Habilitationsschrift

Interactive Visualization and Data Analysis: Visual Analytics With a Focus on Time

ausgeführt in den Jahren 2006–2013
zum Zwecke der Erlangung der *venia docendi* (Lehrbefugnis)
im Habilitationsfach "Praktische Informatik"

eingereicht im Januar 2013
an der Technischen Universität Wien
Faktultät für Informatik

von

Wolfgang Aigner
Weinbergstr. 19f, 3512 Mautern/Donau
geboren am 23. Mai 1977 in Steyr
aigner@ifs.tuwien.ac.at

Wien, 16. April 2013

Technische Universität Wien
A-1040 Wien · Karlsplatz 13 · Tel. +43-1-58801-0 · www.tuwien.ac.at

visualization
data
time
visual
methods
task
time-oriented
evaluation
information
analysis
tasks
different
qualitative
Analytics
user
temporal
design
interactive
users
representation
using
work
Aigner
interaction
quantitative
used
test
one
two
techniques
completion
indexing
also
number
Fig
Miksch
comparison
contingency
experiment
types
variables
based
view

Abstract

The increasing amounts of data available offer great opportunities to promote technological progress and business success in many domains. However, the possible ways of collecting and storing data are increasing at a faster rate than our ability to analyze data and use it for decision making. *Visual Analytics (VA)* is an emerging area of research that addresses this challenge by combining the outstanding visual perception and reasoning capabilities of humans with the strengths of automated data analysis of computers. Thus, it focuses on facilitating the exploration and understanding of large and complex datasets by intertwining interactive visualization, automated data analysis, and human-computer interaction.

Time is an exceptional dimension that is common to many application domains such as medicine, engineering, science, or business. Due to the distinct characteristics of time, appropriate visual and automated methods are required to explore trends, patterns, and relationships.

The main objective of the research presented in this thesis is to account for the complex structure of time in Visual Analytics. From a methodological perspective the research is driven by the principles of human-centered design and development. More specifically, the presented research describes contributions on *intertwining visual and automated methods* (novel methods that closely integrate both), *managing and analyzing specialized data types* (time-oriented and categorical data), *guiding the designers of VA systems* (methodological framework and layout of the design space), *scalability of VA methods* (analysis of large contingency tables), *infrastructure* (software library to support visualization evaluation), and *evaluation* (empirical evidence on the effectiveness of visualization methods). These results are significant contributions to a number of research challenges and are applicable in a variety of domains by allowing a more effective analysis of time-oriented and categorical data.

Zusammenfassung

Menge und Komplexität der uns zur Verfügung stehenden Daten steigen in immer größerem Ausmaß. Obwohl diese Datenfülle völlig neue Möglichkeiten sowohl für den technischen Fortschritt als auch den wirtschaftlichen Erfolg öffnet, können die Methoden um Daten zu analysieren und Entscheidungsprozesse zu unterstützen nicht mit dem rasanten Datenzuwachs Schritt halten. *Visual Analytics (VA)* stellt sich als aufstrebende Forschungsdisziplin dieser Herausforderung. Die Kernidee ist die hervorragenden kognitiven Fähigkeiten des Menschen im Umgang mit visuellen Sinneseindrücken mit den enormen automatischen Verarbeitungsmöglichkeiten von Computersystemen zu verknüpfen. Dabei sollen große und komplexe Datenmengen verständlich gemacht und die Gewinnung neuer Erkenntnisse erleichtert werden. Im Mittelpunkt steht also eine symbiotische Kombination von interaktiver Visualisierung, automatischer Datenanalyse und Mensch-Maschine-Interaktion.

Zeit als komplexe Datendimension spielt in vielen Anwendungsgebieten eine wichtige Rolle. Deren inhärente Charakteristika machen allerdings spezielle Methoden zur Visualisierung und Datenanalyse notwendig, um das Erkennen von Trends, Mustern und Beziehungen zu ermöglichen. Die vorliegende Arbeit hat sich zum Ziel gesetzt, der komplexen Struktur von Zeit in Visual Analytics Rechnung zu tragen und dabei einen methodisch nutzerInnenzentrierten Ansatz zu verfolgen.

