

A Comparison of Static and Dynamic Visualizations for Time-Oriented Data

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Barbara Neubauer

Matrikelnummer 0625063

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.-Prof. Mag. Dr. Silvia Miksch
Mitwirkung: Proj.Ass. Mag. Alexander Rind
Univ.Ass. Dipl.-Ing. Dr. Wolfgang Aigner

Wien, 08.05.2012

(Unterschrift Verfasserin)

(Unterschrift Betreuung)

A Comparison of Static and Dynamic Visualizations for Time-Oriented Data

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Medical Informatics

by

Barbara Neubauer

Registration Number 0625063

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.-Prof. Mag. Dr. Silvia Miksch
Assistance: Proj.Ass. Mag. Alexander Rind
Univ.Ass. Dipl.-Ing. Dr. Wolfgang Aigner

Vienna, 08.05.2012

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Barbara Neubauer
Senfgasse 1, 1100 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasserin)

Acknowledgements

I want to thank my advisors Alexander Rind and Wolfgang Aigner for their great support and assistance during my thesis. They were always available if I needed an advice or any help.

Furthermore I want to thank my family for their financial and emotional assistance during my study at university. They provided the right basic conditions to successfully accomplish my study and to stay the course.

Finally special thanks go to my boyfriend Roman who supported me with proofreading and technical advice. He helped me with his extensive excel-macro knowledge and with the program Lyx which thinks that 511 rows in a table are enough for every one.

Abstract

Visualization techniques for time-oriented data can be divided into two main categories: dynamic and static visualizations. Dynamic visualizations map the time directly to the time and present the data of different time-points or -intervals successively. This means the visualization changes its appearance over time. Compared to that, in static visualizations the factor time is mapped to space (e.g. a line chart where the time is projected to the x-axis). Static visualizations can be further differentiated in diagrams which present only states or diagrams also visualizing the development by e.g. some kind of arrows or lines.

The aim of this thesis was an empirical comparison between a dynamic and two static visualizations which are suitable for analyzing relationships of two variables and their development over time. The chosen dynamic visualization presents the development of the data with an animated scatter plot. Within the static visualizations (“*Small Multiples*” and “*Trace view*”) the time intervals and the data recorded in those time intervals are displayed in small pictures arranged side by side. Additionally data points belonging to an object are connected by lines in the “*Trace view*”. Those connections between the data points are continued over all small pictures and therefore visualize the whole development of the parameters explicitly. To make the comparison possible it was necessary to extend the prototype TimeRider by the “*Small Multiple*” visualization with the possibility to activate the traces resulting in the previously described “*Trace view*”.

The quantitative study was carried out over the internet, where each of the 29 participants had to solve tasks with all visualizations. The tasks covered three topics: the examination of trends, the identification and analysis of outliers as well as the identification of clusters and their development over time. The analysis of the results showed that all visualizations caused similar success rates and completion times for tasks concerning outliers and clusters. Regarding data movements within clusters and small changes in the data the animation outperformed the other visualizations significantly in correctness. The results also showed that participants solved tasks concerning the analysis of specific data points and the comparison between them significantly faster and with higher success rates with the dynamic visualization. However for nearly every hypothesis the results were dataset-dependent and had to be analyzed split by dataset which reduces the significance of the statistical results. But overall the results indicate that animation is better or at least equal to the two static visualizations for the analysis of time-oriented data.

Kurzfassung

Visualisierungstechniken für zeit-orientierte Daten können in zwei Kategorien eingeteilt werden: dynamische und statische Visualisierungen. Dynamische Visualisierungen bilden die Zeit direkt als Zeit ab und präsentieren die Daten der verschiedenen Zeitpunkte oder -intervalle nacheinander. Das heißt, die Visualisierung verändert sich über die Zeit. Im Vergleich dazu wird in statischen Visualisierungen der Faktor Zeit explizit im Raum dargestellt (z.B. ein Liniendiagramm, bei dem die Zeit auf die X-Achse projiziert wird). Statische Visualisierungen können weiters danach differenziert werden, ob nur einzelne Zustände oder auch eine Entwicklung z.B. durch Pfeile oder Linien dargestellt wird.

Das Ziel dieser Diplomarbeit war der empirische Vergleich zwischen einer dynamischen und zwei statischen Visualisierungen, mit denen die Möglichkeit besteht, das Verhältnis von zwei Variablen zueinander und deren Entwicklung über die Zeit zu analysieren. Die dynamische Visualisierung zeigt die Entwicklung der Daten in einem animierten Punktdiagramm. In den beiden statischen Visualisierungen (*“Small Multiples”* und *“Trace View”*) werden die Zeitintervalle und die in den Zeitintervallen aufgenommenen Daten in kleinen, nebeneinander liegenden Bildern dargestellt. In der *“Trace View”* werden zusätzlich zusammengehörige Datenpunkte eines Objekts durch Linien miteinander verbunden. Diese Verbindung wird über alle Bilder der einzelnen Zeitintervalle fortgeführt. Dadurch wird die zeitliche Entwicklung der Parameterwerte explizit visualisiert. Um den Vergleich durchführen zu können wurde der Prototyp TimeRider um die *“Small Multiple”* Visualisierung erweitert mit der Möglichkeit zur Aktivierung der Linien-Darstellung für die *“Trace View”*.

Die quantitative Evaluierung bestand aus einer über das Internet durchgeführten Studie, bei der jeder der 29 Teilnehmer verschiedene Aufgaben mit allen Visualisierungen lösen musste. Die Aufgaben deckten dabei drei Gebiete ab: die Untersuchung von Trends, die Identifikation und Analyse von Ausreißern sowie die Identifikation von Daten-Clustern und deren Entwicklungen. Die Analyse der Erfolgsraten und der Antwortzeiten der Fragen zeigte, dass die drei Visualisierungen bei Ausreißer- und Cluster-Aufgaben ungefähr die gleichen Ergebnisse erzielten. Bei Datenbewegungen innerhalb von Clustern und kleineren Änderungen in den Daten lieferte die Animation signifikant bessere Erfolgsraten als die beiden anderen Visualisierungen. Auch bei der Analyse von einzelnen Datenpunkten und Vergleichen zwischen solchen ergab die statistische Auswertung, dass die Teilnehmer mit der dynamischen Visualisierung schneller und öfter korrekt antworteten. Es konnte allerdings eine Datensatz-Abhängigkeit bei allen Hypothesen festgestellt werden, die dazu führte, dass die Ergebnisse getrennt nach Datensatz ausgewertet werden mussten, wodurch die Signifikanz der Ergebnisse eingeschränkt wird. Insgesamt deuten die Ergebnisse aber darauf hin, dass Animation besser oder zumindest gleich gut wie die beiden statischen Visualisierungen für die Analyse von zeit-orientierten Daten ist.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Problem Statement	3
1.3	Research Objectives	4
1.4	Methodological Approach	4
2	Related Work	7
2.1	Visualization of Time-Oriented Data	7
2.2	Coordinated Multiple Views	14
2.3	Evaluation of Visualizations	15
2.4	Discussion	18
3	Problem description	21
3.1	Medical Scenario	21
3.2	Requirements	22
3.3	Status Quo	22
3.4	Research Agenda and Implementation Objectives	30
4	Design	33
4.1	Introduction	33
4.2	Requirements	34
4.3	Mockups	35
4.4	Final User Interface Design	41
5	Implementation	45
5.1	Features	45
5.2	Programming Environment and Software Dependencies	47
5.3	Program Structure	47
5.4	Summary	62
6	Evaluation	63
6.1	Introduction	63
6.2	Hypotheses	66
6.3	Method	66

6.4 Results	76
7 Conclusion and Outlook	95
7.1 Limitations	97
7.2 Future Work	98
Bibliography	101
List of Figures	105
List of Tables	107
A Introductory Presentation	109
B Questionnaire	115
C Training Tasks	119
D Evaluation Tasks	123

Introduction

Nearly every person was in medical treatment anytime in one's life and furthermore related to medical parameters which were collected during therapies and treatments. "Medical parameters" is a generic term for a variety of body related measurements, for example blood parameters, pulse rate, body temperature or respiration rate all refer to the group of medical parameters. These parameters reflect the patient's current health status and indicate the kind of disease of the patient. For example tests performed on blood samples give information about the physiological and biochemical state, e.g. a disease or organ functions. In former times every parameter had to be examined manually and thus analysis took a lot of time. Today's laboratory and medical equipment make it possible to analyze specimens automatically and record more parameters in less time. Therefore the amount of recorded data values had increased enormously.

During a person's life the same parameters may be collected over and over again, mostly in irregular time intervals. This means every medical parameter is time-oriented because its value usually changes over time. Often it is necessary to analyze the long-term development of these parameters in order to make valuable assertions about the course of disease or organ functions. The medication and the further treatment depend on how specific parameters evolved during the past.

Another topic in health care concerning the collected time-oriented data is the analysis of correlations and dependencies between different parameters and factors. For example a possible research objective could be the examination of effects of a specific medication or treatment on the blood parameters. Additionally it is necessary to analyze the development of such parameter correlations for recognizing the overall trend.

To solve these tasks advanced tools and techniques are needed. Usually statistical methods are used to analyze research questions but visualization techniques could be used to examine the whole data visually and to discover unexpected trends in the data. Especially in research, sophisticated visualization tools are needed to get insights of the medical data which would have been overseen by only applying statistical methods.

Information visualization is “the use of computer-supported, interactive, visual representations of abstract data to amplify cognition” [9]. Well designed visualizations work because of the huge cognitive capabilities of humans. Many visualization techniques have been developed during the last 30 years and every visualization has different characteristics and strengths. The question is which kind of visualization is best for which kind of task or more precisely for the medical tasks defined before. For example a simple line plot is very useful to observe the development of one specific parameter (see Figure 1.1).

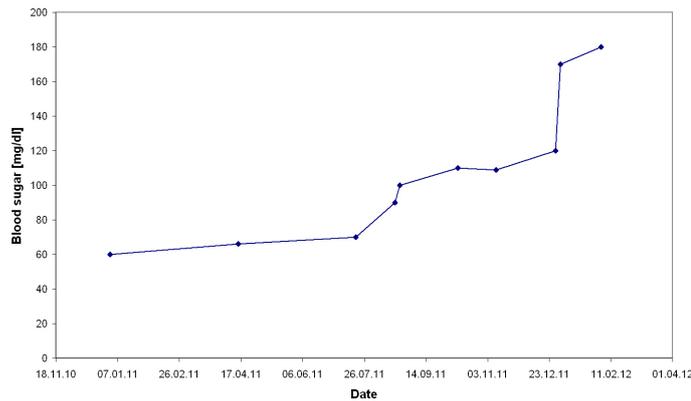


Figure 1.1: Line plot visualizing the blood sugar development of a person.

This often used, static graphic is easy to understand and interpret. But sometimes the correlation and dependencies between parameters should be examined. For such tasks a scatter plot is the typical used solution (see Figure 1.2).

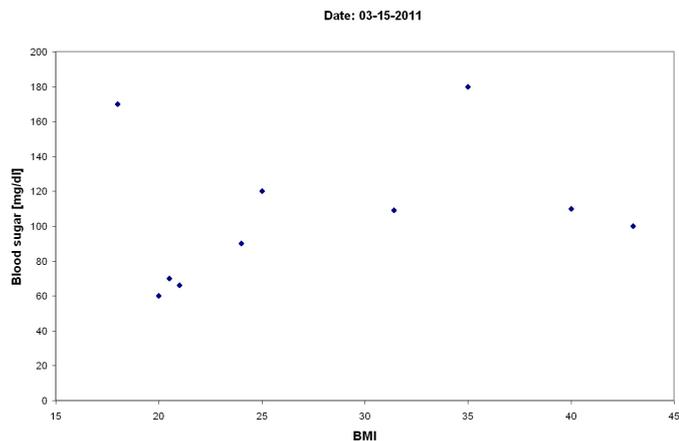


Figure 1.2: Scatter plot visualizing the correlation of blood sugar and BMI for several persons on one specific date.

Scatter plots are static snapshots of two parameters and in that way not applicable to visualize the time dependent, medical parameters. One possible solution could be an animated scatter plot presenting the plots of different dates successively. But such an animated scatter plot does not take the irregularity and the data wear of medical parameters into account. Of course there exist also other possibilities to visualize time-oriented data (see Chapter 2). Further research is necessary to find the most appropriate visualization technique for every kind of task.

1.1 Motivation

In health care interesting tasks involving time-oriented data exist. Visualization techniques could help to analyze many medical parameters in order to get insights and make valuable assertions about the patient's health status. The challenge for visualization techniques in the medical field is that they have to deal with 1) a big amount of parameters, 2) the irregularity and 3) the data wear.

A variety of possible visualization techniques does exist. Aigner et al. [5] presented about 100 different examples of visualization tools and techniques for time-oriented data, reaching from simple bar graphs to complex 3D views. The decision which data representation should be used within an interactive visualization tool is a big challenge for the designers of the systems. Every kind of task may need a different data visualization.

Coordinated multiple views combine different visualizations into one user interface. This concept may help to overcome the limitations of single visualizations if the right visualization techniques are merged. For example combining a line plot presenting the development of a parameter over time with a scatter plot displaying the correlation of two parameters provide two different views on the same data. This may lead to new interesting insights which would have been overseen if the two plots were not integrated into one user interface. Especially the coordination possibilities within coordinated multiple views could help to recognize new relationships and dependencies in the data.

The large quantity of possibilities for data representation makes it necessary to advance research on visualization. Especially medical data constitutes an interesting basis for sophisticated visualization techniques.

1.2 Problem Statement

During two courses in my study I developed the TimeRider prototype providing a solution for the visualization of medical, time-oriented parameters (see Chapter 3). TimeRider provides an animated scatter plot with many interaction possibilities. The tool was evaluated in a qualitative study (see Chapter 2) with the result that TimeRider is useful to gain interesting insights by physicians [21]. This study gives evidence for the effectiveness of animation in data visualization.

Although TimeRider is a well designed tool, an extension by another visualization technique would enable the user to observe the data from different perspectives. The advantage of such a

multiple view system is that it combines the benefits of multiple visualizations and can support users by solving tasks in the medical domain.

Because of the fact that different visualization types may support different kind of tasks, a comparison between them could give interesting insights of the capabilities of the visualization techniques. With a quantitative study the effectiveness of the visualizations could be measured and furthermore could advance research on data visualization.

1.3 Research Objectives

The main objective of this thesis was the comparison between the different visualization techniques by a quantitative study. Altogether three visualizations were compared in the study regarding *time* and *correctness*: the animated scatter plot, the Small Multiples and the Trace view. The study design and results are described in chapter 6 in detail.

For the comparison of the visualization techniques it was necessary to implement a Small Multiple visualization for the TimeRider. The existing visualization, the animated scatter plot, and the Small Multiples should be combined and coordinated within one user interface, following the coordinated multiple view concept. The previous TimeRider version also provided traces where data points are connected with each other. This trace mode should be also applied to the Small Multiples to get the third visualization: the Trace view. Additionally the user interface should be highly customizable, i.e. composed of modules which can be (de)activated separately. The design of the coordinated multiple view is introduced in chapter 4 and the implementation details are described in chapter 5.

1.4 Methodological Approach

At the beginning it was essential to study literature about different visualization techniques and coordinated multiple views to get an overview of the state of the art. The focus of the literature study was on the static visualization technique “Small Multiples” and evaluations regarding animation and Small Multiples. The research results were the basis for the planning and design of the new graphic user interface. The design process was carried out by paper prototyping¹, a fast technique for identifying different design possibilities with their advantages and disadvantages. The next part of this thesis was the implementation phase, integrating the new visualization technique into TimeRider and the coordination between the two plots. The program language Java was used to achieve platform independency and the Java visualization toolkit Prefuse provided the basis for the scatter plots.

After implementation, the evaluation of the extended TimeRider was planned. The choice of the right study design is essential for the success of the evaluation and the validity of the results. Therefore it was necessary to study literature about different evaluation techniques and methodologies. Carpendale [10] discussed methods for evaluations of visualization techniques.

¹The paper prototyping was carried out on the computer with a Mockup program (Balsamiq Mockups <http://www.balsamiq.com/products/mockups>, last accessed on 11th of March 2012).

She describes qualitative (e.g. observations or interviews) and quantitative (e.g. the laboratory experiment) evaluation techniques in her paper. Part of the experiment design and therefore, part of the quantitative evaluation, was the development of hypotheses and the identification of independent and dependent variables as well as statistical methods.

Based on the chosen study design appropriate tasks had to be defined and adequate datasets had to be prepared. The quantitative data composed of the participants' performance and user interactions were automatically recorded in log files. The qualitative part of the study included a questionnaire containing questions about the users' demographic data and their experiences.

The collected data of the study was analyzed statistically according to the defined hypotheses. The study results and the conclusion combined with suggestions for further development and research will be discussed at the end of this thesis (see Chapter 7).

Related Work

This section deals with basic techniques for visualizing time-oriented data. The role of animation in data visualizations will be discussed and the visualization technique called “Small Multiples” will be described in detail. The user interface concept “Coordinated Multiple Views” will be presented to combine both concepts into one user interface. Finally I will discuss the evaluation of information visualizations in general.

2.1 Visualization of Time-Oriented Data

In many application domains time-oriented data plays an important role. We only have to think of examples like medical records, project management plans, simulation or financial data. All these examples contain temporal information. Aigner et al. [5] present a systematic classification of techniques for visualizing time and time-oriented data. They also deal with the question how time-oriented data can be presented visually and define two major criteria to categorize these visualizations:

- the dimensionality of presentation space
- and the mapping of time.

The former divides visualizations into two main categories: visualizations which provide 2-dimensional presentation space and visualizations which provide 3-dimensional presentation space. The dimensionality will be not discussed in detail because only 2-dimensional visualizations are topic of this thesis. The decision to only discuss 2-dimensional visualizations is based on the fact that the previous TimeRider only provides a 2-dimensional animated scatter plot. The evaluation carried out during this thesis should be focused on the comparison between static and dynamic 2-dimensional visualizations and not on visualizations with different dimensionalities.

For the mapping of time Aigner et al. [5] also specified two categories. The first possibility is the mapping of *time-to-space*. This means that time and data is presented in the same visual system.

Such visualizations are also called static because they do not change over time. Usually one dimension of the display is used for the time. For example point plots, line plots or bar graphs use one axis to visualize the time. Because of the fact that time dependency is presented explicitly the visualizations are easy to interpret and therefore widely used. Another static visualization is called Small Multiples. Within a Small Multiple visualization the data is presented in small snapshots arranged side by side, one for every time point or object.

The second possible mapping of time is the *time-to-time* mapping. Visualizations with time-to-time mapping present the visualizations or frames for different time steps successively. Such visualizations are also called dynamic because of the changes in their appearance over time. This is useful if the screen space is required to visualize more than one aspect of the data items, e.g. for multivariate data where we have more than one parameter to visualize. With Small Multiples it is also possible to visualize two parameters but the pictures are small and details get lost. Depending on the frame rate the visualization is called slide show or animation. A slide show presents few frames successively in discrete steps whereas the animation is used to visualize the data fast.

In the following sections I will describe the two concepts *Small Multiples* and *Animation* in detail and I will present some studies, which compare the two techniques.

Static Visualizations – Small Multiples

Small Multiples are a common visualization concept where many small visual representations are arranged directly next to each other. Every small representation presents the same configuration (e.g. value ranges) but data of different time points or objects. Tufte [25] presented the concept of Small Multiples with the physiological background of human perception. He argued that human eyes can make distinctions within a small area very well, e.g. discriminating lines which are placed near each other or locating 100 points in one square centimeter is easy.

He also introduced the data density of a graphic to classify its graphical performance. The data density is the number of entries in the data matrix, which are the numbers that go into a graphic organized into a data matrix of observations by variables, divided by the area of the graphic. He found out that graphics differ very much in their data density and that maps contain the most information. The conclusion of Tufte's analysis [25] was that graphics should be based on large data matrices because it is better to present more information than less. The effort to interpret and handle additional information is low. Low information designs may arouse the impression that information was left out or that important facts were omitted. Data-rich design on the other hand gives credibility of statistical evidence. So the goal by designing graphics is to maximize the data density which is achieved by increasing the size of the data matrix. Tufte presented two ways to enlarge the data density. The first way is to increase the volume of data which leads to shrinking the shape of the data objects, e.g. using smaller dots for scatter plots or thinner lines in line plots. The second way is to reduce the area of the graphic. This way leads to the concept of Small Multiples which is according to Tufte's conclusion, an effective and powerful graphical design.

One possible implementation of Small Multiples for time-oriented data is to present the frames of a movie. This means every picture shows the same combination of variables but different portions of time. The viewer is able to concentrate on the change in the data during time. Another possible implementation of Small Multiples is to present the developments of different objects in the small pictures. In such a visualization it is easy to compare the data of different objects at a glance. Gemignani [12] presented some important requirements on designing Small Multiple visualizations that should be fulfilled. First of all the arrangement of the small pictures should follow a logical order, e.g. the common used read order: top left to bottom right. Another requirement is to keep the small visualizations as simple as possible. Otherwise the observer would be overstrained by the amount of information presented within a lot of small visualizations. The last important thing to consider is that all small representations have to share the same scale and size. Figure 2.1 demonstrates an example of a Small Multiple visualization created with R [3] and the ggplot2¹ library using data about diamonds packaged with ggplot2.

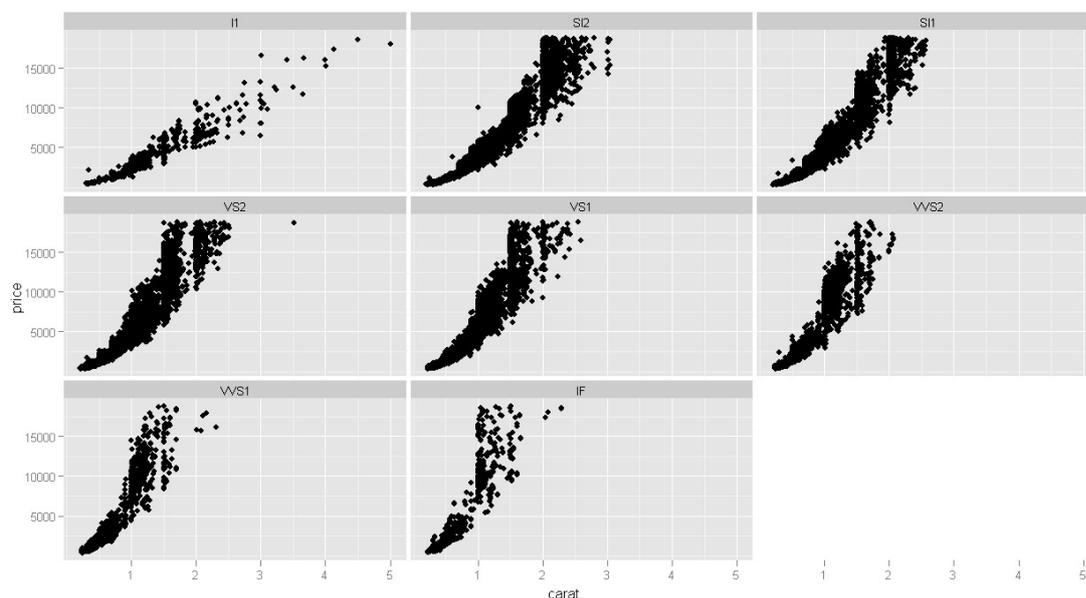


Figure 2.1: Small Multiple visualization presenting data about diamonds. The price and the carat are mapped to the axes. The images show the dependency of price to carat related on the clarity of the diamond (the diamonds get more and more clear from top left to bottom right). This visualization was created with the statistics program R [3] and the ggplot2 library. The diamonds data is part of the ggplot2 library.

The conclusion of Tufte was that Small Multiples are well-designed shrunken, high-density graphics which are easy to use and efficient in interpretation. The advantage of Small Multiples is that the user gets an overview of the data and that it is possible to compare the data at different

¹The ggplot2 library was downloaded from <http://cran.r-project.org/>, last accessed on 26th March 2012.

time points or the data of different objects. This visualization technique can be applied to any existing visualization technique, it is only necessary to create snapshots for different time points or different objects. On the other side there are also disadvantages regarding the Small Multiple visualization. The number of Small Multiples is very limited to the available screen size. Too many Small Multiples make the display confusing and hard to overview the whole development of the data.

Dynamic Visualizations – Animation

Animation is still a highly controversial topic in the field of information visualization. Many studies have been done to investigate the benefits and disadvantages of animation.

Few [11] described the capabilities of an animated visualization compared to a static line plot. He introduced four characteristics to describe the development of time-oriented data: the magnitude of change (the difference between two values of different time points), the shape of change (the path which is formed by the data values developing over time), the velocity of change (the speed or rate at which changes occur) and the direction of change (the overall trend of the data). These four characteristics are easy to extract from line plots. But the main disadvantage of line plots is that they are limited to one parameter at a time. It is often necessary to analyze the correlation between two parameters. Few proposed an animated scatter plot to examine such correlations. But he recognized that this animated scatter plot also cause some problems. It is not possible to compare data values of different time points very easy. Furthermore the observer is not able to remember all steps of the development of the whole data. In order to overcome those problems and to successfully apply all of his characteristics to the animated visualization he introduced *trails*. These trails connect data points which belong together. Altogether with trails it is possible to determine all of the four data characteristics and to compare data of different time points easily.

A similar example of an animated bubble chart is the Trendalyzer [24]. An animated bubble chart is the same as an animated scatter plot but the marks can have different sizes, i.e. the size also visualizes the values of a parameter. The Trendalyzer was built to make geographical statistic data more interesting and understandable for a broader audience. Rosling is famous for his impressive presentations, demonstrating the capabilities of the Trendalyzer.² The Trendalyzer provides an interactive animated bubble chart displaying four data dimensions simultaneously (see Figure 2.2). The bubbles smoothly move across the plot according to the adjusted axis parameters. Additionally two further data parameters are mapped to size and color of the bubbles. The user is able to interact with the visualization by clicking on the bubbles. Selected bubbles are highlighted and are displayed with a trail where the trail visualizes all steps the bubble made during animation. Trendalyzer also provides some additional functions to help the user investigating the data, e.g. changing the axis scale to logarithm or adjusting the maximal size of the bubbles.

²Presentations with Trendalyzer can be found on <http://www.gapminder.org/videos/>, last accessed on February the 14th, 2012



Figure 2.2: The Trendalyzer provides an animated bubble chart to visualize the development of four different parameters over time. Two parameters are mapped to the axes of the plot, one parameter to the size and finally one parameter is mapped to the color of the bubbles. This example shows the correlation of life expectancy to income per person over time. The country bubbles are colored according to the continents they belong and the bubbles' sizes illustrate the countries' population sizes. Source: <http://www.gapminder.org/world/>, last accessed on February the 14th, 2012

Evaluations

Robertson et al. [23] analyzed the capabilities of an animated bubble chart similar to the Trendalyzer in contrast to static visualizations for presentation and analysis tasks with multidimensional data. They compared the animated chart with a Trace and a Small Multiple visualization. Within the Trace visualization all bubbles of different objects are connected with each other, similar to the trails in the Trendalyzer. Different transparencies indicate where the traces start and end. The Small Multiple visualization displays all objects with their traces separately in different frames. The Small Multiple visualization was implemented to deal with the clutter problem of the Trace visualization where often many lines lie across each other and make it difficult to identify trends of specific objects. They assumed that animation is very effective for presentation as it was demonstrated by Rosling and the study was aimed to explore the usefulness of animation for analysis tasks. The results of the study confirmed their assumptions. The study showed that trend animation was least effective for analysis tasks but very good for presentations. They

found out that with both, Small Multiples and Traces, the participants solved analysis tasks significantly faster and more accurate than with the animation.

Another study investigating the benefits of animation for analysis tasks was made by Rind et al. [21]. They analyzed the TimeRider [17] which was built to visualize medical data of diabetes patients. TimeRider provides an animated scatter plot similar to Trendalyzer, but with the possibility to visualize irregular sampled medical data (see Chapter 3). More precisely TimeRider also provides an animated bubble chart because it is possible to map the size to a parameter.³ The study was carried out with physicians to investigate the usability of TimeRider and the usefulness of the animation concept for medical staff to get insights from the data. The results showed that TimeRider provides an effective technique to observe the medical data where the participants were able to find and predict trends.

In the study of Griffin et al. [13] animation was also compared with the static visualization Small Multiples. The animation showed colored hexagons against a background noise combined to a cluster moving from one side of the plot to the other side. In the Small Multiple visualization six pictures containing six time steps of the moving cluster were presented. They set their main focus on the factors pace, cluster coherence and gender in their study and found out that the participants answered in most cases more quickly and more accurate using the animated maps. The results also showed that different paces had an effect on the effectiveness of animation depending on the cluster coherence.

A similar study comparing an animated dynamic graph with a Small Multiple visualization was done by Arachambaut et al. [7]. The first visualization animated the evolvement of a dynamic graph and the participants of the study had the possibility to stop, replay and move the time slider individually. In the Small Multiple visualization the development steps of the graph were placed beside each other chronologically.

They found out that their participants solved the tasks significantly faster with the Small Multiples, but the accuracy depended on the type of the task. Tasks where the participants had to determine sets of nodes or edges added to the graph or removed from the graph over time caused significantly more errors by using the Small Multiples. The animation supported the users by recognizing changes like adding or removing nodes within the graph over time.

Table 2.1 summarizes the studies presented so far concerning the visualization techniques, the focus of the evaluation, the used data and the results of the study.

³In further descriptions I will use “animated scatter plot” because the focus is on the animation of the scatter plot and not on the animation of the mark sizes.

Authors	Robertson et al. [23]	Rind et al. [21]	Griffin et al. [13]	Arachambaut et al. [7]
Visualizations	<ul style="list-style-type: none"> • animated bubble chart • trace visualization (bubble chart with trails) • Small Multiples (one picture for every object with trails) 	<ul style="list-style-type: none"> • animated scatter plot, animated trace visualization 	<ul style="list-style-type: none"> • animation of colored hexagons against a background noise • Small Multiples containing six time steps of the moving clusters 	<ul style="list-style-type: none"> • animated dynamic graph showing the evolution of a graph over time • Small Multiples presenting the steps of the evolution of the graph
Focus	Comparison with respect to presentation and analysis tasks.	Evaluate usability and the capability to get insights and to find trends in the data.	Comparison of the effectiveness of both visualizations to observe moving and growing patterns.	Comparison of both visualizations with respect to accuracy and time.
Data	Yearly recorded data of countries, e.g. GDP per capita growth	Irregular sampled medical data of diabetes patients, e.g. BMI	Random pattern generation	Graph data
Results	Animation worked very well for presentation tasks. Trace and Small Multiples were better for analysis tasks. Small Multiples leaded to fewer errors.	Many participants were able to find and predict trends in the data and therefore animation was useful to solve the tasks.	Using animated maps leaded to more accurate results. Different paces had an effect on the effectiveness.	Participants solved tasks faster with Small Multiples. Accuracy depended on the type of task: using animation to detect disappearing or appearing nodes and edges entailed more accurate results

Table 2.1: Summary of four studies concerning static and dynamic visualizations.

An animation critical study is the one of Tversky et al. [26]. They analyzed studies about animation in the field of teaching complex systems. The common background of all of these studies was that they had published that they got significant better results with an animated graphic compared to a static graphic. Tversky et al. found out that the source of these results could lie in other effects than the performance of animation. For example the animated graphics and the static graphics did not present the same information or did not provide the same interaction possibilities. It is a common known theory that interactivity supports learning, so the advantage of animation in some studies could be reduced to the advantage of interactivity. Furthermore the animated graphics often presented more detail information than their static opponents. Tversky et al. concluded that animation often fail for visualizing complex systems because animations are often difficult to observe and understand and so take more time to handle. They proposed that interactivity could help to overcome the perception difficulties of animation. But further research is necessary where both, animation and static graphics, provide equal interactivity possibilities to decide whether animation is helpful in visualizations or not.

