figure 1-3: Minard's map of Napoleon's Russian campaign

What started as the *Age of Enthusiasm* [71] for graphics ended with what can be called the *Golden Age*, with unparalleled beauty and many innovations in graphics and thematic cartography [35].

1900–1950: The modern dark ages

Friendly & Denis [35] characterize this period as mainly dormant, although now statistical graphics were becoming mainstream and entered textbooks. There were few innovations in graphics, and the emphasis scientific methods was on formal and statistical models of phenomena.

There is one bright landmark in information visualisation in this era. In 1933, Henry C. Beck¹⁰ designed a new version of the London Underground

⁸Florence Nightingale (*1820–†1910) was a celebrated English social reformer and statistician, and the founder of modern nursing.

⁹Charles Joseph Minard (*1781–†1870) was a French civil engineer noted for his inventions in the field of information graphics.

¹⁰Henry Charles Beck (*1902–[†]1974) was an English technical draughtsman.

Diagram (Figure A-4, p84). It is perhaps one of the most imitated information visualisation designs in the world. Though the original still has “unsurpassed visual distinction and proven usefulness” [40].

1950–1975: Re-birth of data visualisation

In this period, the first mass-produced computer, the IBM 650, arrived, and computers with rudimentary graphics became available to scientists. Several new graphical ideas for representing multidimensional data were introduced. Anderson proposed circular glyphs with outward-pointing “rays” [5] (Figure A-5, p85). Siegel, Goldwyn, and Friedman [84] proposed, still widely used, star-shaped glyphs (Figure A-6, p85). Chernoff [21] introduced the more controversial but interesting idea of using cartoon faces to represent multivariate data (Figure A-7, p86).

Tukey¹¹ began the work on *Exploratory Data Analysis* [104], producing a wealth of ideas concerning how to carry out statistical analysis visually. One of the results was PRIM-9, the first statistical system that could perform interactive 3D rotations and allow interaction with multidimensional data in up to nine dimensions.

1975–2000: The beginning of modern visualisation

During the last quarter of the 20th century, information visualisation has blossomed into a mature, vibrant and multi-disciplinary research area. However, it is hard to keep the overview brief when it comes to recent developments in information visualisation. This because they are so varied, have occurred at an accelerated pace, and across a wider range of disciplines. It is also more difficult to highlight the most significant developments, that may be seen as such in a subsequent history focusing on more recent periods.

Cartographer Jacques Bertin¹² published the first theory of graphical symbols and modes of graphical representation in his book *Graphics and Graphic Information Processing* [10]. The focus in Bertin’s work was on developing a general theory of graphics for cartographers, but, as a by-product, he also developed the first interactive visualisation method for multidimensional data, the reorderable matrix.

In 1983 data visualisation aficionado Edward Tufte published his groundbreaking book *The Visual Display of Quantitative Information* [102], which showed us that there were effective ways of displaying data visually and then

¹¹John Wilder Tukey (*1915–†2000) was an American mathematician best known for development of the FFT algorithm and box plot.

¹²Jacques Bertin (*1918–†2010) was a French cartographer and theorist.

there were the ways that most of us were doing it, which were sadly lacking in effectiveness. After this Tufte published two more books on design of graphics and information displays [101, 103].

One year later, in 1984, Apple Computer introduced the first popular and affordable computer that focused on graphics as a mode of interaction and display. This paved the way for the use of data visualisations that we could view and interact with using a computer.

The transition from vector-based graphics to bit-mapped displays made new kinds of ideas easier to implement. The use of image distortion as a visualisation technique was developed independently by Furnas in semantic and graphical *fish-eye views* [37, 38] (Figure A-8, p87) and by Spence & Apperley in the *Bifocal Display* [90]. These ideas later were generalized to cover a wide range of different distortion functions.

Inselberg and Dimsdale [51] developed a new approach called *parallel coordinates* to visualize multidimensional data in a manner that allows a wide variety of interactions. Originally, parallel coordinates were developed to take computational geometry into higher dimensions, but the possibilities in the field of interactive visualisation were soon discovered.

In 1999, at the end of this period, the first textbook on information visualisation appeared: *Information Visualization – Using Vision to Think* by Card et al. [16].

2000–present: Becoming discipline and commodity

The new millennium brought two novel textbooks on information visualisation, by Ware [110] and Spence [88]. Courses on information visualisation became more generally available at universities. At the time of writing, a course, fully dedicated to information visualisation, was given for the first time at the Vrije Universiteit Brussel (VUB).

Furthermore the first large-scale commercial success, *Spotfire* (2006) [1], appeared.

The price of hardware decreased steadily, and a standard PC has sufficient 3D graphics and texture mapping capability to produce complex visualisations.

To summarize

When we look back at the visualisation time line, there have been static 2D drawings of information for the past 500 years, 3D computer graphics for about 40 years, scientific visualisations for about 20 years, and interactive, computer-augmented information visualisation for about 15 years.

1.4 Visualisation Types

In the previous section we have already come across a couple of visualisation types. Shneiderman defines a taxonomy of seven data types [82] reflecting an abstraction of the reality. Many variations and/or combinations on these themes are possible.

Linear: 1-dimensional data types include textual documents, program source code and alphabetical lists of names which are all organized in a sequential manner. Interface design issues mostly include what fonts, colour or size to use. These are commonly retained in text and not visualised.

Planar: map data, 2-dimensional, include geographic maps, floor-plans or newspaper layouts. Each item in the collection covers some part of the total area and may be rectangular or not. Many systems adopt a multiple layer approach to dealing with map data, each layer is 2-dimensional.

Examples: choropleth, cartogram, dot distribution map, proportional symbol map, contour map, dasymetric map, self-organizing map.

Volumetric: 3-dimensional, real-world objects, such as molecules, the human body and buildings, with volume and some potentially complex relationship with other items.

Examples: 3D computer models, surface and volume rendering, computer simulations.

Temporal: items have a start and finish time and may overlap. A distinction is made with time lines, which are widely used to create a data type that is separate from 1-dimensional data.

Examples: time line, time series, connected scatter plot, Gantt chart, stream graph, arc diagram, polar area, sankey diagram, alluvial diagram.

Multidimensional: items with n attributes become points in a n -dimensional space. The interface representation can be 2-dimensional scatter-grams, with each additional dimension controlled by a slider, or by a 3-dimensional scatter-gram but disorientation and occlusion can be problems.

Examples: pie chart, histogram, tag cloud, bubble cloud, bar chart, tree map, scatter plot, bubble chart, line chart, step chart, area chart, heat map, parallel coordinates, radar chart, box plot.

Hierarchical: hierarchies or tree structures are collections of items with each item having a link to one parent item (except the root). Items and the links between parent and child can have multiple attributes.

Examples: general tree visualisation, dendrogram, radial tree, hyperbolic tree, tree map, sunburst, partition chart.

VISUALISATION TYPES

Network: relationships among items cannot be conveniently captured with a tree structure and it is useful to have items linked to an arbitrary number of other items. Many special cases exist: acyclic, lattices, rooted vs. unrooted, directed vs. undirected.

Examples: matrix, node-link diagram, dependency graph, hive plot, alluvial diagram, subway map.

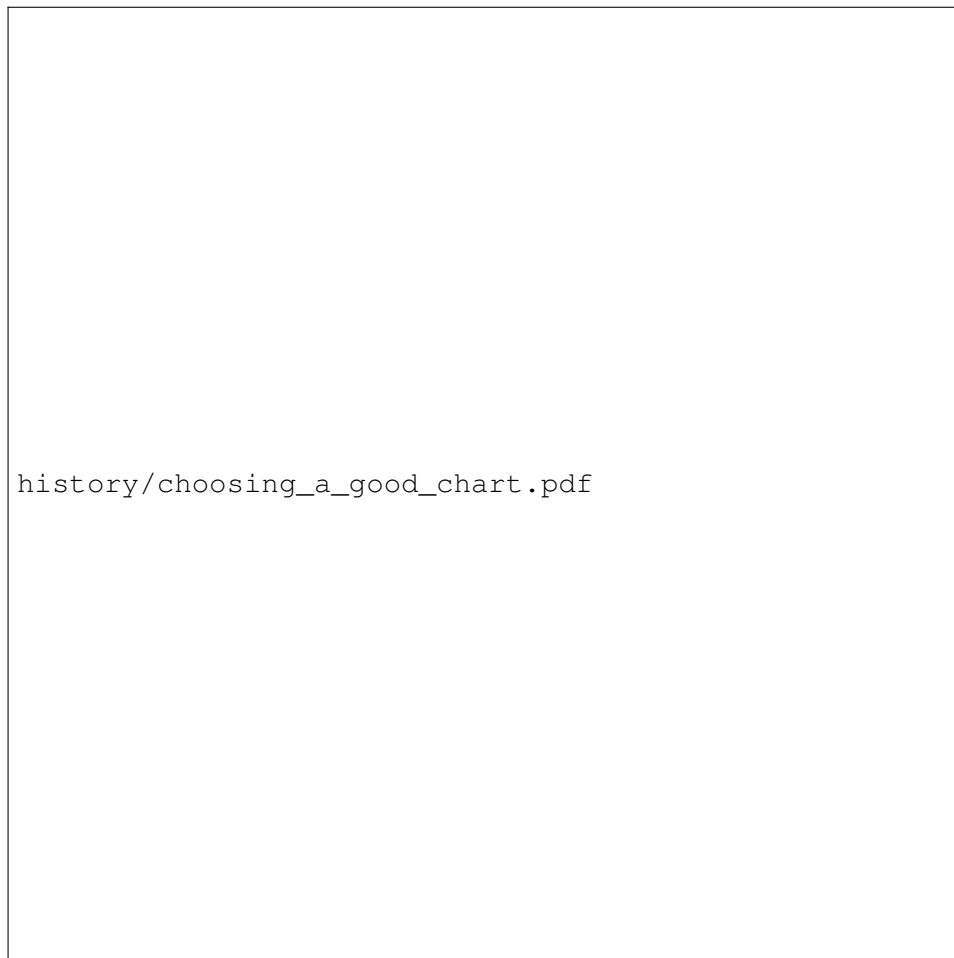


Figure 1-4: Chart chooser. Created by Dr. Andrew Abela, professor of marketing and presentation design consultant.

2

Interactive Visualisation

An important aspect of the information visualisation process concerns the support for forming a mental image of data. This mental image can not be restricted to at most a schema or a structural description of the data. It is an *insight* into the potential story behind the data, basically transforming a clump of data into information. The formation of this mental image can be augmented by allowing the user to interact with the data [111].

2.1 Interaction

The significance of interaction in information visualisation is assessed to be substantial. Gibsons's cookie-cutter experiment [42] is often used as a classic example to prove the relevance of interaction in information acquisition. Participants had to recognize the shape of a cookie-cutter in three different conditions:

- passive: the cutter placed on their hand without movement,
- passive rotation: the cutter rotated after placement,
- active: free interaction with the cutter.

Recognition rates were 49%, 72% and 96%, respectively. Even though the experiment was focusing on haptic touch. Nonetheless, it still indicates a relevance into the importance of interactions for information acquisition tasks. The differentiation between *passive* and *active interaction* made by Gibson has a similarity in the area of visualisations. A visualisation is using passive interaction whenever it does not allow any mode of interaction other than watching [86].

CHAPTER 2. INTERACTIVE VISUALISATION

The distinction between *discrete* and *continuous interaction* is another important one [89]. Common examples for discrete (or stepped) interactions are making a menu selection or following a hyper-link, i.e. a user’s action produces a discrete system response.

Continuous interactions on the other hand, are a special form of discrete interactions. The user’s actions produces a series of system responses. The interaction is observed as continues when the output flow blends into one continuous percept. This interaction mode is important for direct-manipulation (or manual) interfaces. Where, in general, the illusion of being in direct contact with the data is pursued [81]. Both interaction modalities, discrete and continuous, are implemented in the same interface for the majority of information visualisation user interfaces.

Shneiderman [82] observed a reoccurring pattern in designing user interfaces for information visualisations. This pattern, widely known as the “*Visual Information Seeking Mantra*”, consisting of three basic steps: “Overview first, zoom and filter, then details-on-demand”, does not on its own guarantee a decent user interface. Though failure to implement its directive will almost certain produce problems. The mantra has been criticized for a lack of empirical verification and for its apparent high-levelness [25].

Ware [111] models the interaction in information visualisation as three loops. The **problem solving loop**, highest level, models the human memory system, attention, and the low-level functionality of the human eye. This loop captures the processes of forming, refining and falsifying hypotheses about the data. It involves combining several activities to accomplish cognitively more demanding tasks, e.g. laying out pieces of information for finding relations. At the intermediate level, the **exploration loop**, models movement in information space with analogies and metaphors of physical navigation. The user combines basic low-level operations of exploration and navigation of large visual data spaces, e.g. adjusting the visual encoding or visiting different parts of the data. The **low-level interaction loop** contains data selection and manipulation. Interaction is concerned with basic operations of recognising, pointing at, or manipulating objects, e.g. moving the mouse and performing a click to pinpoint a data item of interest. [86, 99]

For the rest of this chapter, we focus on the interaction techniques for data selection and manipulation we deem relevant to the subject of interactive visualisations in the context of presenting information and telling stories about the data.

2.2 Interaction Techniques

Interaction techniques in information visualisation can be seen as the features that provide users with the ability to, directly or indirectly, manipulate and interpret representations. Complying to this view concedes with the idea that static images or autonomously animated representations do not have associated interaction techniques. However menu interfaces, for changing from a scatter plot to a parallel coordinates view, or sorting a bar chart in descending order by their values, are also interaction techniques since they allow users to manipulate a representation, even though they may be less interactive or direct. [118]

Several studies have proposed taxonomies relevant to the examination of interaction techniques. Despite many of these studies share common units, the taxonomies have significantly different levels of granularity to describe interactions. A selection of these studies, with their defined units, can be found in table A-3 in appendix A on page 83.

This section presents Siirtola’s [86] overview of interaction techniques, comparable to the categories proposed by Yi et al. [118].

Based on the notion of user intent, this sort of overview conforms with the “task by data type” taxonomy of Shneiderman [82]. Shneiderman’s taxonomy lists seven *information actions* that users wish to perform:

- **Overview:** a view of the total collection.
- **Zoom:** a view of a single item. This may be either at the object or attribute level.
- **Filter:** removing unwanted items from the displayed set.
- **Details-on-demand:** getting the detail of a selected group, sub-group or item.
- **Relate:** viewing the relationships between a selected group, sub-group or item.
- **History:** the actions of undoing, replaying and refining using a store of historic information.
- **Extract:** the extraction or focusing in on sub-collection and other parameters from a given set.

The overview action is very technique-specific. Essentially many visualisation techniques *are* overviews, though some require to implement it separately. This by the use of focus and context views, or detail and overview displays. The history and extract tasks, on the other hand, are at a different level, and perhaps something expected in more modern applications.

CHAPTER 2. INTERACTIVE VISUALISATION

Next to Shneiderman’s task by data type taxonomy, Dix and Ellis [28] emphasize on two important principles in interacting with visualisations.

The first, “*same representation, changing parameters*”, meaning interactive change of some parameter of the presentation. Examples for implementing this principle are VICKI [26] and Spotfire [1].

The second principle, “*same data, changing representation*”, means switching between conceptually different displays of the same data. Various representations can be appropriate for different types of data, and each representation needs to be tuned for its purpose. Examples are systems with multiple coordinated representations for the same data, Spotfire and Ggobi [94].

After discussing the different interaction techniques we will see these concepts return by means of transitions.

2.2.1 Select and Highlight

Select and highlight is the most simple interaction on a visual representation of a set of objects [115]. By highlighting the selected information, focus is set on the subset of data, allowing visual comparison between the subset of interest and other objects. Most common methods used are changing colour and shape of the selected objects.

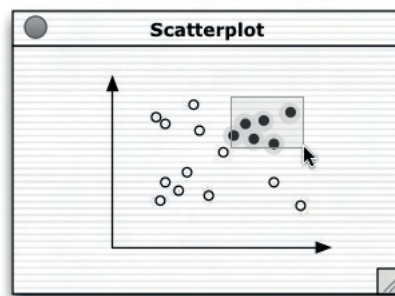


Figure 2-1: Selecting and highlighting a set of objects. (Source: Siirtola [86])

2.2.2 Brush and Link

Brushing and linking builds upon select and highlight, with the distinction that the same set is also selected in a number of linked views [9]. In a special case, the brush and link within one view, reduces to selecting and highlighting. Brush operations may also manipulate existing selections in different ways, e.g. replacing, adding, subtracting, intersecting, or toggling the state [115]. Brushing and selecting can also be compound operations to build more complex selection by use of logical connectors from a prior selection [108].

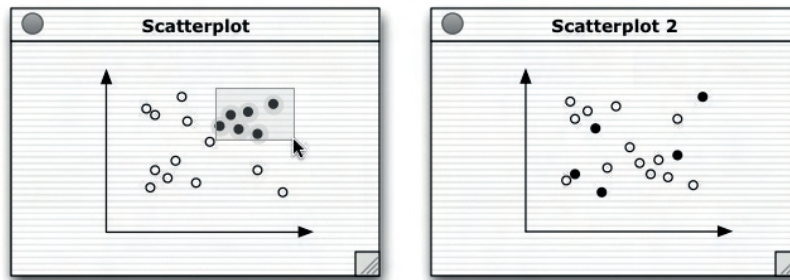


Figure 2-2: Brushing: selecting a set of objects in one view and highlighting them in another. (Source: Siirtola [86])

Brushing technique variations for specific visualisation methods have appeared as well, e.g. angular brushing for parallel coordinate visualisations [44]. A generalisation for compound brushing based on hi-graphs where various components can be linked together via logical operations and expression was presented by Chen [19].

2.2.3 Reorder

Reordering [88] is a very natural interaction operation, in which a set of objects is arranged for easier processing.

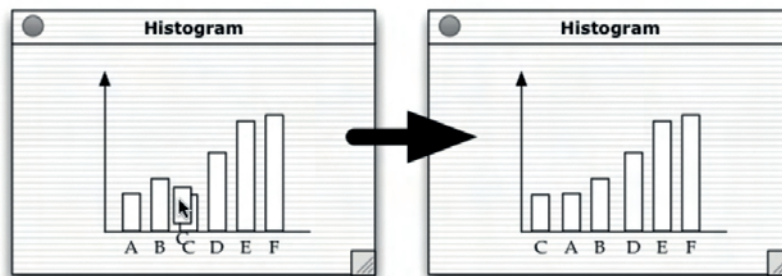


Figure 2-3: Reordering a set of objects for a consistent pattern. (Source: Siirtola [86])

Rearranging objects can be, a simple sort operation based on attribute values, a subset, or their characteristics [11], a manual operation, or driven by some algorithm. Automatic data reordering can reduce clutter, although the process is very method-dependent [73].

