BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

ATTRIBUTE ANALYSIS OF DATA PRESENTATION USING INFORMATION DASHBOARDS

ANALÝZA VLASTNOSTÍ DAT PREZENTOVANÝCH V INFORMAČNÍCH DASHBOARDECH

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **Ing. NATALYA LOGINOVA**
AUTOR PRÁCE

SUPERVISOR — **prof. Ing. TOMÁŠ HRUŠKA, CSc.**
VEDOUCÍ PRÁCE

BRNO 2017

Brno University of Technology - Faculty of Information Technology

Department of Information Systems Academic year 2016/2017

Bachelor's Thesis Specification

For: **Loginova Natalya, Ing.**
Branch of study: Information Technology
Title: **Attribute Analysis of Data Presentation Using Information Dashboards**
Category: Information Systems

Instructions for project work:

1. Get acquainted with the *Information Dashboard* tool and the principles of its usage.
2. Analyze the most frequently used kinds of data which are visualized by information dashboards (e. g. comparison of values). Specify a set of possible graphical representations with their visual attributes.
3. Design a model describing selected kinds of data.
4. Implement a tool which will generate graphical representations of data specified by the designed model.
5. Generate sufficient amount of graphical representations (by changing its visual attributes) and perform a usability testing of generated graphical representations.
6. Evaluate the results of the user testing and propose set of design recommendations.

Basic references:

- Few, S.: Information Dashboard Design: The Effective Visual Communication of Data. Sebastopol [MA]: O'Reilly, 2006, ISBN 978-059-6100-162.
- Harris, R. L.: Information Graphics: A Comprehensive Illustrated Reference. Oxford: Oxford University Press, 2000. ISBN 978-0-1951-3532-9.
- Hynek J., Informační dashboardy. Skriptum pro předmět Pokročilé informační systémy, VUT Brno, 2014.

Detailed formal specifications can be found at http://www.fit.vutbr.cz/info/szz/

The Bachelor's Thesis must define its purpose, describe a current state of the art, introduce the theoretical and technical background relevant to the problems solved, and specify what parts have been used from earlier projects or have been taken over from other sources.

Each student will hand-in printed as well as electronic versions of the technical report, an electronic version of the complete program documentation, program source files, and a functional hardware prototype sample if desired. The information in electronic form will be stored on a standard non-rewritable medium (CD-R, DVD-R, etc.) in formats common at the FIT. In order to allow regular handling, the medium will be securely attached to the printed report.

Supervisor: **Hruška Tomáš, prof. Ing., CSc.**, DIFS FIT BUT
Consultant: Hynek Jiří, Ing., FIT VUT
Beginning of work: November 1, 2016
Date of delivery: May 17, 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

Dušan Kolář
Associate Professor and Head of Department

Abstract

This thesis deals with the study of data visualization usability aspects in information dashboards. The aim of this work is to design a model describing certain kinds of data most frequently used in dashboards, along with their possible graphical representations, and implement a tool for automatic generation of data visualizations based on the designed model. The tool will be used by researchers to examine various aspects of graphical data representations and to conduct user tests. A sample research of this kind has also been carried out under the scope of this project.

Abstrakt

Tato práce je věnována studiu některých aspektů použitelnosti vizualizace dat v informačních dashboardech. Cílem této práce je navrhnout model popisující vybrané druhy dat, které jsou nejčastěji používané v informačních dashboardech, a jejich možné grafické reprezentace, a poté implementovat nástroj na automatické generování vizualizací dat podle navrženého modelu. Nástroj mohou využít výzkumníci zabývající se zkoumáním různých aspektů grafické reprezentace dat k provedení uživatelských testů. Ukázkový průzkum uživatelů byl rovněž proveden v rámci tohoto projektu.

Keywords

information dashboards, data visualization, information graphics, user interface, usability, chart, data modeling, user testing

Klíčová slova

informační dashboardy, vizualizace dat, informační grafika, uživatelské rozhraní, použitelnost, graf, modelování dat, uživatelské testování

Reference

LOGINOVA, Natalya. *Attribute Analysis of Data Presentation Using Information Dashboards.* Brno, 2017. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor prof. Ing. Tomáš Hruška, CSc.

Attribute Analysis of Data Presentation Using Information Dashboards

Declaration

Hereby I declare that this bachelor's thesis was prepared as an original author's work under the supervision of prof. Ing. Tomáš Hruška, CSc. The supplementary information was provided by Ing. Jiří Hynek. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

. . . .
Natalya Loginova
May 15, 2017

Acknowledgements

I would like to thank my supervisor prof. Ing. Tomáš Hruška, CSc. and consultant Ing. Jiří Hynek for their time, support and valuable advice that helped me to write this thesis.

Contents

Chapter 1

Introduction

Present time is characterized by rapid growth in the field of information technologies and, as a consequence, in the amounts of data that current information systems have to deal with. Under such circumstances, effective representation of the extensive data becomes one of the crucial concerns. One of the tools that have been developed in the effort to discover a way to present large sets of generally diverse data efficiently, is *information dashboard*. This instrument aims at displaying only the most important information in a limited, typically small, amount of space to provide a quick overview at key metrics at a glance. Information dashboards can be a simple yet very powerful solution, helping to get the immediate notion of what actions have to be done to solve urgent problems rapidly and efficiently. However, in order to serve this purpose, they should be well thought-out and properly designed taking into account principles of effective visual communication, human perception of graphical information and other aspects that influence human interaction with digital interfaces.

Stephen Few states in his pioneer work on dashboards' visual design, entitled Information Dashboard Design [10]: „*To serve their purpose and fulfill their potential, dashboards must display a dense array of information in a small amount of space in a manner that communicates clearly and immediately. This requires design that taps into and leverages the power of visual perception to sense and process large chunks of information rapidly. This can be achieved only when the visual design of dashboards is central to the development process and is informed by a solid understanding of visual perception, what works, what doesn't, and why.*“

As Few further demonstrated, information dashboards designed without thorough consideration, fail to fulfill their large potential. Unfortunately, up to the present time graphical design of many dashboards used in business focuses rather on a fancy visual appearance than on usability. It also lacks understanding of how people communicate with graphical data.

This work investigates some typical attributes and properties of data, usually presented by means of information dashboards. My goal was to implement a tool for generating visual representations of dashboard data in order to be able to test different representations on real users and, as a result, propose a set of design recommendations for clearer and more understandable representation of dashboard information.

The rest of this chapter explains the overall concept of the work. The second chapter focuses on theoretical overview. Chapter three specifies the kinds of data and graphical representations to be further analysed in this work. The fourth chapter describes the data model and architecture of the generator. The fifth chapter explains the implementation process. The last chapters are dedicated to usability research and evaluation of results.

1.1 Concept of Work

In this bachelor thesis I applied the following stepwise approach.

1. Comprehend the principles of dashboards' usage and analyse the most frequently used kinds of data which are visualized by information dashboards.
2. Specify a set of possible graphical representations of these data with their visual attributes.
3. Design a universal model describing these most common types of data along with their graphical representations.
4. Design and implement a tool for generating graphical representations of data specified by the model from the previous step.
5. With the help of the implemented tool generate sufficient amount of different graphical representations by changing their visual attributes.
6. Perform usability testing of the generated graphical representations with real users.
7. Based on the evaluation of the results of the user testing, propose a set of recommendations for better information dashboards design.

Chapter 2

Information Dashboard

2.1 Definition and Main Characteristics

Information dashboard is a powerful medium for visualizing large volumes of manifold data for analytical, strategic, operational and other purposes. The concept of information dashboard originates in a real-life monitoring panel, which is usually a flat area where monitoring instruments are shown. These monitoring instruments typically represent a variety of display mechanisms, such as gauges and meters. A common example is, for instance, an automobile's dash, composed of fuel gauges, speedometers and other instruments. These panels allow monitoring and controlling the functioning of complex systems, such as manufacturing machines and production lines, vehicles, scientific or medical devices, computers and many others.

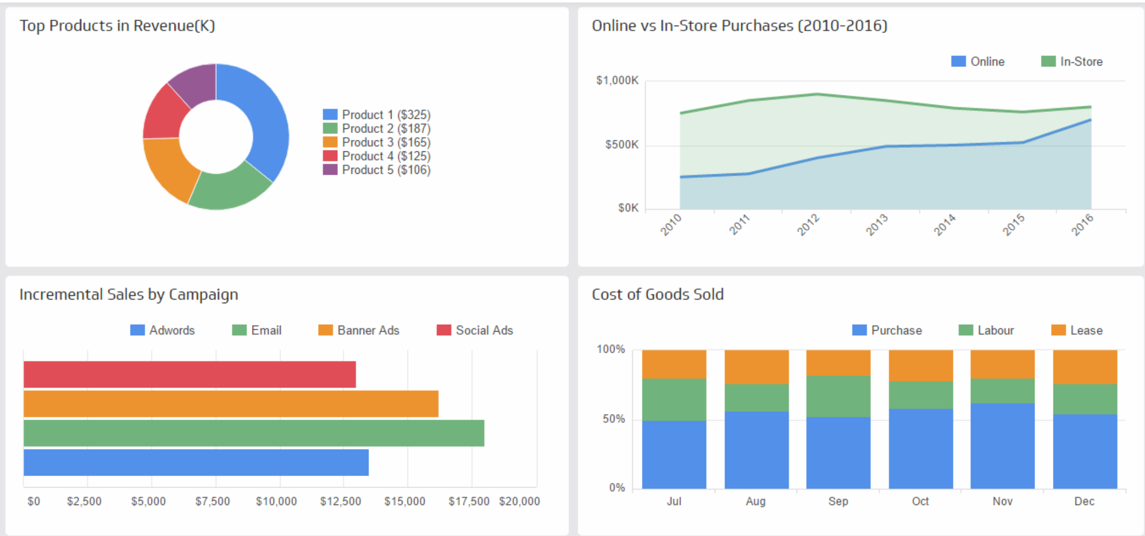

Figure 2.1: Example of information dashboard. [5]

Generally, modern information dashboards, in spite of great diversity in what they look like and what they are used for, serve the same purpose: to provide a medium for quick and hassle-free overview of the system functioning. To accomplish this requirement, information dashboards must be clear and transparent, as well as occupy quite a modest amount of space, typically no more than the size of a single screen.