Konkret trägt die Forschung *zur Verflechtung von visuellen und automatischen Methoden* (neue Methoden, die beides integrieren), *dem Management und der Analyse von spezialisierten Datentypen* (zeitorientierte und kategorische Daten), *der Anleitung von Designern von VA Systemen* (methodisches Framework und Beschreibung der Gestaltungsmöglichkeiten), *der Skalierbarkeit von VA Methoden* (Analyse von großen Kontingenztabellen), *Infrastruktur* (Software Bibliothek zur Unterstützung der Evaluierung von Visualisierungen), sowie *Evaluierung* (empirische Belege über die Effektivität von Visualisierungsmethoden) bei. Die beschriebenen Ergebnisse leisten einen wertvollen Beitrag zu einer Reihe an wissenschaftlichen Herausforderungen in Visual Analytics. Sie ermöglichen eine effektivere Analyse von zeitorientierten und kategorischen Daten und können in vielen Anwendungsbereichen eingesetzt werden.

Contents

All the pieces are there – huge amounts of information, a great need to clearly and accurately portray them, and the physical means for doing so. What has been lacking is a broad understanding of how best to do it.

— Wainer [1997, p. 112]

CHAPTER 1

Introduction

This thesis presents a subset of the author's research work in the field of Visual Analytics carried out at Vienna University of Technology and Danube University Krems, Austria in the years 2006–2013. The current chapter provides a brief introduction to the field and highlights some major research challenges in order to provide the context for the articles presented in the following chapters (Section 1.1). Following this, the publications that constitute the body of this thesis are listed (Section 1.2) and briefly outlined along with an overview of the author's scientific contributions and his role in carrying out the research (Section 1.3).

1.1 BACKGROUND

One of the most significant challenges of our Information Age is to effectively harness the immense wealth of the ever-growing amount of data recorded and processed by modern information systems. During the last decades, capabilities to both generate and collect data have seen an explosive growth. Advances in scientific and business data collection (e.g., from remote sensors, from electronic health records, from network traffic, or from retail and production devices as well as from increasingly complex simulation systems) have generated a flood of data and information. Advances in data storage technology such as faster and cheaper storage devices with higher capacity, better database management systems, and data warehousing technology have allowed us to transform this data into "mountains" of stored data. These increasing amounts of data offer great opportunities to promote technological progress and business success. But we also face the challenge of getting lost in information that is irrelevant, inappropriately processed, or poorly presented [Keim et al., 2010].

Visual Analytics (VA) denotes "the science of analytical reasoning facilitated by interactive visual interfaces" [Thomas and Cook, 2005, p. 28] and addresses this challenge. It aims to make complex information structures more comprehensible, facilitate new insights, and enable knowledge discovery. It is an emerging discipline that aims to combine the outstanding visual perception and reasoning capabilities of humans with the strengths of automated data analysis, i.e., the enormous processing power of computers.

Analytical reasoning for real-world problem solving usually involves the analysis of huge amounts of heterogeneous, possibly incomplete, conflicting, inconsistent, and dynamic information. For this, human judgement is required to deal with ill-defined problem solving, to synthesize knowledge, and to make decisions based on complex data. Thus, a major tenet of Visual Analytics is that analytical reasoning is not a routine activity that can be automated completely [Wegner, 1997]. Instead it depends heavily on analysts' initiative and domain experience. Considering its current practice, Visual Analytics can be defined more specifically as the "[combination] of automated analysis techniques with interactive visualisations for an effective understanding, reasoning and decision making on the basis of very large and complex datasets" [Keim et al., 2010, p. 7]. Thus, the discipline puts its focus on the information discovery process and aims to enable the exploration and the understanding of large and complex datasets combining interactive visualization, automated data analysis, and human-computer interaction.

Visual Analytics is an inherently multi-disciplinary field that aims to combine the findings of various research areas such as human-computer interaction (HCI), usability engineering, cognitive and perceptual science, information visualization, scientific visualization, databases, data mining, statistics, knowledge discovery, data management, and knowledge representation. Application domains benefiting from Visual Analytics are for example health care, biotechnology, security & disaster management, environmental science, and climate research.