2.2 Coordinated Multiple Views

Coordinated Multiple Views (CMV) is a visualization technique that supports different views, which are somehow coordinated, on the same data. This approach enables designers to combine the strengths of different visualizations in one user interface [19]. Coordination between different views means that user actions in one view, e.g. selecting an item, also affect the other views, e.g. coloring the selected item in the other view. Some common used coordination techniques used for information visualizations were presented by North et al. [18]. They defined a taxonomy which organizes the coordination strategies into two dimensions. The first dimension describes the three possible combinations of coordination (activating event \longleftrightarrow resulting event in other views): “Selecting Items \longleftrightarrow Selecting Items”, “Navigating Views \longleftrightarrow Navigating Views” and “Selecting Items \longleftrightarrow Navigating Views”. The second dimension classifies whether the information collections are the same or different in the coordinated views. By using different information collections a mapping of data items is needed to connect items to each other, e.g. a kind of relational database. For example a document can be interrelated with a dictionary but they do not represent the same data.

With these two dimensions, six cases can be distinguished depending on the coordination and the data collection of the views. A typical example of “Selecting Items \leftrightarrow Selecting Items” where two views have the same data collection is the following. The user selects, respectively highlights, an item in one view and the same item is immediately also highlighted in the other view. This coordination form is also referred to as *brushing and linking* [19]. An advanced form of selection is filtering where unselected items are removed and only selected items stay in the view. If the views use different data collections the related item(s) in the other view are highlighted, for example selecting a student in one view results in highlighting all his written papers in another view. The case of “Selecting Items \longleftrightarrow Navigating Views” using the same data collection for both views is also called *overview and detail view* [19]. This means that a selection of an item in an overview initiates that the other view navigates to the corresponding details of that

item. Using the coordination technique “Navigating Views \longleftrightarrow Navigating Views” means that the navigation, e.g. scrolling or zooming, is synchronized between the views. An example for *synchronized scrolling* is an HTML text editor where scrolling the markup text representation causes an automatic scrolling of the layout representation.

Another aspect of CMV systems is the kind of presentation of the different views within the user interface [8]. One type of presentation is called dual view which means that two views are visible *side-by-side* simultaneously. Roberts [22] described some typical variants of such *dual view* systems. Two variants of dual view systems are the *overview & details* view and the *focus & context* view. The principle idea of both is that one view displays an overview of the data and a second view presents a part of the data in detail. Another variant of dual view systems is the *difference* view focusing on the difference in the observed data. The *master & slave* view provides two representations where one view controls the other. Finally Roberts described Small Multiples as a variant of *dual views*, although they are not limited to two views.

The benefits of such coordinated multiple views are improvement of user performance, reducing the necessary user actions and information access time. Furthermore it supports the user by discovering unforeseen relationships between the data.

The OECD Regional Statistics

One example of a well designed CMV system is the OECD Regional Statistics Explorer [2]. It was developed to analyze social economic data of countries. Furthermore it should help to make insights about regional differences and about countries’ performances. The user interface is divided into three coordinated and linked views which contain altogether seven visualizations (see Figure 2.3). The first view contains a choropleth map, the second view a scatter plot, a data grid or a table lens. The last visualization component includes a dynamic histogram, a parallel coordinates chart or a time graph. The views are divided by interactive splitters to enable the user to scale the sizes of the views. The visualization components use the same data model and coloring scheme to provide the coordination between them. A selection in one of the visualization component is propagated to all the others.

2.3 Evaluation of Visualizations

The evaluation in the human computer interaction field is very important to examine the usefulness and characteristics of implemented visualizations and tools. The challenges arising in evaluating information visualizations are versatile. It is difficult to allocate the right focus, to ask the right questions and to choose the most appropriate methodology [10]. The overall aim is to get meaningful reports of studies and controlled experiments that could help to understand the potential and limitations of visualizations [20].

Carpendale [10] dealt with the topic of evaluations for information visualizations and proposed to do more evaluations in general and to be aware of the many different types of empirical methodologies. She summarized some typical methodologies for evaluations and described that

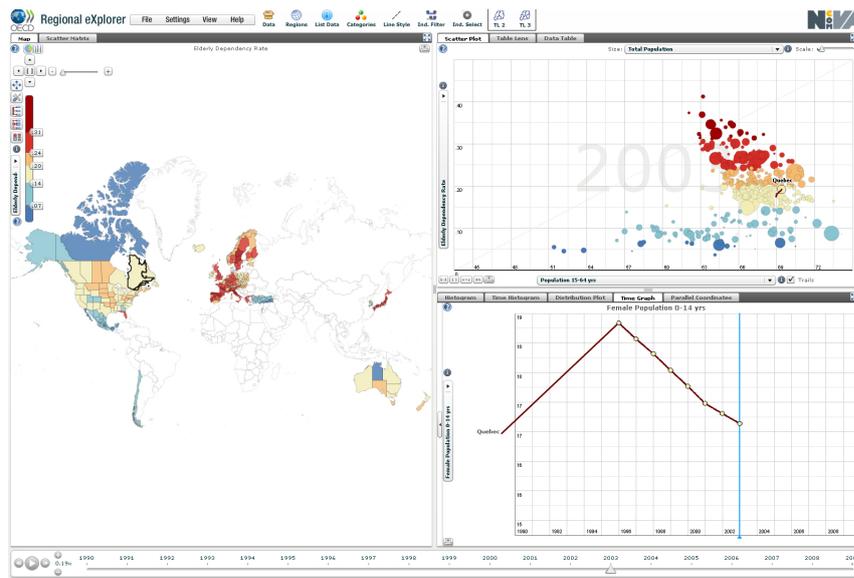


Figure 2.3: The OECD regional statistics explorer is an example of a well designed coordinated multiple view. In this example the country “Quebec” is selected: all of the three visualizations highlight Quebec or display data of that country (i.e. the development of the female population size). Source: <http://stats.oecd.org/OECDregionalstatistics/> accessed on March the 7th 2012

the choice of the right methodology is very important to achieve good results regarding a particular goal. There are three factors that influence the methodologies and the choice of them: the generalizability (i.e. can the results be applied to other people or situations), the precision (i.e. the precision of the measurements that were taken), and the realism (i.e. is the study context like the context where it will be used). There is always a trade-off between the factors when a specific method is chosen. For example a field study is conducted in a real situation while the observer tries to minimize his or her impact during observation and provides therefore most realism but not particularly precise and likely not particularly generalizable. On the opposite side there is for example the laboratory experiment where the experimenters fully design the study (i.e. the tasks, the setting and the procedure is planned) and thus realism is largely lost. Laboratory experiments or studies are typical quantitative evaluation methodologies [10]. Regarding the factors described before the quantitative studies have a relatively high precision and it is possible to find declarations about the generalization to a larger population. During the preparation of a quantitative study it is necessary to develop hypotheses, to identify independent variables (i.e. factors that may affect the hypotheses and that have to be studied) and to control variables which also influence the results. Hypotheses should be defined in precise and lucid language, should be focused on a problem testable within one experiment and should clearly state the conditions of the study [16]. The complexity should be eliminated by controlling other factors in the environment of the study. The dependent variables are measured during the study, e.g. the speed or accuracy. Afterwards there are statistical methods applied to the recorded measurements to analyze the correctness of the hypotheses and the significance of the results.

Qualitative evaluation methodologies are based on more realistic settings to get an improved understanding of existing practices and tasks [10]. There are primary two methods to get qualitative data: observations and interviews. There are many different techniques to collect the data, e.g. with video and audio tapes, logs, traces or screen captures.

Also within quantitative studies it is common to use qualitative methods. For example an experimenter could observe the evaluation and take notes about it or the participants could speak their thoughts as they progress through the experiment (*Think-Aloud Protocol*). Another possibility is to use questionnaires to collect the participants' opinions and experiences about the evaluated subjects.

Another paper with the topic information visualization evaluation was written by Plaisant [20]. She summarizes current evaluation methodologies and challenges which are specific to information visualization.

The evaluation practices summarized by Plaisant cover four main areas of evaluation: controlled experiments comparing design elements, usability evaluation of a tool, controlled experiments comparing two or more tools and case studies of tools in realistic settings. Plaisant also deals with possibilities to improve user testings. One important aspect to consider in evaluation are the used tasks. Empirical evaluations often only include simple tasks to decrease the complexity of the study. But studies should also cover tasks requiring users to compare, associate, distinguish, rank, cluster, correlate or categorize data. Plaisant suggests to report results separately for every task instead of reporting an overall performance for a set of tasks. This could help potential adopters to match tools or visualization techniques with their own tasks.

The selection of data and tasks could be improved by the development of task taxonomies (e.g. like it was presented by Andrienko & Andrienko [6]) and benchmark repositories of datasets and tasks. Tasks used in user studies should be "*simple enough to be accomplished in a short – or at least predictable – amount of time, and specific enough for performance to be measured*" [20]. To handle that challenge Plaisant proposed to allow users to examine the data freely on their own and to afterwards ask them about their experiences and insights they made. This procedure may lead to informative results but only if participants are motivated.

Lazar et al. [16] provide an overview of research methods and state of the art evaluations in human computer interaction. They describe the characteristics of the two possible study designs: within-subject design (every participant is exposed with every experiment condition) and between-subject design (participants are only confronted with one of the experiment conditions). The advantage of a between-subject design is that the participants do not learn from different task conditions and therefore learning effects can be avoided. Furthermore the time to accomplish all tasks is shorter for every participant. The main disadvantage of the between-subject design is that there are two completely different groups compared and that differences in the results could be caused by individual differences of the participants. Another disadvantage is that there is a larger sample size needed.

The characteristics of the within-subject design are constituted by the opposite of these advantages and disadvantages. With a within-subject design a smaller sample size is possible and the performances of the same participants under different conditions are measured. Therefore the

impact of individual differences of the participants is limited. An evaluation using a within-subject design has to deal with learning effects and fatigue. The time to complete the whole experiment can be quite long and participants get tired and bored and may cause an unexpected difference between the conditions.

Lazar et al. proposed to choose the within-subject design when tasks are complicated (e.g. reading, information retrieval and problem solving) where the individual differences of the participants would be too large.

2.4 Discussion

I have presented the concept of Small Multiples and studies about the use of animation for visualizing time-oriented data in the previous section. As I mentioned before there are different opinions about using animation and the benefits thereof. Some studies found evidence for the advantages of animation compared to static visualization techniques and other found the opposite.

The report of Few [11] introduced the main advantage of an animated scatter plot compared to a simple line plot: the capability to visualize multi-dimensional data. Furthermore he described the advanced animated scatter plot with traces and proposed that this solution is better than the scatter plot without traces. With the trace visualization described by Few the problem of clutter comes into play. The visualization is well suited to compare a few countries, but not a lot of countries where the lines of the traces may lie across each other. In the study of Robertson et al. [23] a similar scatter plot with traces was explained. They found the same clutter problem and solved it with a Small Multiple visualization where every object is visualized separately in one frame. This solution works also well for a small number of objects. If there are a lot of objects in the data the frames get very small and fewer details are visible, e.g. the labeling of the axes is not readable any more. Furthermore it is not possible to observe and compare the objects in the same context. For example if we want to identify outliers, a visualization where all objects are displayed in one plot, e.g. the animated scatter plot or the trace visualization, is better suited. However Robertson et al. concluded after their study that animation is useful for presentation tasks and Small Multiples and Traces are better suited for analysis tasks. The Trendalyzer is an example for the huge presentation capabilities of animated scatter plots which was often demonstrated by H. Rosling.

Arachambaut et al. [7] showed the advantage of animation very clearly with their study. Changes in the data can be recognized more accurately with animation than with static Small Multiple graphics. The results of the study of Griffin et al. [13] give another evidence for that theory. It is comprehensible that moving clusters on a pattern can be identified with animation more easy, of course dependent on the cluster size and the tempo of the animation. This leads to the important insight that animated visualizations should be highly interactive. The user must have the possibility to control the animation, i.e. stopping, forwarding and backwarding the animation as well as adjusting the tempo. Tversky et al. [26] came to a similar conclusion. As I described before they criticized some studies which published that animated graphics are better than static

ones. But Tversky et al. [26] proposed that interactivity is needed to overcome the perception problems of animation.

The presented studies indicate that the benefits of using animation depend on the task. The goal of this thesis is to find out which kind of tasks can be solved with animation more efficient than with static techniques visualizing time-oriented data. Specifically this thesis deals with the comparison of three different visualizations:

- an animated scatter plot,
- a Small Multiple visualization displaying different time intervals and
- a trace visualization composed of Small Multiples with traces (Trace view).

I have presented some typical approaches for the evaluation of information visualizations and described the difference between quantitative and qualitative studies and measurements. The comparison of the effectiveness, regarding time and correctness, of three different visualization techniques is typically done with a quantitative study.

I also presented the CMV concept for combining different views within the same program. The possibility to combine different visualizations and coordinate them leads to a solution where the positive aspects of different visualizations are merged and their limitations are reduced. We have heard so far about pros and cons of dynamic and static visualizations for time-oriented data. The idea of the diploma thesis is to provide a coordinated multiple view which combines dynamic and static visualizations into one program. The coordination of the visualizations is the most important point to consider within CMV systems. I presented the taxonomy of North et al. [18] for coordination possibilities between different views. Some of the described coordination techniques are also important for this master thesis. *Brushing and linking* is one of the most important coordination form – a selection in the animated scatter plot should be forwarded to the Small Multiple visualization and vice versa. Additionally all filter and mark properties should be applied to all visualizations.

The Small Multiple visualization is somehow a special type of a coordinated multiple view. Clearly *brushing and linking* should be implemented, but additionally *synchronized zooming* should be part of the implemented coordination techniques. Gemignani [12] and Tufte [25] defined Small Multiples as small images presenting different portions of time or different objects but with equal scale, size and view on the data. This means if the user zooms into one of the small images all other images should display the same detail of the data and therefore should be zoomed simultaneously.

Problem description

This chapter describes the previous TimeRider prototype. TimeRider provides a visualization to solve tasks in the medical field, i.e. to analyze data of diabetes patients. Therefore I will give an introduction to the medical scenario of diabetes patients and the challenges arising when medical time-oriented data should be visualized. Afterwards I will describe the visualization and the features of the TimeRider prototype. Furthermore I will introduce the aims and research agenda of this thesis constituted on the medical scenario, the previous TimeRider prototype and the related work.

3.1 Medical Scenario

The focus of the TimeRider prototype [17] and therefore of this thesis was on the visualization of parameters collected during treatments of diabetes mellitus diseased people. Diabetes mellitus is a widespread metabolic disease. Affected people have a high blood sugar because their body does not produce enough insulin or their cells do not react on the distributed insulin correctly. The disease causes symptoms like increased thirst and hunger or frequent urination. Usually diabetes mellitus can not be cured and therefore patients need to be treated for the rest of their lives. Additionally to the treatment the patients have to adapt their life style, e.g. by stopping smoking and by maintaining a healthy body weight. If patients are not treated they have high risk of many complications like cardiovascular diseases. During the diabetes treatment the patients' health status has to be checked regularly. The interval of these clinical checks depends on the patients' conditions and thus can reach from some weeks to several months. During the medical controls many medical parameters are collected, e.g. cholesterol or blood pressure. Additionally the prescribed therapy is recorded for every examination.

3.2 Requirements

The medical scenario I have described so far has an influence on a visualization tool which should help physicians to accomplish their tasks. There are three main aspects to consider by implementing an appropriate visualization tool: the *tasks*, the *data* and the *users*. We have heard so far that the data consists of many parameters collected during the patients' treatment. More precisely there are 35 time-oriented parameters representing the patients' health status and their therapy.¹ They can be further divided into numerical and binary parameters, e.g. the cholesterol is one of the numerical parameters whereas the therapy "Insulin" is part of the binary parameters. Table 3.1 presents all parameters (except the main data of the patient: name, gender and date of birth) collected during an examination. Caused by the irregular intervals the patients visit the clinic, the data is also irregularly sampled. Additionally the collected parameter values lose their medical relevance over time, i.e. data wear. For example the systolic blood pressure recorded one year ago has low explanatory power for the current health status.

The main task of the medical staff is to observe the long-term development of the parameters because they want to recognize the patients' courses of disease. For research circumstances it is necessary to consider the development of several parameters at the same time. Finding correlations and dependencies of different parameters are also part of the tasks. Different patient cohorts may react unequally for therapies and medications. For example it could be possible that female patients react differently to male patients in the correlation of a medication with some of the blood parameters. In succession a visualization tool should be able to visualize such differences between patient cohorts. Another type of medical task is to locate patients whose parameter values lie outside the typical ranges of values and it is necessary to identify the causes of such anomalies.

Clearly the users are people having enough medical knowledge to interpret the medical data. Furthermore the target group is composed of experts, using the program for sophisticated tasks. Table 3.2 summarizes the facts for the *tasks*, the *data* and the *users*.

A visualization tool, built to support physicians at their work should consider all these requirements. It should be possible to visualize many irregularly sampled data parameters, the data wear, correlations between parameters and the development of the data over time. The TimeRider prototype provides a visualization and many features to fulfill those requirements. In the following section the prototype with all of its features will be described in detail.

3.3 Status Quo

I created the TimeRider prototype during two courses in my study [17]. The aim was to build a highly interactive prototype for visualizing the irregularly sampled data of patients who are treated for diabetes. TimeRider provides a scatter plot where the parameters are displayed and animated. The animation should help to observe the development of the parameters over time. Different transparencies visualize the data wear of an examination point. The program also

¹I used parameters which were collected of diabetes patients in the hospital in Krems (state of 2010). Other hospitals may collect other or additional parameters.

Category	Binary	Numerical
Biometric data	Smoker	Weight[<i>kg</i>] Size[<i>cm</i>] Body Mass Index Systolic Blood Pressure[<i>mmHg</i>] Diastolic Blood Pressure[<i>mmHg</i>]
	Proteine in urine	Blood sugar[<i>mg/dl</i>] HbA1c[%] Cholesterol[<i>mg/dl</i>] Triglyceride[<i>mg/dl</i>] Creatinine[<i>mg/dl</i>]
	OAD therapy	SH, Met, Glit, DPP4, Acarb
	Insuline therapy	VZI, ALT, Misch
	Concomitant medication	ACE, Bbl, BP other, Statin, ASS
Organ damages	Retinopathy Stroke Coronary Heart Disease Peripheral Arterial Occlusive Disease Polyneuropathy Nephropathie	

Table 3.1: Time-oriented parameters collected in the hospital of Krems for diabetes patients split by category (e.g. biometric data) and data type.

provides functions like filtering, showing the normal and risk areas of parameters or mapping the parameters to size, shape and color.

Scatter Plot and Animation

The medical parameters are used for the axes of the scatter plot. A point in the plot represents a patient on a specific date. If two examinations of a patient have different parameter values on different dates the point moves from the coordinates of the first examination to the coordinates of the second examination. To differentiate between examinations which are only interpolated points (i.e. “moving points”²) and examinations of the current visible date they have different transparencies. In Figure 3.2 is a cut out of the plot during the animation. The dates of the ex-

²I will use “moving points” as name for interpolated points in further descriptions.

Tasks	Data	Users
<ul style="list-style-type: none"> analyze correlations between parameters and the development thereof recognize general trends in the data analyze differences between several patient cohorts, e.g. find clusters in the data find outliers in the data 	<ul style="list-style-type: none"> 35 time-oriented parameters representing the health status and the therapy numerical and binary parameters irregular sampling 	<ul style="list-style-type: none"> physicians (experts)

Table 3.2: The requirements on a tool providing a visualization for medical data, divided by *tasks*, *data* and *users*.

amination points can be approximately estimated by comparing the examination’s transparency and the transparencies in the animation legend (see Figure 3.3). If an examination has not got a successive examination the examination stays on the place where it is and will disappear after a predefined number of days (for example 50 days), while the transparency increases. The animation is navigated by media-player like controls and also the tempo of the animation is adjustable by the user.

Additional to the two axes there are three more possibilities to visualize parameters. Binary parameters can be mapped to the color or the shape of the examination points. For example if the color is mapped to “Coronary Heart Disease”, examinations of patients with coronary heart disease and patients without coronary heart disease can be differentiated by the colors of the examination points. The size of the points can be used to visualize a third numerical parameter. The higher the parameter value the bigger is the point in the plot.

TimeRider provides different ways to pan and zoom the plot. First of all the plot can be panned by drag and drop with the mouse and can be zoomed by moving the mouse wheel forward or backward. Another way is to use the two RangeSliders near the plot where the user has the possibility to pan and zoom every axis separately. Finally the last method to zoom the plot is to click the magnifier buttons.

Medical parameters often have risk and normal areas for their values. This means that there exists a value range which is normal for the parameter and an upper and a lower risk area outside the normal range. The TimeRider is able to visualize these areas with a blue background color in the plot for both axes. Furthermore it is possible to adjust the ranges in the program directly. In Figure 3.1 the blue background in the scatter plot visualizes the normal area of the parameter adjusted for the y-axis.

The details of an examination are displayed when the user moves the mouse across an examination point. That “details-on-demand” window contains all the information about the examination including all parameter values and the main data of the patient.

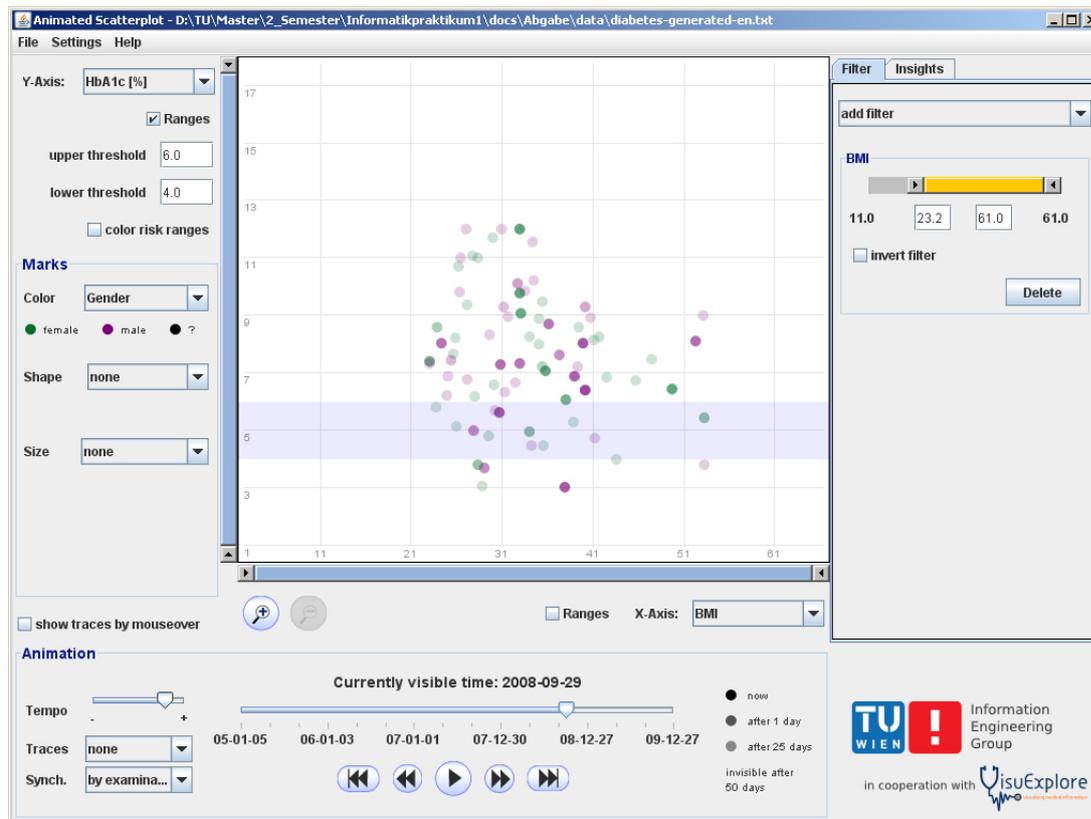


Figure 3.1: The basis for this thesis was the TimeRider prototype – an interactive animated scatter plot to visualize time-oriented medical data.

Finally TimeRider also provides an export function to save the current scatter plot visualization as image.

Traces

TimeRider also provides an alternative dynamic visualization – the Traces. This visualization is comparable to the trace visualization of Few [11] described in Chapter 2. Traces connect the examination points of a patient with each other and make it possible to observe the whole development till the current visible date at a glance. There are two possible trace views available in the program. The first view shows the traces combined with only the last examinations of the patients. This means that past examinations disappear if a new examination gets visible. In Figure 3.4(a) an example is pictured. All three images in this figure belong to the same patient. On the 16th of May is an examination point visible and the trace expands to the next examination. The next examination's date is the 9th of June. On the 9th of June the examination appears and the first examination disappears. In the image of the 9th of June the examination's point is bigger. This means that the examination's date is the current visible date of the animation. On

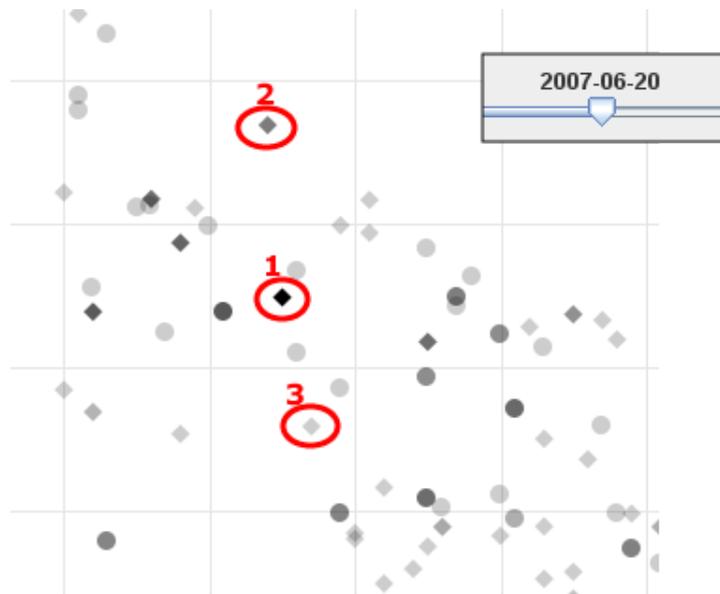


Figure 3.2: The small image in the right corner shows the time slider with the current visible date. In the red circle number 1 is an examination of date 2007-06-20. Because of the fact that this is a current examination the transparency of this point is zero (the point's color is black). The examination in circle number 2 has a higher transparency. This means that the last examination of this patient was about one month ago. The examination in the red circle number 3 has a higher transparency than number 2 and so the patient's last examination is about 2 or 3 months ago.



Figure 3.3: Legend for the animation transparencies.

the next day, the 10th of June, the second examination's point gets small again and indicates that this examination is a past one. The trace grows to the next examination.

The second trace view displays the traces combined with all past examinations of the patients. When this view is activated an examination will never disappear – all points stay visible in the plot. Figure 3.4(b) shows an example. As in the last example there is an examination with the growing trace pictured in the top image. On the 9th of June the second examination appears and now the first examination does not disappear. The size of the point is also dependent of the current date. This means the point of the second examination is big at the 9th of June and gets small again one day later.

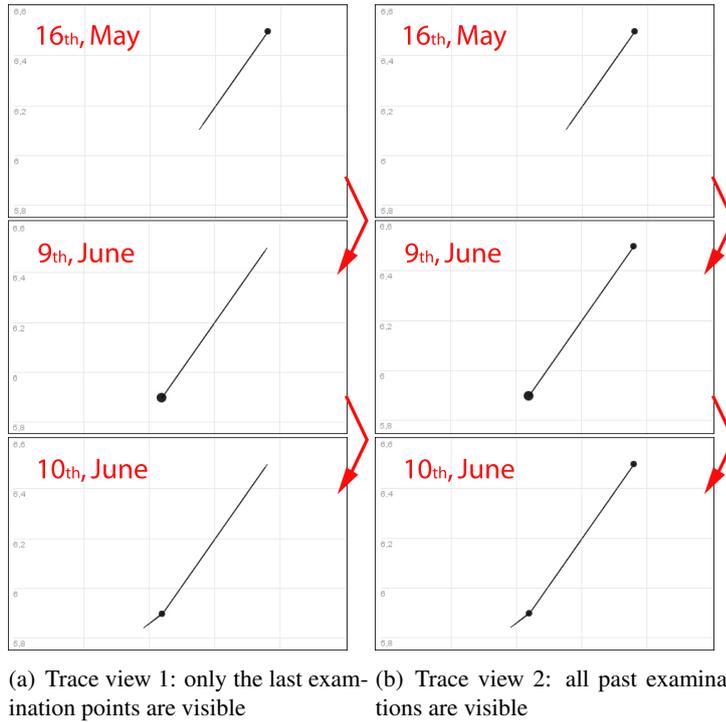


Figure 3.4: Trace view modes

If both trace views are switched off it is possible to activate the “trace by mouseover” function. This means that the whole trace (till the current visible time) of the patient appears in the plot when the mouse is moved across one of the patient’s examinations.

Filter and Selection

It is possible to compare many patients with the TimeRider, but often it is necessary to analyze subgroups of patients or specific parameter values. For that reason TimeRider provides three types of filter. With the name filter one or more patients can be selected which will be visible in the plot. For numerical parameters a value range is adjustable in the filter. Only patient’s examinations which have a parameter value within the filtered range are visible. The last type of filter is used for filtering binary parameters. For example if a “Smoker” filter is added and the selected value is “yes”, only smoking patients stay visible in the plot.

TimeRider also provides a selection feature: one or more patients can be selected by clicking on their examination points in the plot. As a consequence of that the selected patients are highlighted. Patients are highlighted differently in the different visualization modes (standard scatter plot vs. trace views). When the trace view is active the selected patient’s examinations and traces are black and all other traces and examinations have a high transparency (see Figure 3.5).

If there is no trace view switched on, selected patients are drawn as usual and all other points are drawn with a thin border (see Figure 3.6).

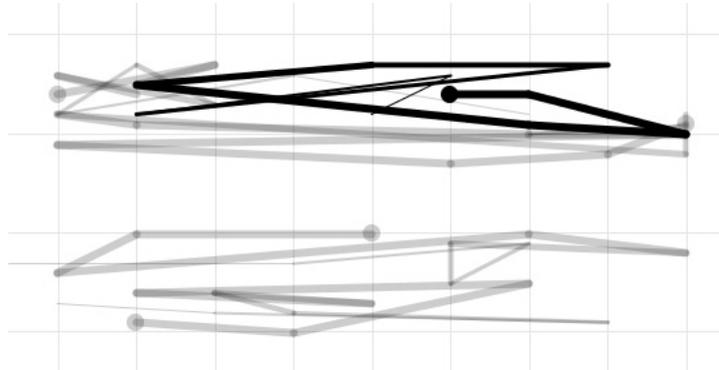


Figure 3.5: Selected patients' traces and examination points are black, all other traces and points have a high transparency.

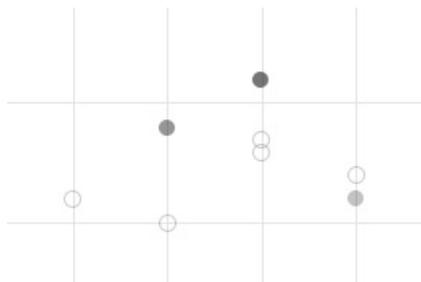


Figure 3.6: Selected patients' examination points are displayed as usual, all other points are empty with a thin border.

Synchronization

Another feature of TimeRider is the possibility to synchronize the data in four different ways. The first synchronization mode is the usual synchronization by the examination date as we have it seen in the previous sections. The second mode is the synchronization by the first examination date. This means that the first examinations of all patients will start at the same virtual "date". With that synchronization the curves of the different patients can be compared easily with each other. In Figure 3.7 the "time" parameter is selected for the x-axis. In this example it is possible to see that all examination curves start at the same time unit even though the dates of the first examinations of the patients are not equal in reality. The third mode works similar to the second synchronization but the data is synchronized by the last date. In this mode all last examinations of all patients have the same "virtual" date. Finally in the last mode the examinations are syn-

chronized by the ages of the patients at their examination dates. This means that values of, for example 50 year old patients, can be compared with each other.