Few introduces the following definition of information dashboard in [10]: „*A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.*“

Thus, if we want to give the most general definition specifying only the very essential characteristics, information dashboard at its core is a single-screen display medium that allows efficient monitoring of the information needed to achieve one's objectives. Information dashboard is not a particular computer technology or specific kind of information, it's a form of representing the information, independently from its source and meaning. An example of information dashboard (specifically, Sales dashboard) can be found at Figure 2.1.

According to Few, there are several most characteristic properties of information dashboards, which we will further examine.

- **Dashboards are visual displays.** Typically the information on dashboards is represented as a composition of both text and graphics, but with the predominance of visual means. The emphasis on graphics guarantees the quickness and effectiveness of the dashboard data perception. As Few notices: „*Dashboards are highly graphical, not because it is cute, but because graphical presentation, handled expertly, can often communicate with greater efficiency and richer meaning than text alone.*“ [10]

- **Dashboards display the information needed to achieve specific objectives.** Achieving a goal can often require to familiarize oneself with different kinds of data from various sources, but aggregated and organized in one place. Information dashboards are often associated with business performance, but this is only one typical application of information dashboards: in general, they can be used by anyone who has to achieve some objectives requiring access to information; and the kinds of data that can be displayed on a dashboard are not limited to only business performance metrics.

- **A dashboard fits on a single computer screen.** This is the very essential characteristic of information dashboards: availability of all information within the viewer's eye span helps to comprehend it immediately and without unnecessary efforts.

- **The most typical medium for displaying information dashboards is a web browser.** However, there also are other ways to present them (for example, in desktop applications).

- **Dashboards are used to monitor information at a glance.** Few formulates the one main similarity of all the diverse information that can be displayed on a dashboard: it is abbreviated in the form of summaries or exceptions [10]. Indeed, dashboards tend to emphasize only the most important data that require user's attention and are necessary to take further actions.

- **Dashboards have small, concise, clear, and intuitive display mechanisms.** Taking into account that space of the information dashboard screen is truly a precious commodity due to the one-screen limitation, the applied display mechanisms must communicate their message as clearly and intuitively as possible.

- **Dashboards are customizable.** To be able to entirely fulfil the requirements of their particular users, information dashboards must accommodate themselves to their needs, otherwise, they won't serve their purpose.

Since the beginning of 21st century, information dashboards have been provoking growing interest in the business sphere. Eckerson refers to a study conducted by The Data Warehousing Institute (TDWI) that showed that in 2004 most of organizations had already used information dashboards or had currently been implementing one [9]. The reason why dashboards have become so widespread could be easily explained: they offer numerous practical benefits to everyone in an organization, starting from senior management to ordinary staff.

Eckerson lists the following main benefits of performance dashboards [9].

- **Dashboards communicate strategy.** Thanks to information dashboards, employees with different responsibilities can get a clear notion of the state of affairs in the organization and what actions they need to undertake in their areas to achieve organization's strategic objectives.
- **Dashboards refine strategy.** With the help of performance information dashboards executive managers can adjust the corporate strategy as the situation evolves.
- **Increased visibility.** Information dashboards enable higher visibility into the everyday operations by gathering relevant data in one place, and at the same time they offer the possibility to make predictions and forecast trends based on the information about past activity.
- **Increased coordination.** By presenting performance data clearly and evidently, performance information dashboards allow employees of different departments and hierarchy levels to work more closely and productively together. They also permit management to carry out more constructive and informative performance reviews on staff's productivity.
- **Increased motivation.** By getting business activities measured and providing comparisons, dashboards motivate employees to work proactively and more efficiently.
- **Consistent view of the business.** Performance dashboards consolidate the information in one place, giving all the members in an organization consistent view on the data, providing the unified set of performance metrics across the whole organization and helping to avoid misunderstandings.
- **Reduced costs and redundancy.** A single dashboard can potentially substitute multiple independent reporting systems, documentation files and data warehouses.
- **Dashboards empower users.** Dashboards make information easily accessible for average employees, allowing them to analyse data and undertake required actions without unnecessary delay.
- **Dashboards deliver actionable information.** Performance dashboards provide users with timely data, so information obtained from a dashboard, allows users to take the right actions at the right time to achieve their objectives.

2.2 Classification

Information dashboards can be categorized in several ways based on different taxonomic criteria. We will further examine some of the most common classifications.

- **By type of data**

  As a first approximation, by the type of presented data, dashboards can be divided into quantitative and non-quantitative. Quantitative data can be expressed numerically and non-quantitative data require text representation (for example, simple lists, tasks, due dates, order details, the people responsible) or representation using schemes (for example, entity relationship diagrams). It's worth mentioning that overwhelming majority of information dashboards primarily present quantitative or mostly quantitative data [10].

  The further categorization of data types typically displayed in information dashboards is particularly important for the purposes of this work. It will be examined in detail in the third chapter.

- **By data domain**

  Information dashboards can be applied in various fields of human activity, such as sales, finance, marketing, manufacturing, human resources, customer support, project management, to name just a few most common domains where dashboards data can source from.

- **By interactivity**

  In most cases, information dashboards provide static display of data, but some of them also offer interactive display mechanisms, such as drill-down, filters, providing more detailed information by hovering over some piece of dashboard data.

- **By display mechanisms**

  Information dashboards can use primarily graphical or primarily text display mechanisms, as well as some combination of graphics and text. At that, most of the dashboards rely rather on graphical means (or combination of graphics and text with the emphasis on the graphics), since this way of presenting information is in most cases more intuitive, explicit and efficient.

- **By purpose**

  Based on the type of supported business activity information dashboards can perform 3 roles: strategic, analytical (Eckerson in [9] refers to this role as *tactical*), and operational. According to Few [10], this is one of the most useful ways to classify information dashboards, therefore this characteristic will be examined more particularly.

  1. **Strategic dashboards.** Dashboards of this type, as follows from its name, serve to formulate strategic objectives and monitor their execution. Few mentions that this type of dashboards is the most frequently used of all three [10]. Most information dashboards used by executives and managers in an organization are, in fact, strategic. These dashboards provide a high-level view on the key performance metrics, giving decision-makers the possibility to focus on long-term strategic objectives, rather than immediately react on short-term changes.

Hence, strategic dashboards do not require real-time data. Strategic dashboards typically benefit from the simplest display mechanisms and do not need to support further interactions with data (such as drill down). An example of strategic dashboard can be found at Figure 2.2.

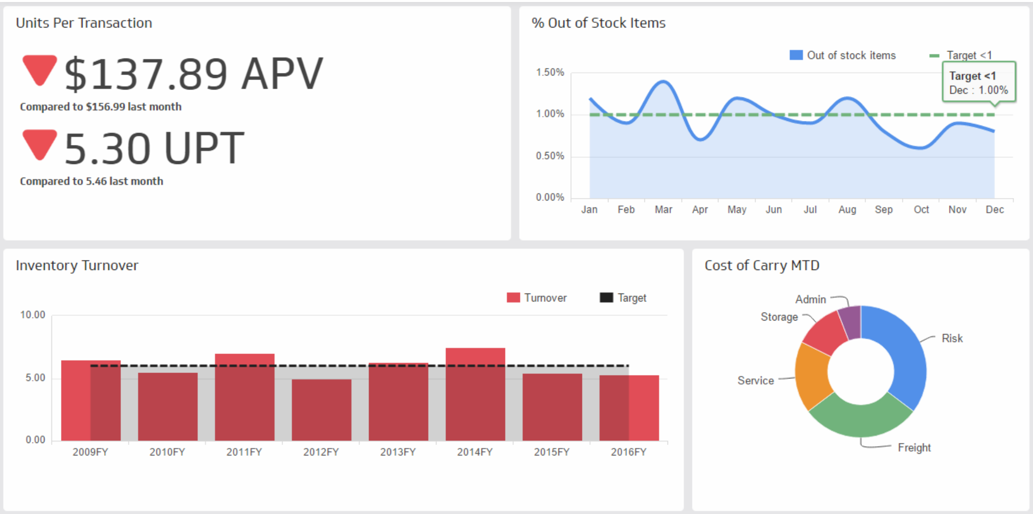

Figure 2.2: Example of strategic dashboard. [2]

2. **Analytical dashboards.** Analytical dashboards are used to perform deeper analysis of information, examine complex data and relationships between them, make comparisons, plans and forecasts. Dashboards designed for these purposes, as well as strategic dashboards, usually reflect static snapshots of data and do not need real-time updates, but unlike them, they typically require more sophisticated display mechanisms supporting interactions with the data, to provide more profound details, that enable the user to make thorough research. A sample analytical dashboard is presented at Figure 2.3.