2.2.4 Query

Because of the lack of space and the quantity of items being displayed, details are often unable to be shown in a visualisation. A query functionality, direct or indirect, can provide detail on demand. Likewise, when the number of objects being visualised has been trimmed down, details can appear.

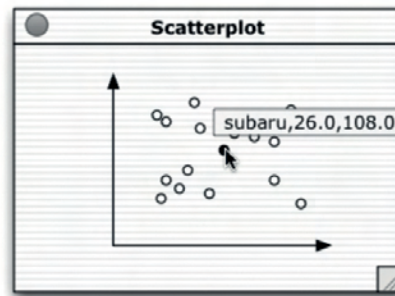


Figure 2-4: Querying an object for more detail. (Source: Siirtola [86])

2.2.5 Filter

Reducing the number of objects being visualised is done with a filtering technique. A reduction can be accomplished indirect, via a control (e.g. a slider), or direct by manipulation (e.g. selection).

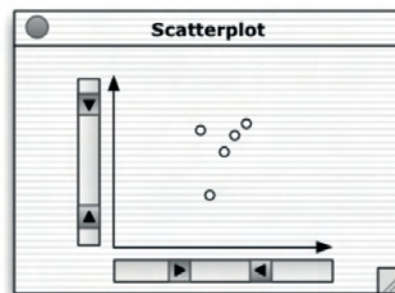


Figure 2-5: Filtering a set of objects (Source: Siirtola [86])

2.2.6 Zoom

Applications with graphical user interface commonly implement zooming and panning. Panning provides for operation on larger information spaces than prohibited by the screen real estate. Zooming in and out allows to view details or the whole information space.

By using two coordinated views, it is possible to see both overview and detail in an effective way, provide that the ratio between views (zoom factor) is not too large (Shneiderman [83] suggests between 5 and 30).

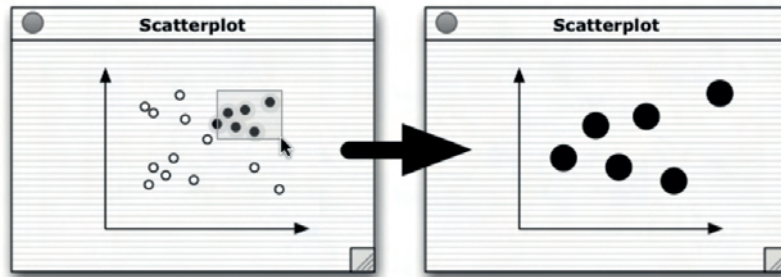


Figure 2-6: Zooming in on a set of objects. (Source: Siirtola [86])

According to the level at which objects are viewed, appearances of the items of interest can be changed accordingly. Causing zoom to be a semantic operation.

2.2.7 Abstract

Abstracting is a common technique to cope with complexity. As Miller [65] noted in his classic survey, there is a limit to the number of items we can comprehend at one time, about seven, plus or minus two.

Bypassing this limit can be achieved through abstraction, by creating higher-level “chunks” of information with greater semantic content.

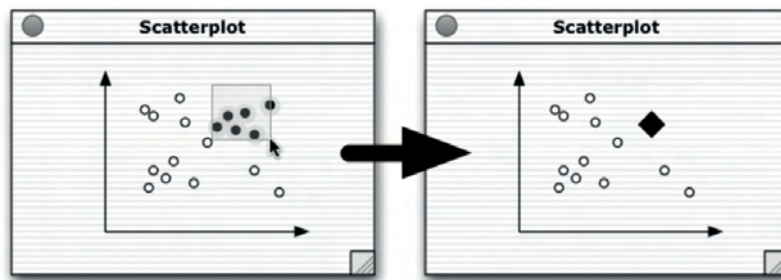


Figure 2-7: Abstracting a set of objects. (Source: Siirtola [86])

Abstraction can be used in a large number of fashions, even though visualisations are already abstractions themselves. Lowering the number of data item being dealt with, can be done by abstracting groups of cases or their variables into higher-level objects.

2.3 Transitions

“Transitions between graphics can be modelled as state changes” [47]. Interactions change the semantic model of the data graphic. Consecutively resulting in changes to the graphical syntax. While, static transitions simply replace the original form with the new one, “The challenge of designing animations is to visually interpolate the syntactic features such that semantic changes are most effectively communicated.” [47]

To better inform the design of animated transitions, Heer & Robertson [47] crafted a taxonomy of the various types of transitions between data graphics. They identified the following transition types by considering the syntactic or semantic operators one might apply to a data graphic.

2.3.1 View Transformation

Changes in viewpoint are expressed by view transformations. This type of transformation is often modelled as the movement of a camera through a virtual space. The view transformation operator is purely syntactic, thus schemas and visual encodings do not change.

2.3.2 Substrate Transformation

Changes to the spatial underlying layer, in which marks are embedded, are handled by substrate transformations. E.g. axis rescaling, log transforms, bifocal and graphical fish-eye distortions.

2.3.3 Filter

Filter transitions apply a predicate specifying which elements should be visible. Visible items are then added or removed from the display.

Filtering does not change visual encodings or data schemas, though a substrate transformation such as axis rescaling may be desired.

2.3.4 Order

Spatially rearranging ordinal data dimensions are done with ordering transitions. E.g. sorting on attribute values or manual reordering.

2.3.5 Time-step

Time-step transitions apply temporal changes to data values. Aside from the sample point from which data is drawn, the data schema does not change. E.g. The transition between sales figures for current and previous years. For some changes of value, axis rescaling may be desirable.

2.3.6 Visualisation Change

Visualisation transitions consist of changes to the visual mappings applied to the data.

E.g. Data represented in a bar chart may instead be represented in a pie chart, or changes made in colour, size, or shape encodings.

2.3.7 Data Schema Change

Data schema transitions change the data dimensions being visualised. E.g. Starting from a bar chart showing only one variable. Adding an additional data column, resulting in multiple possibilities for bivariate graphs.

These transitions may be accompanied by changes to the visual mappings. The resulting bivariate graph may be presented as a stacked or grouped bar chart, a scatter plot, etc.

The schema changes may be orthogonal, in which an independent dimension is added, removed, or nested. The schema change then traverses a hierarchical relation between dimensions of the data table, e.g. roll-up and drill-down operations.

2.4 Tasks

The process of information visualisation is always used as a means to an end. For this, it is important to understand and support the most common user tasks and goals related to them. The following goals of information visualisation were identified by Hearst [45]:

- Making large and complex datasets understandable
- Presenting information from various viewpoints
- Presenting information at several levels of details
- Supporting visual comparisons
- Telling stories about the data

Attaining an overview of a dataset more easily can be done by presenting a dataset compactly. Though, reading these visual presentations serially, as one would with textual information, will not exploit the built-in parallelism of the human visual system [111].

Akin to using multiple viewpoints to rhetorically argue something, one can use multiple points of view in information presentations. Interactions, e.g. panning, zooming, or filtering, are used to create a multitude of viewpoints in visualisations.

Adjusting the amount of detail uses a similar approach. Essentially one of the standard human strategies for dealing with complexity is abstraction, as mentioned in section 2.2.7. According to the task at hand, the abstraction level is adjusted.

CHAPTER 2. INTERACTIVE VISUALISATION

Supporting visual comparison, again, builds on human perceptual capabilities with the use of appropriate visual representations.

Finally there is the task of telling the story behind a dataset. A challenge we will be focusing on in this thesis. Information visualisation techniques, as the ones discussed in sections 2.2 and 2.3, can help us hereby. They can be very effective for a presenter to guide the observer through the dataset being visualised. “It makes the data come to life, and emphasizes the critical results of an analysis.” [76]

The need for general data analysis, reasoning about data, explaining and communicating the information, or some other requirement to gain insight, are objectives at the highest level for visualisations. These tasks can be categorized in 3 types: *exploratory*, *confirmatory*, and *presentational*.

When talking about exploratory tasks, a well-defined goal for actions can not be determined. As a substitute, new sub-goals arise while the process advances. Confirmatory tasks consider the refutation or confirmation for a stated hypothesis for the data. The apparent goal is to test a hypothesis about the data. Presentational tasks ultimately have the goal to find the visual representation that is most effective for communicating the facts. [86]

Wilkins [112] translates these higher-level goals into more concrete user-level tasks:

- Finding items
- Looking at item detail
- Comparing items
- Discovering relationships between items
- Grouping or aggregating items
- Performing calculations
- Identifying trends

The close connection between these user-level task types and the interaction techniques appears to be self-evident.

Zooming and filtering support finding items. Looking at items in detail can be done by querying them, and comparisons are facilitated by highlights and linked views. Building groups or aggregating items are permitted by abstractions, and trends can be explored through reordering. The only task type, not directly supported in the interaction types, is the execution of calculations.

For presenting a dataset, these user-tasks reflect to the intentions a presenter may have to clarify a message, insight, or demonstrate a point being made. The interactions can be used as instruments to shape and support the narration of the underlying story behind the data being visualised.

3

Narrative Visualisation

As seen in previous chapters, the focal point for visualisation research has traditionally been the exploration and analysis of data. Initially, a lot of attention was aimed at novel techniques. This in turn led to the pursuit on which approach to use for which task. Giving rise to evaluation papers, comparing approaches, and investigating cognitive methods behind specific techniques.

Even though more research might be needed in this area, at the moment, there is an adequate comprehension for the design space of visualisations, leading to a competence for employing suitable techniques for most datasets and tasks. [57]

As mentioned, these techniques cover the exploration and analysis of data. However, means for presenting and communicating data have only recently been gaining the much needed attention they deserve.

“Tying facts together into a story is one of the most effective ways of presenting them and making a point.” [57]

The main reason for bringing stories into the equation, is the fact they demonstrate to be an incredibly popular way of conserving information and passing it on.

People have been passing information along via storytelling for as long as humans have had a rich language to draw from.

Not only do they preserve and advertise information, they also perform as an adhesive between facts to make them memorable [6].

3.1 Story Definition and Model

In the Oxford English Dictionary a *narrative* is defined as “an account of a series of events, facts, etc., given in order and with the establishing of connections between them.” Kosara & Mackinlay [57] “define a story as an ordered sequence of steps, with a clearly defined path through it. Each step can contain text, images, visualizations, video, or any combination thereof.”

For both definitions, *order* is an important characteristic. In traditional literature¹, order more or less conforms with the chronological sequence of events as they actually occurred in the time-space universe of the narrative being told. This is crucial to understand causality: occurrences in a earlier moment of time may influence subsequent events, though not the other way around [6]. By breaking up the story in segments, using analeptic references, prolepses², or other literary devices, stories can be told in a non-linear fashion. Still, within each segment, the order must remain consistent. Additionally, to make the story comprehensible, the chronological order of the various segments needs to be apparent. [57]

The working model presented by Kosara & Mackinlay, is based on the way journalists work [80]. Through research, interviews, etc., information is collected and put into essential facts. These facts are then connected into a story. By *not directly* using any of the source materials, e.g. data collected, recorded interviews, the composed story may look very different. Because goals differ during the research and writing phases, so will the tools. Materials from the research phase, e.g. quotes, pictures or pieces of video, might end up in the final story. Though most of the resources only serve as the raw material for the written piece.

In the current model, visualisation is used for both the exploration and the presentation of data. However, the manner in which they are used can be very contrasting. Choices for techniques will vary, as does how much and which data is shown. Visualisation researchers often assume that the tools used for analysis are usable for presentation just as well as for their original purpose. Kosara & Mackinlay [57] state this to be a very limiting assumption, which we can concur with. Robertson et al. [76] likewise conclude the value, for analysis as for presentation afterwards, no longer to be equal.

¹Traditional literature consists of stories that have been passed down orally and in written form throughout history. Traditional literature is found in cultures throughout the world reflecting the norms of it's people.

²Analepsis and Prolepsis, commonly referred to in film as “flashback” and “flashforward” respectively.

3.2 Storytelling in Visualisation

When looking back to the history of information visualisation, chapter 1.3, notice the focus on exploration and analysis in its 20-odd years of existence. Nevertheless, the earliest examples of data visualisation were created, not to analyse data, however to show and explain their meanings.

Florence Nightingale did not use her charts to analyse data about the causes of death in the Crimean War. She first made her analysis, and then employed her visualisation to make the numerically illiterate politicians of the time aware of the size of the problem.

Likewise, Dr. John Snow's map of Cholera was not used to identify the water-born cause of the disease. The map was produced to present the evidence after identifying the most likely cause, using means other than visualisation. In the same way, Charles Minard's famous map of Napoleon's march on Moscow is primarily meant as a medium for telling the story of the soldiers' difficult circumstances, and less as an analytic tool for understanding complex data. [57]

Going forward to more modern visualisation literature, the earliest work on storytelling, in regards to information visualisation, is a paper by Gershon & Ward [41], describing the use of storytelling techniques to show the development of a hostage situation. While the paper describes many interesting points about the power of storytelling, it never explains any actual visualisations. This due to it mostly being based on map view without numerical data. However, stories are argued to be a good means to fill the need in communicating key information in a brief and understandable manner.

Segel & Heer [80] on the other hand, classified the patterns and approaches used by news media to tell stories visually. Identifying a number of approaches that are commonly used, including different layouts and semantic story structures. Likewise, Hullman & Diakopoulos [50] identified a number of approaches to communicate authority and completeness of data in their exploration in the role of rhetoric in narrative visualisation. More importantly, it also shows how these cues can be used to prioritize particular interpretations.

Robertson et al. [76] evaluated the effectiveness of animation trend visualisation, for presentation as well as analysis. In particular the type used by Gapminder³, concluding its efficiency to be limited. However, Gapminder does show off the capability of animated transitions for explaining visualisations and getting people interested in the data [57]. In general, a presentation's goals is to get a point across and have the audience remember it. Considering the impact a visualisation has on recollection, Bateman et al. [7] studied

³Gapminder. <http://www.gapminder.org>

CHAPTER 3. NARRATIVE VISUALISATION

the effects of enhancement on memory and found more embellished charts, *infographics*, to be easier to remember. As with stories, embellishments add context to the presented information, making it easier to remember and recall. Lee et al. [60] employ this by the use of icons and other visual elements that are customized to the dataset being visualised.

When used in a collaborative setting, telling stories about data is a normal consequence [57]. In a more structured context [106, 114], stories can be used not only to support discussion and decision-making, but also the analysis process. Hence, stories can serve as part of the source for a finding, similar to the narrated history of an event [57].

3.3 Visual Narratives

Storytellers, especially on-line journalists, have increasingly been integrating visualisations into their narratives, in some cases allowing the visualisation to function in place of a written story. As mentioned above, Segel & Heer [80] conducted a study covering news media to visualisation research, identifying distinct *genres*, *visual narrative tactics* and *narrative structure tactics* of narrative visualisation.

3.3.1 Genres

Narrative visualisation can be characterized by 7 basic genres (Figure 3-1):

- Magazine style
- Annotated chart
- Partitioned poster
- Flow chart
- Comic strip
- Slide show
- Film, video or animation

These genres vary primarily in terms of the number of frames that each contains, distinct visual scenes, multiplexed in time and/or space, and the ordering of their visual elements. They are not mutually exclusive and can be combined to produce more complex visual genres.

Each of them can be used to tell a story, though different genres work better for different story types. Choosing the appropriate genre will depend on a variety of factors: complexity of the data, complexity of the story, the intended audience and the intended medium. In general there is no “right choice”. Several possibilities will arise, each with their advantages and disadvantages.

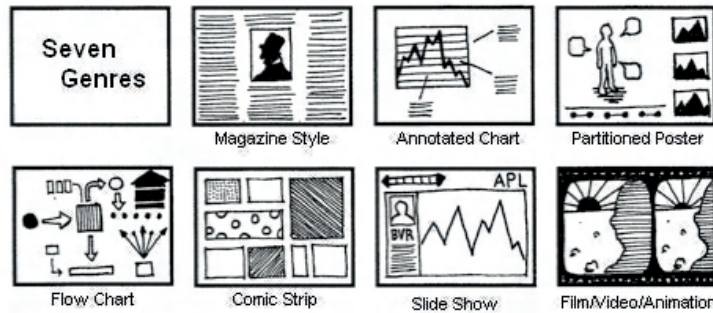


Figure 3-1: Genres of Narrative Visualisation. (Source: Segel & Heer [80])

3.3.2 Visual Narrative Tactics

A second division identifies *visual narrative tactics*. These are the visual devices used to assist and facilitate the narrative. This division is subdivided into three sections.

Visual structuring: the mechanisms that communicate the overall structure of the narrative to the viewer and allow him to identify his position within the larger organisation of the visualisation. These design strategies help orient the viewer early on and allow the viewer to track his progress through the visualisation.

Examples: establishing shot, check-list, consistent visual platform, progress bar, time line slider.

Highlighting: visual mechanisms that help direct the viewer's attention to particular elements in the display and augment the salience of an element relative to its surroundings.

Examples: colour usage, motion, framing, size, audio.

Transition guidance: techniques for moving within or between visual scenes without disorienting the view.

Examples: animated transitions, object continuity, camera motion.

3.3.3 Narrative Structure Tactics

Narrative structure tactics, further divided into three sections, are the non-visual mechanisms that assist and facilitate the narrative.

Ordering: ways of arranging the path viewers take through the visualisation. These can be prescribed by the author (*linear*), have no suggested path (*random access*), or the user is able to select a path among multiple

CHAPTER 3. NARRATIVE VISUALISATION

alternatives (*user-directed*).

Interactivity: refers to the different ways a user can manipulate the visualisation and as well as how the user learns those methods.

Examples: filtering, selecting, searching, navigating

Messaging: involves the techniques a visualisation communicates observations and commentary to the viewer.

Examples: labels, captions, headlines, annotations, articles, introductions, summaries.

3.3.4 Author-Driven and Reader-Driven

The visual narrative genres must balance a narrative, intended by the author, with story discovery, on the part of the reader. A visualisation can thus be placed along a spectrum of *author-driven* and *reader-driven* approaches.