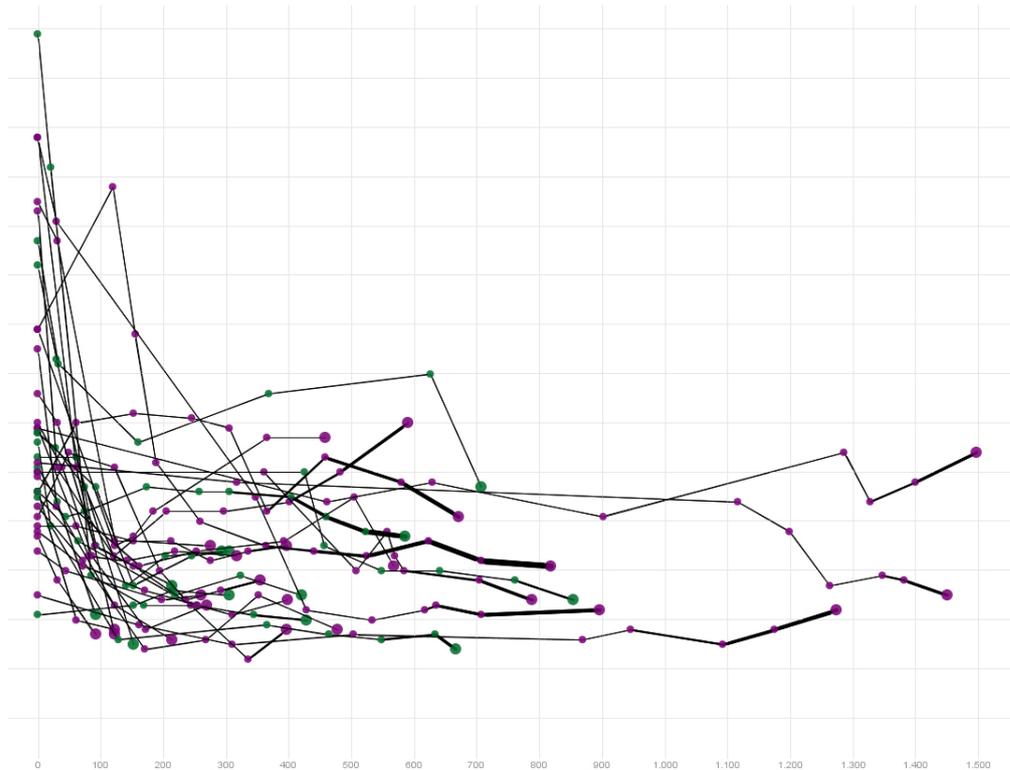


Figure 3.7: The scatter plot displaying data synchronized by first date. Due demonstration concerns the date was mapped to the x-axis (instead of using two medical parameters for the axes). With the synchronization modes it is possible to compare the development of the parameters of different patients very easy.

Insights

Insights made during analyzing the medical data can be easily stored with the TimeRider. They can be typed in a text field within the program and stored with a screenshot. All saved insights of the current session are displayed in a text area. Every time the program is started a new session starts and insights are stored in a new file. The storage location where insights and screenshots are stored is selectable by the user.

Settings and Language

All program settings like axis configuration or selected filters can be stored and afterwards loaded. This function is useful if the user wants to get the same view on the data again or

to show other people the data with a special program configuration. With the Reset-function it is possible to reset all adjusted program settings and to get back to the initial program state.

The language dependent data is saved in an xml-file. Therefore, the user can change the language by selecting a different language file in the program parameters at start-up. At the moment only English and German files have been created but it is easy to create a new language file. Also the risk areas of the parameters are stored in a configuration file. To change the risk areas it is only necessary to start the program with an adjusted configuration file.

3.4 Research Agenda and Implementation Objectives

The TimeRider, described in this section, provides a well designed solution for analysis tasks in the medical field. The animated scatter plot in TimeRider is able to handle the large quantity of parameters, the irregularity and the data wear. In the related work section I have already introduced the evaluation of the main concept of TimeRider. This qualitative study carried out with physicians showed that the animated scatter plot was very useful to get interesting insights of the data [21]. The TimeRider prototype and the study built the background for this thesis. Furthermore the study of related work of visualization techniques for time-oriented data had shown that the opinions about the benefits of animation are very controversial. In the related work section I also introduced static visualizations. I have discussed the pros and cons of dynamic and static visualizations and presented studies regarding this topic. The idea of this thesis was to combine both concepts, static and dynamic visualizations within one user interface to advance the possibilities for analysis tasks in health care and to be able to carry out a comparative evaluation of the visualizations.

The focus of the evaluation was on the comparison of the effectiveness of the dynamic animated scatter plot and two different Small Multiple visualizations. Because of the fact that animation is still a controversial topic in information visualizations this thesis was aimed to find out which tasks can be solved better or worse with animation compared to the static visualizations. In order to make the evaluation possible it was necessary to extend TimeRider by Small Multiples. To get the Trace visualization mentioned in the related work section³ the existing Trace view of TimeRider should be also developed for the Small Multiples. For the evaluation process TimeRider had to be adapted in a way that the user interface would be highly customizable, e.g. presenting only a specific visualization or disabling unnecessary functions. Furthermore the evaluation framework EvalBench [14] had to be integrated. Additionally both visualizations (animated scatter plot and Small Multiples) should be combined within one user interface. According to the CMV concept presented in the related work section both visualizations should be coordinated. At the end the extended TimeRider should provide more different possibilities to get insights of the medical data and furthermore to advance research on visualization of time-oriented data.

³The Trace visualization mentioned in the related work section consists of Small Multiples and traces connecting the records of the subjects with each other.

Another topic of the TimeRider extension was the missing granularity functionality. The initial TimeRider version did not provide different granularities, e.g. months or years, which would be necessary for long term evaluations of the parameters. Additionally the program settings had to be combined into one xml-file to provide a better adaptability and customizability.

In summary the goal was to achieve an extended TimeRider providing a customizable coordinated multiple view with a static and a dynamic visualization, adaptable for the evaluation and useful for presenting data in different views at the same time. The focus of the implementation was on the evaluation process but TimeRider should be also suitable for further research and applicable to solve the tasks in the medical field described at the beginning of this section.

CHAPTER 4

Design

During the design process many different ideas were worked out until the final user interface was chosen. In this section I will give a short introduction and describe the requirements on the user interface. Then some of the mockups and their pros and cons are presented. Finally the chosen user interface design is described in detail.

4.1 Introduction

The main focus of the first mockups was on the design of the coordinated multiple view and the collaboration of the integrated visualizations. Therefore the customizability was not part of the initial design process. Furthermore at the beginning of the design process it was planned to implement an additional static visualization: a line plot where time is mapped to the x-axis. There was one Small Multiple visualization planned for the scatter plot and one for the line plot displaying the data of the objects separately in different pictures. These conceptual ideas (i.e. the line plot and the two different Small Multiple visualizations) were not implemented because in the course of the design process it has been emerged that the comparison between the different visualizations is an interesting objective for the evaluation. Therefore the focus was transferred to the comparison between the visualizations instead of the collaboration of all visualizations. Another fact that supported that decision was that a qualitative study of TimeRider was already done by Rind et al. [21] and the evaluation of an exploration tool where all visualizations are combined into one user interface would also end up in a qualitative study. On the other hand the comparative evaluation of the visualizations usually is done by a quantitative study.

Nevertheless some mockups concerning the coordinated multiple view are described in the next section to summarize the insights made during the design process. The final user interface described in the last section shows the result of the design process and therefore does not include the line plot and the two different Small Multiple visualizations any more.

4.2 Requirements

The extended TimeRider should provide an adaptable CMV system. Although the coordinated multiple views were not part of the evaluation carried out during this thesis it was an aim to get a well designed CMV system for further research. During the design process several basic decisions had to be made. Roberts [22] described three points to decide when a CMV system should be created: first of all which form the visualizations should have, secondly how the data will be mapped to the selected form and finally how the user interacts with the system. Baldonado et al. [8] defined Multiple View systems by a model consisting of three similar dimensions: selection of views, presentation of views and interaction among views. Consequently it was necessary to choose the visualization forms respectively the selected views for the extended TimeRider. As I described in section 4.1 the visualization forms changed during the design process and at the end the animated scatter plot and one Small Multiple visualization were selected.

The presentation of views, described by Baldonado et al. [8], defines how the multiple views are presented in the user interface respectively on the screen. For example two views could be visible simultaneously beside each other or sequentially with the possibility to switch between them. CMV systems which use two simultaneous visible *side-by-side* views are called *dual view* systems [22]. In the resulting CMV system it should be possible to compare the data in the different visualizations easily. Therefore the extended TimeRider should also consist of such a dual view, presenting the two different visualizations side-by-side, both visible at the same time: the animated scatter plot and the Small Multiples.

Another important design issue, also described by Baldonado et al. [8] and Roberts [22], is the selection of appropriate exploration, interaction and coordination techniques. Many different interaction strategies, like filtering or navigating (e.g. zooming) have been developed. In principle there exist two types of interaction, *indirect* and *direct* [22]. Indirect interaction is carried out by buttons or sliders whereas direct interaction is carried out directly with the visualization. One example for direct interaction is *brushing*: when elements are selected and highlighted in one display, the same information is highlighted in any other linked display. The extended TimeRider should provide direct and indirect interaction possibilities similar to the previous TimeRider version. The coordination of the views should be based on the principle that both views should always contain the same information. Thereby it is possible to observe the same part of the data with different presentations. *Brushing and linking* is one example which supports that principle – a selected/highlighted item should be selected/highlighted in every visualization.

The requirements on the user interface with respect to the evaluation were concentrated on the flexibility and customizability of the system. For every visualization it should be possible to display it alone. Furthermore it is necessary to disable functions individually and to make them invisible. The focus should lie on the presentation of the visualizations. Therefore the user interface should be clearly arranged and not overcrowded to get space for the visualizations as much as possible.

4.3 Mockups

The following mockups show some design ideas for the coordinated multiple view including the scatter plot, a line plot and two Small Multiple presentations. For every mockup the main user interface structure, the different visualizations and the coordination between them are described. Additionally the pros and cons are presented. All mockups were created with the tool Balsamiq Mockups.¹

There are two main concepts which were used in the user interface designs: *superimposition* and *juxtaposition*. Superimposition means that different variables are visualized within one plot and within the same coordinate system. In contrast to that juxtaposition means that different variables are visualized in different plots which are placed next to each other.

Superimposition Variant 1

In the first mockup (see Figure 4.1) two plots are placed on top of each other. The line plot is placed on the top and the scatter plot is placed below the line plot. The parameters for the axes can be selected for the scatter plot. According to the selected parameters in the scatter plot, two checkboxes, labeled with the parameter names, are visible left beside the line plot. The user is able to turn on one or both parameters which should be visualized in the line plot. One further difference to the TimeRider described in the previous chapter is that the configuration of the mark settings moved into the tab panel on the right side and that there is a new feature available to select the granularity of time in the animation panel.

To get more flexibility within the user interface it is possible to hide the animation panel on the bottom and the tab panel on the right side. Furthermore it is possible to change the size of the two views by moving the horizontal splitter between the plots.

Line Plot

In the line plot, one or both parameters of the selected scatter plot axis parameters can be visualized superimposed. This means that if both parameters are turned on, for every patient two lines are visible. Both of the two variables have their own axis in the plot, one on the left and one on the right side. Different line colors should help to differentiate between the different parameters. Additionally two different colors are used to visualize the risk and normal ranges of the two parameters. Another difference between scatter plot and line plot is that the *trace by mouse over* feature is not supported in the line plot because the trace of the development is always visible in the line plot.

Small Multiples

Each of the both views (scatter plot and line plot) can be exchanged by their Small Multiple view. Every Small Multiple picture visualizes one object respectively patient and is a small copy

¹<http://www.balsamiq.com/products/mockups>, last accessed on February the 20th, 2012

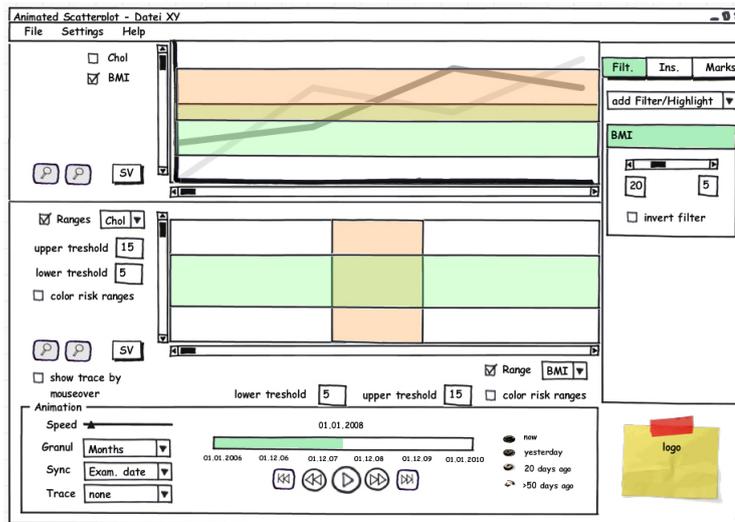


Figure 4.1: Mockup with superimposed variables in the line plot. The line plot is placed above the scatter plot.

of the “parent” view. This means the Small Multiples also provide the same functionality as the “parent” visualizations, e.g.

- the Small Multiples of the animated scatter plot are animated,
- the selection of patients is possible by clicking on the Small Multiple plot of the patient,
- *trace by mouse over* is only available for the Small Multiple of the scatter plot.

Coordination

The mapping of size, shape and color to parameters works similar in both plots respectively for all of the four views. For example selecting the parameter “Gender” for the property “color” means that the marks are colored in all views in the same style, e.g. female patients are colored red and male patients are colored green.

The animation is coordinated between the plots. Within the line plot (and the Small Multiple line plots) the current time is visualized by a vertical line which moves from the left to the right side. The synchronization of the data is also carried out in the line plot. This means that a synchronization by first date causes that all lines start at the first date.

Filters affect all views in the same way. A name filter removes or adds the patients’ marks and lines in the scatter plot and line plot or pictures in the Small Multiple views. If one of the logical or quantitative filters affects an examination, the examination point will disappear in the scatter plot and the line plot. If all examinations are filtered out, the whole line will disappear (in the line plot and in the trace view of the scatter plot). In the Small Multiple views the picture will

disappear if all examinations of a patient are filtered out. The selection of patients is coordinated between the different views. If the user selects one or more patients in the scatter plot the selection is also transferred to the other plot and vice versa. The selection in the line plot is visualized in the same way as the selection in the scatter plot with traces: unselected patients are light grey without colored lines and marks and selected patients have normal color and are not transparent.

The features pan and zoom are not coupled between the views. It is possible to zoom both plots separately. Only the pictures of the Small Multiple visualizations are coordinated for panning and zooming.

Pros and Cons

An advantage of the presented mockup is that the x-axis in the line plot has enough space to visualize the time. On the other side the most common used screens are widescreens and therefore does the y-axis get very little space compared to the x-axis. One positive aspect of superimposition can be found in the Small Multiples of the line plot. Every Small Multiple contains the data of both axis parameters and enables the user to analyze correlations between them easily. Nevertheless superimposition also causes problems. The double y-axis may induce wrong interpretations of the data when the values of the parameters lie in completely different intervals [27] [4]. Another problem with superimposition is that many colors and lines are displayed in one plot. If there are many patients to visualize, the plot is overcrowded and confusing. The layout may also cause problems by selecting the axis parameters. The selection of the parameters is placed near the bottom scatter plot view, but users read from left to right and from top to bottom in Western culture. Therefore it is maybe a bit confusing that the user has to adjust parameters for the line plot near the bottom placed scatter plot. If the plots switch their place, i.e. the scatter plot is placed on the top and the line plot on the bottom, the controls for selecting the x-axis would be placed between the two plots and make it more difficult to compare the two visualizations.

The Small Multiples visualizing every patient within a separate frame also cause a problem. It might be difficult to compare two patients when their Small Multiples are far away from each other.

Superimposition Variant 2

The second superimposed mockup (see Figure 4.2) works similar to the first one. The coordination between the plots equals the coordination described for variant 1. The difference between them is the placement of the two plots. In the second variant the two plots are placed side by side, the scatter plot on the left and the line plot on the right side. Furthermore the *trace by mouse over* feature and the selection of the trace mode moved in the Settings menu. Another difference is that the configuration of the mark settings moved to the left side again, similar to the previous TimeRider version. This mockup also introduces another possible Small Multiple visualization for the scatter plot. The different Small Multiple variants and their advantages and disadvantages are described in the following subsections.

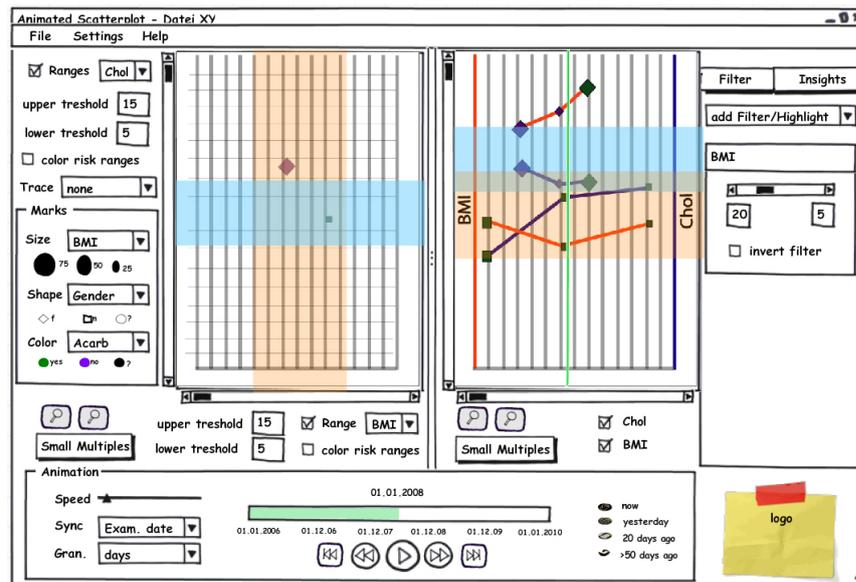


Figure 4.2: Superimposition variant 2: user interface with superimposed variables in the line plot. The scatter plot is placed on the left side and the line plot on the right side.

Small Multiples by Patient

Again the two views can be replaced by Small Multiple visualizations which provide the same functionality as the big view (see Figure 4.3). Both Small Multiple visualizations display the data of the patients separately in different frames. This variant has the advantage that both Small Multiple visualizations display the same information in their small representations – the data of different patients. Consequently the user has not to think about differences in the visualizations. A disadvantage of this variant is that recognizing overall trends in the data is quite hard because of the fact that the data of different patients is split into several frames. On the other hand it is possible to analyze the whole data in the scatter plot view and to use the Small Multiples only to compare some patients' data.

Small Multiples by Time

The second Small Multiple variant of the scatter plot visualizes time points instead of patients, i.e. every picture shows the scatter plot at a specific time point. The advantage of this variant is that the user is able to compare all patients at different time points better than with the first variant and recognizing overall trends is easier. But the number of time points is bigger than the number of patients. In consequence only a few time points can be compared because of the limited screen space. It is more useful to compare some patients than to compare only a few time points. Another disadvantage of such a Small Multiple visualization is that both Small Multiple views do not visualize the same information (patients vs. time points) and the user has to recognize that difference.

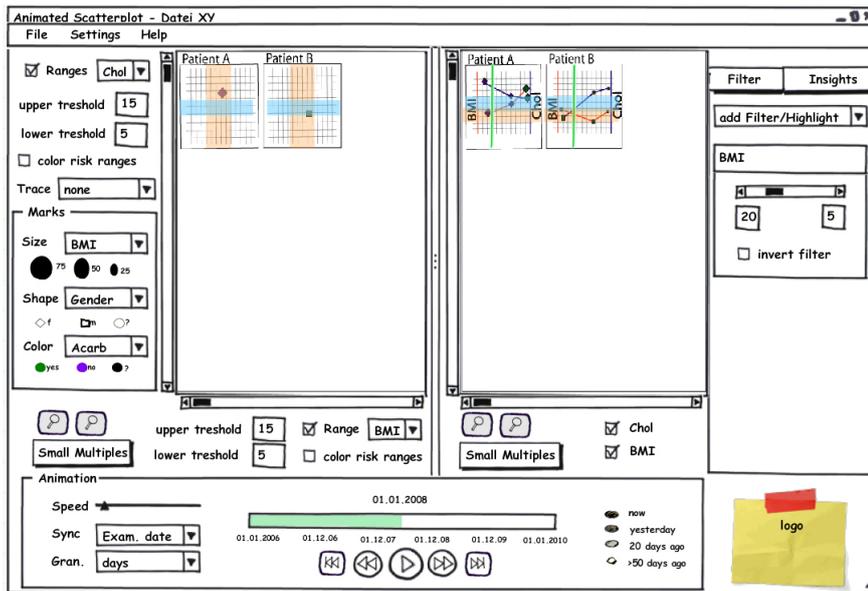


Figure 4.3: Superimposition variant 2 displaying the two Small Multiple visualizations.

Pros and Cons

The positive and negative aspects of superimposition described for the previous mockup also hold for this mockup: the superimposition works well for the Small Multiples but cause problems in the line plot where many colors and lines are drawn across each other and make the plot overcrowded. One disadvantage could be that the line plot has very little space to visualize the time axis but as I mentioned before today usually widescreen displays are used and this disadvantage could be compensated. Compared to the previous presented mockup this version does support more the natural reading direction (from left to right and top down) for selecting the axis parameters.

Juxtaposition

In the third mockup the arrangement of the plot equals the arrangement of the superimposed mockup variant 2. On the left side is the scatter plot and on the right side the line plot. The difference is that the line plot is split into two line plots, separately visualizing the two axis parameters (see Figure 4.4). These two line plots are juxtaposed to make it possible to compare them easily.

Small Multiples

The Small Multiple visualization of the line plot is also juxtaposed (see Figure 4.5). The Small Multiples of the two axis parameters are placed on top of each other. The two different possible

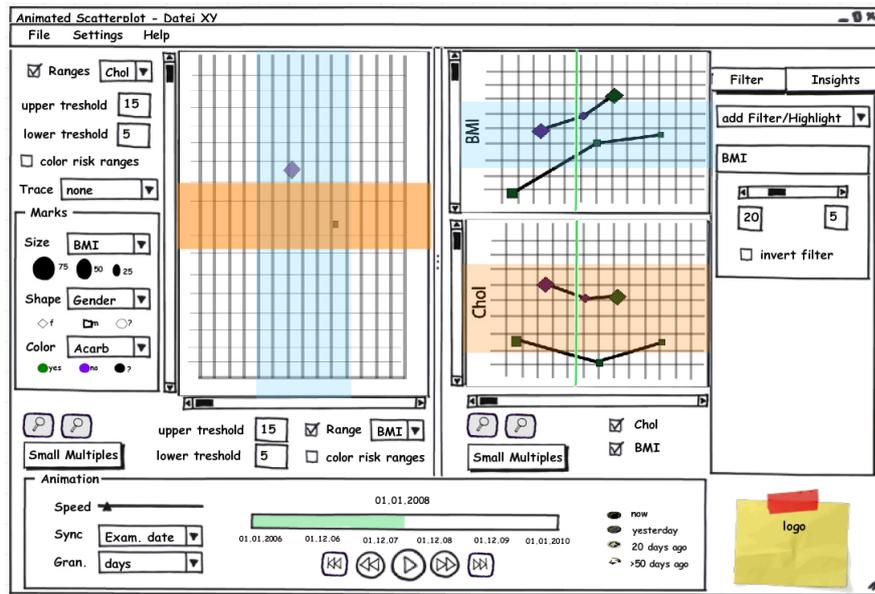


Figure 4.4: The mockup with the juxtaposed line plots. On the left side is the scatter plot and on the right side the two line plots visualizing the two axis parameters.

Small Multiple visualizations for the scatter plot discussed in the previous mockup section were also possible variants for this mockup.

Coordination

The coordination for this mockup was planned similar to the coordination described for the first mockup except one point. In the juxtaposed mockup panning and zooming is coordinated. If the scatter plot is panned or zoomed, the y-axes of the two line plots are also panned or zoomed. Furthermore if the time axis in one of the line plots is zoomed both line plots are affected because they visualize the same time axis and should always display the same range of time.

Pros and Cons

The juxtaposed line plots support the comparison of one parameter between different patients but make it hard to recognize the correlation between the two parameters. The advantage of juxtaposition is that there are not as many lines in the plot as it is in the superimposed line plots. Therefore the plot is more clearly arranged and not overloaded with information. One clear disadvantage of this mockup is that there is less space for the Small Multiples of the line plot. Fewer patients can be visualized and compared without scrolling. Furthermore the two pictures of one patient are relatively far away from each other.

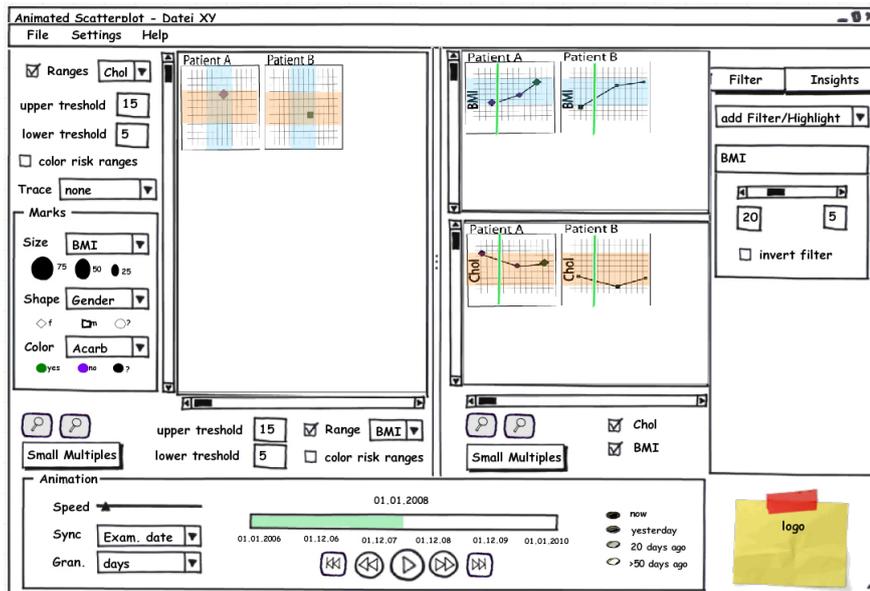


Figure 4.5: The mockup with the juxtaposed line plots displaying the Small Multiple visualizations for both plots.

4.4 Final User Interface Design

The final user interface consists of the animated scatter plot and one Small Multiple visualization. The final Small Multiple visualization works similar to the Small Multiple by time variant described before but displaying time intervals (e.g. months or years) instead of time points (days). The user interface is split into several modules which can be integrated dynamically. Figure 4.6 shows the user interface with all modules and all features. In the final user interface the configuration of the mark settings is placed in the tab panel and the configuration of the parameter for the y-axis is positioned above the two plots.

Small Multiples

Every picture in the Small Multiple visualization contains the data of an adjustable time interval. For example if the time interval is set to years, every Small Multiple represents one year and displays the data which was collected in that year. If one of the trace modes is activated, the record points are connected across the Small Multiples. The trace which started in the previous picture is continued and starts in the current picture where it has ended in the previous one. The traces get thicker within a picture to indicate where it starts and ends but they always start with a thin line in every new Small Multiple.

To evaluate the advantages and disadvantages of the different visualizations it was necessary to make the user interface as flexible as possible. All features and user interface parts are modules. For every module it is decidable if it is visible or not. Figure 4.7 shows the interface when the

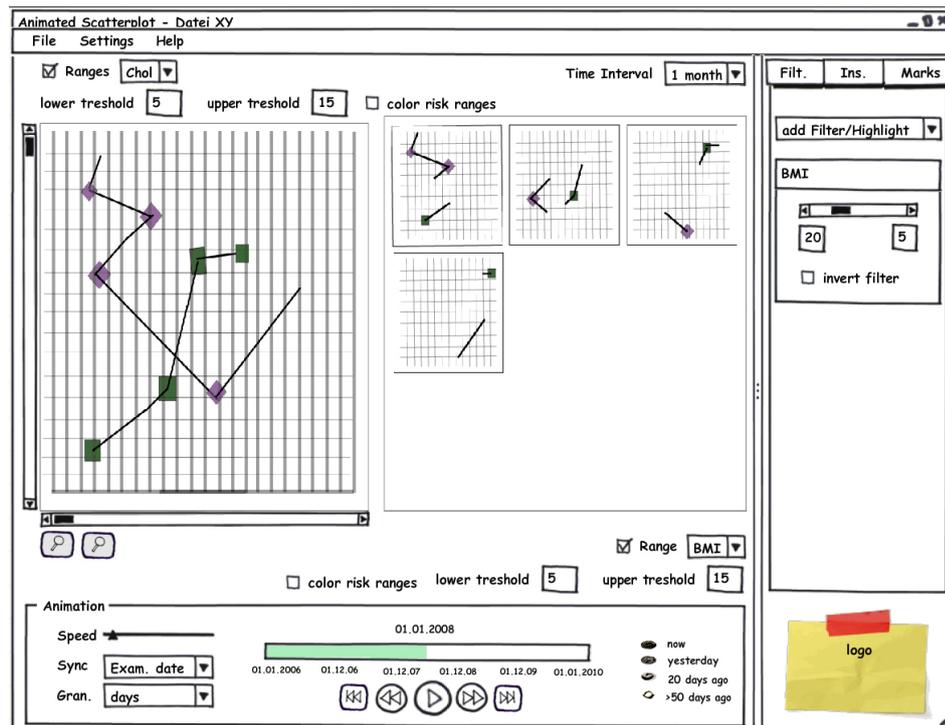


Figure 4.6: The final user interface with the animated scatter plot and a Small Multiple visualization displaying different portions of time.

scatter plot is deactivated and only the Small Multiple visualization with all available functions are visible.

The following user interface parts are customizable modules and can be (de)activated:

- scatter plot
- Small Multiples
- animation configuration panel
- filter configuration
- mark settings
- insights
- x-axis parameter configuration
- y-axis parameter configuration
- risk area configuration for the x-axis

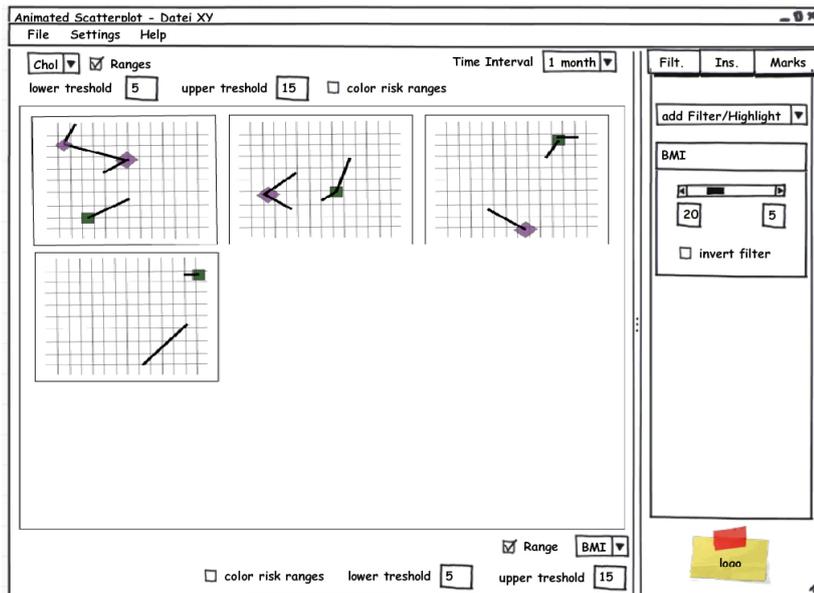


Figure 4.7: The final user interface with the Small Multiple visualization. The scatter plot is deactivated.

- risk area configuration for the y-Axis
- synchronization settings
- trace settings
- *trace by mouse over* function
- the whole menu of the program including File, Settings and Help menu

The settings and values of the modules are adjustable by a configuration file. If one of the modules is switched off, the program uses that predefined values. For example if the mark panel is not available in the user interface, size, color and shape of the points have predefined values, e.g. no mapped parameters.

Coordination

The coordination between the two views covers the following aspects:

- The selected axis parameters are equal in every visualization.
- The adjusted mark properties including the mapping of color, size and shape to parameters, are used in every visualization.
- Filters are applied in every visualization in the same way. Records which are affected by a filter will disappear. A trace where all records were filtered out will also disappear.

- The selection of one or more patients in one view is propagated to the other view.
- The risk ranges are displayed in both views in the same way: by a blue background color.
- The selected trace mode is applied to both visualizations.
- The selected synchronization mode is applied to both visualizations.
- The Small Multiples always have the same axis range and visualize the same part of the image. Therefore the pan and zoom function is also coordinated between all Small Multiples.

The following aspects are not part of the coordination:

- The adjusted granularity only affects the scatter plot and not the Small Multiples.
- Panning and zooming is not coordinated between the plots: the two plots can be zoomed or/and panned separately.