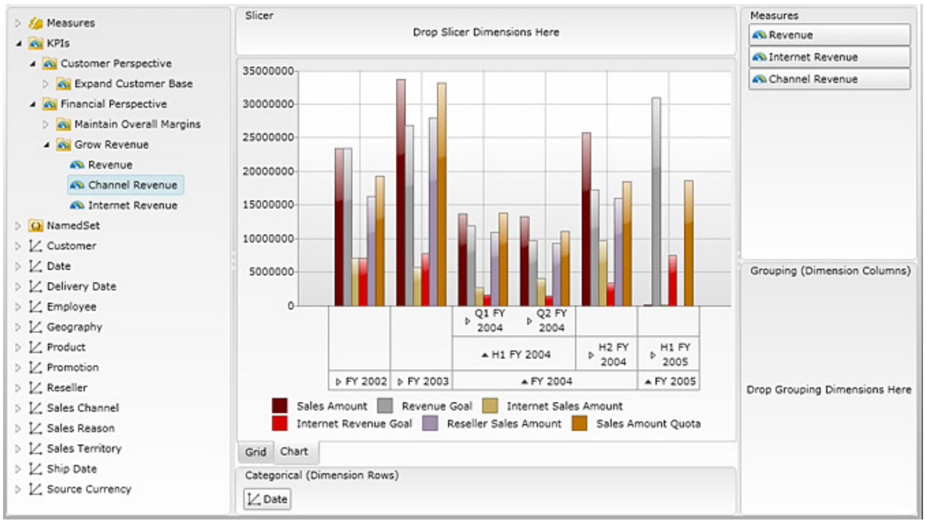

Figure 2.3: Example of analytical dashboard. [6]

3. **Operation dashboards.** According to Eckerson, this type of dashboards is used primarily by front-line workers and their supervisors to monitor operations. Operation dashboards usually provide detailed information which is only slightly summarized [9]. These dashboards require different design approach than those that serve strategic or analytical purposes. As Few states: „*When you monitor operations, you must maintain awareness of activities and events that are constantly changing and might require attention and response at a moment's notice.*“ [10] For the reasons set forth above, operation dashboards benefit from very simple and clear display mechanisms, require constant updates of information and must have the means to immediately draw the attention of the user if the value of some metric falls above or beneath the acceptable limits. Dashboards of this type might also need to support interactions with the data – in order to provide more detailed information required to react to the events properly. An example of operation dashboard is presented at Figure 2.4.

Figure 2.4: Example of operational dashboard. [4]

This research pays special attention to the most popular type of information dashboards – the strategic one. Dashboards of this type are especially suitable for the purpose of examining dashboard data, as they use quite simple display mechanisms, do not require constant data updates and are not interactive.

2.3 Nature of Typically Presented Data

Information dashboards find application in a wide range of fields. Several examples are given below.

- **Sales.** Dashboards in this field typically display number and amount of orders, revenue growth, sales per different periods of time, sales per region or per representative, number of leads, lists of top selling products, selling prices and billings.
- **Marketing.** This type of dashboards most often includes such data as market shares, various online and SEO metrics (number of visitors and of page views, average time spent on a page, visit durations), ROI (Return On Investment), customer demographics, sources of leads traffic, comparisons with analogous metrics of competitors.

- **Finance.** Financial dashboards usually display such economical parameters as revenues, expenses, profits, assets and other financial performance metrics.
- **Customer support.** To monitor the activity of a technical support call center the following metrics can be useful: number and duration of incoming calls, number of resolved cases and of cases that are still in progress, customer satisfaction parameters, average time to resolve a case.
- **Manufacturing.** Dashboards applied in the manufacturing industry typically display production characteristics, such as number of fabricated units per different time slots, average time to produce one unit, statistics of defects.
- **Project Management.** This type of dashboards provides an overview of project budgets, number and status of tasks, project schedule and progress.
- **Human Resources.** Dashboards used in the field of HR management are usually aimed to display such metrics as number of employees by position or division, number of open positions by division, employee satisfaction and turnover.

These examples clearly demonstrate that information dashboards can be used to present numerous kinds of data that can be found in various fields of human activity. In spite of this fact, if we put aside the meaning of data in the real world and focus on their core structure and nature, we will find out that in most cases dashboards display several most typical kinds of data that will be examined in the next chapter.

Chapter 3

Data and Their Graphical Representations

As mentioned above, in the first place all kinds of data presented on a dashboard can be divided into quantitative and non-quantitative types. It is worth mentioning, that overwhelming majority of dashboard data is quantitative. As we remember, the basic aim of information dashboard is to provide a display medium that allows efficient monitoring of the information needed to achieve one's objectives. And most of the information that is significant enough to provide at a glance comprehension of the current state of affairs is quantitative.

As Few notices in [10]: „*Despite these diverse applications, in almost all cases dashboards primarily display quantitative measures of what's currently going on. This type of data is common across almost all dashboards because they are used to monitor the critical information needed to do a job or meet one or more particular objectives, and most ...*[1] *of the information that does this best is quantitative.*"

However, Few also mentions, that non-quantitative data can be also presented in information dashboards and sometimes play an important role in them. Some characteristic examples of non-quantitative dashboard data provided by Few include simple lists (such as „Most recent tasks", „Top 10 sellers", „Most popular website pages"), schedules (plans, events, due dates) and entity relationship diagrams.

Taking into consideration that quantitative data prevail in information dashboards, I decided to focus on this type of data only in this work. Analysing the essence of the most common data that can be found on individual graphical units of a dashboard (which are called *widgets*), I marked out the following several most frequently used categories of quantitative data.

1. Single numeric value.
2. Development of one numeric measure in time.
3. Development of several numeric measures in time.
4. Composition of values (part-to-whole).
5. Multiple instances of a single measure.
6. Multiple instances of multiple measures.

[1] Omitted text: "*(but not all, as we'll see later)*„

These kinds of data will be explained in details below, along with the typical media that can be used to visualize them.

1. **Single numeric value.** This is a quantitative metric that possesses a value, often within some boundaries known in advance. An example of this type of data can be for instance some percentage (customer satisfaction, profit margin, past due invoices as a percentage of total invoices, any percentage of target for arbitrary metric), but it can also be any numeric value in general (number of open customer support tickets, revenue this month, amount of orders). In many cases it will be enough just to display this value on a dashboard as a text, but most often some context for this value has to be provided: users need to know if presented value is satisfactory or not to decide if they have to undertake any actions to improve it, and what actions they should be. A radial gauge chart indicating several states of a metric with different colors can be more suitable for this purpose. Another option for visualization of this kind of data is a bullet graph, which represents a horizontal or vertical scale where several qualitative states are marked using colors or notches.

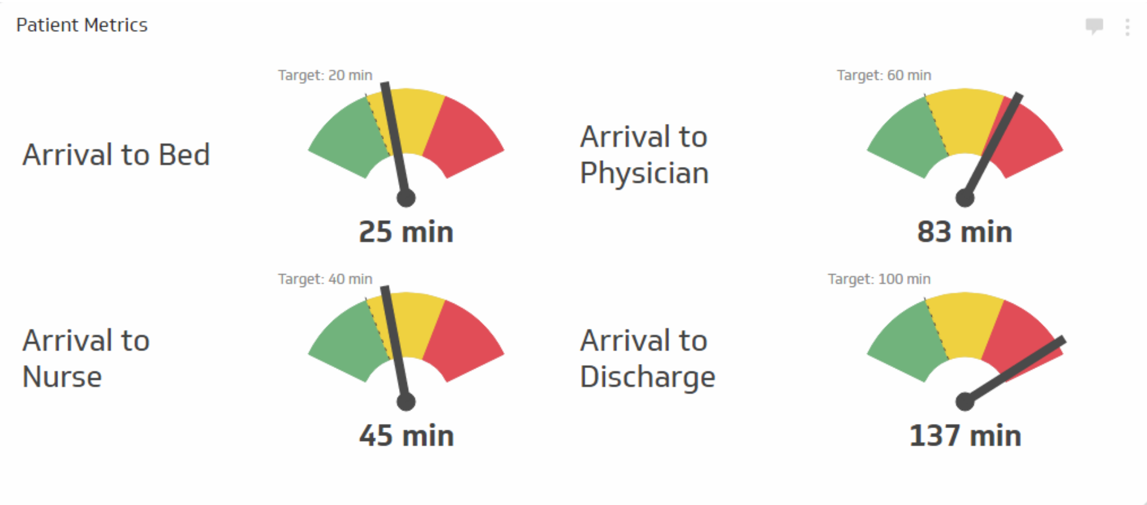

Figure 3.1: Example of several single value metrics. [7]

Figure 3.1 displays a set of 4 single numeric values visualized with radial gauge charts.

2. **Development of one numeric measure in time.** This metric represents a parameter that changes over some period of time. Typical examples can be sales last year, daily website visits or number of manufactured units per month. There are two most common ways to visualize such kind of data: first is a line chart, which presents data in continuous manner; second is a bar chart, which presents a metric as a set of discrete values.

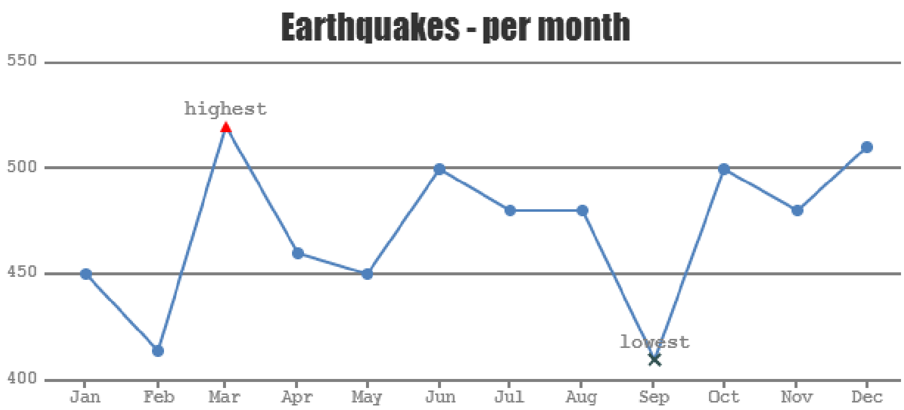

Figure 3.2: Example of development of one numeric measure in time. [1]

Figure 3.2 displays development of one numeric measure in time visualized with line chart.

3. **Development of several numeric measures in time.** Sometimes it can be useful to display the progress of several measures in one widget at once, for example to compare sales this year to the last year's, or to display the dynamics of the revenue from different sources.This type of data, as well as the previous one, is typically visualized with line or bar charts.