Author-Driven: linear ordering, heavy messaging, no interactivity.

Reader-Driven: no prescribed ordering, no messaging, free interactivity.

For visualisations explored by a reader, a strongly author-driven approach works best when the goal is storytelling or efficient communication. A reader-driven approach supports tasks such as data diagnostics, pattern discovery, and hypothesis formation.

Historically, visualisations fall into the author- or reader-driven dichotomy. However, narrative visualisations fall somewhere in-between. An important attribute these visualisations possess is the flexibility in balancing both elements. Providing room for limited interactivity within the context of a more structured narrative.

Although there is a large range of possible combinations, Segel & Heer [80] defined 3 distinct schema's that have become most common.

3.3.4.1 Martini Glass Structure

The Martini Glass visualisation structure begins with an author-driven approach, initially using questions, observations, or written articles to introduce the visualisation. Occasionally no text is used at all, as the visualisation instead relies on an interesting default view or annotations. Once the author's intended narrative is complete, the visualisation opens up to a reader-driven stage where the user is free to interactively explore the data.

3.3.4.2 Interactive Slide-show

The Interactive Slide-show structure follows a typical slide-show format, but incorporates interaction mid-narrative within the confines of each slide. This structure allows the user to further explore particular points of the presentation before moving ahead to the next stage of the story. Contrary to the martini glass, an interactive slide-show allows for interaction mid-narrative, giving rise to a more balanced mix of author-driven and reader-driven approaches.

3.3.4.3 Drill-Down Story

The Drill-Down Story visualisation structure presents a general theme and then allows the user to choose among particular instances of that theme to reveal additional details and back-stories. This structure puts more emphasis on the reader-driven approach, letting the user dictate what stories are told and when. Nevertheless, it still requires significant amounts of authoring to determine the possible types of user interaction, what candidate stories to include, and the details included for each story.

3.4 Storytelling Scenarios

Kosara & Mackinlay [57] provide us with three different scenarios in which diverse requirements for used techniques, ways in which the presentation is structured, amount of interaction is anticipated, etc., are presented.

Live presentation by a speaker in front of an audience. This model, exemplified by Hans Rosling and his Gapminder presentation [79], is similar to the way many business presentations are given today. The main difference is in how the presenter can respond to questions, which partially obviates the need for an *open exploration part* at the end, but also poses other challenges.

A presentation based on a live visualisation allows the presenter to pause the story and interact in response to questions. It is also possible for the presentation to adapt to changes made at one point that carry forward through different presentation steps. In addition to the usual kinds of interactions used in visualisation, there is an additional layer of annotation, highlighting, etc., which can be useful in presenting.

Individual (or small-group) presentation of results. Though not so different from the first situation, in this case there might be more interaction involved between the presenter and the audience. This requires the presentation tool to be more flexible than a simple slide show, making it possible to answer questions that come up during the presentation. For

CHAPTER 3. NARRATIVE VISUALISATION

example, in a discussion of quarterly results, questions about specific sales or marketing measures might come up that were not part of the story, but are of interest to people in the audience.

A third scenario is a **self-running presentation to a large audience**, typical for news media presentations. A presentation is created once and then viewed by many people independently and without the ability to interact with the author or ask questions. The goal of these stories is similar to that of a written feature story, getting a point across and explaining it in sufficient detail. A key concern is often how to draw the reader in. Trying to peak the reader's curiosity to make them read the rest of the article. Also additional hints or affordances are needed to direct the user to certain interactions, this to get the story started or to advance to the next step.

4

Visualisation and Presenting

Presenting and communicating insights to an audience, telling a story, is one of the main goals of data exploration.

In the previous chapters we already noted the predominant focus on the exploration and analysis of data in information visualisation. We have also discussed the next logical step in research, namely presenting and communicating data.

Though research is starting to focus on this aspect, it is particularly aiming its attention on narrative visualisations for the web. Scenarios in which a speaker tries to get one or multiple points across to an audience, on the other hand, are being overlooked.

A need for tools, focused on telling stories behind data to a live audience, is clearly present. Though it would be possible to implement this in stand-alone applications. We rather believe it would be optimal for presentation software, e.g., Microsoft PowerPoint¹ or Apple's Keynote², to support these concepts. In this way, when a presentation consist of more than only data, there is no need to switch between tools and thus interrupting the flow of the presentation.

Before elaborating on the mechanisms existing tools currently provide us and defining our vision for the next generation presentation tools, in terms of information visualisation, let us first take a look at a prime example of how data can be presented in an exciting way.

¹Microsoft PowerPoint. <http://office.microsoft.com/en-us/powerpoint>

²Apple's Keynote. <http://www.apple.com/mac/keynote>

4.1 Gapminder

In section 3.4 we already mentioned Hans Rosling’s TED 2006 talk [79] on human development trends.

Gapminder is an animated presentation using bubble charts, scatter plots with point size representing a value, with animation to transition between years. After Rosling’s talk, this technique gained a lot of popularity [57]. While the animated transitions were shown to have a slight detrimental affect on people’s ability to follow trends [76], they are entertaining and captivating, and lend themselves well to live presentation in front of an audience.

In addition to the transitions, Gapminder also showed the effectiveness of building views up gradually so that the audience can follow, even when the visualisation is relatively complex. Figure 4-1 shows several steps that first explain the idea of a distribution and percentiles, then build a stacked area chart, and then turn that area chart into a scatter plot. The entire transition is quite complex, but by breaking it up into small steps and using entertaining but suitable imagery, the audience can follow along with relative ease.

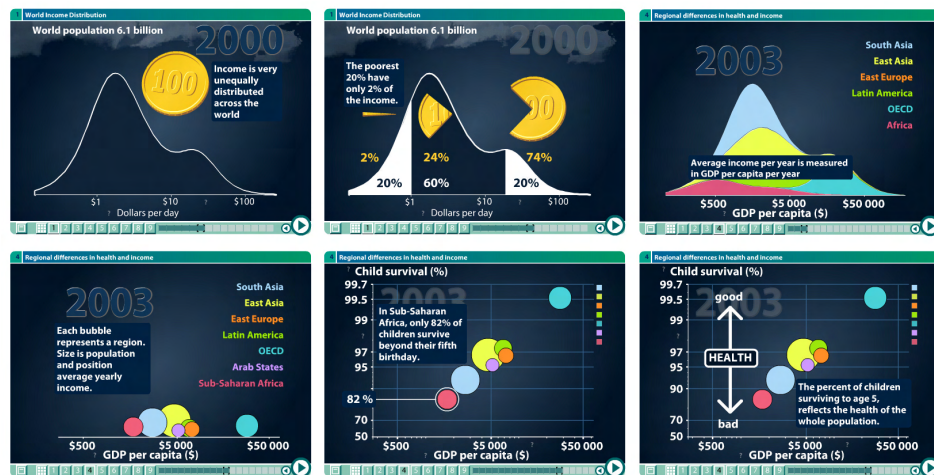


Figure 4-1: Gapminder: one of the most effective stories built entirely using visualisation. This sequence transitions from a stacked area chart to a scatter plot, explaining the visualisation to the user as well as what to look for. (Source: Kosara & Mackinlay [57])

The transition from the stacked area chart to the scatter plot/bubble chart is particularly notable. First, the differently-coloured layers, representing continents, turn into small circles or “bubbles”. These bubbles are then slightly rearranged to split off the Arabic countries from Africa and Asia. But the entire time, their horizontal position is determined by an x axis that repre-

sents income, the same axis that the area chart also used. Once the bubbles are explained, a vertical axis is unrolled, which the bubbles stick to. This animation very simply but clearly explained, the idea that the location of the bubbles is determined by not just one value, but two; the second one being a measure for health.

The popularity of Gapminder is certainly partly due to Hans Rosling's personality and energetic style of presentation. But the techniques and ideas used certainly appear effective and useful, even though only a few of them have been studied with any depth.

4.2 Current Implementations

In 2003 Zonker & Salesin [121] estimated a market share of 95% for Microsoft's PowerPoint. A Forrester study in 2009 supported this number, stating that only 8% of enterprise customers use alternative products [66]. In 2012 Thielsch & Perabo [97] again confirmed a usage preference primarily for PowerPoint. With an estimated occurrence of 30 million PowerPoint presentations every day [72, 87] and the use of more than 400 million copies of Microsoft PowerPoint [87] we can not deny the influence of PowerPoint in our everyday presentation habits [39]. Nevertheless, even with PowerPoint's predominant market share, we must also acknowledge alternative software, i.e., Apple's Keynote, OpenOffice Impress³, etc.

Current and/or popular presentation software already provide us with the means to present data to a certain level. In this section we will discuss the components these tools have implemented. Considering PowerPoint's authoritative lead for presenting, we will first elaborate on the standard set by Microsoft and use this as a reference for other presentation software.

4.2.1 PowerPoint

Static visualisations within PowerPoint are possible via an integrated chart tool. Adding a chart or graph into a PowerPoint presentation can be done in one of two ways.

When dealing with lots of data, it is desirable to create the chart in Excel and then copy it into the presentation. This is also the best way if the data changes regularly and the chart always needs to reflect the latest numbers. For this case it is possible to keep the chart linked to the original Excel file.

³OpenOffice Impress. <http://www.openoffice.org/product/impress.html>

CHAPTER 4. VISUALISATION AND PRESENTING

A simple chart, with little data to show, can be created directly by inserting a chart via the ribbon⁴, or the chart icon in the place-holder which needs to contain the chart. After selecting the appropriate visualisation, an Excel spreadsheet appears in which the data can be entered (Figure 4-2). The data is then stored in the presentation itself.

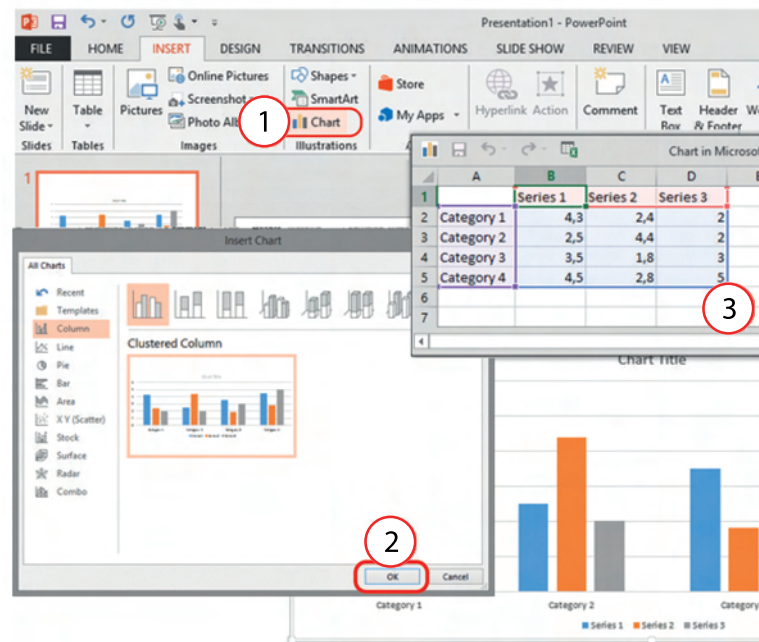


Figure 4-2: Inserting and editing chart data in PowerPoint

Animations can as well be added to a graph. However only with limited functionalities.

PowerPoint provides the use of *entrance*, *emphasis*, and *exit animation effects* for slides and objects as well as *motion paths* for objects. For charts these same effects can be applied to the the complete object or, depending on the type of chart, the different parts of the chart.

The attainable animations do not give a lot of useful functionality for a presentation, nevertheless it does give us the possibility to step by step introduce the audience with the data rather than showing a complete visualisation at once, emphasise on a specific part of the chart when needed, or even visually remove a part of the chart with the exit animation effects. Infinite combinations can be applied to define the linear flow a presenter would like to apply during the presentation.

⁴A ribbon is a set of tool-bars placed on several tabs. Microsoft products released since 2007 a form of modular ribbon as their main interface, where large tabbed tool-bars are grouped by functionality.

The feasibility of combining different animation effects might not be directly clear for the user. This being, an animation can be added by simply selecting the chart and applying a desired animation. When a sequential animation is desired, the *Add Animation* button must be used for not to overwrite the already defined animation.

Animated items are noted on the slide by a non-printing numbered tag, correlating to the effects in the list. These can be selected to manually edit each animation part separately. When opening the *Animation Pane*, an overview of all the defined animations is presented with the order they will be executed. This gives a more comprehensive perspective on which animation is played when and for how long.

Within PowerPoint a variety of **chart types** is offered for use. A list of these types can be found in table A-2 in appendix A on page 81.

4.2.2 Keynote

Static visualisations in Apple's Keynote work with the same principles as PowerPoint. Likewise, it is possible to insert a chart linked to a Numbers table or with the chart data editor in Keynote itself (Figure 4-3). Next to this, a dedicated chart pane pops up in which the chart can be fully customized.

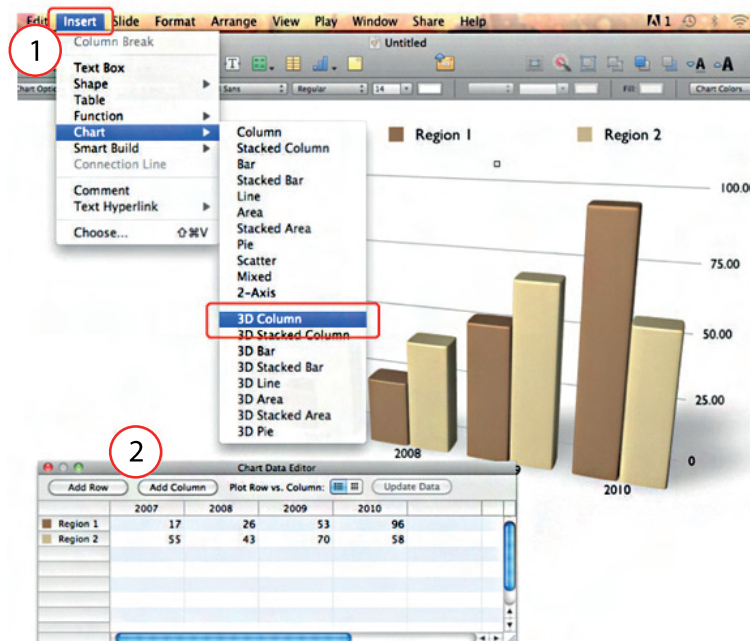


Figure 4-3: Inserting and editing chart data in Keynote

Animations on the other hand, are defined very differently. In the chart pane, different types of animations can be selected. Categorized similar as in PowerPoint, with *Build In*, *Build Out*, and *Action*, they are subdivided in *3D effects* and *2D effects*.

According to the type of chart selected, a different variety of animations can be applied, i.e., a radial animation is provided for a pie chart though it will not appear in the effects supported for a bar chart.

Delivery, as within PowerPoint, is again determined by the type of visualisation.

The range of **chart types** presented by Keynote is equivalent to PowerPoint's, though there are less subtypes available.

4.2.3 OpenOffice Impress

In Impress charts are made, again, in the same fashion as within PowerPoint. Additionally the same types of charts are present.

Then again, **Animating** the charts is impossible. There is no facility in Impress for chart effects, as there is in PowerPoint, such as presenting a series or category at a time. On import, the charts simply appear. A workaround is to have multiple copies of the chart with each one set up to display as desired [70].

4.2.4 Alternative Presentation Software

Prezi

Prezi⁵ is one of the rare radically different tools which has become somewhat known in an environment dominated by classical slide-ware tools. It steps away from the slide paradigm and gives the user an *infinite canvas* to work with. Prezi offers the possibility to add several types of diagrams to a presentation, though not the type of charts as we have seen in PowerPoint or Keynote. When we take a closer look at the diagrams available when placed on the canvas, we notice them to be build with basic shapes also available for manual drawing. It is thus achievable to create a visualisation by hand. Animations are not possible, however by usage of the path editor and the zoom-able user interface in a clever and creative way, the illusion of an animation can be simulated.

PowToon

PowToon⁶ takes a different approach than Prezi. Prezi keeps the canvas and items fixed and moves the camera. PowToon keeps the camera steady but instead of moving the canvas, it moves the characters and props. PowToon

⁵Prezi. <http://prezi.com>

⁶PowToon. <http://www.powtoon.com/>

pushes hard into becoming an animated presentations maker.

As with Prezi, in PowToon, chart creation is only available manually with pre-built objects. However, on account of PowToon's movable objects, animations on the other hand can be created with great ease and lots of flexibility.

Beamer

Beamer is an extension for LaTeX, which is provided as a document class for usage within the regular LaTeX tool-set. The main ideology of Beamer is the same as for the LaTeX language: place the *focus on content, not on presentation*. Due to the fact that Beamer is built on top of LaTeX, all of the regular features provided by LaTeX can be used. The downside for this, it is not trivial for users who never had to write any code before. Slides have to be created through code and there exists no WYSIWYG editor.

With the wide range of packages available for LaTeX, there are also some available to create charts, e.g., PGF/TikZ, pgfplots and PSTricks. Keep in mind, these charts are also created in code, meaning even simple visualisations will take some effort to compose. Though possible, let us not even mention the endeavours that should be overcome to create animations.

4.2.5 Workarounds

While not every presentation tool provides the means to create charts in a presentation, there are always workarounds to eventually achieve a similar effect.

A first alternative is to simply create the desired visualisation in a specialised software for data visualisation, spreadsheet, or graphic design software, and export it as an image which can then be imported into the presentation. Animations can be mimicked by placing various images in different slides, comparable to the use of multiple charts in Impress.

A second possibility, when supported by the software being used, is to create a film of the desired visualisation and animations. This can achieve the same linear structure as for instance PowerPoint's animations.

An other option is to design an interactive flash application which can be imported into the presentation. This option will give the full flexibility of interactions in flash, without the limitations of the presentation software being used. Again, provided the support for flash applications.

Some presentation software allows the use of 3rd party plug-ins for advanced charts and animations. Because of PowerPoint's wide usage, the likelihood of finding plug-ins is much greater as for a less known tool.

As a last resort, we could always fall back on visualisation software and simply switch between presentation and visualisation during the presentation.

4.3 The Issue

In a scenario where a speaker presents live in front of a group, with interaction between the presenter and the audience, we are faced with a selection of possibilities that are not always optimal in regards to presenting data.