Implementation

This part of the thesis presents the technical details of the TimeRider program. First I will give a short introduction to the functions and the user interface of the final TimeRider version. After that the programming environment is explained and finally the program structure is described in detail.

5.1 Features

The graphical user interface of the final TimeRider version is strongly related to the mockup presented in the previous section. Figure 5.1 shows a screen shot of the final TimeRider program visualizing a dataset containing medical data. The new TimeRider is a highly customizable program, providing many features and interaction possibilities. Every function is embedded into a module which can be separately switched on or off. With a configuration file it is possible to preconfigure all program settings and parameters. It is possible to run the program in the “normal” working-mode or in the evaluation-mode of the program. In the evaluation mode the evaluation process will be passed through presenting the different visualizations after each other with the predefined tasks.

Table 5.1 summarizes the features and interaction possibilities of TimeRider. The table splits the functions into features of the animated scatter plot, features of the Small Multiple visualization, and general features affecting both visualizations or the entire program.

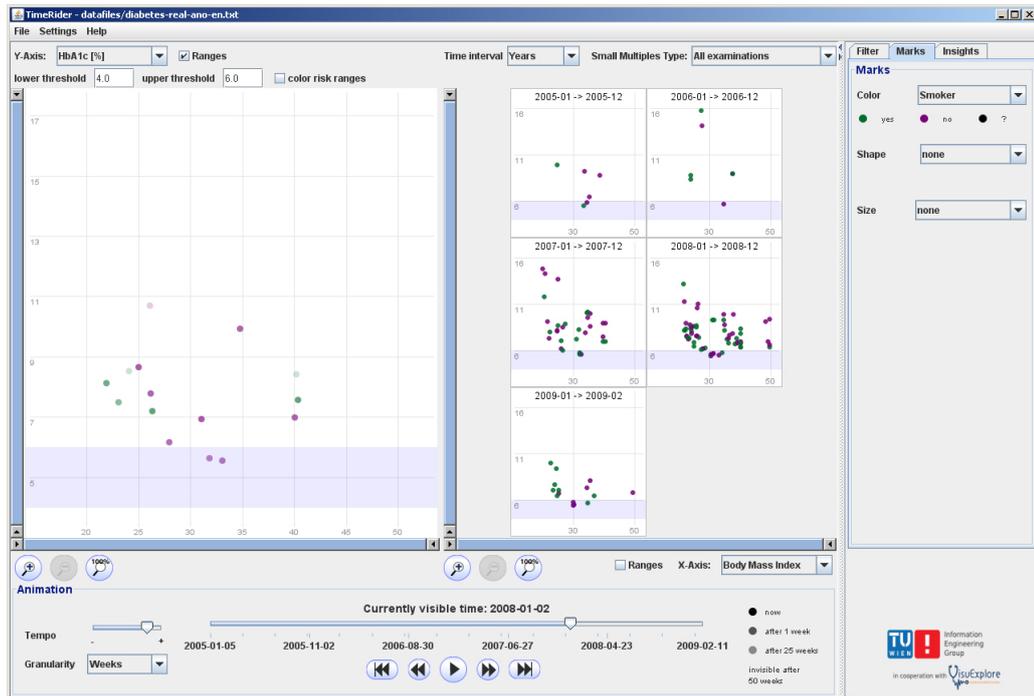


Figure 5.1: This figure shows a screen shot of the final TimeRider version. The TimeRider in the screen shot visualizes the medical data of about 30 patients. The numerical parameters BMI and HbA1c are mapped to the axes. The normal range of HbA1c is visualized in the plots (blue background). The color is mapped to the parameter “Smoker”— green points are examinations of smokers, purple points of non-smokers.

Animated scatter plot	Small Multiple
Media player controls	Defining the aggregation type
Selecting different granularities	Selecting time intervals
Providing a dynamic trace visualization	Providing a static trace visualization
General features	
Reconfiguration of axis parameters	
Panning and zooming	
Mapping parameters to color, size and shape of the points	
Adding filters: name, numerical and logical filters	
Visualizing and adapting normal ranges of the parameters	
Saving insights	
Selection and highlighting of patients	
Loading and storing program settings	
Synchronize data in four different temporal ways	
Exporting plot as a picture	
Configuring user interface modules and properties	

5.2 Programming Environment and Software Dependencies

TimeRider was developed with the Java 1.6 programming language and the Java visualization toolkit *prefuse*.¹ Prefuse is a free and open source software providing a framework for building highly interactive data visualizations. The framework is very flexible and adaptable because of the polyolithic design. Furthermore it uses the Reference Model [9], separating data, visualization and views. This approach makes it possible to use a data source for multiple visualizations and to get different views on the same visualization. The rest of the graphical user interface was created with the Java Swing components.

The Apache *log4j* library was used to display and log the steps of the program execution during the development and was also part of the evaluation framework EvalBench [14] to log the user activities.²

The development environment consisted of *Eclipse*³ and *Subversion*⁴ combined with *TortoiseSVN*⁵ for the version control.

5.3 Program Structure

The program consists of six main packages, five of them refer to TimeRider and one to the extension of the evaluation framework. The *data* package includes the classes for administrating the loaded data and the data needed for the program execution (e.g. constants) and the *gui* package comprises all user interface dependent classes. The *controller* package includes the core piece of the program, the classes with the main functionality of the program. The division of the classes into these packages refers to the MVC (Model View Controller) pattern: the data package represents the Model, the gui package the View and the controller package the Controller. Apart from that there is a package *jaxbsettings* combining all classes which are responsible for loading and storing program settings and configurations. The package *animation* contains the class which is responsible for the animation process. Finally the *evaluation* package includes all classes extending EvalBench.

In the following subsections all class and method names are written with monospaced font and package names as well as keywords with italic fonts.

Data

There are four classes responsible for loading and administrating the data which should be visualized in the TimeRider. The class `UTF8TextTableReader` extends the `AbstractTextTableReader` and provides a UTF-8 based import function for tab-separated data files. The `TableController` stores the table created with the `UTF8TextTableReader`

¹<http://www.prefuse.org/>, last accessed on February the 11th, 2012

²log4j library 1.2.15 <http://logging.apache.org/log4j/1.2/index.html>, last accessed on March the 4th, 2012

³Eclipse Galileo <http://www.eclipse.org/>, last accessed on February the 11th, 2012

⁴<http://subversion.apache.org/>, last accessed on March the 4th, 2012

⁵TortoiseSVN 1.6.16 <http://tortoisesvn.net/>, last accessed on March the 4th, 2012

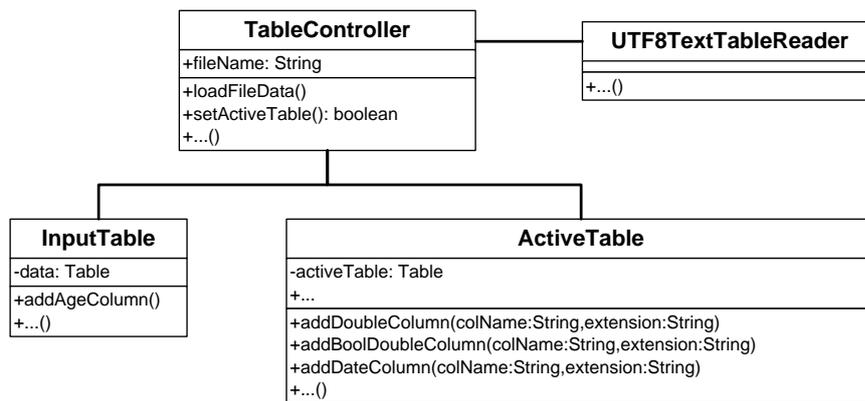


Figure 5.2: The data is loaded in the `TableController` class and stored in an `InputTable` object. The method `addAgeColumn` is used to add a column which contains the age of the patient and is always called when an `InputTable` object is created. With the `setActiveTable` method in `TableController`, the table is prepared for the visualizations by filtering out *null* values and adding new columns of type double (i.e. `addDoubleColumn`, `addBoolDoubleColumn` and `addDateColumn`).

as an `InputTable` object. During the creation of an `InputTable` object the pre-processing of the input data is carried out. Especially important by using medical data is that an age column is added to the input table. The `InputTable` object is the basis for the `ActiveTable` object. The `ActiveTable` provides the data for the prefuse visualizations, e.g. the scatter plot. The loaded data often contains *null* values because it is possible that not for every date and every parameter a data value was stored. It is necessary to filter out such *null* values and to create new columns of type double for every axis parameter. Therefore `ActiveTable` provides different methods to create new columns for numerical and logical parameters as well as a method to create an appropriate date column. Figure 5.2 demonstrates the important methods of the four classes and how they relate with each other.

The `data` package also includes classes which administrate program dependent data. In the `Constants` class important constant variables are stored, e.g. constants for font styles, date formats or names for the prefuse action lists. Another important `data` class is `GuiAttributes`. This class contains all information about the language dependent aspects of the program, like the text for the graphical user interface. That language dependent data is loaded from an `.xml-propertyfile`. Listing 5.1 shows an excerpt of an English property file.

```

<!--Graphical User Interface parameters-->
<!--Main Frame-->
<entry key="mainframe.title">TimeRider</entry>

<!--Axes Configuration-->
<entry key="axis.x.label">X-Axis: </entry>
  
```

```

<entry key="axis.x.tooltip">select the parameter for the x-axis
  </entry>
<entry key="axis.y.label">Y-Axis: </entry>
<entry key="axis.y.tooltip">select the parameter for the y-axis
  </entry>

<!--Menus-->
<entry key="menu.file">File</entry>
<entry key="menu.file.open">Open</entry>
<entry key="menu.file.export">Export</entry>
<entry key="menu.file.export.scatterplot">Export Scatterplot</
  entry>
<entry key="menu.file.export.smallmultiples">Export Small
  Multiples</entry>
<entry key="menu.file.exit">Exit</entry>

```

Listing 5.1: Excerpt of the english xml-property file.

In addition to the graphical user interface texts the language file also contains the information about the numerical and logical parameters, i.e. the information about the column names in the data file. There are always two entries for every numerical and logical parameter. The first entry is the name for the parameter which should be used in the graphical user interface and the second entry is the name for the parameter which is used for the title row in the data file. To differentiate between parameter names which are used in the graphical user interface and parameter names which are used for the data file the entries are named in the following way:

```

<entry key=bool.protein.2.gui>Protein in urine</entry>
<entry key=bool.protein.data>Proteininurine</entry>

```

Gui-entries end with *.gui* and data-entries with *.data*. The split of gui-entries and data-entries makes the implementation more flexible. For example the names of the columns in the data file are fully independent of what is displayed in the user interface. The number in the key of the gui-name indicates the index of the parameter in the ComboBox, i.e. parameters are topically ordered in the GUI by their number.

For the evaluation it was necessary to extend the language file by additional texts, e.g. by the question-texts of the questionnaire. The evaluation was carried out in German, therefore only a German language file for the evaluation was created.

Graphical User Interface

The graphical user interface is composed of several panels integrated in the `MainFrame`. The graphics in Figure 5.3 illustrate the placement of the different panels in the `MainFrame`. The tab panel on the right side contains the `FilterPanel`, the `MarkPanel` and the `InsightsPanel`. The `FilterPanel` includes all graphical components necessary for the filter function and the

MarkPanel contains the components to adjust the mapping of color, size and shape to parameters. The input of insights, i.e. storing insights, making screenshots and displaying the stored insights is handled by the InsightsPanel.

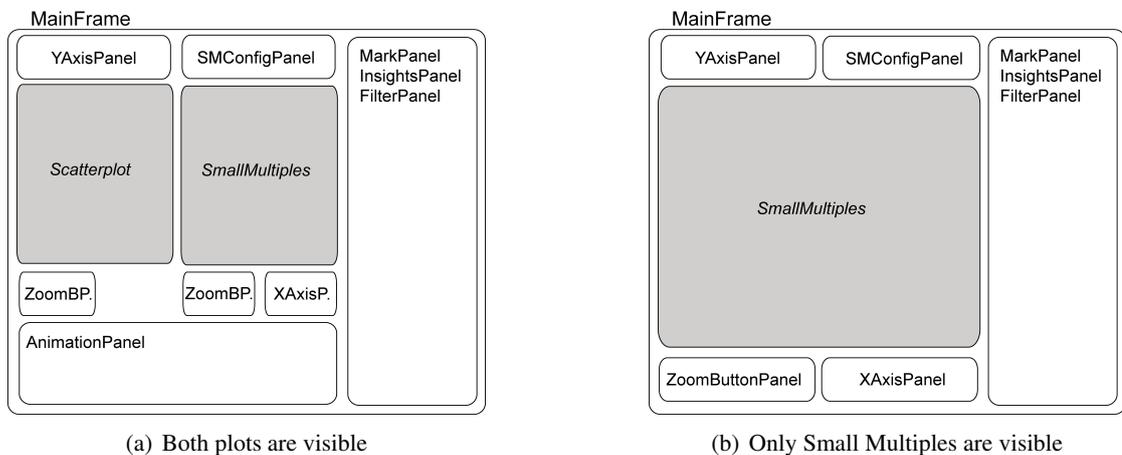


Figure 5.3: There are several panels with different functionality integrated in the MainFrame. This figure demonstrates how they are arranged within the frame. (a) If both plots are visible the scatter plot is always placed on the left and the Small Multiples on the right side (the name of the two ZoomButtonPanels and the XAxisPanel are cut because of the limited space in the graphic). In (b) only the Small Multiple plot is visible and therefore the AnimationPanel is not available.

The AnimationPanel includes the components which are responsible for controlling the animation process of the scatter plot. The configuration of the Small Multiple visualization is placed in the SMConfigPanel. There are two ZoomButtonPanels integrated into the MainFrame if both plots are visible (see Figure 5.3(a)). Figure 5.3(b) shows the MainFrame when only Small Multiples are visible.

The configuration of the axis parameters and the risk and normal ranges of the parameters is contained in the XAxisPanel and the YAxisPanel. When the user changes one of the axis parameters the method `setPlotXAxisData` or `setPlotYAxisData` is called (see Figure 5.4). These methods call the method `setPlotData` in the MainFrame and cause the creation of a new `ActiveTable` object. The new `ActiveTable` is the basis for the creation of new plots presenting the data of the selected axis parameters. The new adjusted axis parameters and the current risk or normal ranges are stored in the `Settings` class (see section 5.3). The changes in the normal and risk ranges are fetched of the `PlotPaint` listener by retrieving the stored values in the `Settings`. Furthermore `PlotPaint` paints the blue background to visualize the ranges in the plots.

The MarkPanel is connected to the `AnimationController` and the

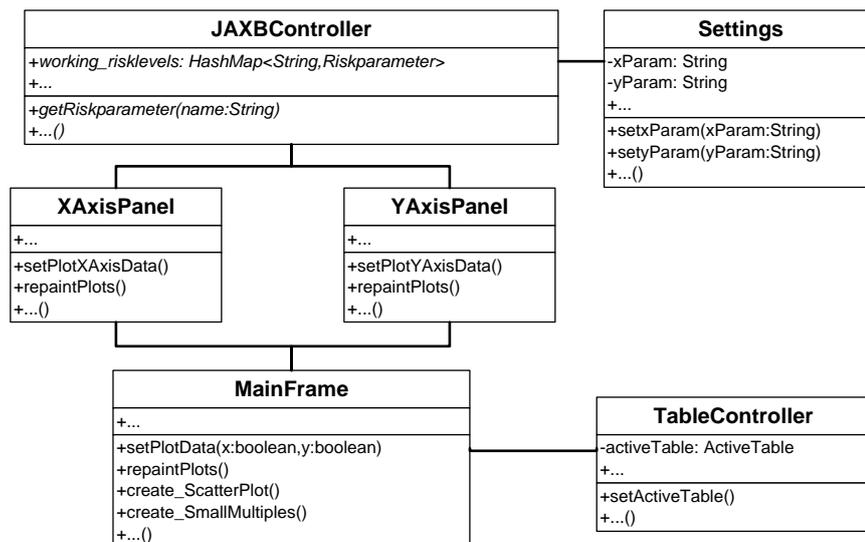


Figure 5.4: The selection of a new axis parameter causes the creation of a new `ActiveTable` object and therefore the generation of new plots. The new axis parameters are stored in the `Settings`.

`SmallMultipleController` to lay out the plots according to the adjusted mark settings. Furthermore the `MarkPanel` gets information of the `TableController` about the maximal and minimal values of parameters to calculate the appropriate values for the size of the points (if a parameter is mapped to the size). Figure 5.5 demonstrates the important methods and class interactions for adjusting the mark properties. Figure 5.5 also shows that the adjusted mark properties are stored in the `Settings` (see section 5.3).

The plots are placed in the middle of the `MainFrame`. The sizes of the plots are calculated using the same predefined width-to-height-ratio, every time the window or the panel on the left side is resized. In that way it is guaranteed that both plots always have the same proportions.

The `MainFrame` also contains a menu bar where several functions are placed: exporting the plots, loading and storing program settings, adjusting trace and synchronization settings and switching on or off the trace-by-mouseover feature.

Filter

Filters are administrated in the `FilterListPanel` by adding or removing different kind of panels. All of the three filter panels, `NameFilterPanel`, `NumFilterPanel` and `BoolFilterPanel`, are subtypes of the abstract class `FilterPanel`. Therefore every specific filter panel provides functions to manipulate them. For example to set new filter properties when the user loads program settings containing filters. The `FilterListPanel` is the interface between the `FilterController` and the added filter panels. Every time a change occurs in one of the filter panels a new `Predicate` is created and transferred to the

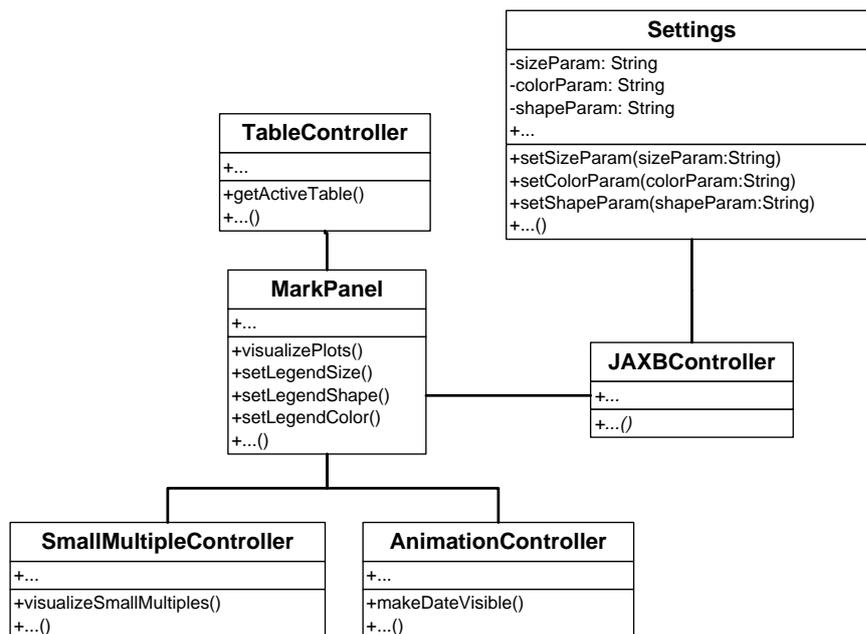


Figure 5.5: The `MarkPanel` calls the `visualizeSmallMultiples` and the `makeDateVisible` method of the two plot control classes when the user changes the mark settings. The adjusted mark properties are stored in the `Settings`. The `MarkPanel` also provides functions to prepare the legend for every mark parameters, e.g. `setLegendSize()`.

`FilterController`. Every type of filter uses another subtype of `Predicate`. For example an inverse numerical filter creates an `OrPredicate` combining two `Range-Predicates`.⁶ The range predicates represent the upper and the lower value range of the filter and examination points stay visible if their parameters either lie in the lower or in the upper range:

```

RangePredicate lowfilter = new RangePredicate(ce, minBound1,
    maxBound1);
RangePredicate upfilter = new RangePredicate(ce, minBound2,
    maxBound2);
filter = new OrPredicate(lowfilter, upfilter);
  
```

The `FilterListPanel` is also responsible to lay out the plots after adding filters to the `FilterController` by calling the `AnimationController` and the `SmallMultipleController`. Figure 5.6 shows how the classes work together and demonstrates the important methods.

⁶An inverted numerical filter is used to visualize patients in a risky parameter range whereas a normal numerical filter visualizes the normal range of the parameter.

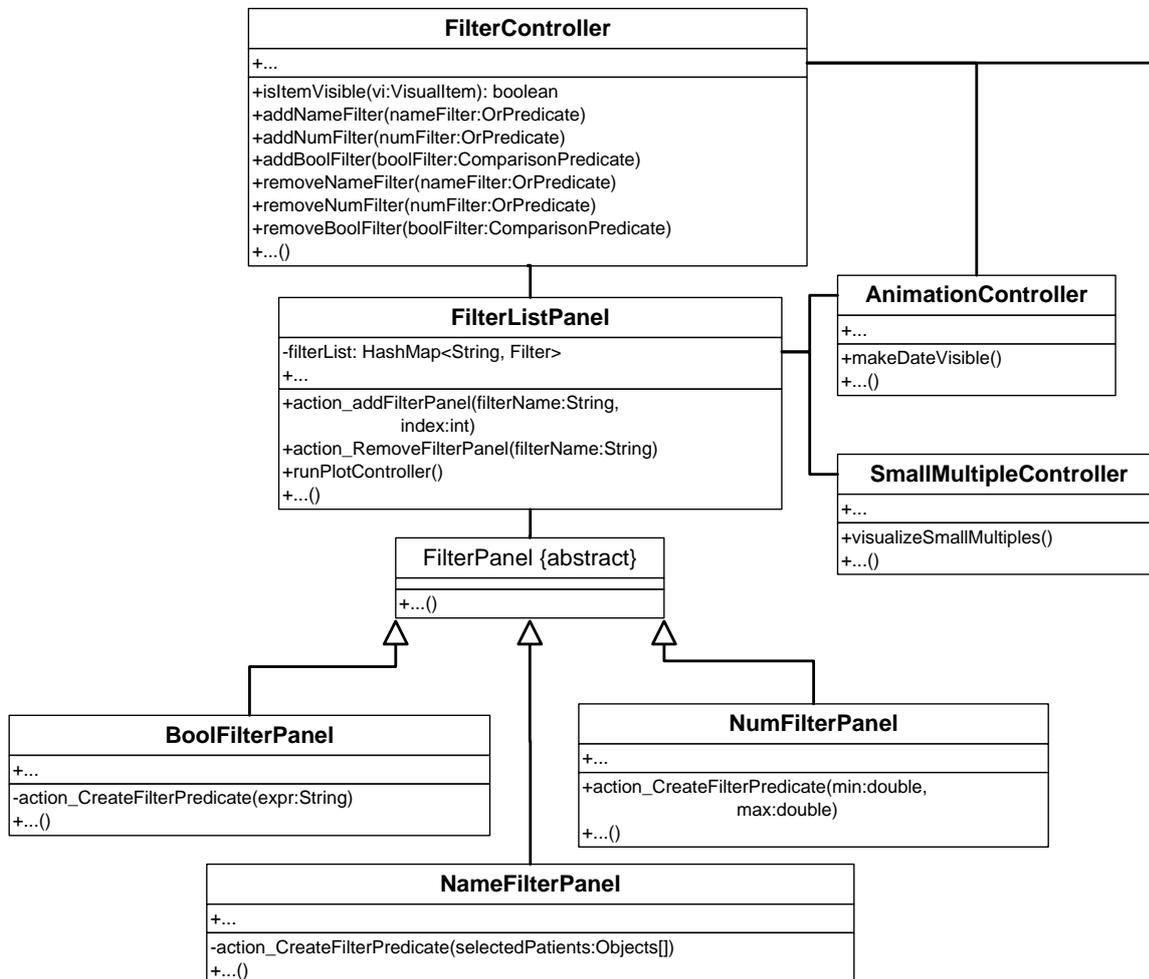


Figure 5.6: All of the three filter panels extend `FilterPanel`. A change in the filter configurations is transferred to the `FilterController` by adding new predicates. The `FilterListPanel` induce the refresh of the plots by calling the methods in the `AnimationController` and the `SmallMultipleController`. The plot controller classes then determine the visibility of the items according to the adjusted filters, i.e. they call the `isItemVisible` method for every item. Clearly the adjusted filters are also stored in the `Settings` but this aspect was left out in this Figure to make the picture more well-arranged.

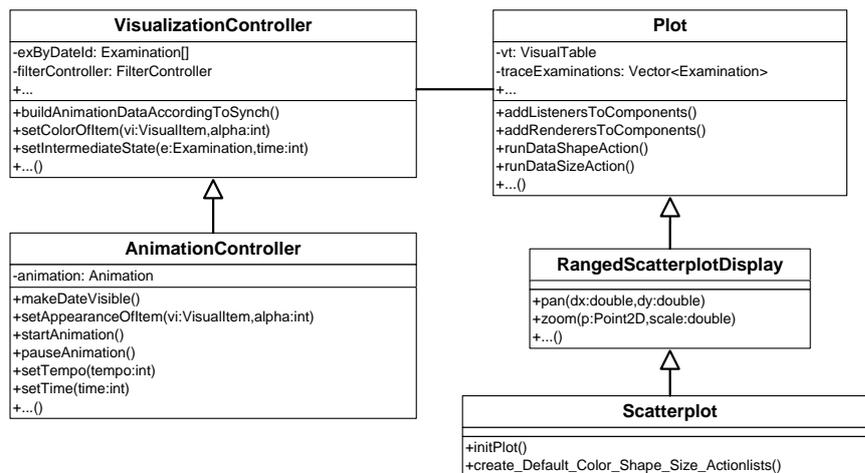


Figure 5.7: Plot provides the main functionality of a plot. The RangedScatterplotDisplay extends the Plot by methods supporting pan and zoom. Finally the Scatterplot is a subtype of the RangedScatterplotDisplay. The VisualizationController and its subtype the AnimationController are responsible for the visibility and appearance of the items in the plot.

Animated Scatter Plot

The scatter plot is implemented by three classes: Plot, RangedScatterplotDisplay and Scatterplot (see Figure 5.7). Plot extends Display of the *prefuse* library and provides some important functionality for a plot, e.g. adding and administrating renderers and listeners. This class also provides methods to run the DataSizeAction and the DataShapeAction to draw the points according to the adjusted mark settings. Furthermore Plot is responsible to arrange the axes and the data rectangles (i.e. the area of the data value ranges within the plot) in the plot. RangedScatterplotDisplay is an extension of Plot and provides methods to pan and zoom the plot. This class is strongly connected to the MainFrame object because the range sliders, which are administrated in the MainFrame, and the zoom function provided by the RangedScatterplotDisplay are coordinated. Finally the class Scatterplot is an extension of the RangedScatterplotDisplay and contains a method to initialize the plot. Within that initialization method (`initPlot`) the VisualTable, which contains the data for the *prefuse* visualization, is filled with the current ActiveTable and the axis data is defined.

The animation process is controlled by the user with the user interface components provided by the AnimationPanel. The link between the AnimationPanel and the animated scatter plot is the AnimationController.

The AnimationController is an extension of the VisualizationController and is the most important class for the animation and the visualization of the items. The VisualizationController is responsible to build and administrate the animation data and provides important methods to define the appearances and the positions of the items. The

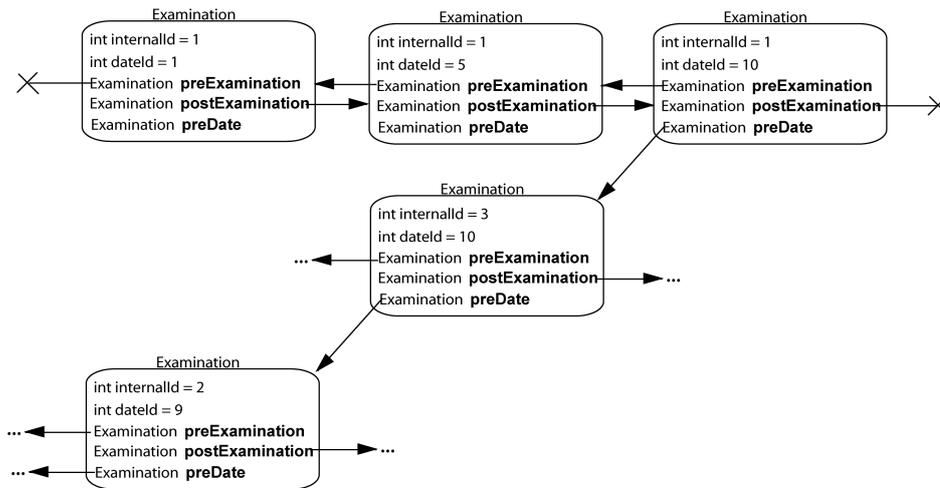


Figure 5.8: The top line shows the linkage between examinations of different dates (*date-Ids* differ) of the same patient (*internalId=1*). The second and third line demonstrate the linkage between examinations according to their date id. Examinations with the same *date-id* are connected to each other - then follows the examination of the previous date.

`AnimationController` controls the `Animation` object and provides a scatter plot specific method (i.e. `makeDateVisible`) to visualize the appropriate items. The `Animation` object implements an `ActionListener` and provides a `Timer` object which runs through the “dates”. Within the `actionPerformed`-method the time counter is increased and the scatter plot gets refreshed by calling the `makeDateVisible` method in the `AnimationController`. The creation of the animation data and the functionality of the `makeDateVisible` method are described in the following subsections in more detail. The complex algorithm to create the animation data and the animation process to handle that data was necessary because of the requirement to be able to deal with the irregularly sampled data.

Animation Data

There are four different methods to create the animation data, one for every synchronization mode, but they all follow the same principle idea. The process is aimed to fill the animation array allocating one entry for every time point filled with `Examination` objects. `Examination` objects represent an examination of a patient at a specific date. Every `Examination` object has a previous and a successive `Examination`, of course except the first and the last one, all referring to the same patient. The date of an examination is specified by a *date-id* which is calculated during the creation process and indicates the entry in the animation array and therefore when the examination should be visualized during animation. The examinations are connected by references to the examinations of the previous dates. Figure 5.8 demonstrates how the `Examination` objects are linked together.

The entries of the animation array always refer the “last” examination of a specific date. For

example a part of the input data looks as follows after ordering the data by their date:

Date	internalId	Patient
2012-01-01	1	A
2012-01-01	2	B
2012-01-03	1	A
2012-01-03	3	C

Then the resulting array contains three entries, one for every date. Dates without examinations (the 2nd January in the example) are filled up with the last occurred examination. In the following table is the result of the animation array presented. In every date-entry always the last examination of the date is stored:

array index	Date	internalId	Patient
0	2012-01-01	2	B
1	2012-01-02	2	B
2	2012-01-03	3	C

The described procedure works well for data synchronized by the examination date but has to be adapted for the other synchronizations. When the data should be synchronized by the first examination date a second, derived date column has to be created and filled. The column is filled by new dates which are calculated by subtracting the difference between the first date of all examinations and the first date of the current considered patient, from examination's date:

derived date of examination = [date of the examination] - ([first date of the patient] - [first date of all examinations])

The following example illustrates the process: the first examination of patient A on the 1st January is the first examination of both patients. Therefore all patient's examinations should start on 1st January. The dates of patient A stays the same whereas the examination dates of patient B have to be calculated with the previous described formula.

Patient A	Patient B	Calculation of new date	Derived dates
2012-01-01	2012-03-01	[2012-03-01] - ([2012-03-01] - [2012-01-01])	2012-01-01
2012-04-01	2012-10-01	[2012-10-01] - ([2012-03-01] - [2012-01-01])	2012-08-01

After preparing this new column, the animation array is created analogically to the procedure described before. The synchronization by last examination date works very similar. The difference is that the calculation of the new dates considers the differences between the last date (of all examinations) and the last dates of the individual patients.

For the synchronization by age, the age column is used instead of the date column. Again, a new column is created and filled by the values of the age column multiplied by 365. The ages are stored as days because the result in the plot gets more precise. The new age column is then used to build the animation array similarly to the procedure described before.