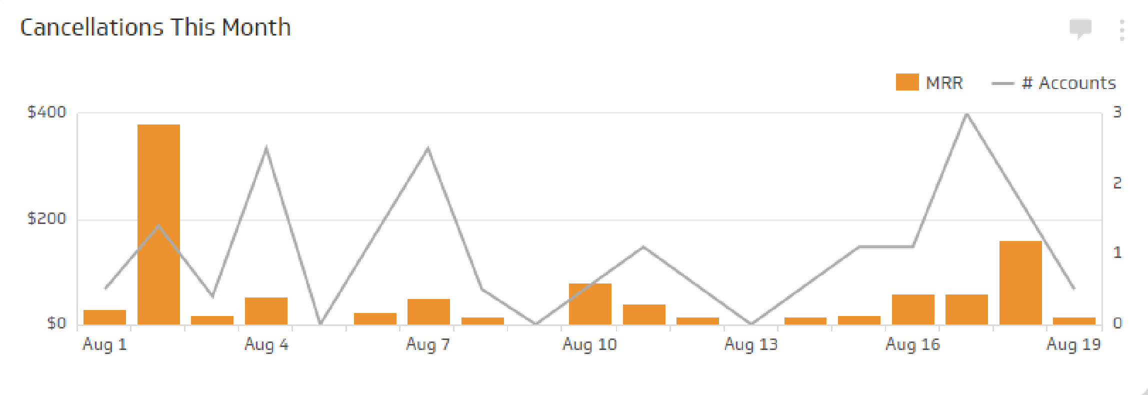

Figure 3.3: Example of development of several numeric measures in time. [7]

Figure 3.3 displays development of two numeric measures in time visualized with line and bar charts simultaneously in one widget.

4. **Composition of values (part-to-whole).** This kind of data represents any set of values, which, being summed up, constitute the entire 100% of some measure. An example of this metric can be distribution of tasks by status, or, for instance, sales per different regions. The most common medium to present part-to-whole data is a pie chart – a circular graph divided into slices to illustrate numerical proportion between the components it consists of. Besides a pie chart, this kind of data can be visualized as a bar chart as well.

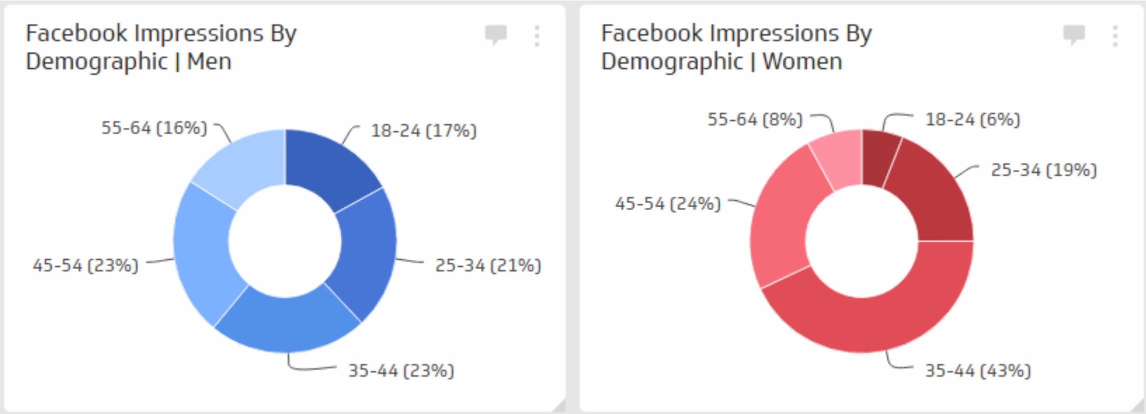

Figure 3.4: Example of composition of values. [3]

Figure 3.4 displays two part-to-whole measures visualized with pie charts („donut“ modification).

5. **Multiple instances of a single measure.** This kind of data represents a measure decomposed into several parameters, which, being summed up, do not form a whole. An example of this kind of data is average time spent on a page for different pages of a website. Another good case is customer satisfaction analyzed by several criteria (such as Product quality, Price, Delivery, Quality of service). The most usual way to present such data is using bar chart. Another option is to present this kind of information as a set of radial gauges or bullet graphs.

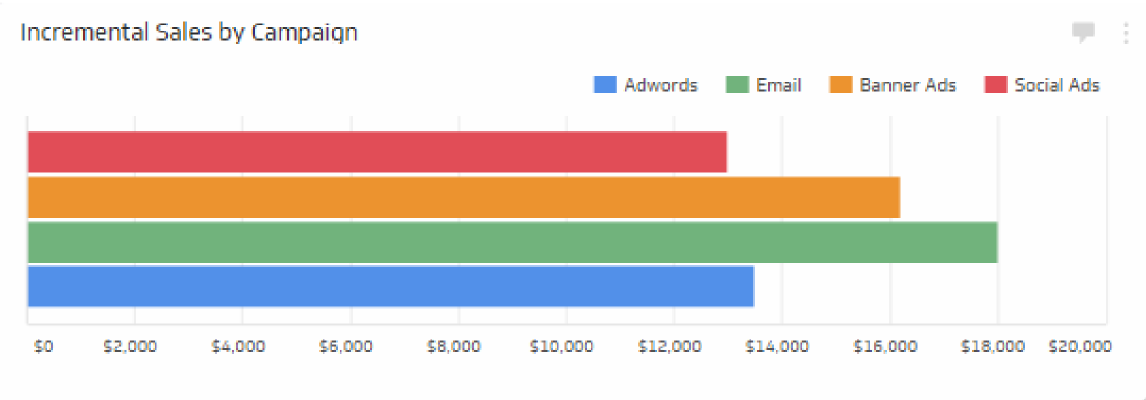

Figure 3.5: Example of multiple instances of a single measure. [3]

Figure 3.5 displays four instances of one metric („Incremental sales by campaign“) visualized with horizontal bar chart.

6. **Multiple instances of multiple measures.** In many cases, it is good to have a view on multiple dimensions of multiple measures on a single widget. An example is a graph of actual and expected sales by region or by department. Another example is a graph of customer satisfaction this month compared to last month by several factors. This type of data is most commonly presented as a bar chart.

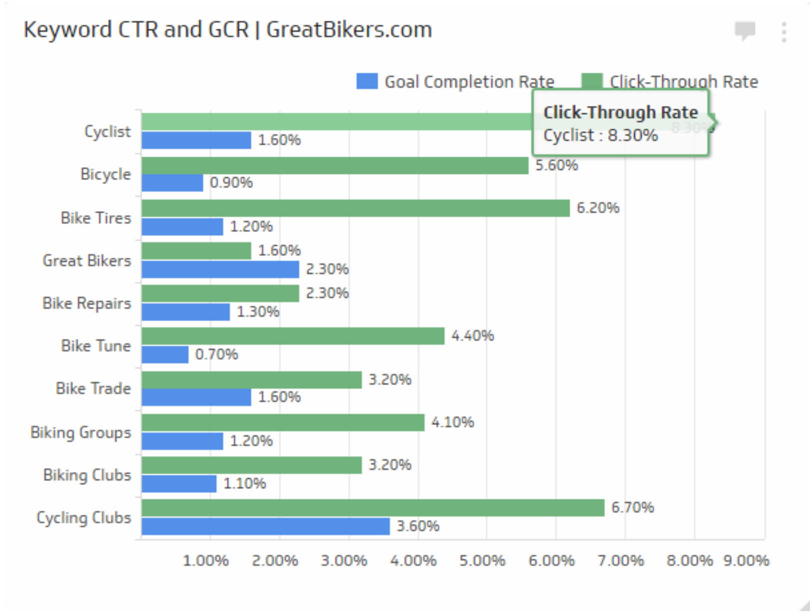

Figure 3.6: Example of multiple instances of multiple measures. [3]

Figure 3.6 displays several instances of two metrics („Keyword Goal Completion Rate" and „Keyword Click-Through Rate") visualized with horizontal bar chart.

I thereby marked out the following several visual media, typically used to present most common kinds of dashboard data.

1. Radial gauge chart.
2. Bullet graph.
3. Line chart.
4. Bar chart.
5. Pie chart.

Undoubtedly, there exist a lot more visual means capable of representing the above-listed kinds of data (e.g. see [11], where Harris lists plenty of graphic media). But for the purpose of this work, only the ones that are most broadened and familiar to a wide range of users, were selected.

To analyse the properties of these media and test their variations on real users, I singled out some visual attributes of each chart type for the subsequent research. These attributes for each medium are described below.

1. **Radial gauge chart.**

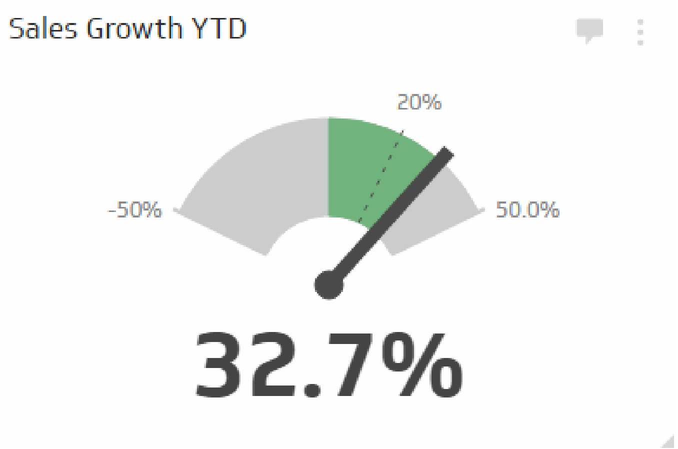

Figure 3.7: Example of radial gauge chart. [7]

An example of radial gauge chart can be found in Figure 3.7, it usually has one or several qualitative states represented by different colors. For this type of chart the following attributes can be examined:

- number of qualitative states;
- color palette (for example, colors of different hues or different tints of the same hue can be used);
- target value (it can be specified in the chart using different ways, but may be not specified at all).

2. **Bullet graph.**

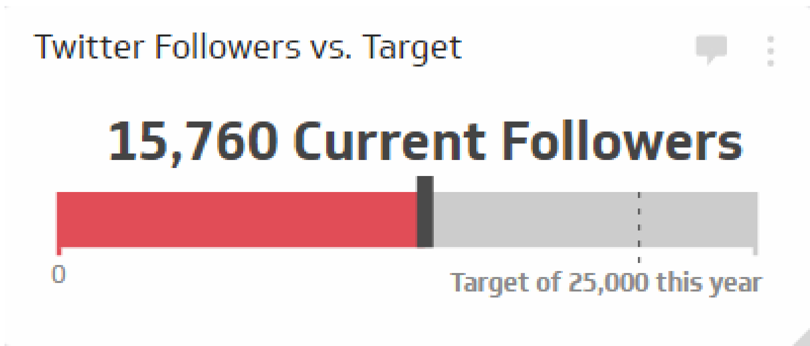

Figure 3.8: Example of bullet graph. [7]

Bullet graph (represented in Figure 3.8) less often than radial gauge displays several qualitative states. Due to its more compact size, it is often used to visualize several single-value metrics on the same dashboard. Bullet graph has basically the same list of attributes as a radial gauge, but, in addition, it can be vertically or horizontally oriented.