Static visualisation might give us a certainty that we are always able to display its content. Though they do not give us much flexibility. We are able to create a needed chart for a given presentation. However when questions arise about the data being visualised, the presenter must rely on the visualisations created in preparation for the presentation. This means, a lot of back-up visualisations for every possible question that might arise, or no visualisation at all to visually help explain a concept.

Let us take the example of a business analyst presenting some product sales. These can be visualised via a bar chart for every product. Other interesting visualisations might be a relative percentages view in a pie chart, or a comparison of sales with profits in a scatter plot. When during this presentation someone is interested in sales per region, and the analyst never made a visualisation for this situation, expecting this was not of interest for the audience, the presenter is forced to communicate raw figures in answer to the question. Not being able to visually show how regions compare, even though the data was available.

When a visualisation immediately presents lots of data, the audience also loses focus on what the speaker is actually telling and will only have focus on the data being presented. Therefore the message of the whole visualisation gets lost because of the overwhelming amount of data the audience is trying to interpret on their own. The speaker is also forced to explain every detail the audience is seeing, thus not being able to focus on the message, but purely on the data and what is being visualised.

As discussed, a workaround could be to build up to a chart with consecutive charts, each presenting more and more detail. This is in essence also what can be achieved with the animations provided in PowerPoint and Keynote. These presentation tools also provide us with another challenge. Because of their linear format, a presenter is forced to run through every slide whenever a question arises concerning a subject covered by a so called back-up slide. The same whenever there is a reason to move back to the front of the presentation. Thus whenever animations are defined on a chart, these also must be executed when passed through. Though each animation can be ended with a simple click or button push, it does provide us with a inconvenience. An other issue can be when two or multiple visualisations have a relationship to each other which is not directly visible when not showed simultaneously.

The presenter must then switch back and forward between the visualisations to visually express the narrative, or display them on the same slide and risk losing details of the message needed to express.

Interactive visualisations provide a solution to these problems. They can be programmed in a way that data can be filtered, explored and changed in representation. Thus making back-up visualisations obsolete. In accordance to the story or a question being asked, the presenter can simply adapt the visualisation to what is needed. Provided of course the visualisation is programmed in such a manner this is possible.

These types of visualisations are time consuming to create, and can potentially be very expensive. They are also mostly custom tailored to a given dataset. Making re-usability limited. The visualisation might get adapted, being able to present an different dataset, giving rise to more costs and time consumption.

We've discussed the example of Gapminder, which falls into this category. Though it is an excellent example of how data can be presented in an exciting and engaging way. Nevertheless, the development was a time consuming engagement. Adaptations from the very first version, supporting 16 indicators, to the current version, over 200 indicators, make it possible to visualise global trends. But then again only the trends provided by Gapminder's dataset can be visualised. It is also not possible to import this tool into a PowerPoint or Keynote presentation. Thus when such a presentation is given, the presenter is forced to switch between his presentation and the Gapminder visualisation tool. Interrupting the flow of the presentation, showing a whole new environment with no connection, on a visual perspective, to the environment the presentation started from.

Narrative visualisations have the same goal as live presentations, communicate a specific message to an audience. Even though the medium on which this is done is very different, and the audience must explore the visualisation themselves without the guidance of a presenter, they do provide us with some tactics on how data can be presented in a live setting. They are in fact stand-alone presentations telling us a story through the data being presented.

4.4 The Next Step

In the latest decade slide-ware tools have been put under a microscope. Ever since Tufte's critiques on PowerPoint [100], researchers have been evaluating slide-ware tools.

CHAPTER 4. VISUALISATION AND PRESENTING

Stepping aside from the linear structure and presenter-oriented features, new tools are being introduced to enhance engagement, liberating interesting conversations from the boredom of one-way presentations.

Though we see a positive evolution in the work being done on presentation software, we do believe these concepts must be applied into every aspect of the software. For the visualisation of data, we notice a lack of adaptation to these new approaches.

By studying the techniques in interactive and narrative visualisations, we believe their mechanisms can be applied to further enhance the conversation between speaker and audience.

The interaction techniques and transitions discussed in chapter 2 could make it possible to adapt visualisations on the fly, according to the needs of the audience at that moment.

Narrative visualisation techniques discussed in chapter 3 can guide in how these interactive visualisations help us in order to keep the audience compelled to the message being told instead of distracted by what is shown on screen. They also help the presenter accompany the audience throughout a large dataset. Making it possible to communicate a story about data in contrary to merely presenting data.

As opposed to specific interactive visualisations, we therefore propose to define these aspects on a higher level, making it possible to utilize these tools during a presentation, regardless to the dataset being visualised, nor the type of visualisation being used. Thus preserving the ease of creating visualisations and removing the limited functionalities.

In this way, we strive for flexibility in regards to how data can be presented. Balancing interactivity and additions in narrative communication of the story. And being able to communicate in a linear fashion whilst remaining capable of digressing from a pre-defined story path.

5

MindXpres: An Extensible Content-driven Cross-Media Presentation Tool

In the previous chapter, we already pointed out some of the issues with information visualisation in current presentations tools. In this chapter we take a step back and introduce ourself to a novel way of presenting in general.

MindXpres [78] is a new presentation tool that clears the way of working with presentations for a more flexible and semantically enhanced future of presentation tools. It introduces a radically new presentation format that solves the lack of some features introduced by the common tools like Microsoft PowerPoint, Apple Keynote or OpenOffice Impress, e.g., the enforcement of linear traversal of slides, the impossibility to display multiple slides at once, and presenter-oriented presentations.

Most of the features offered by the common slide-ware are there to increase the ease of use for the presenter. These features, creating content and aesthetics, publishing and presenting, are all designed with the presenter in mind. MindXpres gives a solution for these problems without any loss for ease of use offered by the common slide-ware and gives us the opportunity to create presentations more audience-oriented. This convenience is made possible because MindXpres is developed with the eye on extensibility.

MindXpres separates the content and visualisation similar to a \LaTeX document. A graphical editor handles the creation of a domain-specific language that focuses on the content, the visualisation is done by the tool (compiler) based on a chosen template and outputted in HTML5 which makes everything very portable and distributable over heterogeneous devices. It

also offers components to facilitate specific types of information and has a plug-in mechanism which can add new future media types. Some more advanced features of MindXpres are non-linear traversal of the presentation, hyper-links, transclusion, semantic linking and navigation of information, multi-modal input, dynamic interaction with the content, and the import of external presentations.

In the following sections, we will take a closer look at how MindXpres is built. We discuss the hypermedia model that is used as a base, the domain-specific language and a short overview of the architecture.

5.1 The RSL Model

The Resource-Selector-Link (RSL) model [85] is a general meta-model for hypermedia that is based on the semantic, object-oriented data model OM [69]. Hypermedia is the extension of hypertext with multimedia facilities like sound or video. This RSL model is used as the base for MindXpres. Features of hypermedia as semantic linking, navigational links, structural links, transclusion, annotation and context awareness can be handled by the RSL model.

Resource, Selector and Link are seen as the three base components of the model and is defined as *the Link Meta Model*.

- **Resource:** a resource is a piece of data that becomes concrete in a domain-specific context. E.g. in MindXpres, this can be text, video, images, or more.
- **Selector:** a selector is an abstract concept that selects a part of the resource. We can have many selectors on one entity. E.g. we select a time-range of a video data file.
- **Link:** the linking concept allows us to link resources directly or via a selector. This is most fundamental in a hypermedia model. The RSL model also supports the concept of multi-source links.

Additional features are introduced by two other concepts in the core model on top of the three previous defined concepts.

- **Property:** a property is a parameter that is associated with an entity. An entity can be any of the previous defined concepts.

- **Context Resolver:** Each entity can be linked with a set of Context Resolvers. The correct entity will be displayed based on a calculation by the Context Resolver.

The preceding concepts form the base for the RSL model. Other features are built around these concepts. The following, user model, layers and structural links, lay down the foundation as how MindXpres does this.

5.1.1 The User Model

The user model describes the ownership and access of information. This explicit feature often lacks in other hypermedia models. It is possible to create rights for a group of users or even an individual user to access a piece of information. When an entity is created, it is owned by its creator. The owner can then assign rights to groups or individuals, allowing sharing and collaboration.

The context resolvers can also be used to help define the rights in certain context, e.g., sharing some extra notes to the audience members while giving a presentation. When the same presentation is published on-line, it may be the choice of the presenter to only let audience members access this extra information. Other people who obtain the presentation that did not participate will not be able to see this information.

5.1.2 Layers

Earlier we mentioned how selectors can be used to access specific parts of a resource. We also stated the possibility to define multiple selectors for one entity, causing the possibility for overlapping.

In the following example [?], we see a conflict in the anchor tag of HTML. When clicking on the link, the browser will not know which anchor tag to select. The layer feature of RSL model provides a solution for this. For every resource, a layer can be defined. Since many selectors can be defined for the same resource, the restriction is that a selector will be defined for a resource for a given layer. This creates a hierarchy of selectors. When a conflict occurs with the selectors, the one on the top layer is selected.

```

1 <a href="target1">
2   This is some text that links to target1, but
3   <a href="target2">
4     this text links to target2!
5   </a>
6 </a>
```

5.1.3 Structural Links

Structural links are a subtype of the first class Link object in the RSL model. These specified links bind information into a structure, e.g., a document has a structure, it contains chapters with sections.

The RSL model also has a navigational subtype of the Link object that specifies the way of navigation. The concrete form depends on the domain.

5.1.4 Conclusion

We have introduced the RSL model that is used as a base model for MindXpres. For completeness, the full model is given in figure 5-1. It provides a solid model for hypermedia.

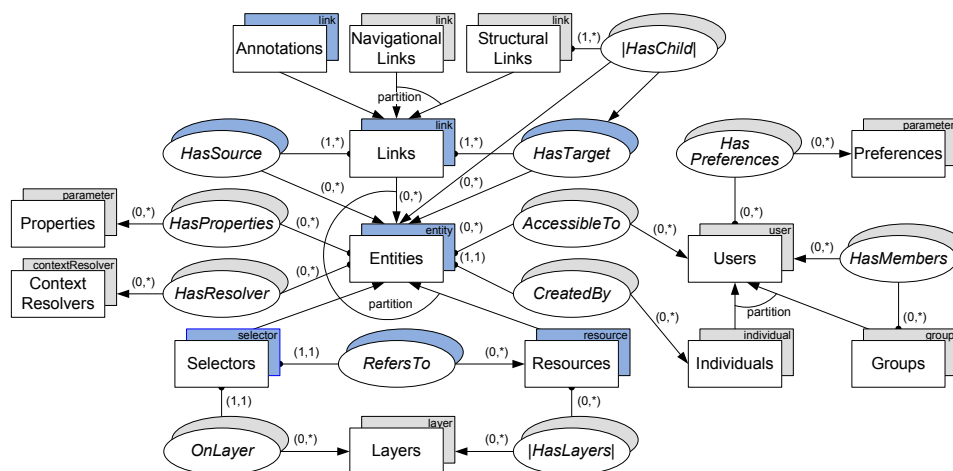


Figure 5-1: Complete RSL model (Source: Signer and Norrie [85])

5.2 Fundamental components

Without going into the detail, we will now discuss how MindXpres is build. For more information we kindly refer to [?]. First we will discuss the XML Authoring language that will be outputted into HTML5 and the visualisation library. In figure 5-2 a general overview for the architecture of MindXpres is visible.

The content, and thus the presentation that it represents, is stored in an XML container format. This is a predefined, machine-readable XML vocabulary to define content that will be used in the presentation. This format can be created manually, or optionally a WYSIWYG editor could generate

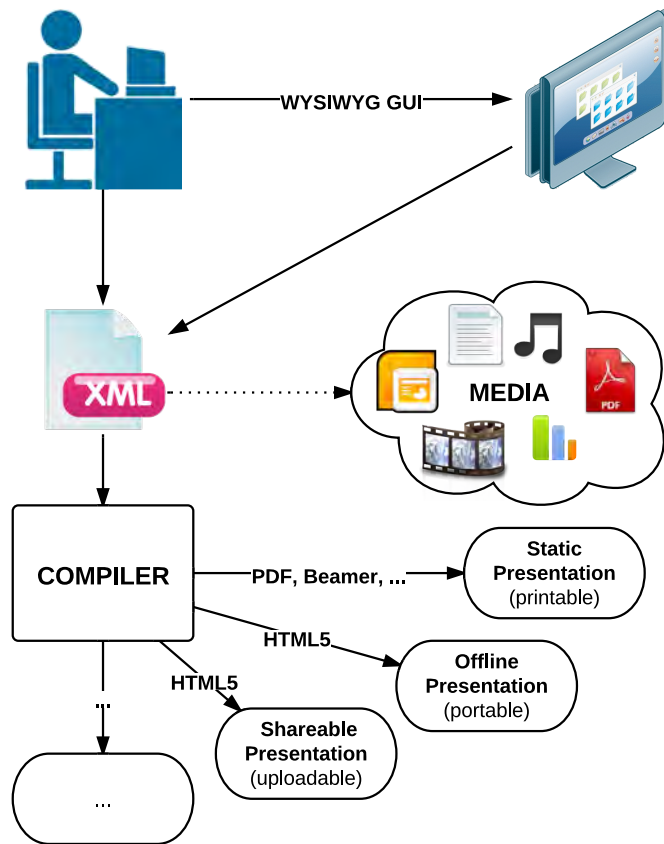


Figure 5-2: MindXpres architecture (Source: Roels and Signer [77])

the XML.

Once the XML Container file has been created, this is passed to the compiler, which will in his turn convert the XML document into an HTML document. This HTML document is basically an HTML representation of the original XML document, which makes it easier to visualise using JavaScript and related technologies. This transformation allows us to add compile-time features to aid both the user and the visualisation layer. An example of a compile-time feature could be the import of data from an exotic document format such as Excel, and convert it to something usable by the visualisation layer.

5.2.1 XML Authoring language

MindXpres uses XML as a base mark-up language. The XML document is validated by the compiler as seen above with the use of an XML schema. When attributes are forgotten by the user, the compiler will correct and add

CHAPTER 5. MINDXPRES

them.

The following is an example of the language used to compile in the desired HTML5 format. It will generate a presentation that contains one slide. This slide displays the title and has an image with two item bullets underneath it.

```
1 <presentation theme="my_theme">
2   <slide title="A small example">
3     <image src="example.jpg">
4     <bulletlist>
5       <item>A first item</item>
6       <item>A second item</item>
7     </bulletlist>
8   </slide>
9 </presentation>
```

5.2.2 Compiler

The compiler for MindXpres is written in Java and has three main goals. First, validating the XML document, made by the user, based on the XML schema. Once a document has been successfully validated, the XML language gets processed via a SAX ¹ parser. The document is processed sequentially and events are generated as elements are encountered. While traversing the XML document, elements are also transformed to valid HTML. The previous given XML example will be compiled to the following HTML code:

```
1 <div data-type="presentation" data-theme="my_theme">
2   <div id="element_1" data-type="slide" data-title="A small
3     example">
4     
5     <ul>
6       <li>A first item</li>
7       <li>A second item</li>
8     </ul>
9   </div>
10 </div>
```

¹SAX (Simple API for XML) is an event-based sequential access parser API developed by the XML-DEV mailing list for XML documents. SAX provides a mechanism for reading data from an XML document that is an alternative to that provided by the Document Object Model (DOM). Where the DOM operates on the document as a whole, SAX parsers operate on each piece of the XML document sequentially.

5.3 Plug-in Mechanism

As previously stated, MindXpres has some interesting presentation tool features. It separates the content and visualisation as in a \LaTeX document. And due to HTML5, which is accepted in all major browsers, portability and distribution over heterogeneous devices is possible. The MindXpres tool also offers components to facilitate specific types of information and has a plug-in mechanism that can add new future media type. Other aspects as:

- Non-linear traversal of the presentation
- Hyperlinks
- Transclusion
- Semantic linking and navigation of information
- Multimodal input
- Dynamic interaction with content
- Import of external presentations

are also features why MindXpres becomes powerful to use.

MindXpres does not have a core with hard-coded components and aesthetics. This is possible due to the plug-in framework. It also allows third parties to replace, modify or even add functionalities. A mechanism that creates expandability in a presentation tool, which often lacks or is fairly limited in other existing tools. Making it possible to download or create plug-ins, adding functionalities that one would desire.

This aspect is one we will fully utilize for implementing our own framework.

In MindXpres there are three major types of plug-ins. These plug-ins focus more on the functionality and implementation details rather than aesthetics. Aesthetics of a plug-in can be changed by templates.

5.3.1 Components

Components are the group of plug-ins that provides visualisations and functionality for a specific content type for different content containers, e.g., a bullet list, images, video, source code, and more. It decides how relevant content is displayed, and defines the interaction with it. Note that everything is a plug-in, including the content containers to the simplest content types (i.e., text). An example of where it would make sense to create a new component plug-in is video. A plug-in components could be created, providing video's and using the HTML5 component. However, it could also be preferred to use a flash player for video. The existing plug-in could then be modified or a new plug-in supporting flash could be created as well.

5.3.2 Containers

Containers are the elements that group or organise components visually. The most obvious container would be a slide. Each slide contains different content and can have other content that reoccurs. For instance, a slide can have a title, a slide number and the author's name. These common elements can be abstracted in a higher level container, taking tedious work out of the users' hands. It is also possible that the container provides functionality to help the user with the layout by defining pre-sets or allowing easy definitions of layout.

5.3.3 Structures

Structures are the elements that lay out components on a larger scale for use with a Zoom-able User Interface, e.g., displaying elements in a grid or structured in sections. The difference with containers, is that structures have ties to the XML language to define presentations. Similar to how \LaTeX provides structure in documents (e.g., chapters, sections and subsections), the structure plug-ins can define structures that can be used from the XML language. This is sometimes necessary for complex visualisation, as with the structured layouts, because the plug-in needs to know how to group the content. Making structures more complex and on a higher level than containers.

5.3.4 Implementing Plug-ins

Plug-ins are contained into individual folders to keep things clean. Such a plug-in folder includes all JavaScript, CSS, Images and other resources that are relevant to the plug-in. This mechanism allows for easy plug-in management, as adding or removing plug-ins is as easy as adding or removing plug-in folders in the plug-in directories.