Animation Process

During the animation process the examination of the current *date-id* is fetched from the animation array. Then the loop runs through all linked examinations by always fetching the examination with the previous *date-id* until the first examination has been reached. If we consider the previous example again and apply the animation process we fetch the examination of patient C on 2012-01-03. This examination is linked by the field *preDate* with the examination of patient A which is also linked to the examination of patient B of date 2012-01-02 and so on. Apart from that the algorithm in the `AnimationController` decides for every examination point whether it is visible, where the point is placed in the plot and how the point should look like, i.e. transparency and color. The method `setIntermediateState` of the `VisualizationController` is used to calculate the transparency and the positions of so called *moving points*. *Moving points* visualize the development of the examination parameters during animation. A patient's *moving point* is always the point of the next examination and moves from the position of the previous examination point to the coordinates of its own parameter values. The coordinates of the *moving points* are calculated by linearly interpolating between the successors x- and y-parameter values and their own values. The transparencies of moving points increase during animation. If it is needed the whole transparency functionality can be switched off completely by adjusting the program configuration file.

This approach has the advantage that the access to the animation array entries and the referenced `Examination` objects works very fast. This is necessary to provide a smooth animation. The disadvantage is that it is memory consuming and every time the plots are created it takes some time to prepare the animation array, i.e. when the axis parameters have changed or a new data file has been loaded.

Small Multiples

The plots of the Small Multiple visualization are also composed of three classes: `Plot`, `RangedSmallMultipleDisplay` and `SmallMultiple`. The `Plot` was already described in the `Animated Scatter plot` section. The `RangedSmallMultipleDisplay` extends the `Plot` and contains methods to pan and zoom the Small Multiples. The difference to the `RangedScatterplotDisplay` is that panning and zooming in one of the Small Multiples is transferred to all the others. Therefore the `RangedSmallMultipleDisplay` interacts with the `SmallMultipleController` to pan and zoom all pictures of the Small Multiple visualization simultaneously. The class `SmallMultiple` extends the `RangedSmallMultipleDisplay`, works similar to the `Scatterplot` and therefore also provides a method to initialize the plot.

The Small Multiple visualization is configurable by the control elements in the `SMConfigPanel`. There are two settings that specify how the visualization should look like. Firstly the time interval or the number of Small Multiple pictures. In the program configuration file it is possible to specify which of the two possibilities is used, but the time interval is more user friendly and coherent and therefore the standard configuration. The time interval refers to the normal calendar structure, e.g. years, quarters, months and so on.

The second possible configuration refers to the aggregation type. The aggregation type determines which data points should be visible if there are more than one record of the same patient in one Small Multiple contained. For example if the time interval is set to years and patient A had three different examinations in year 2009, it is possible to define which of these three examinations should be visible in the Small Multiple of year 2009. Altogether there are seven aggregation types available:

- all examination points are visible,
- only the newest examination of each patient is visible,
- only the oldest examination of each patient is visible,
- the examinations with the lowest x-parameter values of the patients are visible,
- the examinations with the highest x-parameter values of the patients are visible,
- the examinations with the lowest y-parameter values of the patients are visible,
- the examinations with the highest y-parameter values of the patients are visible.

The plots of the Small Multiple visualization are generated and administrated in the `SmallMultipleController`. Changes in the settings, like adding a filter, cause an update of the Small Multiples. This is done by a call of the `visualizeSmallMultiples` method in the `SmallMultipleController`. This method furthermore calls the `makeDateVisible` method in the `SmallMultipleVisController`. The `SmallMultipleVisController` is the equivalent to the `AnimationController` and therefore also extends the `VisualizationController`. The Small Multiple visualization also uses the examination array constructed in the `VisualizationController` which was described in section 5.3. The difference to the scatter plot is that the array is passed through only once for every Small Multiple with the associated end date of the Small Multiple. Additionally it is necessary to define the start date of a Small Multiple to visualize only the data of the associated time interval. Figure 5.9 shows the classes needed for the Small Multiple visualization.

Traces

Traces connect the examinations of a patient with each other. All visible `Trace` objects are administrated in the `Plot`. Every trace has a start point, an end point, a color and a thickness. The thickness depends on how many previous examinations the trace's examination has. Traces start with a thin line and end with a thicker line. The start point is the trace's examination and the end point is defined by the x- and y- coordinates of the post examination. In the animated scatter plot the traces always end on the positions of the current *moving points* but with invisible *moving points*: animated traces expand till they reach the next examination point and then the examination point gets visible. In the Small Multiple visualization it looks like the traces are connected across all pictures because the traces start in the next picture where they have ended in the previous one. The ends of the traces in the Small Multiples

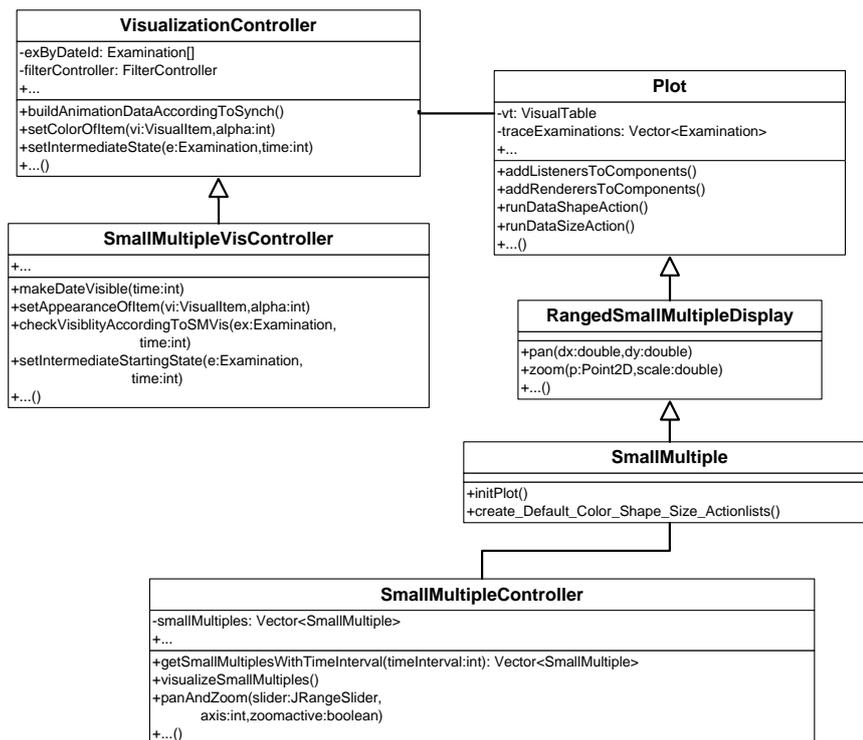


Figure 5.9: SmallMultiples are administrated by the SmallMultipleController. SmallMultiples are subtypes of RangedSmallMultipleDisplays and furthermore subtypes of Plot. The SmallMultipleVisController provides the important method `makeDateVisible` to decide about the item's visibility and appearance.

are positioned by the same mechanism used for the *moving points*. For calculating the position of the starting points another method (i.e. `setIntermediateStartingState`) was implemented in the `SmallMultipleVisController`. This method works similar to the `setIntermediateState` method but without considering the transparency. The positions are also calculated by linearly interpolating between the positions of two successive examination points.

JAXBSettings

The `JAXBSettings` package covers all classes responsible for administrating, loading and storing program settings and configurations. The Java framework JAXB (Java Architecture for XML Binding) is used to map Java classes to XML representations [1]. The procedure where objects are transferred into XML equivalences is called *marshalling* whereas *unmarshalling* means the inverse process. The class `JAXBUtils` provides two methods which are responsible for the marshalling- and unmarshalling-process.

Two independent topics are part of the `JAXBSettings`. First of all the program configuration

containing all information, important for building the graphical user interface. Secondly the current program settings, e.g. the adjusted axis parameters. There are two classes (`Config` and `Settings`) which are responsible for these two topics. The variables in these classes are provided with XML-annotations, to indicate that they should be transferred into XML representations during the marshalling process.

Config

contains all information about the program configuration. For example there are parameters which define the used colors in the program and parameters to define the tempo of the animation. Furthermore with the program configuration parameters it is possible to customize the user interface. There are several modules like the scatter plot and the Small Multiple visualization, which can be deactivated and set invisible by adjusting the configuration parameters. Figure 5.2 presents an excerpt of the configuration file.

```
<color-highlight blue="0" green="88" red="228" />
<color-risklevel blue="255" green="0" red="0" />
<animation-start-tempo>3</animation-start-tempo>
<animation-tempo-multiplier>25</animation-tempo-multiplier>

<risklevels-changeable>true</risklevels-changeable>
<xaxis-changeable>true</xaxis-changeable>
<yaxis-changeable>true</yaxis-changeable>
<smallmultiples-active>true</smallmultiples-active>
<scatterplot-active>true</scatterplot-active>
```

Listing 5.2: Excerpt of the program configuration file. The top four lines are general configuration parameters, e.g. the first line defines the color which should be used when the user moves the mouse across an item. The bottom lines are part of the module configuration. All five modules (Small Multiple, scatter plot, risk level configuration, x-axis configuration and y-axis configuration) are switched on and therefore will be visible in the program.

Settings

comprises the current adjusted program settings, e.g. adjusted parameter mappings to color, size and shape or the adjusted filters. These settings are user dependent and can be stored and loaded within the program.

The risk and normal ranges of the parameters are used in both, settings and configuration. There are values loaded with the configuration file but they can be changed within the program. If the user stores the program settings the adjusted risk ranges are also stored and therefore also included in the `Settings` class. `Riskparameters`⁷ are stored in a `HashMap` indexed with the parameter name and containing `Riskparameter` objects. The class `Riskparameter` comprises the necessary information about a parameter with risky ranges, i.e. the lower and the

⁷I use “Riskparameters” for medical parameters where a normal range was defined in the configuration file.

upper bound of the normal range. To make it possible to serialize the riskparameter-HashMap it was necessary to implement an appropriate XMLAdapter in the JAXBUtills class.

The three kinds of filters are represented by three classes including the necessary JAXB annotations. For example the class NumFilter administrates four variables: the parameter name, the lower and the upper bound of the numerical filter and finally a boolean variable to store if the filter should be inverted. The following example shows the parameter name variable with the appropriate annotation. The parameter “required” indicates that the element is necessary to create the object from the xml-representation.

```
@XmlElement (name=parametername , required=true )
private String parametername;
```

EvalBench

EvalBench is a class library written in Java. It was build by Hoffmann [14] during a master thesis to support experiments and evaluations in the HCI field.

EvalBench provides different kind of tasks, e.g. there exists a ChoiceSelectionTask for single-choice and multiple-choice questions. For the evaluation of TimeRider it was necessary to extend EvalBench by a new kind of task: the ItemSelectionTask. The ItemSelectionTask makes it possible to use the selected items in the plot as answers to the question. The class PlotControlAdapter is an extension of the prefuse’s ControlAdapter and is responsible to fetch and handle the mouse events occurring on the items. This class is also able to communicate the “selection events” to the ItemSelectionTaskPanelStrategy which transfers the answers to the ItemSelectionTask.

The DefaultEvaluationPanelFactory is responsible for fetching the appropriate TaskPanelStrategy to the current task. The EvaluationPanelFactory extends the DefaultEvaluationPanelFactory and sets the current ItemSelectionTaskPanelStrategy object in the PlotControlAdapter and therefore enables the communication between them.

Apart from the new implemented task it was necessary to provide an evaluation frame and appropriate dialogs. The package *evaluation.ui* contains all user interface dependent classes for the evaluation. The EvaluationWindow is the main frame responsible for controlling the evaluation process. This class is an extension of the EvaluationDelegate of the EvalBench library and therefore overrides the methods for preparing the evaluation sessions and tasks as well as the methods which are called when the session or a task was finished. At the beginning of the evaluation process the EvaluationWindow opens the JDialog SessionGroupDialog which is used to display information about the evaluation, to get the student id of the participants and to prepare the evaluation configuration, e.g. to determine the order of the visualizations. Finally the QuestionnaireFrame was implemented to ask participants about demographic data and their subjective preferences.

The second evaluation package is *evaluation.configuration*, containing the visualization dependent constants. The `VisualizationType` of type *enum* contains constants to define the three different visualizations: *SC*, *SM* and *SMT*. The `ConfigSet` provides the constants which assigns the configuration files to the different visualizations. The assignment of data sets to task sets or sessions is done in the `DataSet` enum. Every task can have a different setting files, e.g. to adjust filters or mark settings individually for every task. In the `TaskDataSet` it is possible to define the mapping of specific setting files to tasks.

5.4 Summary

TimeRider is a highly customizable visualization system built with Java and the visualization toolkit `prefuse`. The program provides two different visualizations integrated into one user interface and coordinated. The feature to show traces for the visualizations creates a third visualization technique: the Trace view. TimeRider is highly customizable in its functionality and appearance (i.e. language, program settings and modules). The large quantity of interaction possibilities gives users the possibility to examine the data in versatile ways. Furthermore the evaluation framework `EvalBench` is fully integrated into TimeRider to support the evaluation process.

The functions of the system were tested during and after the development phase. Parts of the usability and functionality were also already evaluated by physicians during a study [21]. The study carried out during this thesis did not test the usability of TimeRider but was focused on the comparison of the visualization techniques. The following chapter will describe this evaluation and the results in detail.

Evaluation

This chapter describes the evaluation carried out during this thesis. I will first give a short introduction and then I will describe my hypotheses and the evaluation method. Finally I will present the results of the study.

6.1 Introduction

The focus of this quantitative study was on the comparison between the following three visualization techniques:

- the animated scatter plot
- the Small Multiples
- and the Trace view.

The goal of the evaluation was to advance research in information visualization generally and to investigate the pros and cons of the chosen visualization techniques. Especially animation is a controversial topic in the visualization domain and this study was aimed to get new insights about its characteristics, advantages and disadvantages in comparison to the chosen static visualizations.

Figure 6.1, Figure 6.2 and Figure 6.3 present the three visualizations within the program during evaluation process.

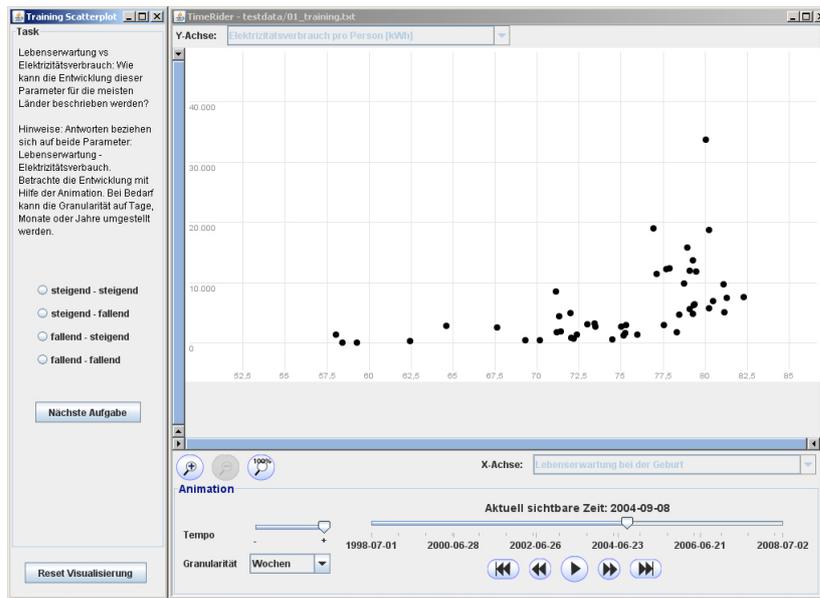


Figure 6.1: The evaluation window with the animated scatter plot.

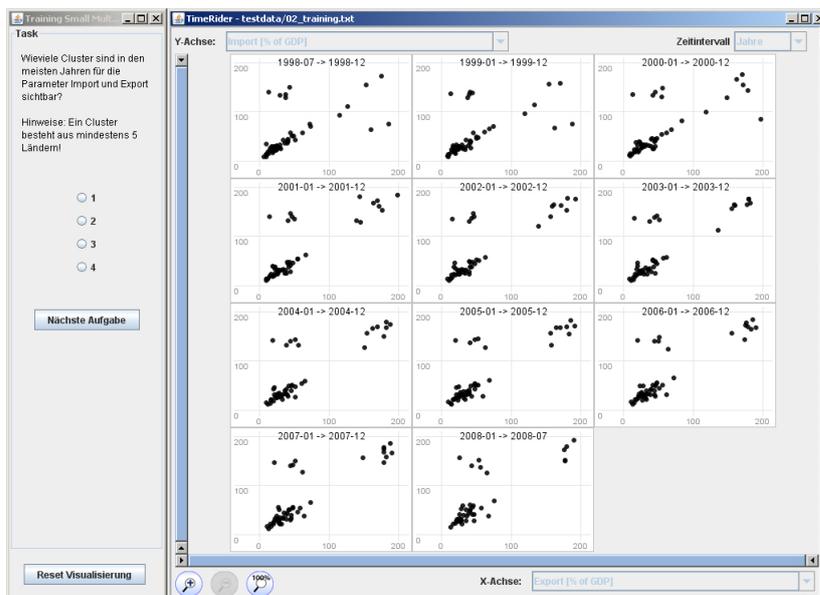


Figure 6.2: The evaluation windows with the Small Multiples.

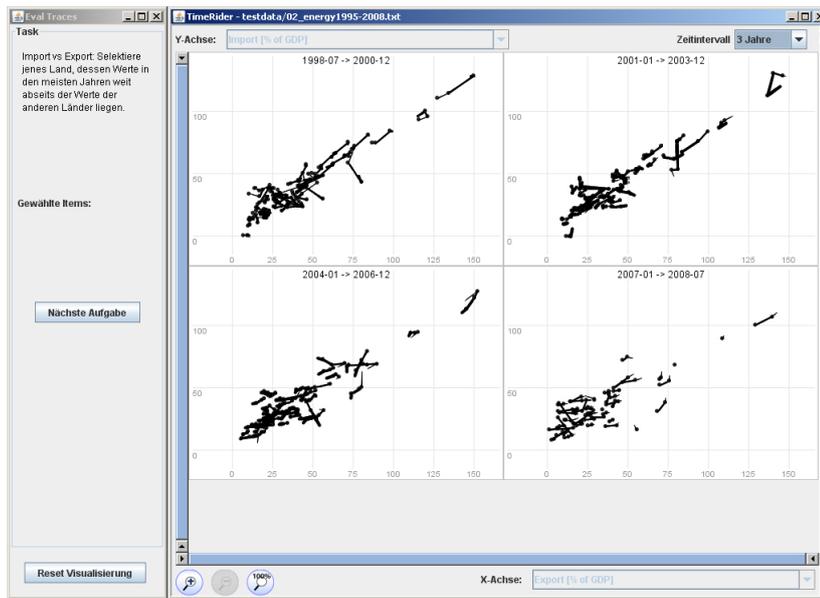


Figure 6.3: The evaluation window with the Trace view.

All of the three visualizations combine different advantages and disadvantages because each visualization works differently. The animated scatter plot presents the data in a dynamic way, whereas the Small Multiples and the Trace view are static visualizations. The main advantage of the animated scatter plot is that the animation supports the user by recognizing changes in the data. Especially small data movements can be observed better with animation. On the other side it is quite time consuming to run through time with the animation to get an overview of the data. The Small Multiple visualization supports users to get an overview on the whole data and to recognize movements of clusters at a glance. The Trace view is somehow categorized between the animated scatter plot and the Small Multiples because it combines their positive aspects. It provides Small Multiple pictures similar to the Small Multiple visualization, presenting the data of time intervals. The difference to the Small Multiple visualization is that the data points belonging to a subject are connected by lines. The lines get thicker over time but always start with a thin line in every time interval picture. Furthermore they are connected across the pictures, i.e. the lines start in the next picture where it has ended in the previous one. So on the one side the Trace view has the capability of the Small Multiples to help the user over viewing the whole data and on the other side the possibility to visualize the data movements with the traces which is quite similar to the animation of the animated scatter plot.

The evaluation was focused on the difference between static and dynamic visualizations, i.e. the animated scatter plot versus the Small Multiples. The second static visualization, the Trace view, was added to the research because of the fact that it combines the advantages of the other two visualizations and would give further interesting insights.

6.2 Hypotheses

Based on my assumptions I created four hypotheses. The first hypothesis refers to the capability of the visualizations to identify and explore the development of clusters. A cluster is a group of points and the development of such a cluster is a movement of all the points in the same direction. The hypothesis assumes that recognizing clusters and their development could be well done with the Small Multiples. The second hypothesis assumes that the Small Multiples would outperform the other visualizations in identifying outliers and their development because the user is able to overview all data points in all time intervals at once. The third hypothesis is concerned with data movements within clusters. This means that clusters can evolve in different directions over a period of time or develop very slow in one direction. The animated scatter plot and the Traces have the capability to point out small changes in the data as well as cluster movements in different directions. Therefore I assumed that they would outperform the Small Multiples in time and accuracy. The last hypothesis deals with the development of specific subjects and the comparison between them, e.g. identifying parameter trends of individual subjects (e.g. patients) or comparing the development of parameter values of different subjects. I assumed that the animated scatter plot and the Trace view are better suited than the Small Multiples to accomplish such tasks because they support the user to perceive movements of individual points better.

For all hypotheses it was necessary to compare all three visualizations. The comparison was based on the recorded times and the correctness of the tasks.

Hypothesis 1: Identifying clusters and exploring developments of clusters over time can be achieved faster with the Small Multiples and the Trace view than with the animated scatter plot. Regarding the correctness the three visualizations do not differ.

Hypothesis 2: Identifying outliers and their development over time can be done faster with the Small Multiples and the Trace view than with the animated scatter plot. Regarding the correctness the three visualizations do not differ.

Hypothesis 3: Animated scatter plot or Trace view are better suited for analyzing data movements within clusters (trends in more than one direction or small data movements in one direction). They will perform better than the Small Multiples in terms of time and correctness.

Hypothesis 4: Tasks involving analyzing and comparing trends of specific subjects can be solved faster and with higher correctness with the animated scatter plot and the Trace view than with the Small Multiples.

6.3 Method

The participants got a short introduction to the program and the evaluation process. They passed through the evaluation process of the TimeRider program with their own computers using Java

Webstart. The process consisted of three main evaluation parts regarding the three visualizations. Before every evaluation part a training session was provided to give the participants the chance to get acquainted with the techniques and the program handling. Finally the participants had to answer questions about their experiences with the visualization techniques.

In the following subsections I will describe the study design and process in detail.

Design

The experiment was set up with a within-subject design. Consequently every participant had to work with all three visualizations. The within-subject design was chosen to reduce individual effects of the participants and to be able to ask every participant for his or her subjective preferences and experiences afterwards. Furthermore a smaller sample size was possible.

There were three independent variables: the visualization technique (animated scatter plot, Small Multiples and Traces), the task type (Trend, Cluster and Outlier) and the task category (Time, Value, Behavior and Identifier). The task type specifies the kind of task whereas the task category characterizes the type of the answer the task questioned. On the other side there were two dependent variables: the time to complete a task and the correctness of a task.

Independent variables	Dependent variables
<ul style="list-style-type: none"> • Visualization techniques: animated scatter plot, Small Multiples, Trace view • Task type: Trend, Cluster and Outlier • Task category: Time, Value, Behavior, Identifier 	<ul style="list-style-type: none"> • task completion time (in ms) • task correctness (1 for correct, 0 for incorrect)

Table 6.1: Independent and dependent variables of the experiment.

Participants

The pool of participants of this user study was restricted to people with reasonable computer experiences as well as basic knowledge about visualization techniques and common user interface controls. The knowledge about visualization techniques and user interface controls was necessary because they had to know how to read simple scatter plots and how to use typical interactive controls (e.g. range sliders).

To match these requirements the students of the course “Informationsdesign und Visualisierung” at the TU Vienna were recruited. This course is part of the Bachelor curriculum in Media informatics, so the students were familiar with using a computer and had a basic knowledge about visualization techniques. 29 students attended the evaluation, 22 male and 7 female persons.

The participants' ages ranged between 21 and 31 (mean age 24.9 years). One of the participants stated a kind of sight disorder but the data of this participant was not excluded from the evaluation because the results were comparable to the results of the other participants.

Apparatus

The evaluation was done over the internet using Java Webstart. The students had to download the Java Webstart-file (.jnlp-file) in a folder they preferred and they had the possibility to work on their own computers. The only constraints were that they had to use a mouse and only one display. But the compliance of these constraints could not be controlled. The size of the program window was fixed to 1024x768 to guarantee that all participants work under the same condition, i.e. the same image resolution.

This automated process was chosen because the number of participants could be easily increased without any additional time requirements. Furthermore it was not clear how many students would attend the course and would carry out the evaluation. Therefore it was necessary to choose a procedure which is flexible in the number of participants.

Interactions and Program Settings

The focus of the evaluation was on the differences between the visualization techniques and not on the usability of the prototype. To ensure that not the time for e.g. picking up the right parameters in the combo box was measured, the interaction possibilities in the program were limited.

The possibility to change the axis parameters was disabled. The axis parameters were adjusted automatically for every task by the evaluation framework. Furthermore the whole menu bar was set invisible. Therefore the following functions were not reachable: the possibility to load data files, export the plot as image, store and load settings as well as to change the trace- and the synchronization-settings. During evaluation the data was always synchronized by date. In the Trace view the second trace mode was activated, i.e. the trace mode where all record points are visible.

Summarized the participants were able to use the following program functions. For all visualizations it was possible to pan by mouse and range sliders and to zoom by mouse, range sliders or zoom-buttons. Furthermore the participants were able to select and highlight items in the plots, to define filters and mark settings and to display the details-on-demand window of a record. Within the animated scatter plot the participants were able to change the time settings, i.e. the granularity, the tempo and the current visible date. The time interval was changeable in the Trace view but not in the Small Multiple view. The decision to switch off the time interval function in the Small Multiple view and in succession to disable the function to switch between different aggregation types was made after the Pretest (see section 6.3). The time interval in the Small Multiple visualization was set to "years" because I used yearly recorded data for the evaluation (see subsection "Dataset" in section 6.3).

Materials

The materials included the evaluation and training tasks as well as the questionnaire and the introductory presentation. The questionnaire was used to get demographic data about the participants and data about the participants' experiences with the visualizations. The demographic data included the age, the gender and sight disorders. The whole questionnaire can be found in the appendix (see Appendix B). The presentation was used to introduce the main features and the evaluation process to the participants (see Appendix A).

Dataset

The subjects of the evaluation were all students of Media Informatics and I assumed that most of them are not familiar with medical parameters. Therefore a dataset containing economic and energy data of countries, composed of data provided on the Gapminder homepage¹, was used. Another advantage of using such real-world data values is that participants are able to become acquainted with it and understand the background of the data easier and faster. Furthermore the economic and energy data of the Gapminder homepage provided the data for longer time intervals compared to the medical dataset.

Altogether the dataset contained 54 countries with data records reaching from 1998 to 2008. There were 13 numerical (e.g. the CO₂ emissions) and 8 logical (e.g. EU member) annually recorded parameters in the dataset. For the evaluation the date of a record was set to the 1st July of the corresponding year. I assumed that every record is valid for the whole year when it was collected.

Because of the within-subject design separate data files for the visualizations were needed to avoid learning effects. Additionally it was necessary to provide another three datasets for the training sessions. I did not create additional datasets to further divide the participants into several subgroups (i.e. to complicate cheating) because producing equal data sets with similar effects on the tasks is very time consuming and hard to achieve. The tasks should be almost similar for all data files and big differences within the data files would make the results less meaningful.

All data files were created by manually adapting the original data values collected from the Gapminder homepage.

The three evaluation sessions were randomly assigned to one of the evaluation datasets whereas the training tasks always get a predefined dataset. The training tasks were assigned to specific datasets because the task instructions included hints about the use of the visualization (e.g. a hint for the animated scatter plot to use the animation to watch the trend of the data).

Evaluation Tasks

There were 22 evaluation tasks covering the topics clusters, outliers and distributions. All tasks were focused on correlations between two numerical parameters and the development of them over time. They did not ask about trends of individual or binary parameters. According to

¹<http://www.gapminder.org/data/>, last accessed on the 28th of February 2012

Plaisent [20] I did not use simple tasks – instead they cover interesting topics and realistic problems. The task descriptions were formulated in German and English, but the German tasks were used for the evaluation. The tasks in German can be found in appendix D.

The tasks were categorized into three different task types: *Trend*, *Outliers* and *Cluster*. These task types specified the main focus of the included tasks. Additionally to the task types there were four different task categories: *Behavior*, *Value*, *Time* and *Identifier*. They reflected the kind of expected answer of the included tasks, e.g. a Time-task asked for a time interval or a specific time point. Table 6.2 shows the distribution of the 22 tasks to task types and categories.

task type	task cat.	sum tasks	hypothesis
Trend	Behavior	4	H1, H3
	Time	2	H3
	Identifier	4	H4
Outliers	Time	2	
	Identifier	3	H2
	Value	1	
Cluster	Time	2	
	Identifier	2	H1
	Value	2	

Table 6.2: The distribution of the tasks to task types, categories and hypothesis.

Every task was assigned to one of the four hypotheses. Approximately every task type referred to one hypothesis except hypothesis 3 and 4 – they both refer to the task type *Trend*.

Hypothesis 1 deals with cluster movements and trends of the entire data. The tasks of hypothesis 1 cover two different task types and four different task categories (see Table 6.3). Task number 1 is the only Trend task in this hypothesis because the question deals with the trend of all countries. The task was assigned to hypothesis 1 because the trend of the data is a clear cluster movement in one direction.

<i>Hypothesis 1</i>			
Nr.	Task type	Task category	Task description
1	Trend	Behavior	Consider the electricity generation and the natural gas production of all countries. What is the trend for the majority of countries?
3	Cluster	Time	Until when are the two main clusters for the parameters energy production and electricity consumption clearly separated if you go forward in time starting in 1998?
20	Cluster	Time	In which time interval can you see two clusters for the parameters hydro production and export?
17	Cluster	Value	How many clusters can you see in the year 1998 for the parameters energy production and energy use?
18	Cluster	Value	How many clusters can you see in most years for the parameters oil production and electricity generation?
19	Cluster	Identifier	Which three countries form a clearly separated cluster in the relationship between population size and electricity consumption that is stable across time?
22	Cluster	Identifier	Which three countries form a clearly separated cluster in the relationship between energy consumption and life expectancy that is stable across time?

Table 6.3: Tasks of hypothesis 1.

The tasks of hypothesis 2 are concentrated on identifying outliers in the data and all refer to task type *Outliers* (see Table 6.4). They cover three different task categories.

<i>Hypothesis 2</i>			
Nr.	Task type	Task category	Task description
10	Outliers	Time	Consider the electricity generation and the natural gas production of all countries. From which year on were the values of the countries Mexico and Turkey far away from the values of all the other countries?
21	Outliers	Time	Import vs. hydro production: in which time interval was Brasil an outlier?
11	Outliers	Identifier	Consider the relationship between natural gas production and energy use. Which country is an outlier over all years?
12	Outliers	Identifier	Import vs Export: select the country which is an outlier over all years.
14	Outliers	Identifier	Does there exist a country which is an outlier over all years considering the parameters nuclear production and energy use?
13	Outliers	Value	How many countries are outliers over all available years considering electricity consumption and hydro production?

Table 6.4: Tasks of hypothesis 2.

Hypothesis 3 deals with the movement of data points within clusters or small cluster movements. The tasks of hypothesis 3 only covers the two task categories Behavior and Time (see Table 6.5).

<i>Hypothesis 3</i>			
Nr.	Task type	Task category	Task description
2	Trend	Behavior	Take a look at the relation of Imports to Exports between 1998 and 2008. Did the majority of the countries develop in similar ways?
8	Trend	Behavior	What is the trend for the majority of countries for the parameters CO2 emissions and energy production?
9	Trend	Behavior	Take a look at the relationship between Life expectancy and Import. Did the majority of the countries develop in similar ways?
16	Trend	Time	Find the time interval where the Import and Export decreased in most countries.

Table 6.5: Tasks of hypothesis 3.

The tasks of hypothesis 4 are focused on identifying individual countries with specific trends and the comparison between countries. Therefore most tasks refer to task category *Identifier* (see Table 6.6).

<i>Hypothesis 4</i>			
Nr.	Task type	Task category	Task description
4	Trend	Identifier	Which country follows a positive correlation for the parameters CO2 emissions and the energy use? India or Hungary?
5	Trend	Identifier	CO2 emissions vs energy production: Which of the following countries (Chile, Ecuador or Kuwait) follows the same trend like in China?
6	Trend	Identifier	Consider the Import related to the life expectancy over all years. Which country (out of the given five countries) develops not in the same way as the other four countries?
7	Trend	Identifier	Which country has a strong growth in natural gas production between the years 1998 and 2008 whereas the population size is very low and nearly stays the same?
15	Trend	Time	Find the time interval where the hydro production decreased and the nuclear production increased in Russia.