3. **Line chart.**

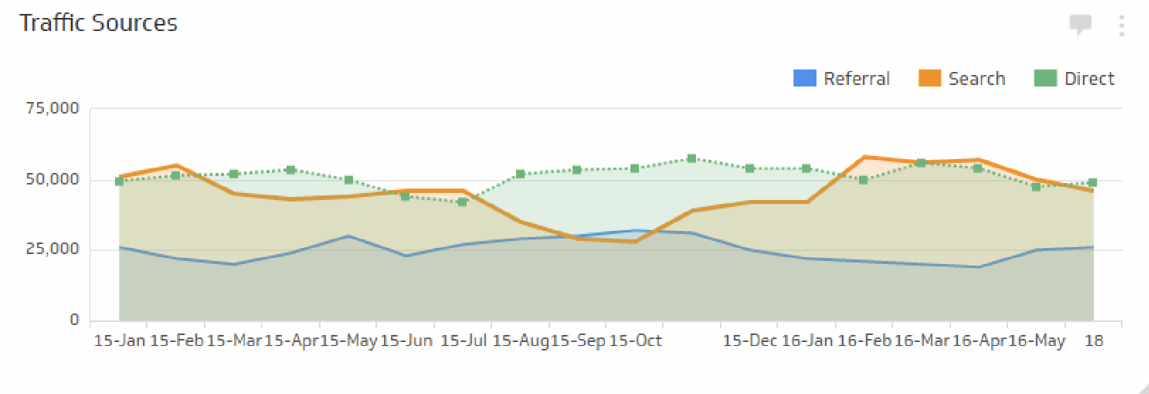

Figure 3.9: Example of line chart. [7]

Line chart (example for three different metrics can be seen in Figure 3.9) is one of the most widely used chart types. It's usually the first choice when visualizing time dependencies. Some of the possible attributes to examine for the line chart are:

- number of lines;
- color palette;
- type of lines (solid, dotted, with or without marks);
- area chart (area under the line can be filled with color or not).

4. **Bar chart.**

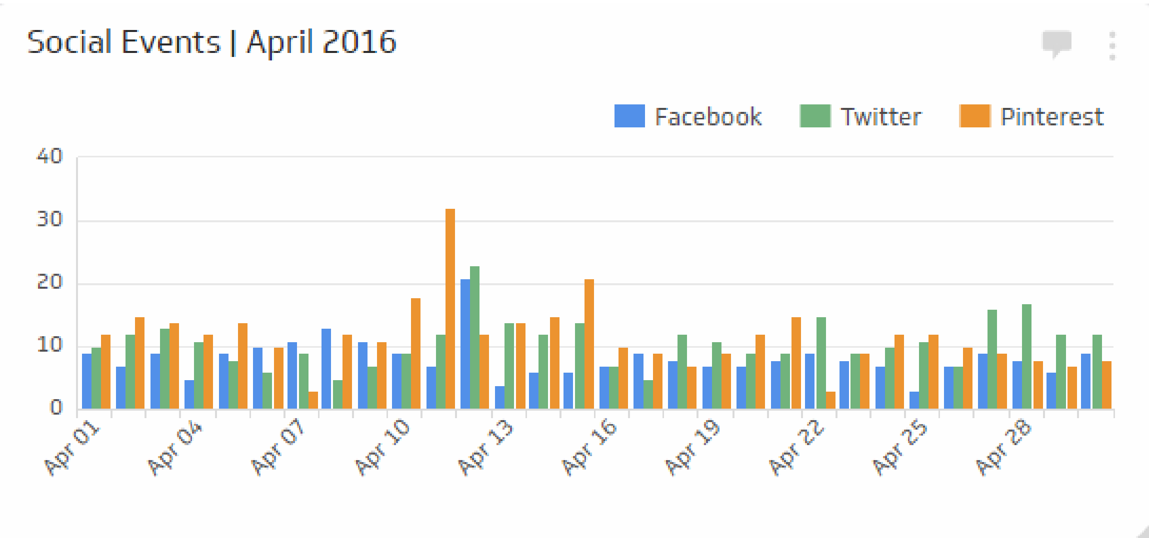

Figure 3.10: Example of bar chart. [3]

Example of multi bar chart (for three different metrics) is presented in Figure 3.10. Some of the attributes of this chart type include:

- number and size of values;

- color palette;
- orientation of chart (vertical or horizontal).

5. **Pie chart.**

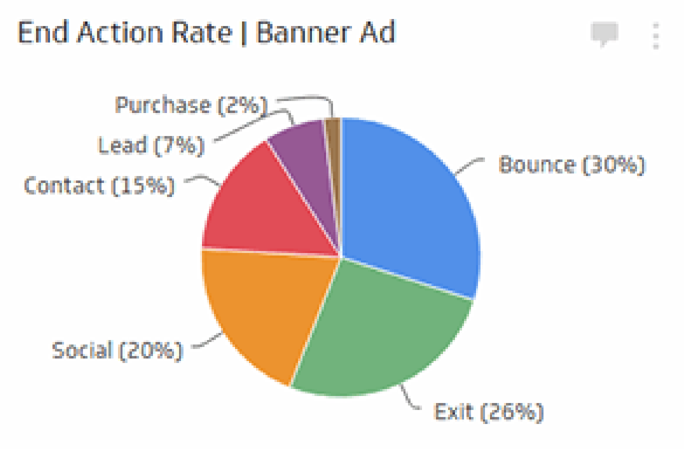

Figure 3.11: Example of pie chart. [7]

Example of pie chart (which is, along with line and bar charts, one of the most widespread in information dashboards) can be found in Figure 3.11. This type of chart is characterized, among others, with the following attributes:

- number and size of values;
- color palette;
- „donut“ modification (donut chart is basically a pie chart with cutout center part).

Chapter 4

Design

4.1 Tool Architecture

The tool for generating graphical data representations, which I worked on, is supposed to be used mainly for different kind of researches on real users. These user researches would aim to investigate various aspects of usability, intuitiveness and comprehensibility of visual data representations typically used in information dashboards. It was also quite advisable that the tool is compatible with the other instruments used in the Department of Information Systems of Faculty of Information Technology to study the topic of information dashboards usability. Taking this into consideration, the main requirements for the tool architecture were the following.

- The presentation of data in the tool must be separated from the presentation of style attributes of graphical elements.
- The tool must support multidimensional data.
- The generation of data inputs for the further generation of graphical representations should be as easy and automatized as possible.
- The tool should be easily extendible (by adding new chart types or new style attributes to already supported charts).
- Data model should support XML syntax.

The first three points directly influence the tool architecture.

The first point, separation of data from graphical style, allows to easily modify either some style attributes for the same data, or, vice versa, data for the same visual representation. This required the generator to have two separate inputs: one for data and one for specifying visual appearance of the widget being generated.

The second requirement, support of multidimensional data, shall be explained in more detail. Generally speaking, data used in modern information systems usually have unlimited number of parameters – we can think of these parameters as different dimensions of data. On the contrary, static visual display mechanisms used in strategic information dashboards are typically able to present only limited amount of data dimensions (one in case of gauge or bullet chart; two for pie chart or simple line and bar chart; three for multi-line or multi-bar chart).

These circumstances will be more clear if illustrated with an example. Let us assume we have a set of data describing population of different Czech cities divided by two population types: man and woman, and also by year when the number of citizens was measured. These four-dimensional data can be illustrated with Table 4.1.

City	**Type**	**Year**	**Population (K)**
Brno	Man	2014	181
Brno	Woman	2014	183
Prague	Man	2014	548
Prague	Woman	2014	592
Ostrava	Man	2014	141
Ostrava	Woman	2014	149
Brno	Man	2015	185
Brno	Woman	2015	183
Prague	Man	2015	571
Prague	Woman	2015	608
Ostrava	Man	2015	142
Ostrava	Woman	2015	146
Brno	Man	2016	189
Brno	Woman	2016	188
Prague	Man	2015	592
Prague	Woman	2016	610
Ostrava	Man	2016	141
Ostrava	Woman	2016	145

Table 4.1: Example of table describing population of Czech Republic cities.

Ideally, the generator should be able to visualize such data, even though the supported display media can handle only smaller number of dimensions. To realize this requirement we need an intermediate layer between data and graphical representation – a third generator input that would specify which data dimensions should be displayed in the widget and how they should be displayed. This intermediate layer can be considered as mapping that maps individual data dimensions on the axes of the chart and also specifies if the data should be aggregated (summed up to reduce the number of dimensions) or not.

For example, we may want to display a bar chart showing the number of citizens in each city grouped by year. In this case, the values for men and women for each city and year should be aggregated. The resulting graph is presented in Figure 4.1.