1.1.1 *Visualization*

Visual interfaces, especially visualizations, are high bandwidth gateways for the perception of structures, patterns, or connections hidden in the data. Visualization[1] can help to alleviate the problem of information overload by taking advantage of the powerful human perceptual system that is extremely efficient in processing visual input to make sense of data, explore complex information spaces, or spot patterns and relationships. It is an interdisciplinary field that is commonly defined as the "use of computer-supported, interactive,

[1] In the context of this thesis, *visualization* is used as an umbrella term. The related field of *Scientific Visualization* is usually defined as involving data with an inherent physical component. In this thesis, the focus is on abstract data, thus, on *Information Visualization*.

visual representations of data to amplify cognition" [Card et al., 1999, p. 6]. Visualization achieves this goal by [Card et al., 1999]

- increasing the memory and processing resources available to the user,
- reducing the search for information,
- using visual representations to enhance the detection of patterns,
- enabling perceptual inference operations,
- using perceptual attention mechanisms for monitoring, and
- encoding information in a manipulable medium.

Particularly the last aspect lends itself to the notion of interactivity as described next.

1.1.2 *Interactivity*

Although a static image might be worth a thousand words, true exploration and reasoning is subject to the highly dynamic processes of pondering problems from different points of view and twisting and turning on the path to discovery. It is the flexibility and fluidity of human thought processes that allow for deep understanding and novel insights, which in most cases cannot be detected straightforwardly using automated methods [Elmqvist et al., 2011]. Supporting these dynamic processes with visualization leads to next generation tools of thought that truly support human reasoning. Limiting the fluidity and flexibility of human analytical reasoning by poor design or lack of interactivity may introduce confusion and errors, while it limits the power of human reasoning considerably [Heer and Shneiderman, 2012]. Therefore, interactivity is one of the most important elements in the sensemaking process and is at the core of visualization [Spence, 2007]. Empirical studies like the one by Saraiya et al. [2006] confirm that users prefer inferior visual representations with interaction to superior non-interactive visual representations. Visual representations provide only an initial direction to the data and their meaning, while through the combination with appropriate interaction mechanisms, users gain insights into the data in a dynamic process. Interaction methods are essential for exploring the data as well as the parameter space. Moreover, it is important that these methods are designed according to the user's demands. Interacting directly with the visual representation and the automated analysis methods provides more control and tighter feedback for the human analyst. This must also include the interactive parameterization of both visual and automated methods. Navigation methods for large information spaces are decisive for analysis environments that support exploration. In parallel, they should allow for visual overviews as well as the ability to drill down into areas of interest while preserving orientation within the information space. A helpful principle concerning user interaction tasks is Shneiderman's Visual Information Seeking Mantra "overview first, zoom and filter, then details-on-demand" [Shneiderman, 1996, p. 337]. To put it in a nutshell, user interaction is pivotal in Visual Analytics and a call for a *new* *science of interaction* [Pike et al., 2009; Thomas and Cook, 2005] has been made to empower analysts to engage with visualizations. This will be "concerned with methods by which humans create knowledge through the manipulation of an interface" [Pike et al., 2009, p. 263].

1.1.3 *Automated Data Analysis*

Facing the huge volumes of data to be analyzed today, applying purely visual techniques is often not sufficient. Visual Analytics systems aim to bridge this gap by combining both interactive visualization and automated data analysis methods. Automated data analysis methods perform computational activities on potentially large volumes of data and thus complement human cognition.

In general, *data mining* and *knowledge discovery* are commonly defined as the application of algorithms to extract useful structures from large volumes of data, where knowledge discovery explicitly demands that knowledge be the end product of the analytical calculations [Fayyad et al., 2001, 1996; Han and Kamber, 2005]. A variety of algorithms and methods are involved in achieving this goal, including disciplines like databases, statistics, artificial intelligence, neural networks, machine learning, information retrieval, pattern recognition, data visualization, and high-performance computing. Major data mining functionalities include

- the discovery of associations or correlations (capturing relationships between attributes),
- classification (determining to which class an item belongs),
- clustering (grouping data into clusters based on similarity),
- prediction (inferring future behavior from data collected in the past),
- trend detection (discovering the most significant changes),
- search & retrieval (searching for a priori specified queries including exact as well as approximate matches), as well as
- pattern discovery (automatically discovering interesting patterns).

The sole use of interactive visual methods is not sufficient for coping with the data analysis problems we are facing, nor is the exclusive use of automated data analysis methods. Because Visual Analytics focuses on ill-defined problems and tasks based on large amounts of complex, often incomplete and conflicting data, human evaluation and judgement is needed in order to achieve meaningful results.