Table 6.6: Tasks of hypothesis 4.

Some task descriptions are very similar but associated to different parameters with different effects caused by individual adaptations in the data file. For example question 1 (hypothesis 1) and 2 (hypothesis 3) are equal but they refer to different hypotheses. For one question the data was adapted in a way that it shows a trend in only one direction and for the other question there is a cluster movement in different directions visible within the whole time interval. Additionally there are some similar questions to balance the task categories within a hypothesis as far as possible (e.g. within hypothesis 1 and hypothesis 2)

Training Tasks

The training tasks should help the participants to get familiar with the visualizations. There were five training tasks for every visualization. They were presented with the same limited user interface and were similar to the evaluation tasks. The difference between training and evaluation tasks was that the correct answers were shown after answering training tasks. The participant also had the possibility to repeat an incorrectly answered training task. For every training task a detailed task instruction was shown to help the participant to accomplish the task, e.g. for the animated scatter plot the student got the advice to use the animation for analyzing the trend of the data. In Appendix C all training tasks and the visualization specific instructions are listed.

Procedure

The students got a 15-minute introduction to TimeRider and how the evaluation would be carried out. The presentation included some detail information about the visualization techniques and the important functions of the program (see Appendix A). The evaluation procedure was demonstrated by solving some example questions. The students were informed to do the tasks as carefully and as quick as possible. To motivate them, the students who solved the tasks best (considering time and correctness) got five extra points for their course.

After the introductory presentation the Java Webstart download was put online. The participants got one week to finish the tasks and to upload their zipped evaluation journal on the e-learning platform of the course. The evaluation journal included the interaction log files and result files as well as the answers to the questionnaire.

The evaluation comprised three evaluation sessions, one for every visualization technique. All evaluation sessions consisted of 22 tasks. Before every evaluation session a short training session with five training tasks was carried out, to give the participants the chance to get familiar with the visualization. The participants got feedback during the training session if they answered the task wrong or correctly. They also had the possibility to start a wrong answered training task again. After finishing all sessions the participants had to answer a short questionnaire. If the student had to stop the experiment prematurely he or she had to start again. In that case new logging files were created. A timestamp associated with the logging file indicated the order of the created log files. The participants were informed of that procedure and they were urged to do the whole evaluation at once. The procedure caused no problems because only one participant aborted the program during evaluation and started again.

To control the systematic error caused by the experimental procedures the evaluation and training sessions concerning the three different visualizations were randomly ordered. All together six different possibilities for the order of the evaluation sessions did exist. The evaluation tasks within an evaluation session were also randomly ordered. Training tasks were always presented in the same order. The datasets were always presented in the same order (i.e. dataset 1 in the first round, dataset 2 in the second round and dataset 3 in the last round).

Pretest

Before the evaluation started, a pilot test was carried out to find difficulties in the tasks and to get an estimation for the time needed to accomplish all tasks. The pilot test was carried out with the teaching assistant of the course “Informationsdesign und Visualisierung”, who never used the program before. I presented him the introduction materials, i.e. the Power Point presentation including the short introduction to all visualization techniques and information about the evaluation process. Afterwards he started the prototype via Java Webstart and passed through all tasks. The test person was observed by three persons during the whole pilot test.

Overall the pilot test did correspond to the planned test, but some problems were identified. I figured out that the time interval of the Small Multiples should be preconfigured to one year

and should be not changeable because otherwise the user would be able to watch a kind of trace composed of the points – ending up with a similar visualization as the trace view provided. To discriminate between the pros and cons of Small Multiples and trace view the time interval function was deactivated for the Small Multiples. On the other hand there were not enough time intervals available for the Trace view, e.g. two and three years were missing. Because of the fact that every Small Multiple picture always contain only records of one year the possibility to switch between different aggregation types, e.g. to show only the last records within one Small Multiple, was also disabled.

The test person mentioned that the different transparencies confused him while he was following the animation. The evaluation data only contained yearly recorded (regularly sampled) data and therefore different transparencies did not make any sense. Therefore I disabled the transparency function of the animation for the evaluation.

The log and result files of the test person were also checked. The logging worked very well, but some detail information was missing, e.g. the logging of the time slider movements. The logging was adapted after the pretest to also record those missing details.

6.4 Results

The evaluation results contained timing and correctness data of 29 participants. The data was assembled and prepared for the statistical analysis. The results of descriptive and inferential statistics were done separately for every hypothesis and are presented in the following subsections. For the three different interfaces/visualizations I used the shortcuts “SC” (animated scatter plot), “SM” (Small Multiples) and “T” (Trace view) during analysis and in the further descriptions.

I used all time and correctness records individually for timing and error analysis. Additionally I calculated success rates (number of correct answers divided by number of subjects) for further correctness analysis.

Hypothesis 1

A Shapiro-Wilk test performed on the completion time for every task showed that timings were not normal distributed but log-normal distributed. With a two-way ANOVA I found a significant effect of dataset on timings ($F(2, 600) = 36.58, p < 0.001$). Also a two-way ANOVA applied on all combinations of datasets (e.g. dataset 1 + dataset 2, dataset 3 separately) showed a significant influence of dataset on timings (dataset 2+3: $F(2, 400) = 32.250, p < 0.001$; dataset 1+2: $F(2, 400) = 7.616, p < 0.01$; dataset 1+3: $F(2, 400) = 72.553, p < 0.001$). Therefore for further analysis the results were considered separately for every dataset.

Success rates were neither normal nor log-normal distributed. A Chi-square test ($\chi^2(2, N = 609) = 0.713, p = 0.700$) performed on the task correctness and a Kruskal-Wallis ($\chi^2(2) = 0.209, p = 0.901$) test applied on the success rates, showed that the dataset had no influence on them. Nevertheless the further analysis of the success rates and correctness was split by dataset because timings showed a dataset dependency.

Timings

A F-Test performed on the timing results showed equal variances between interfaces for all datasets. A Shapiro-Wilk test showed that the data of the different interfaces for all datasets were log-normal distributed. Figure 6.4 shows the completion times for the different interfaces split by dataset and Table 6.7 summarizes the results of descriptive statistics for timings.

To analyze the differences between the interfaces for timings I used a one-way ANOVA separately for every dataset. For all datasets a significant effect of interface on timings was found (dataset 1: $F(2, 200) = 16.02, p < 0.001$; dataset 2: $F(2, 200) = 3.306, p = 0.039$; dataset 3: $F(2, 200) = 5.717, p < 0.01$). Post-hoc tests on completion time using a t-Test with Bonferroni correction showed significant differences between SC and T (dataset 1: $p < 0.001$) and two times between SM and T (dataset 2: $p = 0.022$; dataset 3: $p = 0.016$).

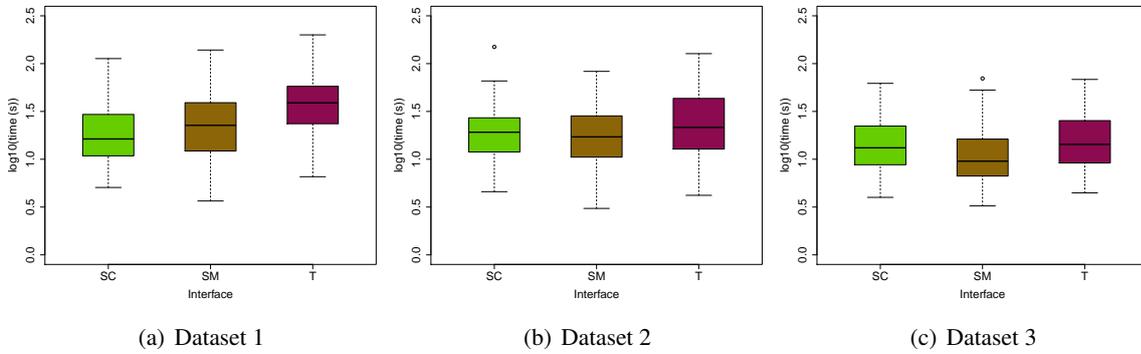


Figure 6.4: Timing Boxplots of H1 for every dataset.

Interface	Dataset	MEDIAN [sec]	MEAN [sec]	SD [sec]
SC	1	16.3	23.4	19.4
SM		22.6	33.9	33.8
T		38.9	46.9	35.2
SC	2	19.2	23.6	22.2
SM		17.1	21.9	16.6
T		21.6	31.9	28.2
SC	3	13.2	17.4	12.6
SM		9.5	13.7	11.5
T		14.3	20.0	15.5

Table 6.7: Descriptive measures for timings of hypothesis 1.

A one-way ANOVA showed a significant effect of task category on timings (dataset 1: $F(3, 199) = 27.08, p < 0.001$; dataset 2: $F(3, 199) = 20.74, p < 0.001$; dataset 3: $F(3, 199) = 49.63, p < 0.001$). Therefore I additionally performed tests on the completion times split by task categories (and dataset). I performed F-tests on the datasets of the different task categories and found out that all had equal variances except task category *Identifier* regarding dataset 1 and dataset 2 and task category *Time* regarding dataset 1. Furthermore I used one-way ANOVA for groups with equal variances and otherwise a Kruskal-Wallis test.

A Kruskal-Wallis test showed a significant effect of interface on completion time within task category *Identifier* for dataset 1 and dataset 2 (dataset 1: $\chi^2(2) = 11.053, p < 0.01$; dataset 2: $\chi^2(2) = 7.523, p = 0.023$). A post-hoc test using a t-Test with Bonferroni correction for dataset 1 showed a significant difference between SC and T (dataset 1: $p = 0.039$). Using a Mann-Whitney test with Bonferroni correction for dataset 2 resulted in a significant difference between SM and T (dataset 2: $p = 0.037$). Figure 6.5 presents the task completion times for dataset 1 and dataset 2.

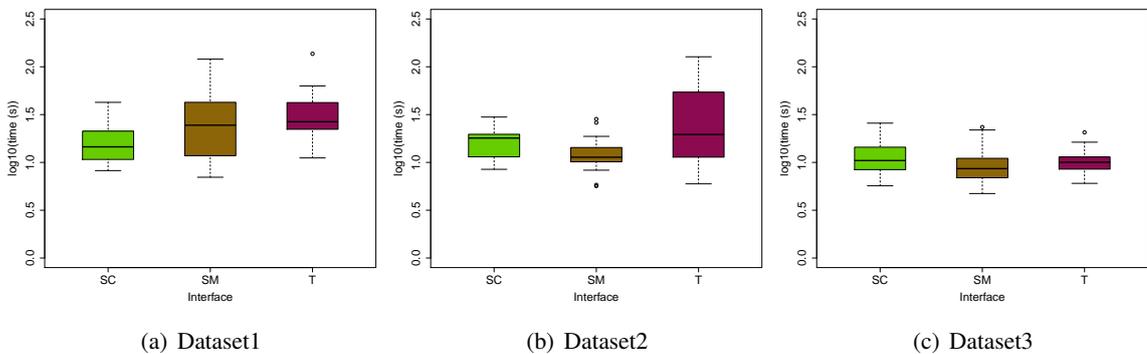


Figure 6.5: Boxplots for H1 of task category Identifier for all datasets.

For task category *Behavior* I found significant effects of interface on time using one-way ANOVA for dataset 1 and dataset 2 (dataset 1: $F(2, 26) = 6.219, p < 0.01$; dataset 2: $F(2, 26) = 3.997, p = 0.031$). A t-Test with Bonferroni correction revealed a significant difference between SC and T for dataset 1 ($p = 0.01$) and SC and SM for dataset 2 ($p = 0.021$). The results for dataset 1 and dataset 2 are shown in Figure 6.6.

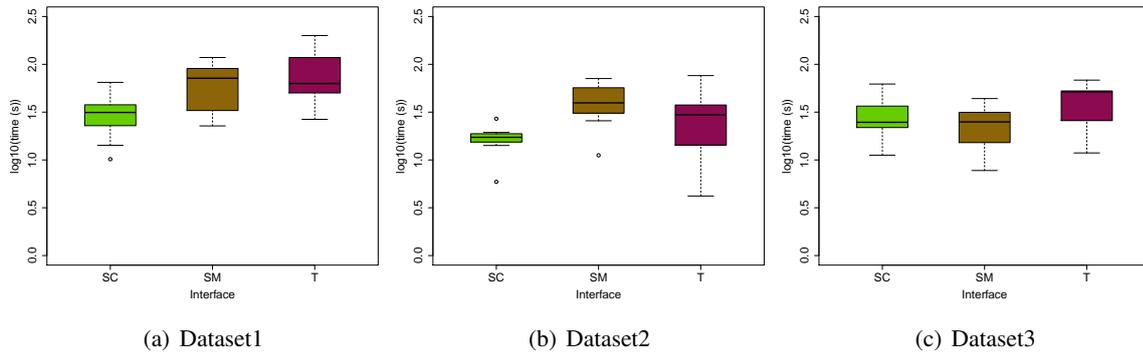


Figure 6.6: Boxplots for H1 of task category Behavior for all datasets.

There was also a significant effect for dataset 1 within task category *Value* using one-way ANOVA (dataset 1: $F(2, 55) = 7.94, p < 0.001$). With the post-hoc test, t-Test with Bonferroni correction, I found a significant effect for SC and T ($p < 0.001$) and SM and T ($p < 0.01$). Figure 6.7 presents the results for dataset 1.

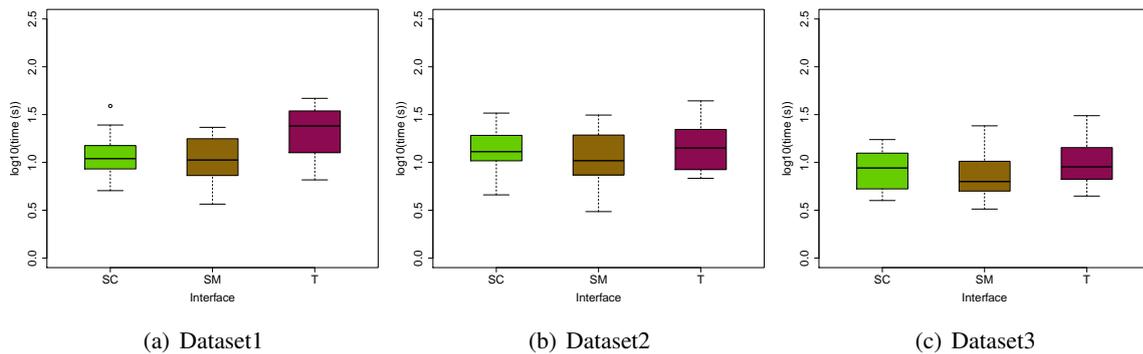


Figure 6.7: Boxplot for H1 of task category Value for all datasets.

With one-way ANOVA I found significant differences between interfaces for task category *Time* for dataset 3 ($F(2, 55) = 6.569, p < 0.01$) and significant difference for dataset 1 with a Kruskal-Wallis test ($\chi^2(2, 55) = 15.3864, p < 0.001$). Using a t-Test with Bonferroni correction I revealed significant differences between SM and T (dataset 3: $p = 0.028$). I ran a Mann-Whitney's test to evaluate the difference between the interfaces for dataset 1 and revealed significant differences between SC and T ($p < 0.001$) and SM and T ($p < 0.01$). The results are shown in Figure 6.8).

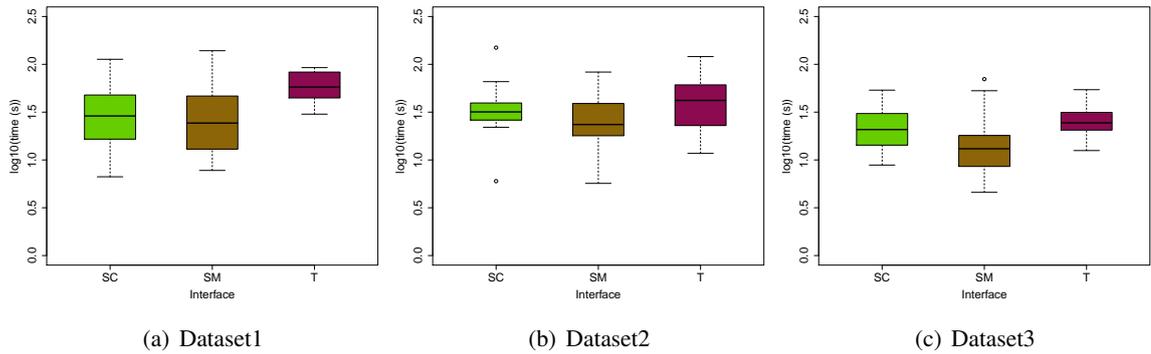


Figure 6.8: Boxplots for H1 of taskcategory Time for dataset 1 and dataset 3.

Correctness

I analyzed the correctness with the binomial Chi-square test separately for every dataset. The Chi-square test showed no significant effect of interface on the results for all datasets.

A Kruskal-Wallis test performed on the success rates of the hypothesis revealed a significant influence of interface on correctness for dataset 1 ($\chi^2(2) = 7.155, p = 0.028$). A post-hoc test using Mann-Whitney tests with Bonferroni correction showed a significant difference between the SM and T ($p = 0.05$). Figure 6.9 presents the success rates of the different interfaces split by dataset and Table 6.8 summarizes the means, variances and standard deviations for the success rates.

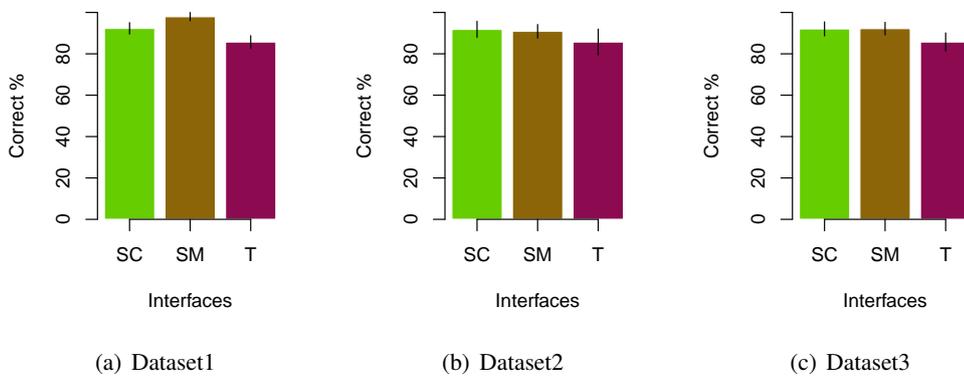


Figure 6.9: Success rates for tasks of H1 split by datasets.

Interface	Dataset	MEAN	VAR	SD
SC	1	0.923	0.005	0.074
SM		0.979	0.003	0.054
T		0.857	0.010	0.101
SC	2	0.918	0.006	0.076
SM		0.909	0.005	0.072
T		0.857	0.004	0.064
SC	3	0.921	0.006	0.075
SM		0.922	0.010	0.098
T		0.857	0.036	0.189

Table 6.8: Descriptive measures for success rates of hypothesis 1.

Hypothesis 2

The completion times of the tasks concerning to H2 were log-normal distributed according to a Shapiro-Wilk test performed on the tasks. A two-way ANOVA test revealed significant influences of the datasets to the completion times ($F(2, 513) = 12.122, p < 0.001$). For the different dataset combinations I also found influences of dataset (dataset 2+3: $F(2, 342) = 5.900, p = 0.016$; dataset 1+2: $F(2, 342) = 5.712, p = 0.017$; dataset 1+3: $F(2, 342) = 26.601, p < 0.001$). Consequently the results were analyzed separately for every dataset.

A Shapiro-Wilk test showed that success rates were neither normal nor log-normal distributed. Furthermore I performed a Chi-square test ($\chi^2(2, N = 522) = 15.636, p < 0.001$) on the task correctness and a Kruskal-Wallis ($\chi^2(2) = 0.853, p = 0.653$) test on the success rates to analyze the effect of dataset on the results. Because of the fact that I have found a significant influence of dataset on timings and on the correctness I also analyzed the correctness and success rates split by dataset.

Timings

I performed a one-way ANOVA test separately for every dataset and I did not find any significant differences between the interfaces. Figure 6.10 displays the results for the different interfaces split by dataset. In Table 6.9 the medians, means and standard deviations of the interfaces for all datasets are summarized.

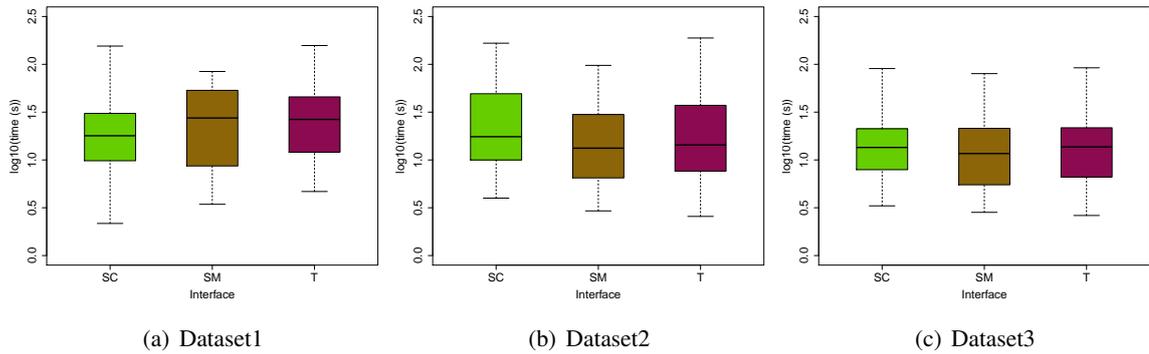


Figure 6.10: Timing Boxplots of H2 for every dataset.

Interface	Dataset	MEDIAN [sec]	MEAN [sec]	SD [sec]
SC	1	17.9	25.9	27.1
SM		27.5	32.1	24.2
T		26.5	34.0	30.6
SC	2	17.5	31.9	32.5
SM		13.3	19.7	17.9
T		14.4	28.9	32.8
SC	3	13.5	19.6	18.5
SM		11.7	16.1	15.0
T		13.7	20.5	21.9

Table 6.9: Descriptive measures for timings of hypothesis 2.

Also a division by task categories revealed no significant differences between the interfaces except for task category *Value* and dataset 1 ($F(2, 26) = 4.032, p = 0.030$) and for task category *Time* and dataset 2 ($F(2, 55) = 4.314, p = 0.018$) using one-way ANOVA tests. A post-hoc test for task category *Value* (dataset 1) using a t-Test with Bonferroni correction showed a significant difference between SC and SM ($p = 0.012$). For task category *Time* the t-Test with Bonferroni correction also resulted in a difference between SC and SM ($p = 0.048$). Figure 6.11 shows the time results for task category *Time* and Figure 6.12 the results for task category *Value*. Figure 6.13 presents the results of taskcategory *Identifier*.

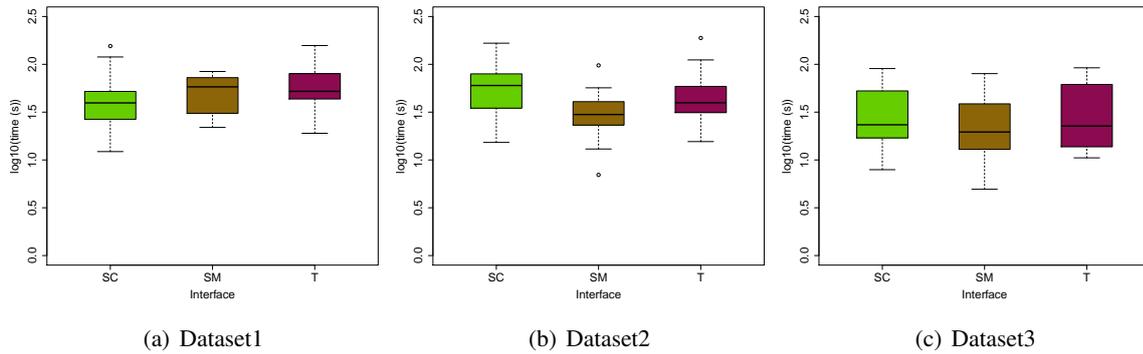


Figure 6.11: Boxplots for H2 of task category Time.

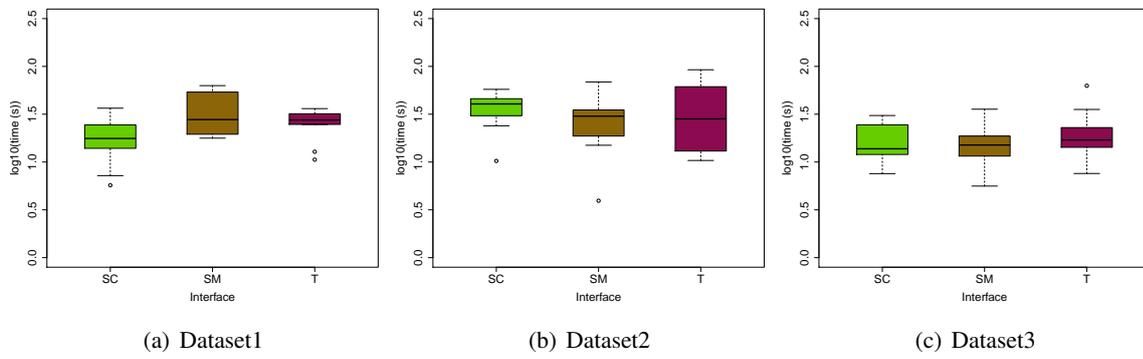


Figure 6.12: Boxplots for H2 of task category Value.

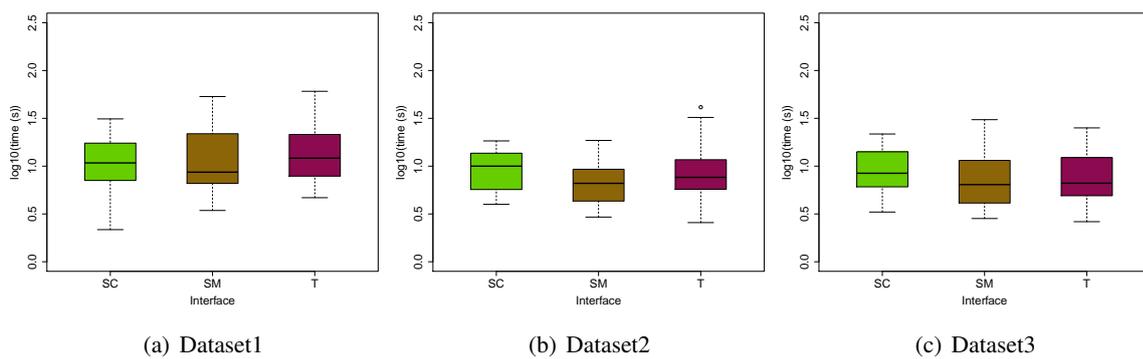


Figure 6.13: Boxplots for H2 of task category Identifier.

Correctness

A Chi-square test applied on the correctness of the hypothesis tasks brought a significant difference between interfaces for dataset 3 ($\chi^2(2, N = 174) = 6.4245, p = 0.040$). A post-hoc test using Fisher's test showed a significant difference between SC and SM ($p = 0.022$, the odds ratio is 6.309).

Using a Kruskal-Wallis test I found a significant effect of interface on success rates for dataset 3 ($\chi^2(2) = 6.5696, p = 0.037$). But a post-hoc test using a pairwise Mann-Whitney test with Bonferroni correction showed no significant difference between the different interface combinations.

Figure 6.14 presents the correctness results in boxplots and Table 6.10 summarizes the descriptive measures for H2.

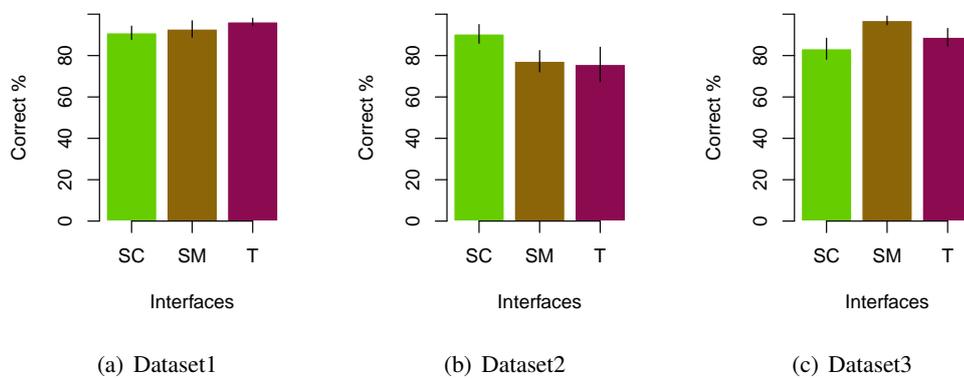


Figure 6.14: Success rates for tasks of H2 split by datasets.

Interface	Dataset	MEAN	VAR	SD
SC	1	0.910	0.017	0.129
SM		0.929	0.017	0.131
T		0.963	0.012	0.111
SC	2	0.905	0.017	0.131
SM		0.773	0.018	0.135
T		0.758	0.019	0.137
SC	3	0.833	0.028	0.167
SM		0.970	0.005	0.067
T		0.889	0.007	0.083

Table 6.10: Descriptive measures for success rates of hypothesis 2.

Hypothesis 3

Completion times were not normal distributed but log-normal distributed. With a two-way ANOVA test I found a significant effect of dataset on time ($F(2, 339) = 7.063, p < 0.001$). Dataset 2 combined with dataset 3 revealed no significant effect of dataset ($F(2, 226) = 0.659, p = 0.418$). For the other two dataset combinations a significant effect was found (dataset 1+2: $F(2, 226) = 7.433, p < 0.01$; dataset 1+3: $F(2, 226) = 14.169, p < 0.001$).

A Shapiro-Wilk test showed that success rates were neither normal nor log-normal distributed. A Chi-square test ($\chi^2(2, N = 348) = 15.261, p < 0.001$) was used to find effects of the dataset on the task correctness and a Kruskal-Wallis ($\chi^2(2) = 2.400, p = 0.301$) test to find influences of dataset on the success rates. The Chi-square test performed on the combination of dataset 2 and dataset 3 also revealed a significant effect of dataset on the correctness ($\chi^2(1, N = 232) = 8.451, p < 0.01$). Therefore the further analysis was done separately for every dataset.

Timings

A one-way ANOVA showed that there was no significant difference between the interfaces regarding dataset 2 and dataset 3 (dataset 2: $F(2, 113) = 1.118, p = 0.331$; dataset 3: $F(2, 113) = 0.329, p = 0.72$). The test performed with dataset 1 revealed a significant effect of interface ($F(2, 113) = 21.37, p < 0.001$). A post-hoc test using a pairwise t-Test with Bonferroni correction showed a significant difference between SC and SM ($p < 0.001$) and SC and T ($p < 0.001$). Figure 6.15 shows the timings for every interface separated by dataset. Table 6.11 summarizes the descriptive values for H3.

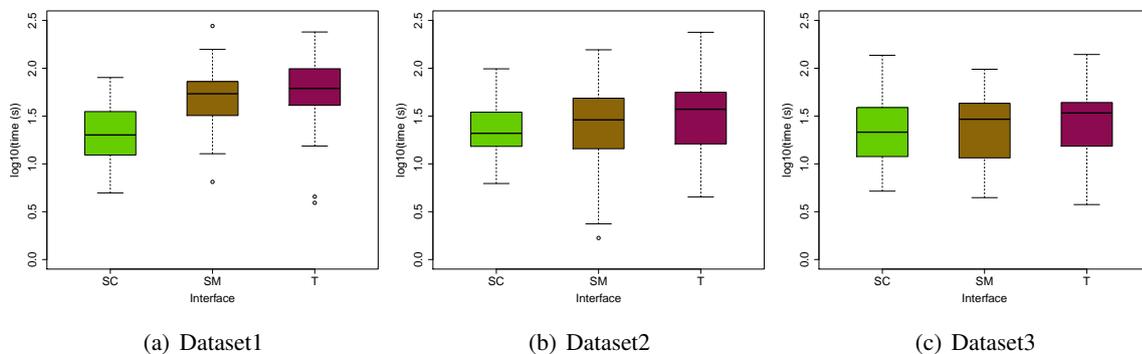


Figure 6.15: Timing Boxplots of H3 for every dataset.