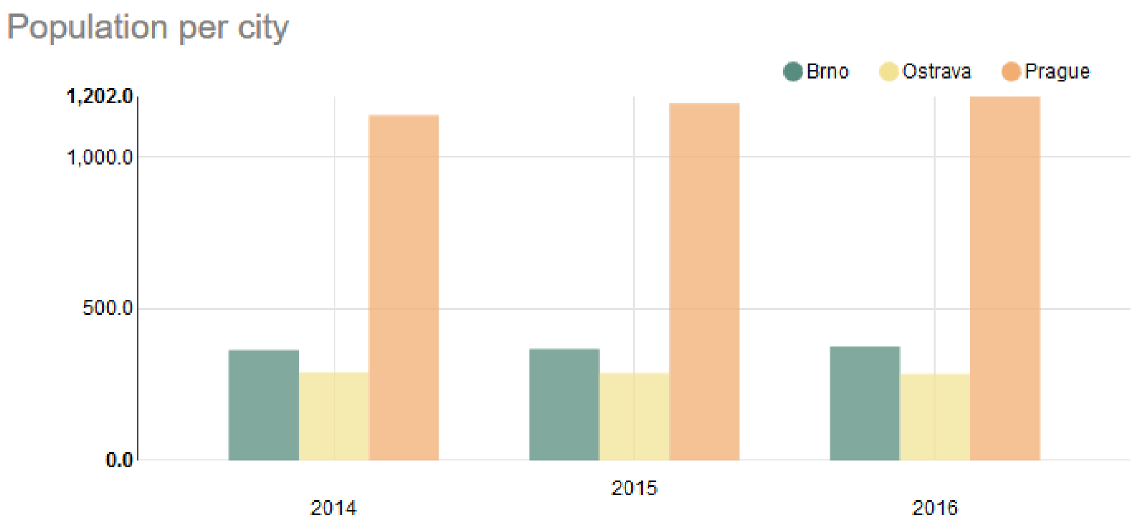

Figure 4.1: Bar chart representing population per city.

Another example of different chart obtained from the same set of data, pie chart showing the average structure of population in all three cities, can be seen in Figure 4.2 (both graphs were generated using the tool implemented under the scope of this work). In this case, the number of data dimensions was reduced to 2 by summarizing the population of all cities for each citizen type.

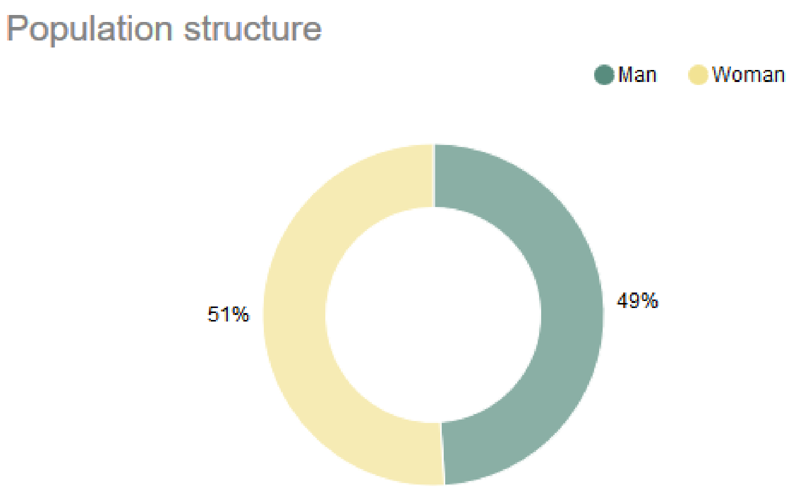

Figure 4.2: Pie chart representing the structure of population.

So, it might be said that mapping input actually specifies the meaning and context of visual representation to be generated.

Finally, the third point that influenced the tool architecture is the need of simple way to create sample test data to make the work with the generator more convenient. That requires to add a data generator, that can create sets of data (based on the parameters specified by user) to be further passed to the data input of the tool.

Scheme of the final architecture of the tool, taking into consideration the requirements mentioned above, is presented in Figure 4.3.

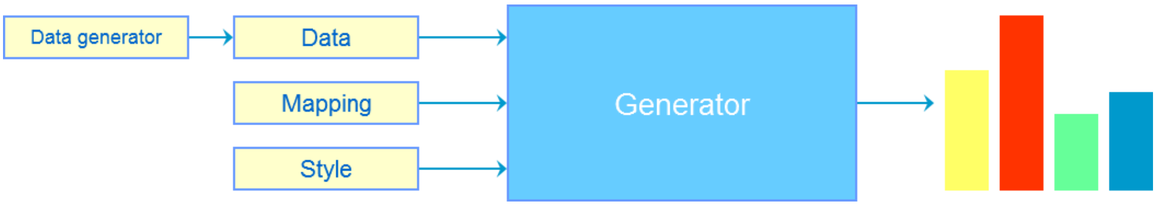

Figure 4.3: Tool architecture.

Data generator creates the Data input, which, along with Mapping and Style inputs, is passed to the Generator. The output of the generator is graphical representation of data provided as the input.

4.2 Data Model

For the syntax of data model used in the tool, several options were considered, namely XML, YAML and JSON languages. Both YAML and JSON have clear and compact human-readable syntax. Moreover, JSON is naturally compatible with JavaScript, which was chosen as the main implementation language. However, after comparing the above-mentioned languages and processing technologies to deal with them, I finally opted for XML. The main reason was that this data format is conventional for the research group of Department of Information Systems dealing with the study of information dashboards. Furthermore, XML has language-defined document schema descriptors, e.g. XML Schema Definition (XSD), allowing to formally describe the elements of XML model.

Complete XSD model of data used in the tool can be found in Appendix A. Here I will provide some particular XML examples of the tool input.

For example, the data presented in Table 4.1 are rewritten to XML in the following manner.

```
<dataset>
  <values>
    <value>
      <city>Brno</city>
      <year>2014</year>
      <population>181</population>
      <type>Man</type>
    </value>
    <value>
      <city>Brno</city>
      <year>2014</year>
      <population>183</population>
      <type>Woman</type>
    </value>
    ...
  </values>
</dataset>
```

The root element <dataset> contains nested element <values>, which, in turn, contains an array of one or more <value> elements. The <value> element can include unlimited number of arbitrary nested elements (which represent data dimensions), and one of them must necessarily be of integer type.

This model is universal for all types of data described in Chapter 3. For example, for a single-value metric, the <values> element will contain only one <value> element consisting of a single numeric nested element.

Now the graphical element in Figure 4.1 used to visualize the data from the example above, has the following XML presentation.

```
<graphicalElement>
  <type>bar</type>
  <width>700</width>
  <height>300</height>
  <style>
    <theme>colored</theme>
    <orientation>vertical</orientation>
  </style>
</graphicalElement>
```

The root element <graphicalElement> contains four obligatory nested elements.

1. <type>, which sets the chart type. This element can have one of five values (by the list of supported chart types): bar, line, pie, gauge or bullet.

2. <width>, which sets the width of generate graphical element in pixels.

3. <height>, which sets the width of generate graphical element in pixels.

4. <style> – an array of style attributes, which is specific for each chart type. However, all 5 chart types include one common style attribute <theme>, which defines the color palette of the chart and can take one of two values: mono (for grey scale palette) or <colored> (for colored palette). The latter was successfully tested against three major types of color blindness.

Other <style> attributes specific for individual chart types are the following.
For bar chart:

- <orientation> – specifies graph orientation and can take one of two values: vertical or horizontal;

- <dateFormat> – optional attribute to specify date format for x-axis, in case the dates are passed as Unix timestamps. The attribute supports five values (by the list of supported date formats): date, time, weekday, month or year.

For line chart:

- <marks> – attribute of boolean type indicating whether tick marks will be added to a graph or not;

- <dateFormat> – has the same syntax and semantic as the one in bar chart;

- <isArea> – boolean attribute indicating whether the areas between x-axis and lines will be filled with color;

- `<fromZero>` – boolean attribute, which, when set on `true`, forces the y-axis scale to start from 0. Otherwise, it starts from the smallest value in the dataset.

For pie chart:

- `<donut>` – boolean modifier indicating whether pie chart will be displayed as donut chart.

For bullet and gauge charts no additional style attributes are supported.

Finally, the mapping input used to display the graphical element in Figure 4.1 has the following XML presentation.

```
<mapping>
  <map>
    <title>Population per city</title>
    <label>city</label>
    <value>population</value>
    <group>year</group>
    <aggregates>false</aggregates>
  </map>
</mapping>
```

The root element `<mapping>` contains one or more nested `<map>` elements (an individual graph will be generated for each map). `<map>` element is an array of the following elements:

- `<title>` – string parameter specifying the graph title;
- `<label>` – indicates the data dimension for labels;
- `<value>` – indicates the data dimension for values;
- `<group>` – for bar and line chart indicates the data dimension for x-axis;
- `<aggregates>` – boolean parameter, which indicates for bar and line chart, whether aggregation to two dimensions should be performed. In any case, aggregation to the maximal number of dimensions for a given chart type (which is 3 for bar and line charts) will be performed implicitly.

In addition, `<map>` element for one-dimensional charts (such as gauge and bullet) contains nested `<ranges>` element, which is an array of one or more `<range>` elements, that take integer values. These elements set the number and extent of qualitative states marked in the graph.

An example of `<ranges>` element indicating 3 equal qualitative states, is presented below.

```
<ranges>
  <range>100</range>
  <range>200</range>
  <range>300</range>
</ranges>
```

Chapter 5

Implementation

5.1 Chosen Technologies

After the careful analysis of several possible technologies to use for the implementation of the dashboard generator, I decided to settle on the ones listed below.

- JavaScript – main implementation language).
- ReactJS library[1] – for building GUI.
- D3.js library[2] – for visualizing charts.

As regards to the implementation language, JavaScript was the best choice for building a purely client-side web application supported by all modern web browsers. JavaScript applications can run directly in web browser, which makes them easily accessible for potential users.

ReactJS library was selected for building the GUI due to the fact that this is a component-based library, which allows to create efficient and easily extendible interactive user interfaces.

D3.js library was used as the core technology for transforming input data into charts. This is a powerful JavaScript library for visualizing data in web applications. The components for each chart type were created with the help of NVD3 – a library allowing to build re-usable and highly customizable charts for D3.js library.

5.2 User Interface

The GUI of chart generator consists of 3 input fields – for Data, Style and Mapping inputs. The input for dataset is a file-select button that accepts XML files. Since an XML for the data input can be quite large, it was preferred to use file-select input more than text field. On the contrary, for the inputs defining graphical element and mapping, text-area fields were selected, as long as it makes manipulations with visual attributes more convenient. A generated chart appears on the same web page next to the input fields.

In addition, one more text field was added, which allows to easily generate XML files for the data input by specifying the parameters of data in config.