A specific naming convention allows the plug-in to be found by MindXpres. Every plug-in must have a file called `plugin_info.js` that is loaded at first. This file shares the plug-in-specific information to MindXpres such as the tags that it provides for the XML authoring language. The information that is shared also contains the name of the main plug-in object, which is initiated by calling its `init` method. In figure 5-3, we see an example of the video plug-in folder.

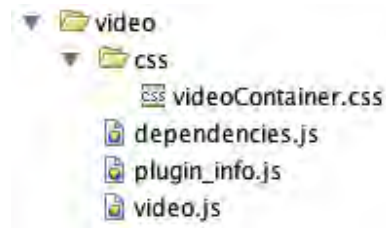


Figure 5-3: Complete RSL model (Source: Roels and Signer [?])

5.4 Conclusion

In this chapter we have introduced MindXpres. A presentation tool that will help us creating next-generation presentations. We have seen that it is highly portable due to HTML5. The visualisation layer of MindXpres is heavily inspired by Zoom-able User Interfaces and spatial hypertext. It also offers a plug-in mechanism through a specific naming convention. This plug-in mechanism offers third parties the possibility to add, modify, and remove features with ease and in a clean way. The indicated flexibility is a feature we must fully exploit when we implement our own framework for interactions on interactive visualisations.

The audience-oriented approach that is embraced by MindXpres also coincides with our pursuit to further enhance the dialogue between presenter and audience.

6

Implementation

In chapter 4, we have detailed some perceptions, opinions, and aspirations about how presentation tools should implement information visualisations. This assessment is based on the evaluation of existing presentation tools, together with the study of techniques used in interactive and narrative visualisations presented in chapters 1 and 3 respectively.

We start this chapter by defining some goals. These are certain technical features that we wish to achieve with our implementation.

6.1 Goals

We start this chapter by defining some implementation goals. These are general technical features that we deem important for the prototype. By explicitly listing these goals, we hope to give the reader an idea of what we are building towards in the more technical parts of this chapter.

Chart Types

For the prototype to prove its value, the implementation of basic most common chart types is a must.

These include, but are not limited to:

- Line Chart
- Bar Chart
- Scatter plot
- Pie Chart

Interactivity

For every chart type basic interactions, as described in chapter 2.2, should be available and followed by the appropriate transition, chapter 2.3.

Linear Traversal in the Chart

The presenter should be able, in some form, to traverse a pre-defined set of interactions in a linear way during the presentation. This way he is not compelled to manually adapt the visualisation whilst narrating his story.

Authoring

Authoring can be done in 2 different stages.

The first stage is set in MindXpres's authoring Language. As with all other components in MindXpres, all settings for a chart should be defined here.

Fiddling with these setting to acquire a desirable result, can be a tedious process. Therefore the author must also be able to directly adapt a chart via a UI. Making adjustments instantaneously visible.

6.2 Architecture

As mentioned in chapter 5, one of the reasons for choosing MindXpres is it's output in HTML5. Allowing us to take full advantage of *document object model (DOM)* manipulations to create and adapt visualisations.

Before jumping into the architecture, we will first shortly describe the Underlying Libraries used for our implementation.

6.2.1 Underlying Libraries

When we speak about DOM manipulation and visualisations Bostock's Data-Driven Documents (D3) [14] instantaneously comes to mind. D3 is a JavaScript library for producing dynamic, interactive data visualisations in web browsers, making use of the widely implemented SVG, HTML5, and CSS standards.

Data-Driven Documents

When D3 came out in 2011, it became clear pretty quickly that it was going to be a powerful tool for creating data visualisations. Embedded within an HTML page, the JavaScript D3.js library uses pre-built JavaScript functions to select elements, create SVG objects, style them, or add transitions, dynamic effects or tool-tips to them. Large datasets can easily be bound to SVG objects using simple functions to generate rich text/graphic charts and diagrams.

Nevertheless, DOM manipulation can be extremely slow for particularly large numbers of entries and SVG also has performance limitations when dealing with extensive quantities of elements. Perhaps the most important part of D3 is the position and approach it takes. It is not a graphics library, nor is it a data processing library. It has tools that make the connection between data and graphics easy. It sits right between the two.

D3-based reusable chart library

Bostock's article, proposing a convention for encapsulating reusable charts in D3 [13] gave rise to several projects applying the notions of *repeatable*, *modifiable*, *configurable* and *extensible* to their charts.

One of these projects is C3.js [95]. An open-source project lead by Masayuki Tanaka. The difference from other charting libraries is that with C3 (stands for *Comfortable*, *Customizable*, and *Controllable*) one can control the state of a chart through its API, such as focusing on a data series, selecting data points, showing/hiding the data series, and updating the data. With these features it is possible to change the chart according to events such as user input, and interaction, server responses, or other events.

Out of the box, C3 initially does not support all of our requirements, though due to its MIT License the framework can be extended to do so.

6.2.2 Initialisation

As with D3, a chart needs to be bound to a division (`<div>`). This is specified in a configuration variable together with all other possible setting. Potential settings and how this is integrated in MindXpres will be discussed later on. For now, acknowledge they are there.

Before being able to draw a graph, an initiation process is started. In this process unique id's for several components are defined, format setting for the axes are set, variables for data-management are created, and the legend is configured.

After this, the data gets loaded and the actual chart is initialized.

In this second initialisation process, above all, data is formatted into targets, calculations for settings and sizes related to the data are made, and groupings for the different elements in the chart are appended to the main chart division.

CHAPTER 6. IMPLEMENTATION

6.2.2.1 Targets

When data is gets loaded, it is formatted into targets. The `targets` variable is an array consisting of objects. These objects are the members formatted with three properties: `id`, `id_org`, and `values`.

Values is yet another array of objects, expressed by the following properties: `id`, `index`, `value`, and `x`. `index` is used to identify a specific value, e.g. for the tool-tip, while `x` defines the placement on the x-axis.

The code below illustrates how a simple dataset is formatted to targets. It contains three members (`data1`, `data2`, and `data3`) with each three variables (A, B, and C). We will continue using this dataset further on in our examples.

Table 6-1: A simple dataset.

	A	B	C
data1	1	2	3
data2	10	20	30
data3	15	25	35

Transforming to targets will result in the following array:

```
[
  {
    id: 'data1',
    id_org: 'data1',
    values: [
      {
        id: 'data1',
        index: 1,
        value: 1,
        x: 0
      },
      {
        id: 'data1',
        index: 2,
        value: 2,
        x: 1
      },
      {
        id: 'data1',
        index: 3,
        value: 3,
        x: 2
      }
    ]
  },
  {
    id: 'data2',
    id_org: 'data2',
```

ARCHITECTURE

```
values: [
  {
    id: 'data2',
    index: 1,
    value: 10,
    x: 0
  },
  {
    id: 'data2',
    index: 2,
    value: 20,
    x: 1
  },
  {
    id: 'data2',
    index: 3,
    value: 30,
    x: 2
  }
],
{
  id: 'data3',
  id_org: 'data3',
  values: [
    {
      id: 'data3',
      index: 1,
      value: 15,
      x: 0
    },
    {
      id: 'data3',
      index: 2,
      value: 25,
      x: 1
    },
    {
      id: 'data3',
      index: 3,
      value: 35,
      x: 2
    }
  ]
}
```

The targets will later be bound to the chart, meaning adaptations to `targets` will be reflected in the chart itself.

CHAPTER 6. IMPLEMENTATION

6.2.2.2 Chart-components

After the targets are set, and sizes and scales are updated, thus now knowing all of the dimensions for the chart components, the various area's making up the chart are appended.

First a main SVG element is appended to the division in which the chart should appear. Next clipping paths¹ for the chart, x-axis, y-axis, grid, and subchart are defined in a `defs`² element.

A first grouping (`<g>`) is added, defining the main region for the chart. The `transform` attribute specifies a transformation applied to all the children in the grouping, corresponding to a set margin. This way, later on there is no need to keep in mind these margins.

The second grouping is for a possible sub-chart, and a third grouping for the legend.

A minimal example of the result, displaying the hierarchy, is shown below.

```
1 <div id="chart" class="c3" style="...">
2   <svg style="...">
3     <defs>
4       <clippath id="c3-...-clip"></clippath>
5       <clippath id="c3-...-clip-xaxis"></clippath>
6       <clippath id="c3-...-clip-yaxis"></clippath>
7       <clippath id="c3-...-clip-grid"></clippath>
8       <clippath id="c3-...-clip-subchart"></clippath>
9     </defs>
10    <g transform="translate(..., ...)"></g>
11    <g transform="translate(..., ...)" style="..."></g>
12    <g transform="translate(..., ...)"></g>
13  </svg>
14 </div>
```

The main region then is appended with groupings for the different subsections of the actual chart:

- A text element, used if no data can be shown.
- A zoom rectangle, on which the zoom events can be specified.
- A regions grouping, in which the regions will be defined.
- A grid grouping, for standard grid-lines
- A chart grouping, contain groupings for chart elements, e.g. bars, , lines, arcs, ...

¹The clipping path restricts the region to which paint can be applied. Conceptually, any parts of the drawing that lie outside of the region bounded by the currently active clipping path are not drawn.

²SVG allows graphical objects to be defined for later reuse. Defining these elements inside of a `defs` element promotes understandability of the SVG content and thus promotes accessibility.

- A grid-lines grouping, for extra grid-lines definitions
- A x-axis grouping, containing all x-axis elements
- A y-axis grouping, containing all y-axis elements
- A y2-axis grouping, containing all y2-axis elements

```

1 <g transform="translate(...,...)">
2   <text class="c3-text c3-empty" ...></text>
3   <rect class="c3-zoom-rect" ...></rect>
4   <g class="c3-regions" clip-path="url(...-clip)" ...></g>
5   <g class="c3-grid" clip-path="url(...-clip-grid)" ...></g>
6   <g class="c3-chart" clip-path="url(...-clip)"></g>
7   <g class="c3-grid c3-grid-lines" clip-path="url(...-clip-
   grid)"></g>
8   <g class="c3-axis c3-axis-x" clip-path="url(...-clip-xaxis)
   " ...></g>
9   <g class="c3-axis c3-axis-y" clip-path="url(...-clip-yaxis)
   " ...></g>
10  <g class="c3-axis c3-axis-y2" ...></g>
11 </g>

```

These groupings are bound to the clipping paths defined earlier via the clip-path property.

As mentioned above, the chart grouping will contain groupings for the chart elements. Giving us an apparent structure for making selections on the elements we later on manipulate.

The order in which these groupings are defined is important for showing multiple types of data presentations in one chart, e.g. a bar chart for the member data1, with an overlapping line chart for data2. In SVG the z-index is defined by the order the elements appear in the document.

Likewise, transforming a member into a different type can then be as simple as removing it from one grouping and adding it to the other.

```

1 <g class="c3-chart" clip-path="...-clip)">
2   <g class="c3-event-rects c3-event-rects-single" ...></g>
3   <g class="c3-chart-bars"></g>
4   <g class="c3-chart-lines"></g>
5   <g class="c3-chart-arcs" ...></g>
6   <g class="c3-chart-texts"></g>
7 </g>

```

The event rectangles grouping (c3-event-rects) is used to define mouse events in particular area's of the chart, e.g. showing a tool-tip container when hovering over a bar.

In the other groupings then there will be a sub-division for each member, which accordingly will contain the specific elements making up for the distinct components of the chart type.

CHAPTER 6. IMPLEMENTATION

In case of the bar type, this being the most straightforward one i.e. the only components are bars, thus needing only the shapes forming these bars.

```
1 <g class="c3-chart-bars">
2   <g class="c3-chart-bar c3-target c3-target-data1" ...></g>
3   <g class=" c3-shapes c3-shapes-data1 c3-bars c3-bars-
4     data1" ...>
5     <path class=" c3-shape c3-shape-1 c3-bar c3-bar-1"
6       ...></path>
7     <path class=" c3-shape c3-shape-2 c3-bar c3-bar-2"
8       ...></path>
9     <path class=" c3-shape c3-shape-3 c3-bar c3-bar-3"
10      ...></path>
11   </g>
12   <g class="c3-chart-bar c3-target c3-target-data2" ...></g>
13   <g class="c3-chart-bar c3-target c3-target-data3" ...></g>
14 </g>
```

An area chart, for example, is a composition of lines, areas, circles and selected circles. Hence when this type of chart is drawn, each shape is grouped according to their embodiment and the member they constitute.

It is only when this preparatory work is done, targets are able to be added to the chart. As a result of the distinct separation in type, means to do this will differ. Given our example, we will only discuss how this is done for the bar type.

```
1 updateBar = function (durationForExit) {
2   var $$ = this,
3       barData = $.barData.bind($$),
4       classBar = $.classBar.bind($$),
5       initialOpacity = $.initialOpacity.bind($$),
6       isHiddenDataPoint = $.isHiddenDataPoint.bind($$),
7       color = function (d) { return $.color(d.id); };
8
9   $.mainBar = $.main.selectAll('.' + CLASS.bars).selectAll(
10     '.' + CLASS.bar)
11     .data(barData);
12   $.mainBar.enter().append('path')
13     .attr("class", classBar)
14     .style("stroke", color)
15     .style("fill", color);
16   $.mainBar
17     .style("opacity", initialOpacity);
18   $.mainBar.filter(isHiddenDataPoint)
19     .style("opacity", 0)
20   $.mainBar.exit().transition().duration(durationForExit)
21     .style('opacity', 0)
22     .remove();
23 };
```

Line 9 of `updateBar` shows how the selection for the bars is made.

We then bind the data to the selection. `barData` is a function that will test for a given member if it is set for the bar type and will return the values of the member. If not a bar type, an empty array will be returned.

D3 provides a general update pattern, where a data-join is followed by operations on the *enter*, *update* and *exit* selections.

As can be seen, the *enter* selection appends a path element to the selection, whilst the *exit* selection uses a transition to first fade-out and eventually remove it.

The update selection only sets the opacity.

At this stage the bars them-self are not drawn yet. Drawing will be done in the `redraw` function. This function is placed at the centre of the framework. Every time an action occurs, it is this function that is called to update the chart accordingly.

6.2.3 The Core

The core of the framework is the `redraw` function. Whenever an adaptation is made to the chart, meaning there is a change in how data is represented, whether it is data gets loaded or unloaded, a (de)selection is made, a chart is resized, ... Eventually `redraw` will be called upon.

On account of the length and all the services this function provides, we will not go into detail of all the steps that are processed. Though we will concentrate on the more important parts for our example.

When calling the `redraw` function some options are available as arguments. With these options one can state if sub-chart, axes, legend, dimensions, etc. should be updated or not. Likewise for transitions, making it possible for `redraw` to be used in every situation, whether transitions are needed or not, or even when only small parts of the chart should be updated.

Every aspect of the chart will reach its turn, if stated so.

As mentioned before in our example, the bars them-self are still to be drawn. With a custom visualisation tailored to a specific dataset this is pretty straightforward. However, as we don't know any specifics in advance this becomes a little more complicated.

As with the `updateBar` function, for every shape, there also exists a separate `redraw` function. These functions use the selections made and stored in their respective update functions, for bars this is `mainBar`, and return them to be executed later in `redraw`.

CHAPTER 6. IMPLEMENTATION

```
1 redrawBar = function (drawBar, withTransition, delay) {
2     var n = this.getMaxDataCount();
3     return [
4         (withTransition ?
5           (delay ?
6             this.mainBar.transition().delay(function (d, i)
7               { return i / n * delay; })
8             : this.mainBar.transition() )
9         : this.mainBar).filter(this.isShownDataPoint.bind(
10           this))
11         .attr('d', drawBar)
12         .style("fill", this.color)
13         .style("opacity", 1)
14     ];
15 }
```

As we can see, depending on what is needed, a selection or transition with or without delay is returned.

The shape of an SVG Path element is defined by the *d* attribute. This attribute should contain a series of commands and parameters in the SVG Path Mini-Language.

The parameter, *drawBar*, on the other hand, is storing a generated function which calculates how a bar is drawn and returns the path as stated in the SVG Path Mini-Language.

As mentioned, these selections aren't directly executed. This is on account of the possibility transitions are being used. If this is the case, we want them all to start at the same time.

redraw finally will gather the selections and execute them.

6.2.4 Interactions

When discussing the chart components we already mentioned a couple of places where events could be defined to handle interactions. The following interactions are available, intuitively, by hovering, clicking and/or dragging with the mouse-pointer in a chart.

Select and Highlight

When selection is enabled, data points can be selected by clicking on them. By setting grouped selections enabled, multiple data points that have the same x-value will also be selected. When it is demanded only one data point can be selected, the multiple data points selection can be disabled. This way when a new data point is selected, others will be unselected.

Also possible by means of a function, the determination for each data point can be made if it is selectable or not.

By default, selecting is done by clicking on a data point. When desired, it is also possible to make selections by dragging. If this option is set, scrolling

on the chart will be disabled because the dragging event will be handled now for selection.

Query

While hovering over a data point, a tool-tip is shown visualising the exact value of the data point. The tool-tip can be grouped, at this point values of all data points with the same x-value will be shown.

Filter

While hovering over the name of a specific member in the legend, if shown, all values for this member will be highlighted, i.e. all other values will get translucent. When clicking on the legend item it reverts the state from shown to hidden or vice versa.

Zoom

If zooming is enabled, scrolling with the mouse-wheel will zoom in and out. Dragging will then pan across the chart.

When desired, a sub-chart can also be used for zooming and panning.

6.2.5 API

Not all interaction can be directly accomplished through mouse inputs. For this the framework provides an API to carry out these interactions by other means. In our MindXpres plug-in we use the API to define an interface for easy access to these interactions.

- **focus, defocus, revert**: used for highlighting and fading out specific members.
- **show, hide, toggle**: showing and hiding specific members.
- **showpoints, hidepoints, filtered**: filter in/out specific data points.
- **load, unload, flow**: load/unload data.
- **select, unselect, selected**: selecting data points.
- **transform**: transform members to a specified type.
- **groups**: group members together, used for stacking.
- **sort**: sorting the set via the values of a specific member.
- **xgrids, xgrids.add, xgrids.remove**: update x-grid lines.
- **ygrids, ygrids.add, ygrids.remove**: update y-grid lines.
- **regions, regions.add, regions.remove**: update regions.
- **data, data.shown, data.values, data.names, data.colors, data.axes**: data manipulators.
- **category, categories**: update categories.