1.1.4 *Human-Centered Aspects*

As explained earlier, the basic idea of Visual Analytics is the integration of the outstanding capabilities of humans in terms of visual information exploration and the enormous processing power of computers to form a powerful knowledge discovery environment. Both visual as well as automated methods are combined in an intertwined manner to fully support this process. Most importantly, the human user is not merely a passive element who interprets the outcome of visual and automated methods, but rather, she is the at the core and drives the whole process. Visualization and Visual Analytics aim to amplify cognition, but simply producing images is no guarantee that complex visualizations will be understood and are useful for gaining insights. Therefore, a human-centered approach is essential and should follow four main principles:

Early focus on users and tasks. Understanding the users, the tasks they perform, and the environment in which users perform these tasks [Kerren et al., 2007; Munzner, 2009].

Design for human perception and cognition. Artifacts (methods, techniques, tools, and systems) need to be designed based on the basic characteristics and constraints of the human visual system and cognitive abilities, such as for example preattentive processing [Ware, 2000], Gestalt principles [Wertheimer, 1938], or sensemaking theory [Pirolli and Card, 2005].

Continuous evaluation. Evaluating Visual Analytics methods involving studies on effectiveness, efficiency, and usability to present measurable benefits and understand the limitations of the developed methods [Lam et al., 2012].

Iterative design & refinement. Correcting problems found by experts and users and improving artifacts continuously throughout the design and development life cycle [Shneiderman and Plaisant, 2004].

Figure 1.1 presents an overview of the major factors to consider when designing interactive visual artifacts. The three main questions to be answered in this context are:

- Who are the users of the systems? (*users*)
- What kind of data are they working with? (*data*)
- What are the general tasks of the users? (*tasks*)

The answers to these questions largely determine which representation and interaction methods are suitable. Along the sides of the triangle, some of the major quality criteria for visualization are listed that need to be satisfied in order to obtain useful results. *Expressiveness* refers to the requirement of showing exactly the information contained in the data; nothing more and nothing less must be visualized [Mackinlay, 1986]. *Effectiveness* primarily considers the degree to which visualization addresses the cognitive capabilities of the human visual system, but also the task at hand, the application background, and other context-related information, to obtain intuitively recognizable and interpretable visual representations [Mackinlay, 1986]. Finally, *appropriateness* involves a cost-value ratio in order to assess the benefit of the visualization process with respect to achieving a given task [Van Wijk, 2006].

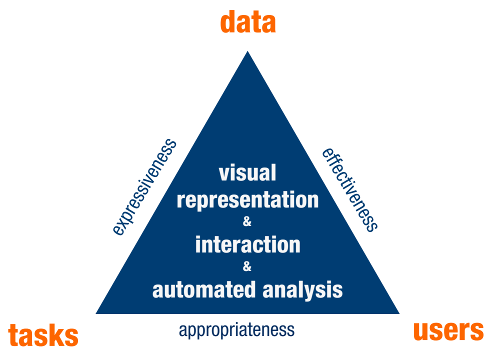


FIGURE 1.1 – **Design triangle.** Major factors to consider when designing interactive VA methods.

1.1.5 *Evaluation*

As emphasized in the previous Section, the evaluation of Visual Analytics artifacts plays a crucial role. This encompasses studies on their effectiveness, efficiency, and usability. These are important for scientific research and practical application. Empirical evaluation of visualization can be challenging [Carpendale, 2008; Plaisant, 2004], particularly for complex processes like sensemaking with interactive visualizations.

There are multiple possible evaluation methods, which cater to different research objectives and have trade-offs in terms of precision, generalizability, and realism [Carpendale, 2008; Lam et al., 2012]. Broadly, they can be divided into quantitative and qualitative methods, whereas the first emphasizes measurable outcomes and the second emphasizes interpretative analysis of observation material. Some commonly used methods in this context are described next.

Controlled experiments allow the precise measurement of performance indicators like speed or correctness for solving tasks in a laboratory environment and, thus, compare different systems.

Usability tests involve subjects working with a visualization system, typically solving tasks in a laboratory environment. Meanwhile, their interactions, insights, and difficulties are observed to judge the system's usability. Post-test questionnaires or interviews are used in addition.

Heuristic evaluations and inspections (often also referred to as *expert reviews*) aim at finding problems and are performed by evaluators who examine the visualization tool [Nielsen, 1994]. Observed problems with the investigated tool are collected, often using a set of heuristics to focus on important aspects of the tool.