[1] Available at `https://github.com/reactjs`.

[2] Available at `https://d3js.org/`.

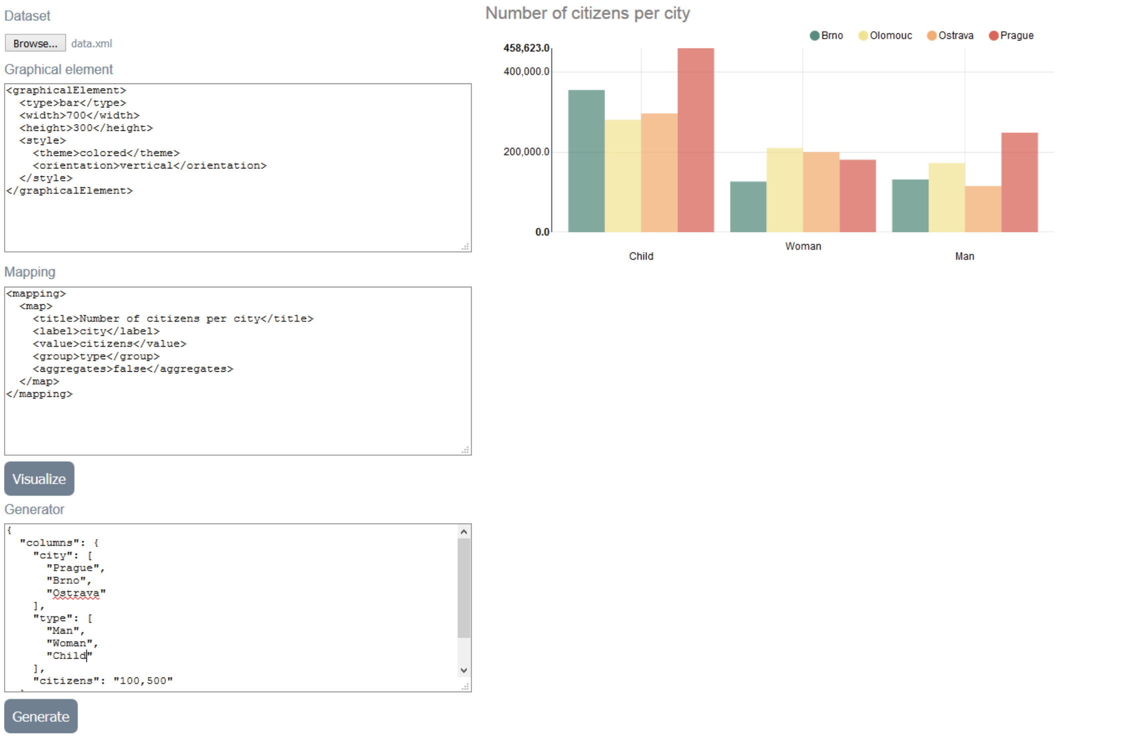

Figure 5.1: User interface of the tool with an output example.

Screen shot of the tool interface (along with a sample output) can be found at Figure 5.1. Several other tool outputs for different types of data are presented in Appendix B.

5.3 Generator

The principle of the chart generator's work includes the following steps.

1. Parsing of inputs using xml2js library.
2. If necessary, aggregating the data according to the type of chart and mapping input.
3. Passing the final data to the appropriate React component for visualizing the required chart type.

It is also worth mentioning, that the tool can generate multiple graphical outputs. Firstly, user can select multiple XML file for the Data input. Secondly, user can provide several `<map>` elements in the Mapping input. As a result, each set of data will be visualized using each mapping provided by user.

5.4 Script for Data Generation

To make using the chart generator more convenient and user-friendly, I added a possibility to generate test data inputs directly in the tool. The desired format and number of values are specified by user in JSON config. The values are then randomly generated based on the specified parameters.

For the config format, compatibility with external tools was not required, which left more freedom in choice of the language. JSON language was selected as the format for describing data because of its minimalistic laconic syntax, which makes it easier for user to play with config parameters. In addition, natural compatibility with JavaScript allows to speed up the process of data generation.

In the config user describes each data dimension with its name and possible values. The format of values can be specified in two ways.

- By providing enumeration of possible values (works for both numeric and string data dimensions).
- By setting the range limits (works for numeric data dimensions only).

Alternatively, the script which generates data can be executed from command line.

5.5 Possibilities for Improvement

Some possible directions for evolution of this tool include the extension of supported chart types and individual style attributes by adding new React components. Another point of interest is the enhancement of the tool interface to allow users to specify inputs using GUI wizard instead of entering them in XML format. Lastly, it would be appropriate to implement a custom tool for making user surveys using the generated graphics.

Chapter 6

Research

6.1 Survey

To carry out a sample user research with the help of the implemented tool, I selected several hypotheses or graph attributes to be tested.

1. The first subject of research was visualization of composition of values. This type of data, as has been already mentioned in Chapter 3, is most often represented with pie chart. However, Few suggests in [10] that in case of large amount of composition values, horizontal bar chart becomes a more effective and intuitive media to display this type of data. I was curious to test this hypothesis and, in case it is confirmed, to discover the boundaries for the number of values, in which the pie chart is still preferable. Some modern user experience design specialists even affirm that pie chart is acceptable only when it displays no more than 5 values (see [8]), but this assumption seemed questionable to me. So, I decided to test the versions of pie and bar charts with 6, 8, 10 and 12 values.

2. Secondly, I was interested to explore different numbers of qualitative states in gauge charts. Traditionally, this type of chart is displayed with 3 qualitative states represented by different colors. I tried to discover, if this number is actually optimal, by asking users to choose between gauge charts with various numbers of qualitative states.

3. Further on, speaking about the means allowing to visualize a single-value metric, it was interesting to define, which graphical presentation is more user friendly among the following: gauge chart, bullet chart, bar chart with a single bar.

4. The next subject of interest was comparison between vertical and horizontal bar charts in respect to number of values. It is said (see [10]), that horizontal bar chart becomes generally more preferable than the vertical one with increasing number of values. I tested the versions of these graphs with 4, 6, 7 and values to try to define the exact limit.

5. Finally, the last point of research was a study of time dependencies visualization. This type of data can be presented in continuous manner (with line chart) as well as in discreet way (with bar chart). It seemed important to compare these two presentations with relation to number of values, as well as to the time scale used (whether it is hours, days or years).

In all cases the point of research was investigated using 3 to 5 test questions, in each of them respondents were offered to choose between two graphical presentations of the same data the one that seems more comprehensible to them. The total of 20 questions were available to respondents in the form of a Google Forms survey[1].

6.2 Evaluation of Results

When it was time to evaluate the results, the survey had been filled out by 120 respondents. The majority of them were IT students or technical specialists at the age of 20 to 35 years.

The results interpretation for individual research topics is summarized below.

1. The hypothesis, that horizontal bar chart becomes more preferable media for visualizing compositions of values with large number of values, has proven to be true. However, according to the survey results, even with 10 composition values, most respondents still preferred pie chart over bar chart. So, I would suggest using horizontal bar chart for visualizing composition of values, when the number of values exceeds 10.

   Some further advisable directions in study of this type of data includes comparison of the above-mentioned chart types in respect to the extent of values (for example, if some composition values are disproportionally small), it would also be interesting to know if the users preferred the slices of pie chart to be labelled with percentage or absolute values.

2. The comparison of gauge charts with different numbers of qualitative states confirmed that the classical number of 3 seems more preferable to the majority of respondents. However, it is advisable to carry out more tests on this topic using different color combinations and extents of qualitative states.

3. Generally speaking, the questions related to visualizing single-value provoked uncertainty in respondents distinctively more often, than any others. The meaning of qualitative states was sometimes not clear to users and they tended to choose as simple display media as possible (for example bar chart with a single bar, which is definitely not a usual way to present such type of data). This leads me to the conclusion that a designer of information dashboard must be certain, that the context of qualitative states is familiar to the potential users, when choosing gauge or bullet charts to visualize single-value metrics. Otherwise, it seems more appropriate just to display a value as a text (probably along with a target value or deviation from a target value in a percentage).

   Besides, it would also be interesting to investigate presentation of a set of single-value metrics, which offers several possible opportunities.

4. Regarding the comparison of vertical and horizontal chart, it was discovered that the first one is preferred by the overwhelming majority of users for visualizing several instances of some measure. However, when the number of instances was large enough, the share of respondents, who selected the horizontal version, significantly increased. That makes me think that the hypothesis which suggested using horizontal bar chart

[1] The link to the survey is https://goo.gl/forms/hwiEiGDf7Dlj9RcI3; summary of the responses is available after filling the form.

for larger sets of values, might be true. The exact limit has not been discovered under the scope of this research, though.

Furthermore, this research took into consideration only the simple version of bar chart. For future studies, it would be interesting to compare the multi versions of both vertical and horizontal bar chart as well. In addition, it is advisable to investigate the influence of descending or ascending sorting of values on the comprehensibility of bar chart (of course, sorting by value is not possible in case of time dependencies; time ordering of values should be applied then).

5. As for visualizing time dependencies, most respondents voted for line chart regardless of the time scale. However, when the number of values was quite small (less than 6), the majority of users preferred the bar chart presentation.

   It is necessary to notice, that as well as in the previous point, only the simple versions of charts were tested. It would be, thus, advisable to explore more complicated time dependencies requiring the use of multi-line or multi-bar charts as well.

Chapter 7

Conclusion

As it has been observed, trendy, impressive and visually appealing graphical design doesn't necessarily have to provide clear and effective visual communication and usability. Despite of the rapt attention that has been drawn to the data visualization over the past decade, a lot of effort has still to be put into the proper investigation of various aspects of information dashboards design.

In this work I focused on studying types of data that can be typically found in information dashboards, their attributes and the ways they can be visualized. After the initial study of expert literature written on this topic, I singled out several kinds of data most frequently used in information graphic and focused the further efforts on examining their properties and possible graphical representations for them. After that, XML/XSD model describing selected kinds of data and visualizations was designed. Based on this model, the tool for automatic generation of visual data representations was implemented. Finally, using the graphics generated with the help of implemented tool, I carried out usability test on real users in the form of survey, where respondents were meant to choose the best among different graphical presentations of the same data.