CHAPTER 6. IMPLEMENTATION

- **x, xs**: update x values.
- **axis.labels, axis.min, axis.max, axis.range**: manipulate axes.
- **legend.show, legend.hide**: specifying if a member is shown in the legend.
- **tooltip.show, tooltip.hide**: turn tool-tip on or off.
- **zoom, unzoom, zoom.enable**: zoom manipulations.
- **resize**: resize the chart to a given height and width.
- **flush**: forces a redraw.
- **destroy**: reset the chart object and removes elements and events completely.

6.2.6 Interface

As mentioned above, not all interactions can be directly achieved via mouse events.

Some will interfere with others, e.g. dragging for selection and zooming, and thus must be able to be switched between. Others, e.g. sorting, are just not defined at all via mouse events.

As a result, we've also integrated an interaction menu to carry out these tasks.

A pop-up button is dedicated for enabling and disabling interactions, these include selecting, highlighting, querying, filtering, and zooming.

A second button is dedicated for transformations. Here we can set all members to one type, or select a specific member for a type.

A third button is used for filtering data points, here we can select per member which values should be shown or not.

A grouping of 2 extra buttons is used to traverse over a predefined set of interactions. One allows us to take the next step in the traversal, while the other resets us back to the beginning.

Finally, there is also an editing menu. Even-though all interactions are now available. The need to adapt a chart in a more detailed manner could always arise. This menu can also be used for authoring. In this fashion, the author is directly able to see his changes and export the configuration to be set in the presentation.

6.2.7 Basic Settings

Even though authoring can be done via the interface. A chart still needs to be defined in the presentation before editing is possible. We will now shortly overview a few basic settings to set-up a chart. A full list with descriptions

can be found in appendix A on page 88.

First of all a chart needs to be appended to a division. The plug-in will automatically handle this when a chart is set in the presentation.

A second necessity is the dataset being visualised. There are 2 ways for doing this; we can enter the data in the presentation or we can load it from a source.

Lastly, we must also define what type of chart we will be using. By default a line chart is set. Thus even though this is not essential, directly starting with the correct type might be desirable.

When preferred, it is also possible to disable the editing menu and or the interaction menu.

6.3 Integration in MindXpres

In section 5.3, we described how the plug-in mechanism of MindXpres works. We now use this mechanism to integrate our files into MindXpres to be able to define charts in presentations.

Since our plug-in does not depend on libraries to be integrated upon initialisation of the presentation, adding it to MindXpres is as simple as counting to four.

1. Create a new folder in the components folder.
2. Add the JavaScript files that contain the plug-in information and the plug-in object following the defined naming convention of MindXpres.
3. Register and include the dependencies of the plug-in and place this in `dependencies.js`
4. Register and include the plug-in with the dependency injector.

MindXpres will now include this plug-in whenever it encounters the defined tag in the HTML document.

7

Use Case

In chapter 5 we introduced the MindXpres presentation tool. Together with all our insights that we gained from chapters 1 to 4, we started with the implementation which is explained in chapter 6. The time restrictions implied by this thesis did not allow us to evaluate our implemented solution by means of user trials and such. In this chapter we walk through a scenario whereby a presenter narrates over a simple dataset, highlighting the benefits for our solution. Additionally, this chapter provides insights on how the implementation will work from a practical point of view. By documenting the scenario, the reader will get a feeling on how interactive charts can be used, and what it looks like.

7.1 The scenario

In section 3.4 we have mentioned 3 different scenarios. For this example we set the scene for a live presentation by a speaker in front of an audience. The speaker will do his presentation, and at the end there is some time left for an open exploration part.

The subject is the results of a study to compare the tax and social burdens of salaried employees in the 27 Member States of the European Union.

Due to his talk being held in front of a Belgian audience he will be focusing on the taxes in Belgium.

In his presentation he introduces a chart showing what employers must spend to pay each net euro to an employee. It is the narration of this chart we will discuss.

7.2 Chart Set-up

The dataset used for this chart can be found in appendix A on page 94. The chart is defined with the following settings:

```

Size Height:      550
Size Width:       1000
Data URL:         /data/employercosts.csv
Data Type:        bar
Data Groups:      [{"1 Euro Income", "Employer Social
                    Security", "Income Tax", "Employee
                    Social Security"}]
Data Order:       null
Data Hide:        ["Total", "Employer Social
                    Security", "Income Tax", "Employee
                    Social Security"]
Data Hidepoints:  {'1 Euro Income': [0, 2-27]}
Data x:           Country
Axis x Type:      category
Axis Rotated:     true
Legend Hide:      ["Total"]
Zoom Enabled:     true
Tooltip Grouped:  true
Bar Width Max:    80

```

This configuration gives us a horizontal bar chart. The grouping that is set provides for stacking, and data order ensures stacking is done via data definition. Even though the member `Total` is being loaded with the data, it is set to be hidden in the chart and the legend.

The country names are set to be used as axis data labels.

Every member is set to be hidden except for `1 Euro Income`. All data points, except the one for Belgium, are also hidden. Though this is only set for one member, due to them being grouped, all others within the group will also be hidden.

7.3 Narration

When the presenter, in his presentation, arrives at the chart he introduces the audience to the study to compare the tax and social burdens of salaried employees in the 27 Member States of the European Union.

The chart only shows the 1 euro net for Belgium (Figure 7-1), thus even if the audience pays more attention to the chart rather instead of the presenter. The distraction will not be lengthy.



Figure 7-1: Use Case Chart: Starting point.

The presenter clicks on the label for Employer Social Security. He then sets about explaining what employer contribution to social security is, what the typical cost range is, and the amount that is set in Belgium. (Figure 7-2)



Figure 7-2: Use Case Chart: Explanation of Employer Social Security.

Next he makes the income tax visible and starts explaining its definition and the correlation with social security costs. (Figure 7-3) Again he introduces only one item to the chart, keeping the attention of the audience to himself and his commentary.

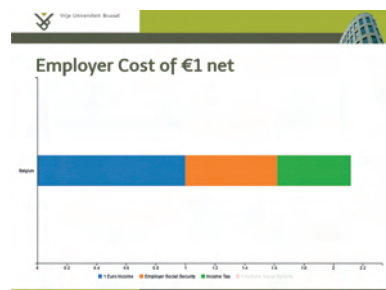


Figure 7-3: Use Case Chart: Explanation of Income Tax.

CHAPTER 7. USE CASE

After this he enables the individual contributions to social security and explains what this is used for, e.g. allowances in event of sickness, unemployment benefits, etc. He then again shortly revisits the numbers for Belgium, visualising this by hovering his mouse-pointer over the bar. (Figure 7-4)



Figure 7-4: Use Case Chart: Explanation of Employee Social Security and all results for Belgium highlighted.

At this point we notice how the speaker introduces the different values to the chart in a step by step fashion. By doing this, he ensures the audience is not distracted by what is being visualised. If he for instance directly had shown all of the results, audience members would be inclined to study the visualisation, therefore perhaps missing out on some of the definitions and examples given by the presenter.

Moving on with his presentation, the presenter then transforms the visualisation into a pie chart. By doing this he emphasizes on the fact that the net total a employee receives only amounts to 42,7% of the entire cost for the employer. (Figure 7-5)



Figure 7-5: Use Case Chart: Belgian results in percentages.

Of course to a layman this doesn't mean anything without something to compare it with. The presenter therefore transforms the chart back to a bar chart, and introduces the EU Averages. (Figure 7-6)

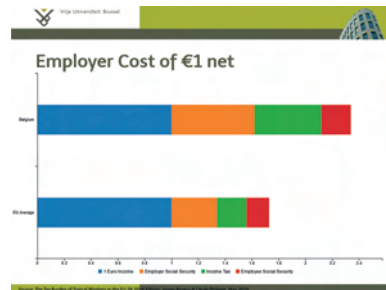


Figure 7-6: Use Case Chart: Comparison with the EU Average.

The audience is able to quickly notice the difference between the costs in Belgium and the EU Average. After going over the detailed numbers of the average European country, the presenter now introduces all other countries. (Figure 7-7) Now he has finished with his explanation on the case for Belgium, he can give the audience a moment to soak in all data being showed. Though, only for a second.



Figure 7-7: Use Case Chart: Comparison with all EU countries.

The presenter now has the ability to zoom in by scrolling with the mouse wheel. Dragging will pan across the chart. This way he can look for an interesting country to discuss in detail. He picks out Cyprus, the country with the lowest employer cost in total, and sets on explaining why this is so. (Figure 7-8)

To conclude his narration of this comparison, he eventually sorts the chart in ascending order by the totals of each country. Giving an overview of most to least expensive country, by means of employer costs, from top to bottom. (Figure 7-9)

Finally the presenter points out the top 3 counties in this list and carries on with the rest of his presentation.

CHAPTER 7. USE CASE



Figure 7-8: Use Case Chart: Exploration of Cyprus.



Figure 7-9: Use Case Chart: All results sorted descending on total.

And the end of the entire presentation, a moment for questions is scheduled. When questions arise on the employer costs for a specific country, the presenter can navigate back to the chart, Zoom in on the country in question, show the values en set about answering the question.

7.4 Conclusion

In this chapter, we demonstrated the usage of our chart plug-in in MindXpres by means of a simple scenario.

We have shown how a presenter can narrate a story whilst introducing data into a chart without it being to overwhelming for the audience to get distracted by. Furthermore, it illustrates changing the type for a chart can shift the emphasis the visualisation expresses. It also shows the advantage of zooming and scrolling over a large dataset, in which not all details can be showed from the start. Likewise, we have indicated the need for only one chart for a certain dataset. Not bulking up the presentation with excessive backup charts.

The reader must also note that what we demonstrate in this use case represents just a limited showcase of what is possible with our plug-in.

8

Conclusion and Future Work

In this final chapter, we would like to recapitulate a number of contributions that have been made in this thesis. Additionally, we outline and discuss some future directions for our chart plug-in and information visualisations in presentation tools in general.

8.1 Contributions

The literature study that has been carried out as part of this thesis shows that there is room for improvements in the field of data visualisation for presentation tools. The market of presentation solutions is still dominated by the classic slide-ware tools that have set the standard since the introduction of digital presentation tools. While there are alternative approaches, one aspect in these tools is most of the time overlooked.

Whenever it comes to presenting data, common, state of the art, and alternative presentation solutions seem to declare static visualisations will suffice. Nevertheless, by analysing and studying these tools, we were able to learn valuable lessons with regard to alternative forms of data presenting, and how they relate to the limitations of classic presentation tools.

By combining the aspects that we believe to be beneficial for data presentation with our own ideas on how visualisations should act, we came up with a plug-in for MindXpres, reapplying known techniques for information visualisation and narrative visualisation into the domain of presentation tools.

CHAPTER 8. CONCLUSION AND FUTURE WORK

During this thesis, we made the following steps and contributions:

- A literature study has been conducted in order to identify problematic areas in of information visualisation in common presentation tools. Additionally, we have taken a closer look at state of the art as well as alternative presentation solutions in order to use some of the findings in our work.
- Based on the insights gained from the literature study and on the investigation of existing presentation tools, we broadened our view in order to explore concept and techniques that might be beneficial in the context of presenting data, and help to overcome some of the limitations of existing presentation tools.
- Based on the techniques found in the fields of interactive and narrative visualisations we illustrated how they could be reapplied in the domain of presenting for an audience.
- Finally, we have implemented a functional prototype of our plug-in, supporting a basic set of chart types to be visualised. Together with interactions and transitions providing an open approach to presenting data in front of an audience.

8.2 Future Work

In this section we suggest and discuss possible expansions and improvements of the results presented in this thesis.

8.2.1 User Study

Though there is a plenitude of articles suggesting story telling could help with data presentation, except for a study by Robertson et al. [47] on animated transitions and staging, there is, as of yet, no body of existing work saying interactions and transitions would benefit to presentation tools in order to present data.

However, we believe with our plug-in as a proof of concept there is reason to do more research in this area.

We therefore suggest a user study, on both ends of the presentation, i.e. for the author/presenter and the audience, to clarify the benefits as well as the disadvantages for using interactive charts during a presentation.

8.2.2 Visualisation types

For now we only support a basic set of chart types to visualise. Although they already give us a conclusive idea on how they can be interactively used,

more advanced types, e.g. geographic maps, will contribute more arguments for the use of interactions.

8.2.3 Linking visualisations

Though the foundation for Brush and link is present, for now this interaction is not implemented. As a basis, we focused on the assumption only one chart is visible at a time during interaction. Though when trying to fully support the techniques used in narrative visualisation, it is essential to be able to link multiple charts and have interactions in one chart effect the others.

8.2.4 Chart traversal

While there is a support system available for linear chart traversal, i.e. step by step executing pre-defined interactions, it is not an optimal solution. Because of time limitations for this thesis we were not able to fulfil this aspect to its full potential. However we felt to have it at least implemented to show how one could resume the traversal, even after being interrupted and having to change some aspects of the chart.



Tables and images

APPENDIX A. TABLES AND IMAGES

Table A-1: How Big is an Exabyte? Source: How Much Information? [62]

Kilobyte (KB)	1 000 bytes OR 10^3 bytes 2 kB: A Typewritten page. 100 kB: A low-resolution photograph.
Megabyte (MB)	1 000 000 bytes OR 10^6 bytes 1 MB: A small novel OR a 3.5 inch floppy disk. 2 MB: A high-resolution photograph. 5 MB: The complete works of Shakespeare. 10 MB: A minute of high-fidelity sound. 100 MB: 1 meter of shelved books. 500 MB: A CD-ROM.
Gigabyte (GB)	1 000 000 000 bytes OR 10^9 bytes 1 GB: a pickup truck filled with books. 20 GB: A good collection of the works of Beethoven. 100 GB: A library floor of academic journals.
Terabyte (TB)	1 000 000 000 000 bytes OR 10^{12} bytes 1 TB: 50 000 trees made into paper and printed. 2 TB: An academic research library. 10 TB: The print collections of the U.S. Library of Congress. 400 TB: National Climactic Data Center (NOAA) database.
Petabyte (PB)	1 000 000 000 000 000 bytes OR 10^{15} bytes 1 PB: 3 years of EOS data (2001). 2 PB: All U.S. academic research libraries. 20 PB: Production of hard-disk drives in 1995. 200 PB: All printed material.
Exabyte (EB)	1 000 000 000 000 000 000 bytes OR 10^{18} bytes 2 EB: Total volume of information generated in 1999. 5 EB: All words ever spoken by human beings.

Table A-2: Chart Types supported by Microsoft PowerPoint.

Column chart

- (3D) Clustered column
- (3D) Stacked column
- (3D) 100% stacked column
- 3D column

Pie and doughnut chart

- (3D) Pie
- Pie of pie
- Bar of pie
- Doughnut

Area chart

- (3D) Area
- (3D) Stacked area
- (3D) 100% stacked area

Stock chart

- High-low-close
- Open-high-low-close
- Volume-high-low-close
- Volume-open-high-low-close

Radar chart

- Radar (with markers)
- Filled radar

Line chart

- Line (with markers)
- Stacked line (with markers)
- 100% stacked line (with markers)
- 3D line

Bar chart

- (3D) Clustered bar
- (3D) Stacked bar
- (3D) 100% stacked bar

XY (scatter) and bubble chart

- Scatter
- Scatter with smooth lines (and markers)
- Scatter with straight lines (and markers)
- (3D) Bubble

Surface chart

- 3D surface
- Wireframe 3D surface
- Contour
- Wireframe contour

Combo chart

- Clustered column - line (on secondary axis)
- Stacked area - clustered column
- Custom combination

APPENDIX A. TABLES AND IMAGES



Figure A-1: Claudius Ptolemy: The World, Source: Scanned by Scott Ehardt from Decorative Maps by Roderick Barron.

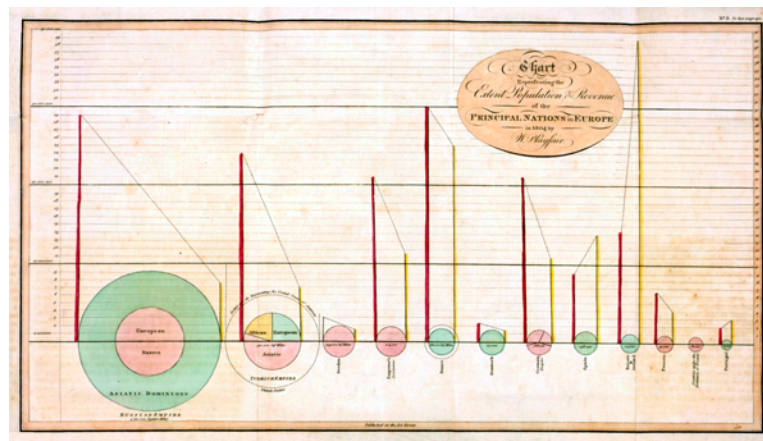


Figure A-2: A data visualization by William Playfair from his book *The Commercial and Political Atlas and Statistical Breviary*. It shows population size versus land area, all compared to taxation.