Interface	Dataset	MEDIAN [sec]	MEAN [sec]	SD [sec]
SC	1	23.3	28.1	18.1
SM		61.4	72.5	53.2
T		62.2	92.8	69.8
SC	2	24.9	29.7	20.8
SM		31.5	38.4	30.4
T		43.6	50.5	44.9
SC	3	25.1	31.7	27.0
SM		31.3	34.0	22.7
T		34.9	40.1	32.4

Table 6.11: Descriptive measures for timings of hypothesis 2.

Performing F-Tests on the task category subsets showed that they had equal variances. Furthermore I analyzed the timings according to the task categories with ANOVA tests. With a one-way ANOVA I found a significant effect of interface for task category *Time* for dataset 1 ($F(2, 26) = 12.46, p < 0.001$). A post-hoc test using the t-Test with Bonferroni correction resulted in a significant difference between SC and T ($p = 0.004$). Considering only tasks of category *Behavior* also showed a significant effect of interface on timings for dataset 1 ($F(2, 84) = 13.84, p < 0.001$). A t-Test with Bonferroni correction performed on dataset 1 for *Behavior* tasks yielded a significant difference between SC and SM ($p < 0.001$) and SC and T ($p < 0.001$).

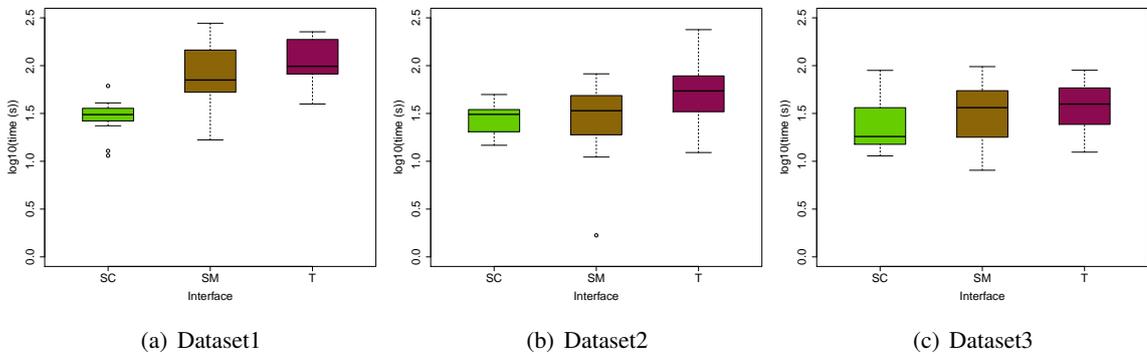


Figure 6.16: Boxplots for H3 of taskcategory Time.

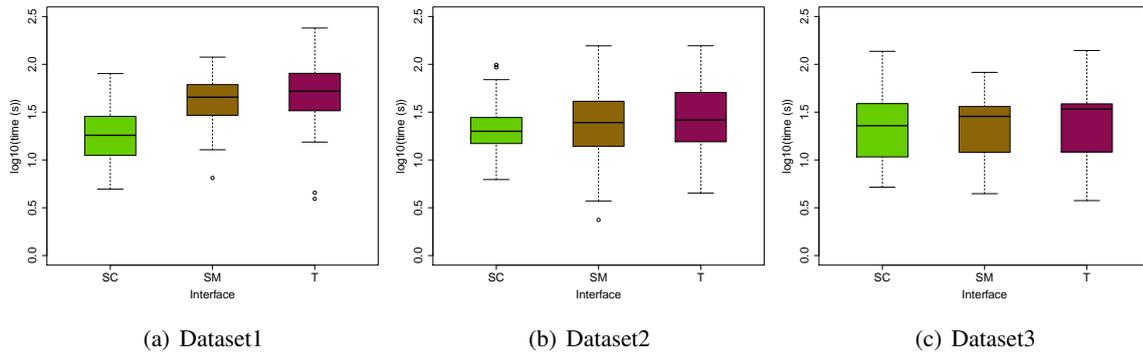


Figure 6.17: Boxplots for H3 of taskcategory Behavior.

Correctness

The Chi-square test performed on the task correctness' resulted in a significant effect of interface for all datasets (dataset 1: $\chi^2(2, N = 116) = 11.693, p < 0.01$; dataset 2: $\chi^2(2, N = 116) = 6.289, p = 0.043$; dataset 3: $\chi^2(2, N = 116) = 8.517, p = 0.014$). A post-hoc test using Fisher's test showed a significant difference between SC and SM ($p < 0.01$) and between SC and T ($p = 0.022$) for dataset 1. Regarding dataset 2 and dataset 3 I also got a significant difference between SC and SM (dataset 2: $p = 0.016$; dataset 3: $p = 0.004$).

Analyzing the success rates by using a Kruskal-Wallis test resulted in significant effects of interface for dataset 1 ($\chi^2(2) = 7.782, p = 0.020$) and dataset 3 ($\chi^2(2) = 8.085, p = 0.018$). A post-hoc test using Mann-Whitney test with Bonferroni correction showed a significant difference between SC and SM for dataset 1 ($p = 0.038$) and dataset 3 ($p = 0.02$).

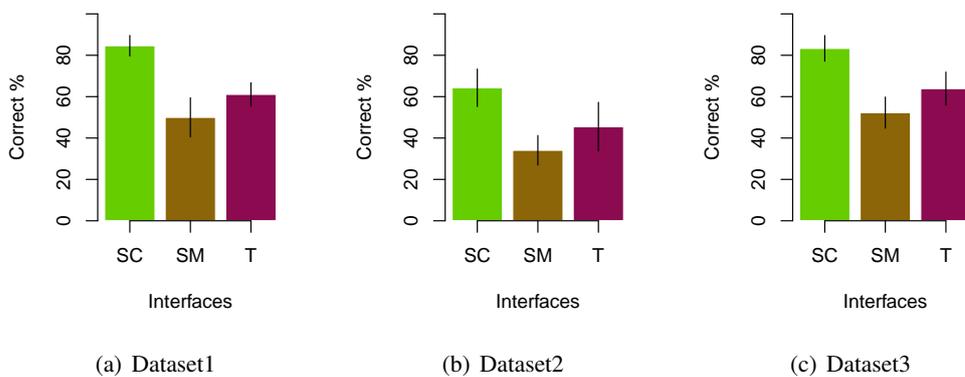


Figure 6.18: Success rates for tasks of H3 split by datasets.

Interface	Dataset	MEAN	VAR	SD
SC	1	0.846	0.047	0.217
SM		0.500	0.083	0.289
T		0.611	0.095	0.309
SC	2	0.642	0.060	0.244
SM		0.341	0.078	0.280
T		0.455	0.135	0.368
SC	3	0.833	0.047	0.217
SM		0.522	0.031	0.175
T		0.639	0.064	0.253

Table 6.12: Descriptive measures for success rates of hypothesis 3.

Hypothesis 4

The timings and error rates for hypothesis 4 were tested for normal distribution and log-normal distribution. The Shapiro-Wilk test showed that timings of all tasks were not normal distributed but log-normal distributed. The error rates of the individual tasks were neither normal nor log-normal distributed.

To find out the influence of the dataset on the timings a two-way ANOVA test was performed. According to the results of ANOVA ($F(2, 426) = 0.627, p < 0.535$) there was no significant difference between the data sets for the timings.

A Pearson's Chi-square test was performed on the task correctness (sum of correct answers for each task split by dataset) and a Kruskal-Wallis test was done on the success rates to analyze the dataset dependency of the results. The Kruskal-Wallis test ($\chi^2(2) = 4.297, p = 0.117$) revealed that there was no significant influence of the dataset on the success rates but a Chi-square test ($\chi^2(1, N = 435) = 15.723, p < 0.001$) showed a significant influence of dataset on the correctness. The combination of dataset 2 and dataset 3 revealed no significant influence of dataset ($\chi^2(1, N = 290) = 0.283, p = 0.595$). Therefore I analyzed dataset 2 and 3 together and dataset 1 separately.

Timings

An F-Test was done to test the equality of variances between the timing results of the different interfaces. The results of the F-Test showed that only the interfaces SM and T had equal variances but not interface SC with the others. Furthermore I used a Friedman test for the analysis of the timing results.

Kruskal-Wallis tests performed on the timings revealed significant effects of interface on the timings for both dataset combinations (dataset 1: $\chi^2(2) = 60.486, p < 0.001$; dataset 2 + dataset 3: $\chi^2(2) = 29.006, p < 0.001$). With a post-hoc test, pairwise using a Wilcoxon test, I found significant differences between SC and T ($p < 0.001$) and SC and SM ($p < 0.001$) for

both dataset combinations. Figure 6.19 illustrates the differences in time between the interfaces. Table 6.13 summarizes the descriptive results for hypothesis 4.

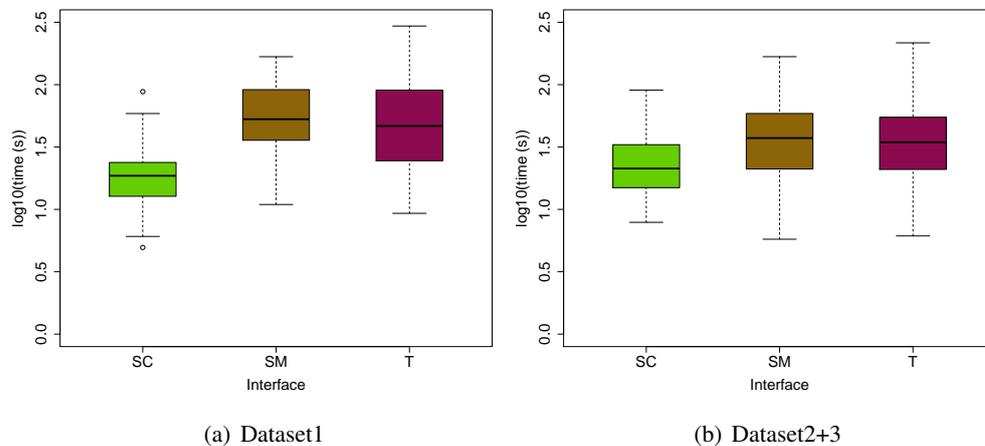


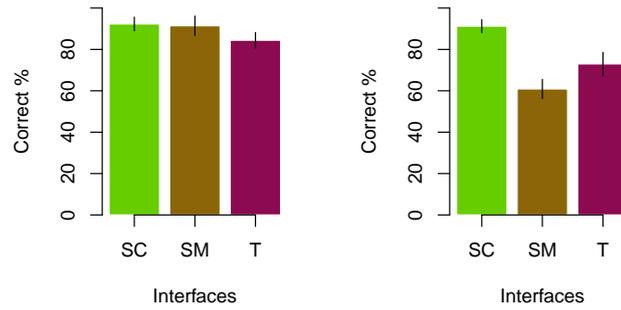
Figure 6.19: Timing results separated by interface

Interface	Dataset	MEDIAN [sec]	MEAN [sec]	SD [sec]
SC	1	18.6	21.2	14.4
SM		52.8	66.8	39.7
T		46.6	69.4	61.1
SC	2+3	21.3	25.6	15.6
SM		37.3	45.9	33.5
T		34.5	45.5	38.4

Table 6.13: Descriptive measures for timings of hypothesis 4.

Correctness

A Chisquare-test performed on the task correctness revealed a significant effect of interface for dataset 2+3 ($\chi^2(2, N = 290) = 21.878, p < 0.001$). A post-hoc test using Fisher's test showed a significant difference between SC and SM ($p < 0.001$) and between SC and T ($p < 0.01$). I also found a significant effect for dataset 2+3 of interface on the success rates using a Kruskal-Wallis test ($\chi^2(2) = 16.023, p < 0.001$). A post-hoc test using Mann-Whitney test with Bonferroni correction showed a significant difference between SC and SM ($p < 0.001$) and SC and T ($p = 0.012$).



(a) Dataset1

(b) Dataset2+3

Figure 6.20: Success rates for tasks of H4 for the different interfaces.

Interface	Dataset	MEAN	VAR	SD
SC	1	0.923	0.024	0.154
SM		0.914	0.011	0.107
T		0.844	0.038	0.194
SC	2+3	0.913	0.016	0.126
SM		0.609	0.070	0.265
T		0.730	0.039	0.198

Table 6.14: Descriptive measures for success rates of hypothesis 4.

Summary Results

The following table summarizes the statistically significant results of the evaluation for all hypotheses.

Hypoth.	Time	Success rates	Correctness
H1	dataset1: SC vs. T t-test $p < 0.001$ dataset2: SM vs. T t-test $p = 0.002$ dataset3: SM vs. T t-test $p = 0.016$	dataset1: SM vs. T Mann Whitney test, $p = 0.05$	no difference
H2	no difference	no difference	dataset3: SC vs. SM Fisher's test, $p = 0.022$
H3	dataset1: SC vs. SM t-test $p < 0.001$ SC vs. T t-test $p < 0.001$	dataset1: SC vs. SM Mann Whitney test, $p = 0.038$ dataset3: SC vs. SM Mann Whitney, $p = 0.02$	dataset1: SC vs. SM Fisher's test, $p < 0.01$ SC vs. T Fisher's test, $p = 0.022$ dataset2: SC vs. SM Fisher's test, $p < 0.016$ dataset3: SC vs. SM Fisher's test, $p = 0.004$
H4	dataset1: SC vs. T Wilcoxon test, $p < 0.001$ SC vs. SM Wilcoxon test, $p < 0.001$ dataset2+3: SC vs. T Wilcoxon test, $p < 0.001$ SC vs. SM Wilcoxon test, $p < 0.001$	dataset2+3: SC vs. SM, Mann Whitney test, $p < 0.001$ SC vs. T Mann Whitney test, $p = 0.012$	dataset2+3: SC vs. SM Fisher's test, $p < 0.001$ SC vs. T Fisher's test, $p < 0.01$

Table 6.15: Summary of the evaluation results for every hypothesis.

Questionnaire Results

28 participants answered the questionnaire (one questionnaire was missing). The results are summarized in Table 6.16). The overall favorite visualization technique of the subjects was the animated scatter plot with 27 subjects. Only one person preferred the Small Multiples and nobody favored the Traces. For analysis tasks considering correlations over time and trends the participants answered equally to the previous question. For identifying outliers over time 16 subjects preferred the animated scatter plot, 10 subjects the Small Multiples and 2 subjects the Traces. The same was true for the analysis of clusters over time.

Question		SC	SM	T
Favorite visualization		27	1	0
Analysis of trends and correlations over time		27	1	0
Identifying outliers over time		16	10	2
Analysis of clusters over time		16	10	2
Understandability	++	28	8	2
	+	0	8	5
		0	8	6
	-	0	4	12
	--	0	0	3
Usability	++	28	9	4
	+	0	7	4
		0	5	10
	-	0	2	8
	--	0	5	2

Table 6.16: Overview of the questionnaire results.

For the last two questions I have calculated a final grade for every visualization: ++ = 2 points, + = 1 points, ~ = 0 points, - = -1 points, -- = -2 points. The sum of the points is the result of the final score.

The animated scatter plot was the leader regarding understandability and usability – all participants gave the animated scatter plot the best grading for both characteristics (56 points for both characteristics). Most of the participants graded the Small Multiples moderate for understandability (20 points) and usability (13 points) whereas the Trace view was assessed very bad (-9 points for understandability and 0 points for usability).

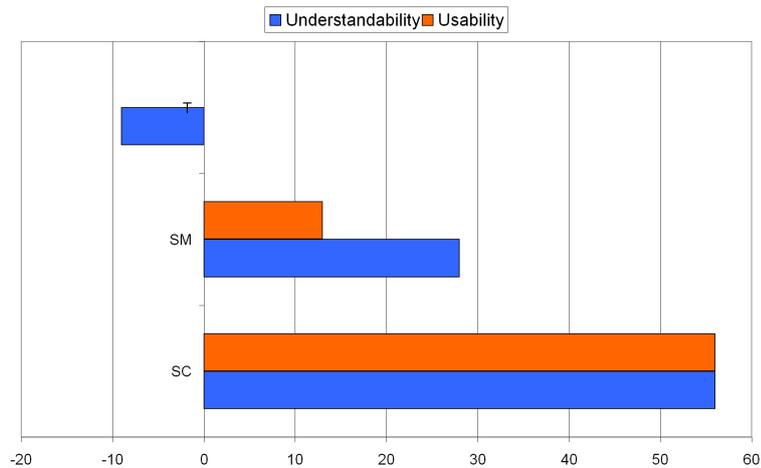


Figure 6.21: Calculated grading of the visualizations for understandability and usability.

The last question of the questionnaire was designed for additional comments of the participants. Some comments of the test persons pointed out difficulties with the trace view. One person mentioned that it was hard to recognize the direction of the traces and suggested to use a color gradient to visualize the direction. Another participant proposed to use arrows to highlight the trace direction. The different thicknesses of the trace divisions were hard to observe without zooming because they were too short to recognize the gradient of thickness. But a person mentioned that zooming the trace view caused that the overview got lost. Another problem was caused by overlapping traces because then individual countries could not be distinguished. Overlapping points were also a problem within the Small Multiple visualization. One person proposed to make data points transparent as solution for that problem. Another participant mentioned that the Small Multiples were not applicable for all tasks, e.g. to find differences between time points.

Two test persons criticized that the filter panel was not available during evaluation. The filter panel was activated during evaluation but the tab panel was hidden by setting the panel splitter on the right side.

One positive feedback was given regarding the animated scatter plot. A person wrote that the time slider was very useful to control the animation and furthermore very helpful for the interpretation of the data.

Log-file Results

I analyzed the log files separately for every visualization. My focus was on the general use of the program, e.g. how many participants adjusted mark settings or how many participants used the zoom functionality of the program.

Two-third of the participants never or rarely used the play, pause and stop button to control the animation. They used instead the time slider to adjust the time. Only a few persons who passed through the animation with the play button adapted the tempo. Nobody used the mark settings to highlight groups of countries in the animated scatter plot and only one person adjusted a

name filter. About one-fourth of the participants changed the granularity and only a few persons zoomed and/or panned the plot. If they zoomed the plot, the zoom buttons and the zoom by mouse was used more often than the possibility to zoom with the range sliders.

Within the Small Multiples the selection feature was used very often for non-selection tasks. Most participants used the mouse to pan and zoom the plot but some persons also used the range sliders and the zoom buttons. Three participants used the mark settings to highlight countries and four also adjusted filters (mostly name filters).

All participants used to change the time interval in the Trace view during evaluation for about half of the questions (MEAN=10.9, MEDIAN=12). Only three persons used filters and only one used the mark settings. Nearly every participant also selected countries in non-selection tasks to highlight them. For zooming most persons used the mouse, but one-third preferred the range slider and the zoom button (or they did not know that they can zoom with the mouse).

Animated Scatter plot	Small Multiples	Trace view
<ul style="list-style-type: none"> • set time with time slider • select subjects (for non-selection tasks) 	<ul style="list-style-type: none"> • select subjects (for non-selection tasks) • pan and zoom the plots 	<ul style="list-style-type: none"> • change time interval • select subjects (for non-selection tasks) • pan and zoom the plots

Table 6.17: This table summarizes interactions with the program which often occurred during evaluation.

Conclusion and Outlook

The evaluation study carried out during this thesis was aimed to compare three visualization techniques. There were 29 participants who carried out 22 tasks for every visualization. The task correctness and completion times were recorded and afterwards analyzed with statistical methods. Furthermore the questionnaire was evaluated to get insights about the participants' preferences and the log-files were analyzed to get information about how the subjects used the program and the visualizations.

Hypothesis 1 stated that the Small Multiples and the trace view would outperform the animated scatter plot when identifying clusters and their development over time. The results of the statistical tests did not confirm that assumption. I did not find a significant difference between the Small Multiple visualization and the animated scatter plot for time. Furthermore using the Trace view was less efficient than the Small Multiples. The split by task categories revealed similar results. Regarding the two task categories Value and Time the participants were significantly slower if they used the Trace view.

Hypothesis 1 was confirmed with respect to the correctness. The hypothesis assumed that all visualizations would perform equally and the resulting success rates were nearly the same for all interfaces and revealed for dataset 2 and 3 no significant difference.

Hypothesis 2 dealt with the identification of outliers and their development over time. I assumed that such outlier tasks can be done faster with the Small Multiples and the Trace view than with the animated scatter plot. I have to reject that assumption because the statistical tests showed no significant difference between the interfaces.

Regarding the correctness hypothesis 2 was right with the assumption that all visualizations would cause similar correctness results. Only considering dataset 2 I found a significant difference between the interfaces and therefore in most cases there was no significant difference.

Hypothesis 3 stated that analyzing data movements within clusters is better suited for the animated scatter plot or the Trace view. Although timing results showed that mean completion

times of tasks were lower for the animated scatter plot for all datasets, statistical tests only showed a significant difference regarding dataset 1. Dividing the data by task categories I also only got significant differences for dataset 1. Furthermore the participants did not achieve similar results with Trace view compared to the animated scatter plot. Therefore I reject hypothesis 3 considering the time.

The evaluation of the correctness results partly confirmed the assumptions of hypothesis 3. The success rates using the animated scatter plot were significantly higher than using the Small Multiples but the Trace view did not lead to significantly better results than the Small Multiples.

Hypothesis 4 assumed that analyzing and comparing trends of specific subjects can be solved faster and with higher accuracy with the animated scatter plot and the Trace view than with the Small Multiples. This hypothesis could be partly accepted. The animated scatter plot was significantly better than the Small Multiples in completion time and correctness. I had to reject the part of the hypothesis assuming that using the Trace view leads to better results than the Small Multiples. The statistical tests showed that there was no significant difference between Trace view and Small Multiples. Furthermore the Trace view was significantly slower and led to more errors compared to the animated scatter plot.

The evaluation of the questionnaire revealed interesting results. The participants somehow confirmed the assumptions of the hypothesis by specifying their favorite visualization for different kind of tasks. The question about the analysis of correlations and trends over time can be referred to hypothesis 3 and 4. Nearly every participant preferred the animated scatter plot for this kind of task which supports the hypotheses and the results of the statistical evaluation.

I assumed that identifying outliers and analyzing clusters is well suited for the Small Multiples. The results of the questionnaire indicate that this assumption is not wrong because more participants would use the Small Multiples for outlier-tasks than for tasks concerning trends. On the other hand two-third of the participants preferred the animated scatter plot instead of the Small Multiples indicating that animation is subjectively favored to find outliers and clusters.

Overall the evaluation showed that the animated scatter plot often outperformed the Small Multiples and the Trace view in time and correctness. I believe that the animated scatter plot is more comfortable to use than the other visualization techniques. My opinion is confirmed by the results of the questionnaire – the animated scatter plot is the favored visualization for all kind of tasks and exceeded the other visualizations in understandability and usability.

Furthermore the results of hypothesis 1 and 2 showed that using animation to get an overview of the data is not slower than using Small Multiple visualizations. This conclusion gives further evidence for the usefulness of animation in information visualization and disagrees with the results of Robertson et. al [23] who concluded that animation is not as useful for analysis tasks as Small Multiples or a visualization with traces. The animation performed better or at least equally to the other visualizations for all kind of tasks.

Another interesting insight is that the Trace view overall performed rather poorly. Nearly every participant was not able to use the traces efficiently to accomplish the tasks. I assumed that some tasks could be done better with the Trace view than with the Small Multiples because changes in the data over time are not visualized explicitly within the Small Multiples. But

the questionnaire showed that participants preferred the Small Multiples compared to the Trace view and statistical analysis showed that the Trace view was often less efficient than the Small Multiples. A difference between Trace view and Small Multiples was that the time interval of the Small Multiples was fixed to one year whereas it was changeable in the Trace view and set to three years by default. Therefore there was more data visualized within the Trace view and maybe increased the complexity. On the other side the pictures of the Trace view were larger than the pictures in the Small Multiple visualization if the time interval was longer than one year, which could compensate the larger amount of data in the pictures.

The experiences made with the automated evaluation process were all over positive. Except one person all participants did not have any problems with the evaluation framework and the upload of the log files. The advantage of that procedure was that the number of participants was not limited because of time circumstances. Furthermore the participants were not influenced by observers. They were able to choose their favorite place and time to accomplish the evaluation tasks. On the other side there could not be all influencing factors controlled, e.g. it was not possible to control the usage of a mouse and that participants were interrupted in the evaluation process by disturbing factors (i.e. noise).

7.1 Limitations

A limitation of the study is that the results of nearly every hypothesis were dataset-dependent. The split of the data into the three datasets decreased the number of samples and therefore limited the explanatory power of the results. Additionally the data was often dependent on the task category which led to another subgroup of samples, even smaller.

Every dataset was linked to a specified round, i.e. the order of the datasets was not systematically diversified (dataset 1 was always the first dataset, then came dataset 2 and finally dataset 3). Only the visualizations were randomly assigned to one of the datasets. This error in the study design maybe was one of the reasons for the dataset-dependency of the results. The order can have an impact on results because of learning effects, i.e. the participants accomplish tasks slower in the first round because they are not familiar with the program handling. On the other side the results indicate that the order caused not the strongest effect for the dataset-dependency. The datasets seemed to be not equally which could be reduced to the fact that the three datasets were created by manually adapting the values of the original data file. Although the datasets were controlled by some persons for equality the generation of the dataset was a subjective process and therefore maybe induced individual variances in the data.

The use of a real-world dataset had some advantages and disadvantages. The real-world economic and energy dataset provided a solid basis for the evaluation because the data covered a relatively long time interval where many values of different parameters were collected and the data also contained some interesting trends for some parameters. The advantage of such a dataset is that participants would work with realistic data which they could easily understand and handle. On the other side a synthetic dataset created by an algorithm may include data effects which participants do not understand and maybe confuse them during evaluation. But of course the trends of the original data values covered not all hypotheses and tasks. Therefore

the manual adaption was necessary. The advantage of synthetically created datasets is that they probably would have equal impacts on the tasks and the dataset dependencies of the results could be excluded.

The samples of the different orders (or datasets) were not equally distributed because of the random assignment of the visualizations to the datasets (e.g. dataset 2 was 13 times assigned to the animated scatter plot, but dataset 3 only 9 times). This also influenced the statistical tests and reduces the significance of the results.

Another limitation could be that the technique to split up one picture into several small pictures containing subgroups of the data, like it is used in the Small Multiples, are a quite unusual visualization technique whereas a scatter plot is a well known, often used technique. The problems with the Trace view could be also attributed to the Small Multiple concept. Furthermore the Trace view used in this evaluation is a quite complex visualization technique because traces are connected across the Small Multiple pictures. The traces also had the disadvantage that they often lie across each other and hinder the user to observe the development of several countries. This problem was also mentioned by the participants in the questionnaire. Another problem was that the directions of the traces were hard to observe and maybe also influenced the results.

Another problem was figured out by evaluating the log files and questionnaires. The tab panel was retracted and therefore not initially visible for the participants. Most of the subjects did not use the mark settings and the filters. Furthermore some persons stated that they missed the filter panel which I had presented in the introductory presentation. It seems that I did not make it clear that the size of the tab panel is adjustable and can be retracted. Maybe some participants would have been able to answer some questions better if they had known that fact.

7.2 Future Work

This section summarizes some ideas for TimeRider improvements and extensions collected during planning and implementation phase of this thesis. I will also make some suggestions for further research concerning the visualization techniques implemented in TimeRider.

TimeRider

The implemented coordinated multiple view provides the display of two coordinated visualizations at the same time. An extension of TimeRider where all three visualizations are placed next to each other would enlarge the possibilities for analyzing the data with different views. Furthermore it would be beneficial if it is possible to activate the trace mode separately for both visualizations (Small Multiples and animated scatter plot).

The evolvement of the traces during animation is visualized by the extension of the trace line. The end of these lines could be better highlighted by adding a kind of cone at the end, e.g. an empty circle. This extension could help to recognize the direction of the traces better. Some suggestions to improve the Trace view were also made by the participants of the study (see

Chapter 6), e.g. visualize the direction of the trace by a color gradient. Other possibilities to visualize the direction are the following: traces get thicker between two points (encoding the direction not the passed time), mark the ends of the lines with small arrows or with a kind of drop (cp. [15]).

If the trace-by-mouseover function is activated it would be helpful to provide the possibility that the whole trace is visualized (in the animated scatter plot) and not only the trace until the current time, i.e. a trace reaching in the future. The trace-by-mouseover function could be further extended by the possibility to let traces stay visible if the subject was selected, comparable to the traces in the Trendalyzer.¹

The Small Multiple visualization could be improved by the possibility to adjust offsets to define the start of the Small Multiples. For example if there is a radical change in the trend in the year 2000 the user should be able to set the offset in a way that 2000 is exactly between two Small Multiple pictures.

The following objectives are some general improvement ideas. Now it is possible to select subjects per mouse click. An improvement could be to enable selection by rectangular brushing or by a freely choose able form. This was also an aspect mentioned by one of the participants. Another idea is to display a name list of all selected subjects or to display the name next to the selected subject (like it is done in the Trendalyzer). Usually the name filter is the most important filter. Therefore should be the name filter on top of the filter list if there are several filters adjusted.

One general problem for analysis tasks concerning a lot of data is that marks often overlap each other. This effect increases if there is no proper mark size adjusted. Therefore an extension could provide the possibility to configure the normal mark size and the minimal and maximal mark sizes when a parameter is mapped to size.

An additional zoom button to set the focus on filtered or selected data could be a possible extension. Finally it would be nice to calculate and visualize the linear regression for the selected axis parameters.

Research

The significance of the study results is relatively low because of their dataset dependency. A follow-up study to approve the results is needed. In my opinion a between-subject evaluation could give further interesting insights of the pros and cons of the visualization techniques. The advantage of the between-subject design would be that there is only one dataset needed – all participants work with the same data and therefore under the same condition. With the within-subject study made during this thesis the user preferences and experiences were already investigated which is impossible with a between-subject design.

Another objective of a follow-up study could be to focus on tasks of specific task categories, i.e. Behavior, Time, Identifier and Value. The evaluation showed that the characterization by task types was not specific enough. The visualization techniques may support tasks of different task categories differently.

¹ see <http://www.gapminder.org/world/>, last accessed on February the 14th, 2012

The evaluation carried out during this thesis analyzed the capabilities of the visualizations to provide users by solving tasks based on yearly recorded data. The characteristic of the animated scatter plot to visualize irregularly sampled data and the data wear was left out. Therefore another possible follow-up study could compare the implemented visualization techniques regarding their performance with irregularly sampled data.

Another focus could be on the huge possibilities of the coordinated multiple view for getting new insights in the data. The benefits of TimeRider regarding the coordinated multiple view were not tested within this thesis. A qualitative study would be necessary to figure out, how users interact with the system if all views are visible and the user has to solve analysis tasks concerning all visualizations.

Bibliography

- [1] JAXB – java architecture for XML binding. <http://www.oracle.com/technetwork/articles/javase/index-140168.html>. Accessed: 2012-03-05.
- [2] OECD regional statistics explorer. <http://stats.oecd.org/OECDregionalstatistics/>. Accessed: 2012-02-14.
- [3] R – statistic program. <http://www.r-project.org/>. Accessed: 2012-04-01.
- [4] Wolfgang Aigner, Kainz, Ma, and Silvia Miksch. Bertin was right: An empirical evaluation of indexing to compare multivariate time-series data using line plots. *Computer Graphics Forum*, 30(1):215–228, 2011.
- [5] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data*. Springer, London, 2011.
- [6] Nataldia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, 2006.
- [7] Daniel Archambault, Helen C. Purchase, and Bruno Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, 2011.
- [8] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces (AVI'00)*, pages 110–119. ACM, 2000.
- [9] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, 1999.
- [10] Sheelagh Carpendale. Evaluating information visualizations. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, number 4950 in LNCS, pages 19–45. Springer, Berlin, 2008.
- [11] Stephen Few. Visualizing change: An innovation in time-series analysis. White paper, SAS, 2007. <http://www.sas.com/reg/wp/corp/3774>, Accessed: 2012-04-21.