This usability research aimed to investigate some selected aspects of visualizing the kinds of data being examined in this work. After the results of the survey were carefully analysed, few usability metrics and recommendations were proposed, that can be applied when designing information dashboards to improve the quality of data readability.

Some possible ways to evolve the results of this work in future studies have been already suggested in chapters 5 and 6. Among others, they include the following options.

1. Further researches on the already implemented types of data and visualizations.
2. Adding support of some other, more sophisticated, types of charts and data.
3. Improvement of the tool interface allowing users to specify the parameters of data visualizations with the means of GUI controls, not only by specifying inputs in the XML format.
4. Extending the functionality of the tool by adding the possibility to generate entire dashboards, not only individual widgets.
5. Implementing a custom tool (instead of Google Forms) for conducting user surveys using the generated graphic.

The results obtained in this project can be further expanded and utilized to put together a framework containing a set of metrics and tests, revealing usability problems of information dashboards design. Thus, some usability and design issues of this type of user interface can be detected automatically and at the early stages of implementation, without costly and time-consuming usability testing on real users.

Bibliography

[1] *HTML5 & JavaScript Line Charts*. [Online; visited on 30.04.2017].
Retrieved from: http://canvasjs.com/html5-javascript-line-chart/

[2] *KPI Dashboard*. [Online; visited on 29.04.2017].
Retrieved from: https://www.klipfolio.com/resources/dashboard-examples/supply-chain/kpi-dashboard

[3] *Marketing Metrics and KPIs*. [Online; visited on 30.04.2017].
Retrieved from: https://www.klipfolio.com/resources/kpi-examples/marketing

[4] *Operational Dashboards and Business Activity Monitoring (BAM)*. [Online; visited on 29.04.2017].
Retrieved from: http://www.informationbuilders.com/products/webfocus/operational_dashboard

[5] *Sales Product Performance*. [Online; visited on 29.04.2017].
Retrieved from: https://www.klipfolio.com/resources/dashboard-examples/sales/product-sales

[6] *Types of Dashboards: The Analytical Dashboard*. [Online; visited on 29.04.2017].
Retrieved from: http://www.dundas.com/support/blog/types-of-dashboards-the-analytical-dashboard

[7] *What are Business Metrics?* [Online; visited on 29.04.2017].
Retrieved from:
https://www.klipfolio.com/resources/articles/what-are-business-metrics

[8] Bales, R.: *Designing Charts - Principles Every Designer Should Know*. [Online; visited on 07.05.2017].
Retrieved from: https://uxdesign.cc/designing-charts-principles-every-designer-should-know-5bd3969a0150

[9] Eckerson, W. W.: *Performance dashboards: measuring, monitoring and managing your bussiness*. Hoboken: Wiley. 2006. ISBN 978-0-471-72417-9.

[10] Few, S.: *Information dashboard design: the effective visual communication of data*. Sebastopol: O'Reilly. 2006. ISBN 978-0-596-10016-2.

[11] Harris, R. L.: *Information Graphics: A Comprehensive Illustrated Reference*. New York: Oxford University Press. 1999. ISBN 0-19-513532-6.

Appendix A

XSD Model of Data

Dataset model:

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="dataset">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="values">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="value" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:any maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Mapping model:

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="mapping">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="map" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="title"/>
              <xs:element type="xs:string" name="label"/>
              <xs:element type="xs:string" name="value"/>
              <xs:element type="xs:string" name="group"/>
              <xs:element type="xs:boolean" name="aggregates"/>
              <xs:element name="ranges" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:decimal" name="range"
                     maxOccurs="unbounded" minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Graphics model:

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="graphicalElement">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="type"/>
        <xs:element name="type">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="bar"/>
              <xs:enumeration value="line"/>
              <xs:enumeration value="pie"/>
              <xs:enumeration value="gauge"/>
              <xs:enumeration value="bullet"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element type="xs:integer" name="width"/>
        <xs:element type="xs:integer" name="height"/>
        <xs:element name="style">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="theme">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="mono"/>
                    <xs:enumeration value="colored"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:any maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Style extension for bar chart:

```
<xs:element name="orientation">
  <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="vertical"/>
          <xs:enumeration value="horizontal"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="dateFormat" minOccurs="0">
  <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="date"/>
          <xs:enumeration value="time"/>
          <xs:enumeration value="weekday"/>
          <xs:enumeration value="month"/>
          <xs:enumeration value="year"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Style extension for line chart:

```
<xs:element type="xs:boolean" name="marks"/>
<xs:element name="dateFormat" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="date"/>
      <xs:enumeration value="time"/>
      <xs:enumeration value="weekday"/>
      <xs:enumeration value="month"/>
      <xs:enumeration value="year"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element type="xs:boolean" name="isArea"/>
<xs:element type="xs:boolean" name="fromZero"/>
```

Style extension for pie chart:

```
<xs:element type="xs:boolean" name="donut"/>
```

For bullet and gauge charts no style elements, besides color scheme, are supported.

Appendix B

Examples of the Tool Output

Figure B.1 demonstrates the tool output in case of 3 different data inputs for bullet chart.

Figure B.1: Tool output presenting 3 bullet charts.

Figure B.2 shows the tool output for a single-value data visualized with gauge chart.

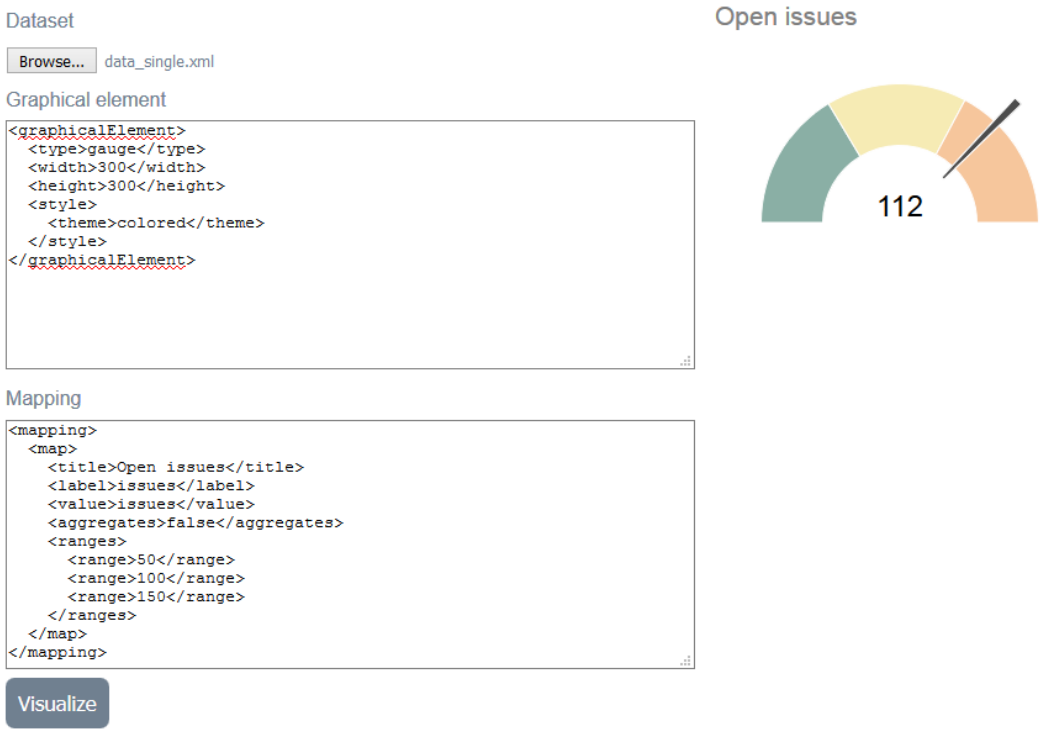

Figure B.2: Tool output presenting gauge chart.

Figure B.3 shows the tool output presenting the development of 2 measures in time visualized with multi-line chart.

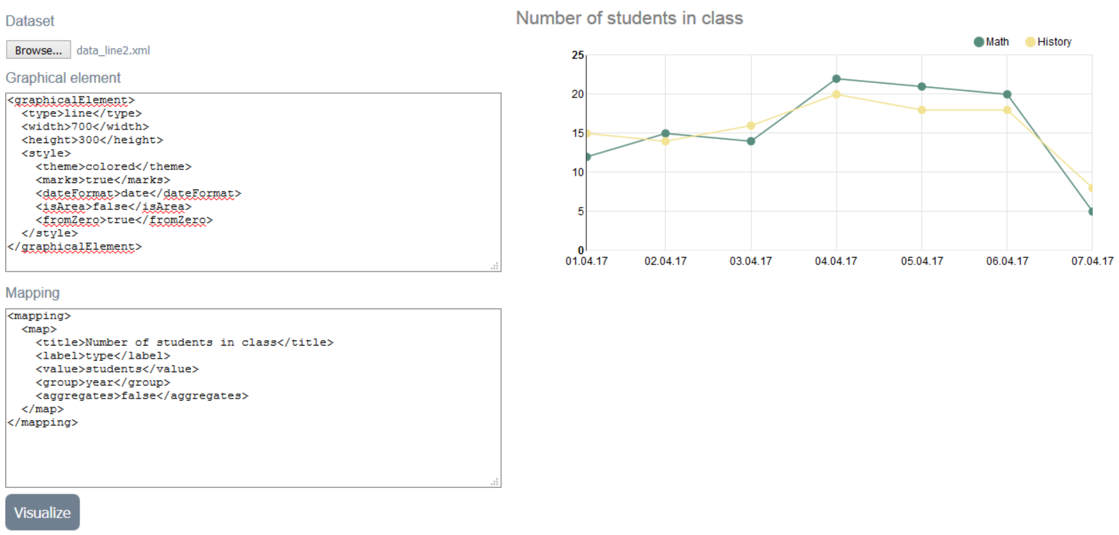

Figure B.3: Tool output presenting multi-line chart.