Table A-3: Information Visualisation Taxonomies Relevant to Interaction Techniques. Source: Yi et al. (2007) [118]

Publication	Taxonomic Units
<i>Taxonomies of low-level interaction techniques</i>	
Shneiderman (1996) [82]	Overview, zoom, filter, details-on-demand, relate, history, and extract
Buja et al. (1996) [15]	Focusing (choice of [projection, aspect ratio, zoom, pan], choice of [variable, order, scale, scale-aspect ratio, animation, and 3-D rotation]), linking (brushing as conditioning/sectioning/database query), and arranging views (scatter plot matrix and conditional plot)
Chuah and Roth (1996) [23]	Basic visualisation interaction (BVI) operations: graphical operations (encode data, set graphical value, manipulate objects), set operations (create set, delete set, summarize set, other), and data operations (add, delete, derived attributes, other)
Dix and Ellis (1998) [28]	Highlighting and focus, accessing extra information – drill down and hyper-links, overview and context, same representation/changing parameters, same data/changing representation, linking representation – temporal fusion
Keim (2002) [54]	Dynamic projections, interactive filtering, interactive zooming, interactive distortion, interactive linking and brushing
Wilkinson (2006) [113]	Filtering (categorical/continuous/multiple/fast filtering), navigating (zooming/panning/lens), manipulating (node dragging/categorical reordering), brushing and linking (brush shapes/brush logic/fast brushing), animating (frame animation), rotating, transforming (specification/assembly/display/tap/2 taps/3 taps)
<i>Taxonomical dimensions of interaction techniques</i>	
Tweedie (1997) [105]	Interaction types (manual, mechanized, instructable, steerable, and automatic) and directness (direct and indirect manipulation)
Spence (2007) [89]	Interaction modes (continuous, stepped, passive, and composite interaction)
<i>A taxonomy of interaction operations</i>	
Ward and Yang (2004) [109]	interaction operators (navigation, selection, distortion), interaction spaces (screen-space, data value-spaces, data structure-space, attribute-space, object-space, and visualization structure-space), and interaction parameters (focus, extents, transformation, and blender)
<i>Taxonomies of user tasks</i>	
Zhou and Feiner (1998) [119]	Relational visual tasks (associate, background, categorize, cluster, compare, correlate, distinguish, emphasize, generalize, identify, locate, rank, reveal, switch) and direct visual organizing and encoding tasks (encode)
Amar et al. (2005) [4]	Retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, and correlate

APPENDIX A. TABLES AND IMAGES

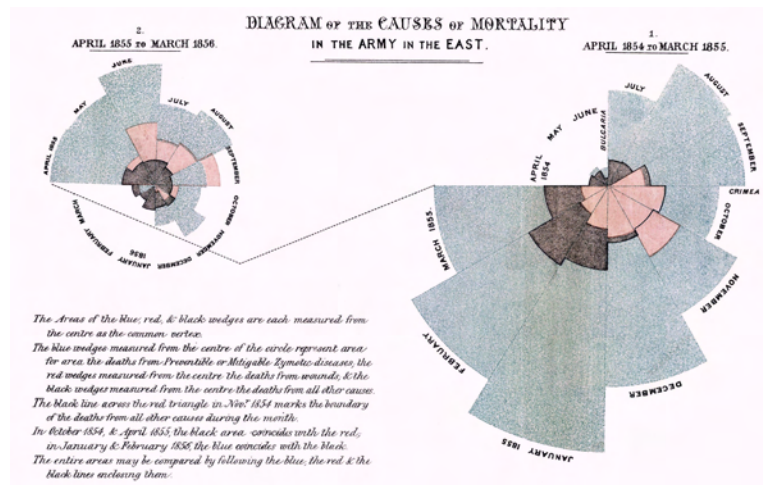


Figure A-3: Example of polar area diagram by Florence Nightingale. This “diagram of the causes of mortality in the army in the East” was published in *Notes on Matters Affecting the Health, Efficiency, and Hospital Administration of the British Army* and sent to Queen Victoria in 1858. This graphic indicates the number of deaths that occurred from preventable diseases (in blue), those that were the results of wounds (in red), and those due to other causes (in black).

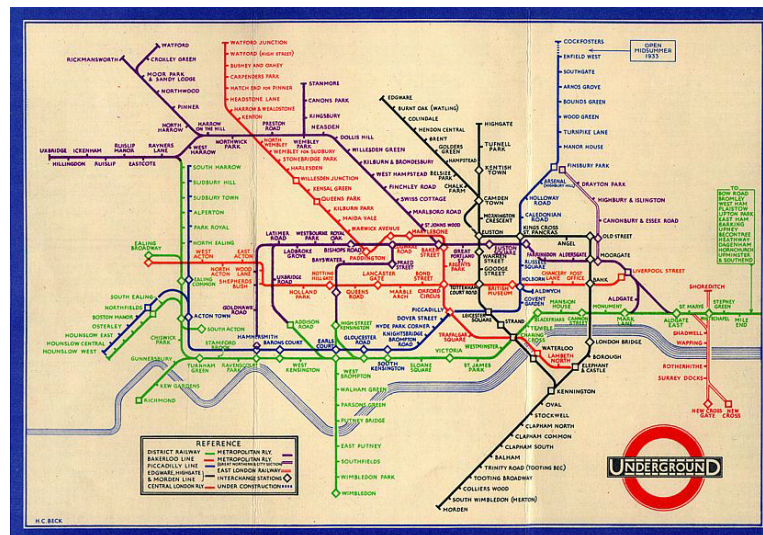


Figure A-4: Henry C. Beck, the London Underground Diagram (1933). Beck's map was ground-breaking for its time as it abandoned true geographic layout in favour of a style similar to electrical circuit-diagrams. It relied on the notion that the physical locations of the stations were irrelevant to the traveller wanting to know how to get to one station from another.

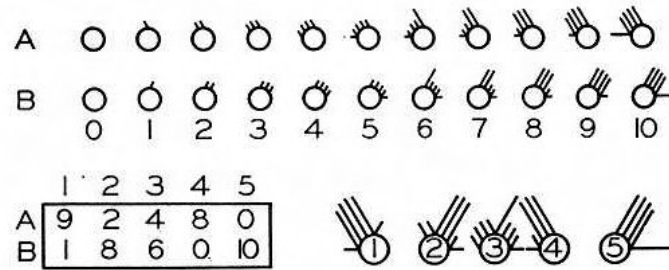


Figure A-5: Edgar Anderson, Circular glyphs, with rays to represent multi-variate data. Top: variables A and B are coded in values from 1 to 10 each. Lower left: the records of five individuals with respect to A and B. Lower right: the corresponding metroglyphs for each of the five individuals.

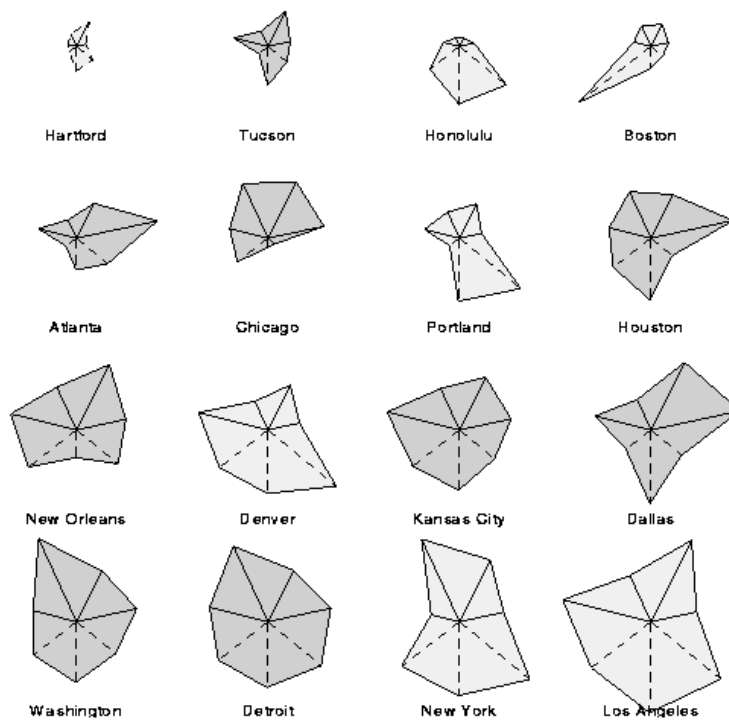


Figure A-6: Star plot of crime rates in US cities. Stars or circle diagrams, where each glyph consists of N lines emanating from a point at uniformly separated angles with lengths determined by the values of each dimension, with the endpoints connected to form a polygon.

APPENDIX A. TABLES AND IMAGES

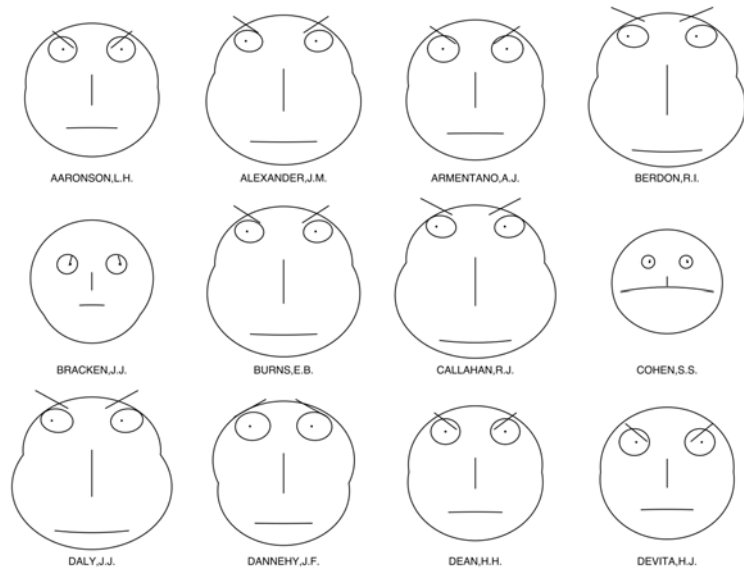


Figure A-7: Chart showing Chernoff faces for data selected from the “US-JudgeRatings” dataset in R, which contains ratings of state judges in the US Superior Court by lawyers who have had contact with them.

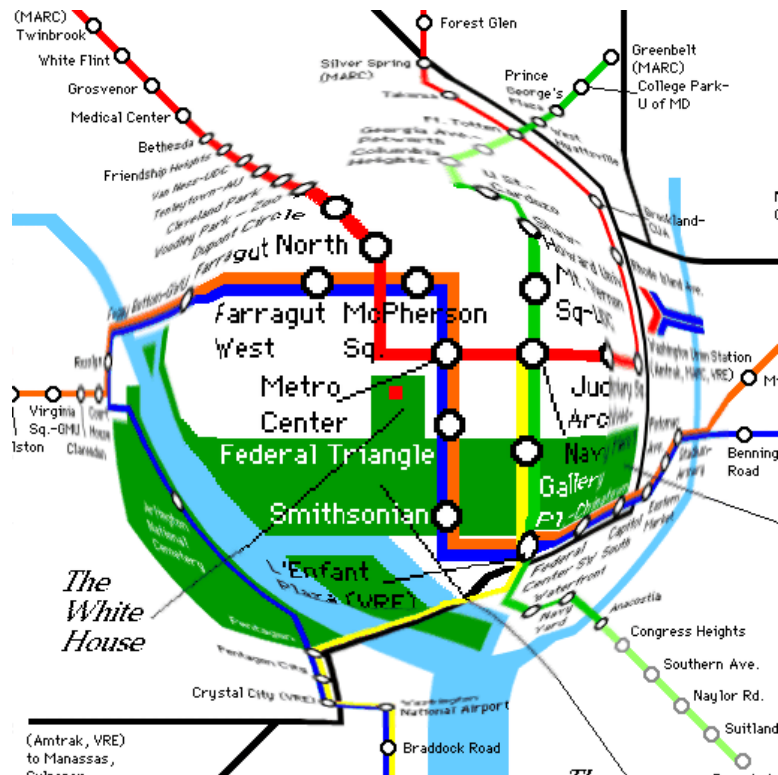


Figure A-8: Fish-eye view of central Washington, D.C.; The fish-eye strategy presents a way to present a small display of a large structure. These structures can be network trees, programs, entire systems, and databases. While the amount of information being presented to the user increases, the viewing space (i.e., computer screen) remains relatively small. This can present an interface design problem, for developers of systems. The idea is to present a large amount of data to users in a way that is searchable, and getting information is not too timely of a task.

APPENDIX A. TABLES AND IMAGES

Table A-4: Configuration options for charts (part 1)

Option	Description	Default
size.width	Set the width of the chart element. If not specified, width will be calculated by the size of the parent element.	
size.height	Set the height of the chart element. If not specified, height will be calculated by the size of the parent element.	
padding.top	Padding on the top of the chart.	
padding.right	Padding on the right of the chart.	
padding.bottom	Padding on the bottom of the chart.	
padding.left	Padding on the left of the chart.	
color.pattern	Set custom color pattern	
interaction.enabled	Indicate if the chart should have interactions.	true
transition.duration	Set duration of transition (in milliseconds) for chart animation.	350
oninit	Set a callback to execute when the chart is initialized.	
onrendered	Set a callback to execute when the chart is rendered (on every redraw).	
onmouseover	Set a callback to execute when mouse enters the chart.	
onmouseout	Set a callback to execute when mouse leaves the chart.	
onresize	Set a callback to execute when user resizes the screen.	
onresized	Set a callback to execute when screen resize is finished.	
data.url	Load CSV or JSON file from a URL.	
data.json	Parse a JSON object for data.	
data.rows	Load data from a multidimensional array. First element containing the data names, following containing related data.	
data.columns	Load data from a multidimensional array. Each element containing an array consisting of a datum name and associated data values.	
data.mimeType	Used if loading JSON via data.url	
data.keys	Set which JSON object keys correspond to desired data.	
data.x	Specify the key of x values in the data.	

Table A-5: Configuration options for charts (part 2)

Option	Description	Default
data.xs	Specify the keys of the x values for each data.	
data.xFormat	Set a format to parse string specified as x.	%Y-%m-%d
data.names	Set custom data names.	
data.classes	Set custom data class.	
data.groups	Set groups for the data for stacking.	
data.axes	Set y axis. Data related to <i>y</i> and <i>y2</i> can be used.	
data.type	Set chart type at once. Applied to every data.	line
data.types	Set chart type for each data. Overwrites data.type setting.	
data.labels	Show labels on each data point.	false
data.labels.format	Set formatter function for data labels.	
data.order	Define the order of the data, relating to stacking and pieces of pie/donut.	desc
data.regions	Define regions for each data.	
data.color	Set color converter function	
data.colors	Set color for each data.	
data.hide	Hide each data when the chart appears	false
data.empty.label.text	Set text displayed when empty data.	
data.selection.enabled	Set data selection enabled.	false
data.selection.grouped	Set grouped selection enabled.	false
data.selection.multiple	Set multiple data points selection enabled.	false
data.selection.draggable	Enable to select data points by dragging. Disables scrolling.	false
data.selection.isselectable	Set a callback for each data point to determine if it's selectable or not.	
data.onclick	Set a callback for click event on each data point.	
data.onmouseover	Set a callback for mouse-over event on each data point.	

APPENDIX A. TABLES AND IMAGES

Table A-6: Configuration options for charts (part 3)

Option	Description	Default
data.onmouseout	Set a callback for mouse-out event on each data point.	
axis.rotated	Switch x and y axis position.	false
axis.x.show	Show or hide x axis.	true
axis.x.type	Set type of x axis.	indexed
axis.x.localtime	Set how to treat the time-zone of x values. True: treat as local-time. False: convert to UTC internally.	true
axis.x.categories	Set category names on category axis.	
axis.x.tick.centered	Centre ticks on category axis.	
axis.x.tick.format	Function to format tick value.	
axis.x.tick.culling	Setting for culling ticks.	
axis.x.tick.culling.max	The number of tick texts will be adjusted to less than this value.	10
axis.x.tick.count	Number of x-axis ticks to show.	
axis.x.tick.fit	Fit x-axis ticks.	true
axis.x.tick.values	Set the x-values of ticks manually.	
axis.x.tick.rotate	Rotate x-axis tick text.	0
axis.x.tick.outer	Show x-axis outer tick.	true
axis.x.max	Set max value of x-axis range.	
axis.x.min	Set min value of x-axis range.	
axis.x.padding	Set padding for x-axis.	
axis.x.height	Set height of x-axis.	
axis.x.extent	Set default extent for sub-chart and zoom.	
axis.x.label	Set label on x-axis.	
axis.x.label.position	Set position label on x-axis. Use axis.x.label.text for the text.	
axis.y.show	Show or hide y-axis.	true
axis.y.inner	Show y-axis inside of the chart.	false
axis.y.max	Set max value of y-axis.	
axis.y.min	Set min value of y-axis.	
axis.y.inverted	Change the direction of y-axis.	false
axis.y.center	Set centre value of y-axis.	
axis.y.label	Set label on y-axis.	
axis.y.label.position	Set position label on y-axis. Use axis.y.label.text for the text.	
axis.y.tick.format	Set formatter for y-axis tick text.	
axis.y.tick.outer	Show or hide outer tick.	
axis.y.tick.values	Set y-axis tick values manually.	
axis.y.tick.count	Set the number of y-axis ticks.	

Table A-7: Configuration options for charts (part 4)

Option	Description	Default
axis.y.padding	Set padding for y-axis.	
axis.y.default	Set default range of y-axis.	
axis.y2.show	Show or hide y2-axis.	false
axis.y2.inner	Show y2-axis inside of the chart.	false
axis.y2.max	Set max value of y2-axis.	
axis.y2.min	Set min value of y2-axis.	
axis.y2.inverted	Change the direction of y2-axis.	false
axis.y2.center	Set centre value of y2-axis.	
axis.y2.label	Set label on y2-axis.	
axis.y2.label.position	Set position label on y2-axis. Use axis.y2.label.text for the text.	
axis.y2.tick.format	Set formatter for y2-axis tick text.	
axis.y2.tick.outer	Show or hide y2-axis outer tick.	
axis.y2.tick.values	Set y2-axis tick values manually.	
axis.y2.tick.count	Set the number of y2-axis ticks.	
axis.y2.padding	Set padding for y2-axis.	
axis.y2.default	Set default range of y2-axis.	
grid.x.show	Show grids along x-axis.	false
grid.x.lines	Show additional grid lines along x-axis.	
grid.y.show	Show grids along y-axis.	
grid.y.lines	Show additional grid lines along y-axis.	
regions	Show rectangles inside the chart.	
legend.show	Show or hide legend.	true
legend.hide	Hide legend.	false
legend.position	Change the position of legend.	bottom
legend.inset	Change inset legend attributs.	
legend.item.onclick	Set click event handler to the legend item.	
legend.item.onmouseover	Set mouse-over event handler to the legend item.	
legend.item.onmouseout	Set mouse-out event handler to the legend item.	
tooltip.show	Show or hide tool-tip.	true
tooltip.grouped	Set if tool-tip is grouped or not for the data points.	true
tooltip.format.title	Set format for the title of tool-tip.	
tooltip.format.value	Set format for the value of each data in tool-tip.	
tooltip.position	Set custom position for the tool-tip.	