- [12] Zach Gemignani. Better know a visualization: Small multiples. <http://www.juiceanalytics.com/writing/better-know-visualization-small-multiples/>, April 2010. Accessed: 2012-03-28.
- [13] Amy L. Griffin, Alan M. MacEachren, Frank Hardisty, Erik Steiner, and Bonan Li. A comparison of animated with static small multiple maps for visually identifying space-time clusters. *Annals of the Association of American Geographers*, 96(4):740–753, 2006.
- [14] Stephan Hoffmann. Empirical evaluation of a visualization technique with semantic zoom. Master’s thesis, Technical University of Vienna, Faculty of Informatics, 2011.
- [15] D. Holten, P. Isenberg, J.J. van Wijk, and J. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *Pacific Visualization Symposium (PacificVis)*, 2011 IEEE, pages 195–202, march 2011.
- [16] Jonathan Lazar, Jinjuan Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction*. Wiley, 2010.
- [17] Barbara Neubauer. Visualizing patient cohorts by animated scatter plots. <http://ieg.ifs.tuwien.ac.at/projects/timerider/>, 2010. Accessed: 2012-03-28.
- [18] Chris North and Ben Shneiderman. A taxonomy of multiple window coordinations. Technical Report HCIL-97-18, University of Maryland, Dep. of Computer Science, College Park, MD 20742 USA, 1997.
- [19] Chris North and Ben Shneiderman. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies*, 53(5):715–739, November 2000.
- [20] Catherine Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI’04)*, pages 109–116, Gallipoli, Italy, 2004. ACM.
- [21] Alexander Rind, Wolfgang Aigner, Silvia Miksch, Sylvia Wiltner, Margit Pohl, Felix Drexler, Barbara Neubauer, and Nikolaus Suchy. Visually exploring multivariate trends in patient cohorts using animated scatter plots. In *Ergonomics and Health Aspects of Work with Computers*, volume 6779 of *Lecture Notes in Computer Science*, pages 139–148. Springer Berlin / Heidelberg, 2011.
- [22] Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV’07)*, pages 61–71, 2007.
- [23] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

- [24] Hans Rosling. Gapminder: Liberating the x-axis from the burden of time. *Statistical Computing and Graphics Newsletter*, 20(1):4–7, 2009.
- [25] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.
- [26] Barbara Tversky, Julie B. Morrison, and Mireille Betrancourt. Animation: can it facilitate? *IEEE Transactions on Visualization and Computer Graphics*, 57(4):247–262, 2002.
- [27] Howard Wainer. *Visual Revelations: Graphical Tales of Fate and Deception from Napoleon Bonaparte to Ross Perot*. Copernicus:New York, 1997.

List of Figures

1.1	Line plot example	2
1.2	Scatter plot example	2
2.1	Small Multiples visualization example presenting diamond data	9
2.2	Trendaylzer	11
2.3	OECD regional statistics explorer	16
3.1	Previous version of the TimeRider prototype	25
3.2	Example describing the animation in the animated scatter plot	26
3.3	Legend for the animation transparencies.	26
3.4	Trace view modes	27
3.5	Visualization of selected patients in the trace mode of TimeRider	28
3.6	Visualization of selected patients without traces in TimeRider	28
3.7	Animated scatter plot with synchronized data	29
4.1	Mockup with superimposed line plots (version 1)	36
4.2	Mockup with superimposed line plots (version 2)	38
4.3	Mockup of the Small Multiple visualization with superimposed line plots	39
4.4	Mockup with juxtaposed line plots	40
4.5	Mockup of the Small Multiple visualization with juxtaposed line plots	41
4.6	Mockup of the final user interface	42
4.7	Mockup of the final user interface with Small Multiples	43
5.1	Screenshot of the extended TimeRider program	46
5.2	Class diagram demonstrating classes responsible for loading data files	48
5.3	Schemata of the MainFrame with integrated panels	50
5.4	Class diagram of the axis configuration	51
5.5	Class diagram of classes responsible for adjusting mark settings	52
5.6	Class diagram of filter related classes	53
5.7	Class diagram of classes responsible for the animated scatter plot	54
5.8	Linkage between Examination objects	55
5.9	Class diagram of the classes necessary for the Small Multiple visualization	59
6.1	The evaluation window with the animated scatter plot.	64

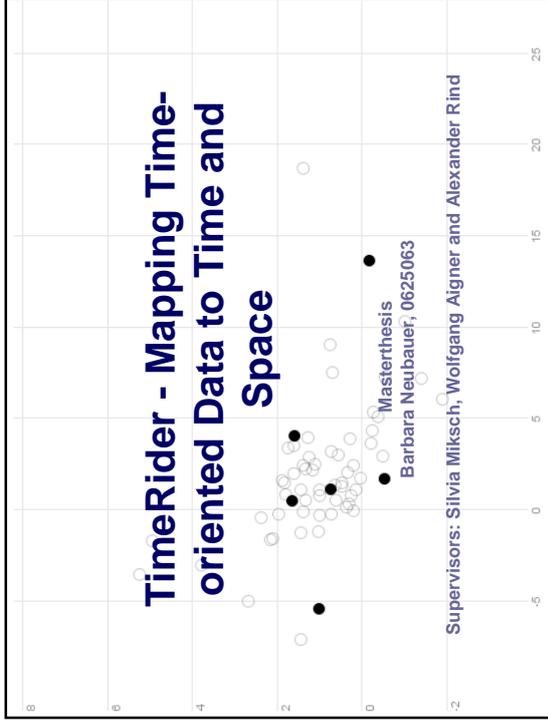
6.2	The evaluation windows with the Small Multiples.	64
6.3	The evaluation window with the Trace view.	65
6.4	Timing Boxplots of H1 for every dataset.	77
6.5	Boxplots for H1 of task category Identifier for all datasets.	78
6.6	Boxplots for H1 of task category Behavior for all datasets.	79
6.7	Boxplot for H1 of task category Value for all datasets.	79
6.8	Boxplots for H1 of taskcategory Time for dataset 1 and dataset 3.	80
6.9	Success rates for tasks of H1 split by datasets.	80
6.10	Timing Boxplots of H2 for every dataset.	82
6.11	Boxplots for H2 of task category Time.	83
6.12	Boxplots for H2 of task category Value.	83
6.13	Boxplots for H2 of task category Identifier.	83
6.14	Success rates for tasks of H2 split by datasets.	84
6.15	Timing Boxplots of H3 for every dataset.	85
6.16	Boxplots for H3 of taskcategory Time.	86
6.17	Boxplots for H3 of taskcategory Behavior.	87
6.18	Success rates for tasks of H3 split by datasets.	87
6.19	Timing results separated by interface	89
6.20	Success rates for tasks of H4 for the different interfaces.	90
6.21	Calculated grading of the visualizations for understandability and usability.	93
B.1	A screen shot of the questionnaire presented within the program.	117

List of Tables

2.1	Summary of four studies concerning static and dynamic visualizations.	13
3.1	Time-oriented parameters collected for diabetes patients	23
3.2	The requirements on a tool providing a visualization for medical data, divided by <i>tasks, data</i> and <i>users</i>	24
5.1	Abstract of the features of TimeRider.	46
6.1	Independent and dependent variables of the experiment.	67
6.2	The distribution of the tasks to task types, categories and hypothesis.	70
6.3	Tasks of hypothesis 1.	71
6.4	Tasks of hypothesis 2.	72
6.5	Tasks of hypothesis 3.	73
6.6	Tasks of hypothesis 4.	74
6.7	Descriptive measures for timings of hypothesis 1.	77
6.8	Descriptive measures for success rates of hypothesis 1.	81
6.9	Descriptive measures for timings of hypothesis 2.	82
6.10	Descriptive measures for success rates of hypothesis 2.	84
6.11	Descriptive measures for timings of hypothesis 2.	86
6.12	Descriptive measures for success rates of hypothesis 3.	88
6.13	Descriptive measures for timings of hypothesis 4.	89
6.14	Descriptive measures for success rates of hypothesis 4.	90
6.15	Summary of the evaluation results for every hypothesis.	91
6.16	Overview of the questionnaire results.	92
6.17	This table summarizes interactions with the program which often occurred during evaluation.	94

Introductory Presentation

The following presentation slides were used to introduce the TimeRider program and the evaluation process to the participants of the study.



Visualizing time-oriented data

- 3 different visualization techniques:
 - Animated Scatterplot
 - Small Multiples
 - Traces

Barbara Neubauer Masterthesis

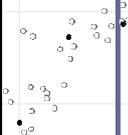
Data

- Economic and energy data (e.g. CO2 emissions or energy use) of different countries
- Annually recorded
- Manually adapted

Barbara Neubauer Masterthesis

Animated Scatterplot

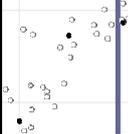
Barbara Neubauer Masterthesis



Animated Scatterplot

- Scatterplot with moving points to visualize the development of the parameters over time
- „Video-player-like“ controls for the animation
- different granularities for animation: days, weeks, months, years
- Pan & Zoom

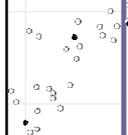
Barbara Neubauer Masterthesis 5



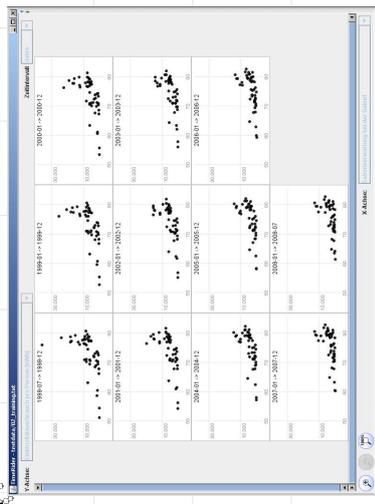
Animation

- Record points move across the plot according to the parameter values.
- A „moving point“ is always the next record of a country
 - Example: the country has 2 records on 31.6.2008 and 31.6.2009
 - current date = 31.6.2008 → the first record is visible
 - current date = 1.7.2008 → first record gets invisible, second record gets visible on the place where the first record disappeared
 - Time till 31.6.2009: second record moves across the plot to its destination place
 - Current date: 31.6.2009 → record reached destination place

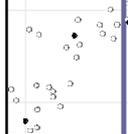
Barbara Neubauer Masterthesis 6



Small Multiples



Barbara Neubauer Masterthesis 7



Small Multiples

- Small Multiples are small pictures containing time segments
- Time interval = years: defines the number of Small Multiples → every Small Multiple contains the records of one year
- Pan & Zoom

Barbara Neubauer Masterthesis 8

Traces

Barbara Neubauer

Masterthesis

9

Traces

- Small Multiple visualization with traces
- Traces connect two records with each other
- Traces end with thicker lines → start and end point can be differentiated
- Traces are connected across the Small Multiples, but they always start with a thin line in every Small Multiple!
- Possible Time intervals: 1,2,3,4 or 5 years

Barbara Neubauer

Masterthesis

10

Traces

- The trace on the left side is split up into several parts within the Small Multiples

Barbara Neubauer

Masterthesis

11

Filters and Mark settings

- Filters help to specify which records should be visible, e.g. only countries with a population size smaller than 3 Mio.
- Mark settings: the colour, the size and the shape can be mapped to logical parameters, e.g. all EU countries have green points, all other countries have purple points

Barbara Neubauer

Masterthesis

12

Evaluation

- Aim: find out which visualization is better for which tasks, which visualization is better to get the right answers to the tasks
- It is not a test of your abilities, it is a test of the visualizations!
BUT: do the tasks carefully and as quick as possible!
 → 5 **extra points** for the 5 students who solve the tasks best (execution time AND correctness counts!)

Barbara Neubauer Masterthesis 13

Evaluation

- 22 tasks for every visualization → 3 sessions
- Every task has to be done with every visualization with different datasets
- Short training session with 5 tasks before every evaluation session
- Execution time and answers are presented after training tasks (not for evaluation tasks!) → it is possible to do wrong answered tasks again
- Short questionnaire afterwards
- Duration: about 1,5h

Barbara Neubauer Masterthesis 14

Explanatory notes

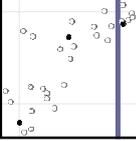
- Requirements:
 - Java 6 (Sun/Oracle, not OpenJDK!)
 - Mouse
 - Minimal Screen resolution: 1024x768 (the window size is 1024x768 – it can not be resized!)
- Try to minimize disturbing factors, e.g. switch off mobile phones
- Do not break within the evaluation otherwise you have to start again!
 Do not pause within a task execution, short break between tasks or sessions is possible.
- Read the task text before clicking OK → execution time is measured
- Do not cheat → we will find out!
- Test results and personal data will be treated as strictly confidential

Barbara Neubauer Masterthesis 15

Evaluation procedure

- Start Java Webstart in TUWEL
 - Download the (.jnlp-) file and start it manually
 - In most cases the evaluation journal will be created in the download folder – if not, look at the last message dialog of the evaluation → shows you the location of the evaluation journal
- Accept the security message!
- Type in your Matrikel-number and start evaluation
- Solve the tasks
- Answer questions in Questionnaire
- Zip the „evaluationJournal“ (Matrikelnr_evaluationJournal.zip) and upload it
- Hint: the location of the „evaluationJournal“ is shown in the last message dialog!

Barbara Neubauer Masterthesis 16



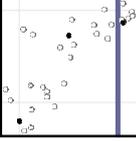
Datasets

- **ATTENTION:**
the datasets also contain generated data –
the values are not all based on realistic
data values.

Barbara Neubauer

Masterthesis

17



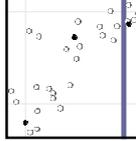
Problems

- If problems occur:
→ Post in TUWEL Forum:
„Help Forum User Study“

Barbara Neubauer

Masterthesis

18



Questions?

Barbara Neubauer

Masterthesis

19

Questionnaire

The questionnaire was used to collect data about the participants and about the participants' experiences with the visualizations. All questions had to be answered mandatory except the last one (question 4).

Question
Alter
Geschlecht (weiblich/männlich)
Farbfehlsichtigkeit (z.B. rot-grün Schwäche)?
1) Mit welcher Visualisierung konnten Sie folgende Aufgaben jeweils am besten beantworten: a) Analyse von Zusammenhängen über Zeit? (Animated scatter plot/Small Multiples/Traces) b) Analyse von Ausreißern über Zeit? (Animated scatter plot/Small Multiples/Traces) c) Analyse von Clustern über Zeit? (Animated scatter plot/Small Multiples/Traces)
2) Wenn sie nur eine der drei Visualisierungen zur Auswahl hätten, welche würden sie zur Beantwortung aller Fragen benutzen? (Animated scatter plot/Small Multiples/Traces)

3) Bewerten sie bitte die Verständlichkeit und Benutzbarkeit der Visualisierungen:

a) Animierter Scatterplot

- Einfach zu verstehen (stimme sehr zu/stimme zu/neutral/stimme nicht zu/stimme gar nicht zu)
- Einfach zu benutzen (stimme sehr zu/stimme zu/neutral/stimme nicht zu/stimme gar nicht zu)

b) Small Multiples

- Einfach zu verstehen (stimme sehr zu/stimme zu/neutral/stimme nicht zu/stimme gar nicht zu)
- Einfach zu benutzen (stimme sehr zu/stimme zu/neutral/stimme nicht zu/stimme gar nicht zu)

c) Traces

- Einfach zu verstehen (stimme sehr zu/stimme zu/neutral/stimme nicht zu/stimme gar nicht zu)
- Einfach zu benutzen (stimme sehr zu/stimme zu/neutral/stimme nicht zu/stimme gar nicht zu)

4) Sonstige Anmerkungen, Schwierigkeiten, Positives, Negatives, Verbesserungsvorschläge, Unverständliches, etc.:

Fragebogen [Minimieren] [Maximieren] [Schließen]

Bitte füllen Sie unten stehenden Fragebogen aus. Alle Angaben werden vertraulich behandelt.

ANGABEN ZUR PERSON

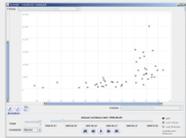
Geschlecht: weiblich weiblich

Alter: Jahre

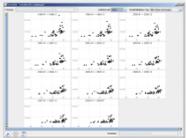
Farbfehlsichtigkeit (z.B. rot-grün Schwäche)? Ja Nein

FRAGEN ZUR EVALUIERUNG

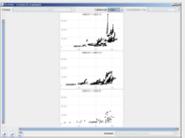
Animierter Scatterplot



Small Multiples



Traces



1) Mit welcher Visualisierung konnten Sie folgende Aufgaben jeweils am besten beantworten:

a) Analyse von Zusammenhängen über Zeit bzw. Analyse von Trends? Animierter Scatterplot Small Multiples Traces

b) Identifikation von Ausreißern über Zeit? Animierter Scatterplot Small Multiples Traces

c) Analyse von Clustern über Zeit? Animierter Scatterplot Small Multiples Traces

2) Wenn sie nur eine der drei Visualisierungen zur Auswahl hätten, welche würden sie zur Beantwortung aller Fragen benutzen?

Animierter Scatterplot Small Multiples Traces

3) Bewerten sie bitte die Verständlichkeit und Benutzbarkeit der Visualisierungen:

a) Animierter Scatterplot

Einfach zu verstehen: stimme sehr zu stimme zu neutral stimme nicht zu stimme gar nicht zu

Einfach zu bedienen: stimme sehr zu stimme zu neutral stimme nicht zu stimme gar nicht zu

b) Small Multiples

Einfach zu verstehen: stimme sehr zu stimme zu neutral stimme nicht zu stimme gar nicht zu

Einfach zu bedienen: stimme sehr zu stimme zu neutral stimme nicht zu stimme gar nicht zu

c) Traces

Einfach zu verstehen: stimme sehr zu stimme zu neutral stimme nicht zu stimme gar nicht zu

Einfach zu bedienen: stimme sehr zu stimme zu neutral stimme nicht zu stimme gar nicht zu

4) Sonstige Anmerkungen, Schwierigkeiten, Positives, Negatives, Verbesserungsvorschläge, Unverständliches, etc. (optional):

Figure B.1: A screen shot of the questionnaire presented within the program.

Training Tasks

The descriptions for the training tasks were equal for all visualizations, but they differed in their task instructions (i.e. hints to accomplish the task). The instructions for the animated scatter plot are labeled with “SC”, the instructions for the Small Multiples with “SM” and instructions for the trace view with “T”. The possible answers are listed in the right column as well as the corresponding correct answers for the different visualizations (also labeled with “SC”, “SM” and “T”).

Task type	Task description amd instructions	Possible and correct answers
Oulier	<p>Description</p> <p>Selektiere im Plot jenes Land, dessen Werte, über den gesamten Zeitraum weit abseits der Werte der anderen Länder liegen. (Wasserkrafterzeugung und Erdgaserzeugung)</p> <p>Instruction</p> <p>SC/SM/T: Hinweise: Ein Land kann selektiert werden, indem auf das Land geklickt wird. Durch nochmaliges Klicken auf das Land bzw. durch Klick in den Hintergrund wird die Selektion wieder aufgehoben.</p>	<p><i>[selection task]</i></p> <p>SC: Norwegen</p> <p>SM: Kanada</p> <p>T: Brasilien</p>

Trend	<p>Description</p> <p>SC: Bei welchem der folgenden drei Länder (Australien, Iran und Kasachstan) kann eine ähnliche Entwicklung wie in Russland beobachtet werden (CO2 Emissionen und Energieerzeugung)? Selektiere das entsprechende Land in der Visualisierung.</p> <p>Instruction</p> <p>SC/SM/T: Hinweise: Bei Bedarf muss in den Plot hineingezoomt werden! Russland ist farblich hervorgehoben. Ein Land kann selektiert werden, indem auf das Land geklickt wird. Die Selektion kann durch nochmaliges Klicken auf das Land bzw. durch Klick in den Hintergrund aufgehoben werden.</p>	<p><i>[selection task]</i></p> <p>SC: Kasachstan SM: Australien T: Iran</p>
Outlier	<p>Description</p> <p>ölerzeugung vs CO2 Emissionen: Welche 4 Länder liegen über den gesamten Zeitraum immer ausserhalb des Bereichs in dem die meisten Länder liegen?</p> <p>Instruction</p> <p>SC: Hinweise: Benutze die Animation um den zeitlichen Verlauf der Parameter zu beobachten. Ein Land kann selektiert werden, indem auf das Land geklickt wird. Durch Halten der Str-Taste können mehrere Länder selektiert werden.</p> <p>SM: Hinweise: Ein Land kann selektiert werden, indem auf das das Land geklickt wird. Durch Halten der Str-Taste können mehrere Länder selektiert werden.</p> <p>T: Hinweise: Bei Bedarf kann das Zeitintervall auf 1, 2, 4 oder 5 Jahre umgestellt werden. Ein Land kann selektiert werden, indem auf das das Land geklickt wird. Durch Halten der Str-Taste können mehrere Länder selektiert werden.</p>	<p><i>[selection task]</i></p> <p>SC: Kuwait, Vereinigte Arabische Emirate, Saudi Arabien, Norwegen SM: Kuwait, Ukraine, Russland, Norwegen T: Russland, Ukraine, Kuwait, Irak</p>

Cluster	<p>Description</p> <p>Wieviele Cluster sind in den meisten Jahren für die Parameter Import und Export sichtbar?</p> <p>Instruction</p> <p>SC: Hinweise: Benutze die Animation um den zeitlichen Verlauf der Parameter zu beobachten. Ein Cluster besteht aus mindestens 5 Ländern!</p> <p>SM: Hinweise: Ein Cluster besteht aus mindestens 5 Ländern!</p> <p>T: Hinweise: Bei Bedarf kann das Zeitintervall auf 1, 2, 4 oder 5 Jahre umgestellt werden. Ein Cluster besteht aus mindestens 5 Ländern!</p>	<p>1,2,3 oder 4</p> <p>SC: 2</p> <p>SM: 3</p> <p>T: 2</p>
---------	--	--

Evaluation Tasks

The evaluation tasks in German. The description, possible and correct answers depend on the dataset (D1=dataset1, D2=dataset2, D3=dataset3).

#Task	H	Task type	Task description amd instructions	Possible and correct answers
1	H1	Trend	<p><i>Description</i> D1/D2/D3: Elektrizitätserzeugung vs Erdgaserzeugung: Wie kann der Trend über alle Jahre, für die Mehrheit der Länder beschrieben werden?</p> <p><i>Instruction</i> D1/D2/D3: Antworten beziehen sich auf beide Parameter in folgender Reihenfolge: Elektrizitätserzeugung - Erdgaserzeugung</p>	<p><i>Possible</i> D1/D2/D3: steigend – steigend steigend – fallend fallend – steigend fallend – fallend</p> <p><i>Correct</i> D1: fallend – fallend D2: steigend – steigend D3: steigend – fallend</p>

2	H3	Trend	<p>Description D1/D2/D3: Import vs Export: Entwickeln sich die Mehrheit der Länder, während des gesamten Zeitraums, immer in ähnlicher Weise?</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: ja, nein oder weiß nicht</p> <p>Correct D1: ja D2: nein D3: ja</p>
3	H1	Identifizier	<p>Description D1/D2/D3: Betrachte die Anzahl an Datencluster im Jahr 1998 (Energieerzeugung und Elektrizitätsverbrauch). Ab welchem darauf folgendem Jahr kann eine veränderte Anzahl an Datenclustern beobachtet werden?</p> <p>Instruction D1/D2/D3: Hinweis: ein Cluster besteht aus mindestens 5 Ländern!</p>	<p>Possible D1/D2/D3: 1999, 2001, 2003, 2005</p> <p>Correct D1: 2003 D2: 2005 D3: 2001</p>
4	H4	Trend	<p>Description D1/D2/D3: Welches Land weist eine positive Korrelation der Parameter CO2 Emissionen und Energieverbrauch auf? Indien oder Ungarn?</p> <p>Instruction D1/D2/D3: Selektiere das entsprechende Land im Plot!</p>	<p>Possible D1/D2/D3: [selection task]</p> <p>Correct D1: Indien D2: Kuwait D3: Indien</p>

5	H4	Trend	<p>Description D1/D2/D3: CO2 Emissionen vs Energieerzeugung: Bei welchem der folgenden drei Länder (Chile, Ecuador oder Kuwait) kann eine ähnliche Entwicklung wie in China, beobachtet werden?</p> <p>Instruction D1/D2/D3: Bei Bedarf muss in den Plot hineingezoomt werden! Hinweis: China ist farblich hervorgehoben.</p>	<p>Possible D1/D2/D3: <i>[selection task]</i></p> <p>Correct D1: Chile D2: Ecuador D3: Kuwait</p>
6	H4	Trend	<p>Description D1/D2/D3: Import vs Lebenserwartung: Wähle im Plot jenes Land (von den gegebenen fünf Ländern) aus, das sich über den gesamten Zeitraum nicht immer in der gleichen Weise entwickelt wie die anderen vier Länder?</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: <i>[selection task]</i></p> <p>Correct D1: Vereinigte Arabische Emirate D2: Kuwait D3: Dänemark</p>
7	H4	Trend	<p>Description D1/D2/D3: Wähle im Plot jenes Land aus, das einen starken Anstieg in der Erdgaserzeugung zwischen 1998 und 2008 hatte, während sich die Population kaum veränderte und immer sehr klein war.</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: <i>[selection task]</i></p> <p>Correct D1: Norwegen D2: Kuwait D3: Vereinigte Arabische Emirate</p>

8	H3	Trend	<p>Description D1/D2/D3: CO2 Emissionen vs Energieerzeugung: Wie kann der allgemeine Trend für die meisten Länder beschrieben werden?</p> <p>Instruction D1/D2/D3: Antworten beziehen sich auf beide Parameter in folgender Reihenfolge: CO2 Emissionen - Energieerzeugung</p>	<p>Possible D1/D2/D3: steigend – steigend steigend – fallend fallend – steigend fallend – fallend kein allgemeiner Trend</p> <p>Correct D1: kein allgemeiner Trend D2: steigend - steigend D3: fallend - steigend</p>
9	H3	Trend	<p>Description D1/D2/D3: Lebenserwartung vs Import: Entwickeln sich die Mehrheit der Länder zwischen 1998 und 2008 immer in ähnlicher Weise?</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: ja, nein oder weiß nicht</p> <p>Correct D1: ja D2: nein D3: ja</p>

10	H2	Outlier	<p>Description</p> <p>D1: Elektrizitätserzeugung vs Erdgaserzeugung: Ab welchem Jahr liegen die Werte beider Länder Norwegen und Vereinigte Arabische Emirate weit abseits der Parameterwerte der anderen Länder?</p> <p>D2: Elektrizitätserzeugung vs Erdgaserzeugung: Ab welchem Jahr liegen die Werte beider Länder Ukraine und Polen weit abseits der Parameterwerte der anderen Länder?</p> <p>D3: Elektrizitätserzeugung vs Erdgaserzeugung: Ab welchem Jahr liegen die Werte beider Länder Mexiko und Türkei weit abseits der Parameterwerte der anderen Länder?</p> <p>Instruction</p> <p>D1/D2/D3: -</p>	<p>Possible</p> <p>D1: 1999, 2000, 2002, 2004</p> <p>D2/D3: 1998, 2000, 2002, 2004</p> <p>Correct</p> <p>D1: 2002</p> <p>D2: 2000</p> <p>D3: 1998</p>
11	H2	Outlier	<p>Description</p> <p>D1/D2/D3: Erdgaserzeugung vs Energieverbrauch: Selektiere jenes Land, das über alle Jahre ein Ausreisser ist?</p> <p>Instruction</p> <p>D1/D2/D3: -</p>	<p>Possible</p> <p>D1/D2/D3: [selection task]</p> <p>Correct</p> <p>D1: Vereinigte Arabische Emirate</p> <p>D2: Ukraine</p> <p>D3: Vereinigte Arabische Emirate</p>
12	H2	Outlier	<p>Description</p> <p>D1/D2/D3: Import vs Export: Selektiere jenes Land, dessen Werte in den meisten Jahren weit abseits der Werte der anderen Länder liegen.</p> <p>Instruction</p> <p>D1/D2/D3: -</p>	<p>Possible</p> <p>D1/D2/D3: [selection task]</p> <p>Correct</p> <p>D1: Luxemburg</p> <p>D2: Malaysia</p> <p>D3: Libanon</p>

13	H2	Outlier	<p>Description D1/D2/D3: Elektrizitätsverbrauch vs Wasserkrafterzeugung: Wie viele Länder gibt es, deren Werte über den gesamten Zeitraum abseits der anderen Länder liegen?</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: 1, 2, 3, keines</p> <p>Correct D1: 2 D2: 1 D3: 1</p>
14	H2	Outlier	<p>Description D1/D2/D3: Nuklearerzeugung vs Energieverbrauch: Existiert ein Land dessen Werte über den gesamten Zeitraum, weit entfernt von den Werten der anderen Länder liegen?</p> <p>Instruction D1/D2/D3: Existiert ein signifikanter Ausreisser über alle Jahre?</p>	<p>Possible D1/D2/D3: ja, nein</p> <p>Correct D1: nein D2: ja D3: nein</p>
15	H3	Trend	<p>Description D1/D2/D3: In welchem Zeitintervall sind in Russland die Wasserkrafterzeugung gefallen und die Nuklearerzeugung gestiegen?</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: 1998-2000 2002-2003 2000-2004 2005-2008</p> <p>Correct D1: 2000-2004 D2: 1998-2000 D3: 2005-2008</p>

16	H3	Trend	<p>Description D1/D2/D3: In welchem Zeitintervall fielen die Werte beider Parameter in den meisten Ländern? (Import und Export)</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1: 1998-2000 2000-2003 2002-2004 2004-2008 weiß nicht</p> <p>D2/D3: 1998-2000 2001-2003 2002-2004 2004-2008 weiß nicht</p> <p>Correct D1: 2000-2003 D2: 2004-2008 D3: 1998-2003</p>
17	H1	Identifizier	<p>Description D1/D2/D3: Wieviele Cluster sind im Jahr 1998 für die Parameter Energieerzeugung und Energieverbrauch sichtbar?</p> <p>Instruction D1/D2/D3: Hinweis: ein Cluster besteht aus mindestens 5 Ländern!</p>	<p>Possible D1/D2/D3: 1, 2, 3, 4</p> <p>Correct D1: 2 D2: 2 D3: 2</p>
18	H1	Identifizier	<p>Description D1/D2/D3: Wieviele Cluster sind in den meisten Jahren für die Parameter Ölerzeugung und Elektrizitätserzeugung sichtbar?</p> <p>Instruction D1/D2/D3: Hinweis: ein Cluster besteht aus mindestens 5 Ländern!</p>	<p>Possible D1/D2/D3: 1, 2, 3, 4</p> <p>Correct D1: 3 D2: 1 D3: 2</p>

19	H1	Identifizier	<p>Description D1/D2/D3: Welche drei Länder bilden einen klar separierten Cluster über den gesamten Zeitraum? (Population und Elektrizitätsverbrauch)</p> <p>Instruction D1/D2/D3: Hinweis: mehrere Länder können durch Halten der Str-Taste ausgewählt werden.</p>	<p>Possible D1/D2/D3: <i>[selection task]</i></p> <p>Correct D1: Indien, Iran, China D2: Vereinigte Staaten, Dänemark, Kanada D3: Indien, Iran, China</p>
20	H1	Identifizier	<p>Description D1/D2/D3: In welchem Zeitintervall sind zwei Cluster für die Parameter Wasserkrafterzeugung und Export sichtbar?</p> <p>Instruction D1/D2/D3: Hinweis: Ein Cluster besteht aus mindestens 5 Ländern!</p>	<p>Possible D1/D2/D3: 2001-2003 2002-2004 1998-2008 nie</p> <p>Correct D1: nie D2: 2001-2003 D3: 2002-2004</p>
21	H2	Outlier	<p>Description D1/D2/D3: Import vs Wasserkrafterzeugung: In welchem Zeitintervall lagen die Werte von Brasilien abseits der Werte der anderen Länder?</p> <p>Instruction D1/D2/D3: -</p>	<p>Possible D1/D2/D3: 1998-2000 2000-2003 2002-2004 nie</p> <p>Correct D1: nie D2: 2000-2003 D3: 2002-2004</p>

22	H1	Identifizier	<p>Description</p> <p>D1/D2/D3: Welche drei Länder bilden über den gesamten Zeitraum einen klar separierten Cluster? (Energieverbrauch und Lebenserwartung)</p> <p>Instruction</p> <p>D1/D2/D3: Hinweis: mehrere Länder können durch Halten der Str-Taste ausgewählt werden.</p>	<p>Possible</p> <p>D1/D2/D3: [selection task]</p> <p>Correct</p> <p>D1: Griechenland, Chile, Costa Rica</p> <p>D2: Serbien, Spanien, Jamaica</p> <p>D3: Belgien, Spanien, Chile</p>
----	----	--------------	--	---