APPENDIX A. TABLES AND IMAGES

Table A-8: Configuration options for charts (part 5)

Option	Description	Default
tooltip.contents	Set custom HTML for the tool-tip.	
tooltip.format.name	Set format for the name of each data in tool-tip.	
subchart.show	Show sub-chart on the bottom of the chart.	
subchart.size.height	Change the height of the sub-chart.	
subchart.onbrush	Set callback for brush event.	
zoom.enabled	Enable zooming.	false
zoom.rescale	Enable to rescale after zooming.	false
zoom.extent	Change zoom extent.	[1, 10]
zoom.onzoom	Set callback that is called when the chart is zooming.	
zoom.onzoomstart	Set callback that is called when zooming starts.	
zoom.onzoomend	Set callback that is called when zooming ends.	
point.show	Whether to show each point in line.	true
point.r	The radius size of each point.	2.5
point.focus.expand.enabled	Whether to expand each point on focus.	true
point.focus.expand.r	The radius size of each point on focus.	point.r * 1.75
point.select.r	The radius size of each point on selected.	point.r * 4
line.connectNull	Set if null data point will be connected or not.	false
line.step_type	Change step type for step chart.	step
area.zerobased	Set if min or max value will be 0 on area chart.	true
bar.width	Change the width of bar chart.	auto
bar.width.ratio	Change the width of bar chart by ratio.	0.6
bar.zerobased	Set if min or max value will be 0 on bar chart.	true

Table A-9: Configuration options for charts (part 6)

Option	Description	Default
donut.label.show	Show or hide label on each donut piece.	true
pie.label.show	Show or hide label on each pie piece.	true
pie.label.format	Set formatter for the label on each pie piece.	
pie.label.threshold	Set threshold to show/hide labels.	0.05
pie.expand	Enable or disable expanding pie pieces.	true
donut.label.format	Set formatter for the label on each donut piece.	
donut.label.threshold	Set threshold to show/hide labels	0.05
donut.expand	Enable or disable expanding donut pieces.	true
donut.width	Set width of donut chart	auto
donut.title	Set title of donut chart	
gauge.label.show	Show or hide label on gauge.	true
gauge.label.format	Set formatter for the label on gauge.	
gauge.expand	Enable or disable expanding gauge.	true
gauge.min	Set min value of the gauge.	0
gauge.max	Set max value of the gauge.	100
gauge.units	Set units of the gauge.	
gauge.width	Set width of gauge chart.	auto

APPENDIX A. TABLES AND IMAGES

Table A-10: Employer Cost of 1 EUR net.

Source: The Tax Burden of Typical Workers in the EU 28 2014 Edition, James Rogers & Cécile Philippe, May 2014

Country	1 Euro Income	Employer Social Security	Income Tax	Employee Social Security	Total
Austria	1	0.5	0.32	0.29	2.11
Belgium	1	0.62	0.5	0.22	2.34
Bulgaria	1	0.22	0.11	0.16	1.49
Cyprus	1	0.07	0.03	0.07	1.17
Czech Republic	1	0.44	0.16	0.14	1.74
Denmark	1	0.01	0.59	0	1.6
Estonia	1	0.44	0.21	0.06	1.71
EU Average	1	0.34	0.22	0.17	1.73
Finland	1	0.32	0.29	0.11	1.72
France	1	0.72	0.09	0.35	2.16
Germany	1	0.33	0.32	0.35	2
Greece	1	0.38	0.12	0.21	1.71
Hungary	1	0.44	0.25	0.28	1.97
Ireland	1	0.13	0.17	0.04	1.34
Italy	1	0.45	0.35	0.14	1.94
Latvia	1	0.35	0.3	0.16	1.81
Lithuania	1	0.42	0.2	0.12	1.74
Luxembourg	1	0.2	0.21	0.17	1.58
Malta	1	0.13	0.14	0.13	1.4
Netherlands	1	0.3	0.27	0.24	1.81
Poland	1	0.27	0.09	0.3	1.66
Portugal	1	0.31	0.15	0.14	1.6
Romania	1	0.41	0.19	0.24	1.84
Slovakia	1	0.46	0.12	0.17	1.75
Slovenia	1	0.23	0.1	0.31	1.64
Spain	1	0.39	0.21	0.08	1.68
Sweden	1	0.42	0.33	0	1.75
United Kingdom	1	0.14	0.2	0.12	1.46

Bibliography

- [1] C. Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25(4):25–29, 1996.
- [2] M. Alley and K. A. Neeley. Discovering the power of powerpoint: rethinking the design of presentation slides from a skillful user’s perspective. In *Proceedings of the 2005 American society of engineering education annual conference and exposition, Portland, Oregon, June*, pages 12–15, 2005.
- [3] M. Alley and K. A. Neeley. Rethinking the design of presentation slides: A case for sentence headlines and visual evidence. *Technical communication*, 52(4):417–426, 2005.
- [4] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 111–117. IEEE, 2005.
- [5] E. Anderson. A semigraphical method for the analysis of complex problems. *Proceedings of the National Academy of Sciences of the United States of America*, 43(10):923, 1957.
- [6] M. Austin. *Useful fictions: evolution, anxiety, and the origins of literature*. U of Nebraska Press, 2011.
- [7] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk?: the effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2573–2582. ACM, 2010.
- [8] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: Enabling interactive multiple-view visualizations. In *Visualization, 2005. VIS 05. IEEE*, pages 135–142. IEEE, 2005.
- [9] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

BIBLIOGRAPHY

- [10] J. Bertin. *Graphics and graphic information processing*. Walter de Gruyter, 1981.
- [11] J. Bertin and S. Bonin. *La graphique et le traitement graphique de l'information*, volume 1. Flammarion Paris, 1977.
- [12] M. Boisot and A. Canals. Data, information and knowledge: have we got it right? *Journal of Evolutionary Economics*, 14(1):43–67, 2004.
- [13] M. Bostock. Towards reusable charts, February 2012.
- [14] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.
- [15] A. Buja, D. Cook, and D. F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.
- [16] S. K. Card, J. D. Mackinlay, and B. Schneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [17] B. Carlson, A. Burgess, C. Miller, and L. Bauer. Timeline of computing history. Retrieved October, 21:2006, 1996.
- [18] C. Chen. Information visualization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):387–403, 2010.
- [19] H. Chen. Compound brushing [dynamic data visualization]. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 181–188. IEEE, 2003.
- [20] M. Chen, D. Ebert, H. Hagen, R. S. Laramée, R. van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *Computer Graphics and Applications, IEEE*, 29(1):12–19, 2009.
- [21] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.
- [22] E. H.-h. Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 63–70. IEEE, 1998.
- [23] M. C. Chuah and S. F. Roth. On the semantics of interactive visualizations. In *Information Visualization'96, Proceedings IEEE Symposium on*, pages 29–36. IEEE, 1996.

BIBLIOGRAPHY

- [24] G. Conti, M. Ahamad, and J. Stasko. Attacking information visualization system usability overloading and deceiving the human. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 89–100. ACM, 2005.
- [25] B. Craft and P. Cairns. Beyond guidelines: what can we learn from the visual information seeking mantra? In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 110–118. IEEE, 2005.
- [26] H. Dawkes, L. A. Tweedie, and B. Spence. Vicki: the visualisation construction kit. In *Proceedings of the workshop on Advanced visual interfaces*, pages 257–259. ACM, 1996.
- [27] T. de Boer and K. Verkooij. Interactive visualization toolkits for rich internet publications.
- [28] A. Dix and G. Ellis. Starting simple: adding value to static visualisation through simple interaction. In *Proceedings of the working conference on Advanced visual interfaces*, pages 124–134. ACM, 1998.
- [29] J.-L. Doumont. The cognitive style of powerpoint: Slides are not all evil. *Technical communication*, 52(1):64–70, 2005.
- [30] P. Felten. Visual literacy. *Change: The magazine of higher learning*, 40(6):60–64, 2008.
- [31] S. Few. The chartjunk debate: A close examination of recent findings. *Visual Business Intelligent Newsletter*, 2011.
- [32] S. Few and P. EDGE. Data visualization: Past, present, and future. *IBM Cognos Innovation Center*, 2007.
- [33] B. A. Frey and D. J. Birnbaum. Learners’ perceptions on the value of powerpoint in lectures. 2002.
- [34] M. Friendly. A brief history of data visualization. *Handbook of Computational Statistics: Data Visualization*, 3, 2006.
- [35] M. Friendly and D. J. Denis. Milestones in the history of thematic cartography, statistical graphics, and data visualization. *Retrieved*, 5(11):2011, 2001.
- [36] H. G. Funkhouser. Historical development of the graphical representation of statistical data. *Osiris*, 3:269–404, 1937.
- [37] G. W. Furnas. The fisheye view: A new look at structured files. Technical report, Bell Laboratories Technical Memorandum, 1981.

BIBLIOGRAPHY

- [38] G. W. Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.
- [39] Y. Gabriel. Against the tyranny of powerpoint: technology-in-use and technology abuse. *Organization Studies*, 29(2):255–276, 2008.
- [40] K. Garland and H. C. Beck. *Mr Beck’s Underground Map*. Capital Transport, 1994.
- [41] N. Gershon and W. Page. What storytelling can do for information visualization. *Communications of the ACM*, 44(8):31–37, 2001.
- [42] J. J. Gibson. Observations on active touch. *Psychological review*, 69(6):477, 1962.
- [43] L. Good and B. B. Bederson. Zoomable user interfaces as a medium for slide show presentations. *Information Visualization*, 1(1):35–49, 2002.
- [44] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 127–130. IEEE, 2002.
- [45] M. Hearst. Information visualization: Principles, promise, and pragmatics. In *Handouts of the tutorial at CHI 2003 Conference on Human Factors in Computing Systems*, 2003.
- [46] J. Heer, S. K. Card, and J. A. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 421–430. ACM, 2005.
- [47] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1240–1247, 2007.
- [48] H. Hochheiser and B. Shneiderman. Using interactive visualizations of www log data to characterize access patterns and inform site design. *Journal of the American Society for Information Science and Technology*, 52(4):331–343, 2001.
- [49] J. Hsu. The secrets of storytelling. *Scientific American Mind*, 19(4):46–51, 2008.
- [50] J. Hullman and N. Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2231–2240, 2011.
- [51] A. Inselberg and B. Dimsdale. *Parallel coordinates for visualizing multi-dimensional geometry*. Springer, 1987.

BIBLIOGRAPHY

- [52] C. Johnson. Top scientific visualization research problems. *Computer graphics and applications, IEEE*, 24(4):13–17, 2004.
- [53] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods—a survey. *ACM Computing Surveys (CSUR)*, 39(4):10, 2007.
- [54] D. Keim et al. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, 2002.
- [55] D. A. Keim. Visual exploration of large data sets. *Communications of the ACM*, 44(8):38–44, 2001.
- [56] R. Kosara, H. Hauser, and D. L. Gresh. An interaction view on information visualization. *State-of-the-Art Report. Proceedings of EUROGRAPHICS*, 2003.
- [57] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. 2013.
- [58] G. Krippel, A. J. McKee, and J. Moody. Multimedia use in higher education: promises and pitfalls. *Journal of Instructional Pedagogies*, 2:1–8, 2010.
- [59] J. Lasseter. Principles of traditional animation applied to 3d computer animation. In *ACM Siggraph Computer Graphics*, volume 21, pages 35–44. ACM, 1987.
- [60] B. Lee, R. H. Kazi, and G. Smith. Sketchstory: Telling more engaging stories with data through freeform sketching. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2416–2425, 2013.
- [61] D. G. Levasseur and J. Kanan Sawyer. Pedagogy meets powerpoint: A research review of the effects of computer-generated slides in the classroom. *The Review of Communication*, 6(1-2):101–123, 2006.
- [62] P. Lyman and H. Varian. How much information 2003? 2004.
- [63] K.-L. Ma, I. Liao, J. Frazier, H. Hauser, and H.-N. Kostis. Scientific storytelling using visualization. *Computer Graphics and Applications, IEEE*, 32(1):12–19, 2012.
- [64] J. D. Mackinlay. Opportunities for information visualization. *Computer Graphics and Applications, IEEE*, 20(1):22–23, 2000.
- [65] G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

BIBLIOGRAPHY

- [66] E. Montalbano. Forrester: Microsoft office in no danger from competitors. *PC World*, 2009.
- [67] T. Munzner. Information visualization. *IEEE Computer graphics and applications*, 22(1):20–21, 2002.
- [68] S. Murray. *Interactive Data Visualization for the Web*. O’Reilly Media, 2013.
- [69] M. C. Norrie. An extended entity-relationship approach to data management in object-oriented systems. In *Entity-Relationship Approach—ER’93*, pages 390–401. Springer, 1994.
- [70] OOo Authors team. Differences in use between impress and powerpoint, 2004.
- [71] G. Palsky. *Des chiffres et des cartes: naissance et développement de la cartographie quantitative française au XIXe siècle*, volume 19. Comité des travaux historiques et scientifiques-CTHS, 1996.
- [72] I. Parker. Absolute powerpoint. *The New Yorker*, 28:76–87, 2001.
- [73] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 89–96. IEEE, 2004.
- [74] G. Reynolds. *Presentation Zen: Simple ideas on presentation design and delivery*. New Riders, 2011.
- [75] T.-M. Rhyne. Does the difference between information and scientific visualization really matter? *Computer Graphics and Applications, IEEE*, 23(3):6–8, 2003.
- [76] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1325–1332, 2008.
- [77] R. Roels and B. Signer. *MindXpres: An Extensible Content-driven Cross-Media Presentation Tool*. PhD thesis, Master’s thesis, Vrije Universiteit Brussel, 2012.
- [78] R. Roels and B. Signer. Mindxpres: An extensible content-driven cross-media presentation platform. In *Web Information Systems Engineering—WISE 2014*, pages 215–230. Springer, 2014.
- [79] H. Rosling. Ted talk: Hans rosling shows the best stats you’ve ever seen, 2006.

BIBLIOGRAPHY

- [80] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1139–1148, 2010.
- [81] B. Shneiderman. 1.1 direct manipulation: a step beyond programming languages. *Sparks of Innovation in Human-Computer Interaction*, 17, 1993.
- [82] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.
- [83] S. B. Shneiderman and C. Plaisant. Designing the user interface 4 th edition. ed: *Pearson Addison Wesley, USA*, 2005.
- [84] J. H. Siegel, E. J. Farrell, R. M. Goldwyn, and H. P. Friedman. The surgical implications of physiologic patterns in myocardial infarction shock. *Surgery*, 72(1):126–141, 1972.
- [85] B. Signer and M. C. Norrie. As we may link: a general metamodel for hypermedia systems. In *Conceptual Modeling-ER 2007*, pages 359–374. Springer, 2007.
- [86] H. Siirtola. *Interactive visualization of multidimensional data*. Tampereen yliopisto, 2007.
- [87] T. Simons. Does powerpoint make you stupid. *Presentations*, 18 (3), 24, 2004.
- [88] R. Spence. *Information visualization*. ACM, 2001.
- [89] R. Spence. *Information Visualization: Design for Interaction*. Prentice Hall, 2007.
- [90] R. Spence and M. Apperley. Data base navigation: an office environment for the professional. *Behaviour & Information Technology*, 1(1):43–54, 1982.
- [91] R. Spence and A. Press. *Information visualization*. 2000.
- [92] M. A. Stankiewicz. A dangerous business: Visual culture theory and education policy. *Arts Education Policy Review*, 105(6):5–14, 2004.
- [93] M. R. Stoner. Powerpoint in a new key. *Communication Education*, 56(3):354–381, 2007.
- [94] D. F. Swayne, D. T. Lang, A. Buja, and D. Cook. Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.

BIBLIOGRAPHY

- [95] M. Tanaka. C3.js| d3-based reusable chart library, 2014.
- [96] M. Theus. Interactive data visualization using mondrian. *Journal of Statistical Software*, 7(11):1–9, 2003.
- [97] M. T. Thielsch and I. Perabo. Use and evaluation of presentation software. *Technical Communication*, 59(2):112–123, 2012.
- [98] J. J. Thomas and K. A. Cook. A visual analytics agenda. *Computer Graphics and Applications, IEEE*, 26(1):10–13, 2006.
- [99] C. Tominski. Interaction for visualization. *Synthesis Lectures on Visualization*, 3(1):1–107, 2015.
- [100] E. Tufte. Powerpoint is evil. *Wired Magazine*, Septembre, 2003.
- [101] E. R. Tufte. Envisioning information. *Optometry & Vision Science*, 68(4):322–324, 1991.
- [102] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.
- [103] E. R. Tufte and E. Weise Moeller. *Visual explanations: images and quantities, evidence and narrative*, volume 107. Graphics Press Cheshire, CT, 1997.
- [104] J. W. Tukey. Exploratory data analysis. *Reading, Ma*, 231, 1977.
- [105] L. Tweedie. Characterizing interactive externalizations. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 375–382. ACM, 1997.
- [106] University of California, Berkeley, VisLab and Minnesota Public Radio. The minnesota employment explorer, July 2008.
- [107] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM, 2000.
- [108] M. O. Ward. Creating and manipulating n-dimensional brushes. In *Proceedings of Joint Statistical Meeting*, pages 6–14. Citeseer, 1997.
- [109] M. O. Ward and J. Yang. Interaction spaces in data and information visualization. In *VisSym*, pages 137–145, 2004.
- [110] C. Ware. *Information visualization*, volume 2. Morgan Kaufmann San Francisco, 2000.

BIBLIOGRAPHY

- [111] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.
- [112] B. Wilkins. *MELD: a pattern supported methodology for visualisation design*. PhD thesis, University of Birmingham, 2003.
- [113] L. Wilkinson. *The grammar of graphics*. Springer Science & Business Media, 2006.
- [114] W. Willett, J. Heer, J. Hellerstein, and M. Agrawala. Commentspace: structured support for collaborative visual analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3131–3140. ACM, 2011.
- [115] G. J. Wills. Selection: 524,288 ways to say “this is interesting”. In *Information Visualization’96, Proceedings IEEE Symposium on*, pages 54–60. IEEE, 1996.
- [116] W. Wojtkowski and W. G. Wojtkowski. Storytelling: its role in information visualization. Citeseer.
- [117] J. Yates and W. Orlikowski. The powerpoint presentation and its corollaries: how genres shape communicative action in organizations. *Communicative practices in workplaces and the professions: Cultural perspectives on the regulation of discourse and organizations*, pages 67–91, 2007.
- [118] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, 2007.
- [119] M. X. Zhou and S. K. Feiner. Visual task characterization for automated visual discourse synthesis. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 392–399. ACM Press/Addison-Wesley Publishing Co., 1998.
- [120] B. Zhu and H. Chen. Information visualization. *Annual review of information science and technology*, 39(1):139–177, 2005.
- [121] D. E. Zongker and D. H. Salesin. On creating animated presentations. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 298–308. Eurographics Association, 2003.