Case studies describe how domain users can use the visualization system for their real work tasks. These studies are conducted in close cooperation between researchers and one or a few domain experts and often run over longer periods of time, which allows the experts to become familiar with the system. Thus, the studies can give a realistic understanding of the system's strengths. Performing multiple case studies in different domains can show the generalizability [Shneiderman and Plaisant, 2006].

Deployment of visualization systems 'in the wild' which is an additional testimonial to their usefulness.

1.1.6 *Conceptual Model and Process*

To summarize the brief introduction to Visual Analytics, Figure 1.2 shows a conceptualization of Visual Analytics systems as a division into five spaces [Sedig et al., 2012]:

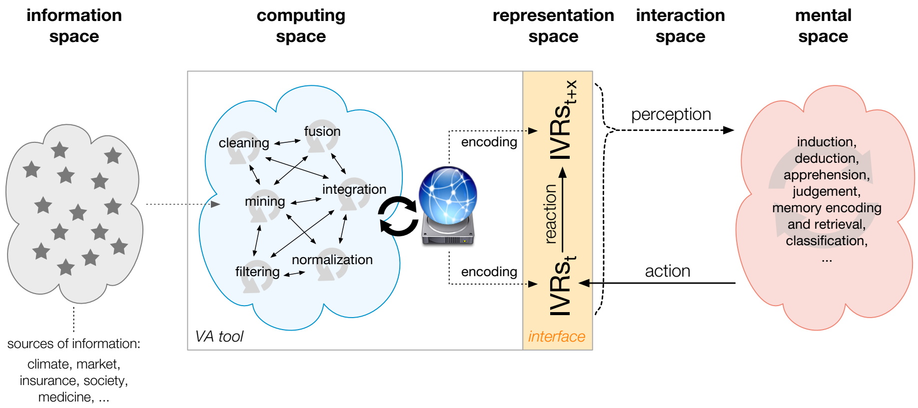


FIGURE 1.2 – **Conceptual model of the structure of Visual Analytics** (based on [Sedig et al., 2012]).

- The **information space** is concerned with the sources of information and aims at modeling, abstracting, and characterizing these.
- The **computing space** deals with encoding and storing internal representations of elements from the information space and includes computational operations carried out on such representations.
- The **representation space** makes the internal representations accessible to users using interactive visual representations (IVRs).
- In the **interaction space** the dyad of action-reaction takes place and perception connects to the mental space.
- The **mental space** is concerned with internal mental events and operations of human analysts.

In addition to this component-oriented conceptualization, a process-oriented view of Visual Analytics is presented in Figure 1.3 [Keim et al., 2010]. It focuses on the tight integration of visual data exploration and automated data analysis and describes the dynamic process of synthesizing knowledge from data. In analogy to Shneiderman's Visual Information Seeking Mantra [Shneiderman, 1996], Keim [2005] proposed the *Visual Analytics Mantra* with a particular focus on the tight integration of automated and visual methods as "Analyze First – Show the Important – Zoom and Filter, and Analyze Further – Details on Demand."

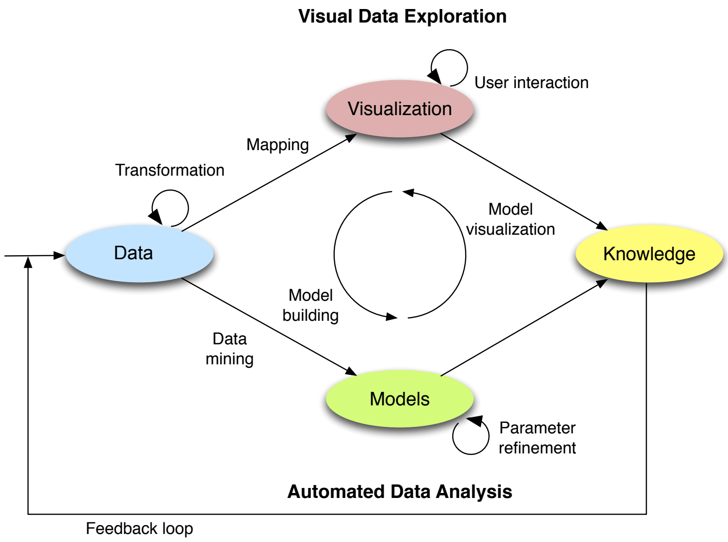


FIGURE 1.3 – **The Visual Analytics process** [Keim et al., 2010].

1.1.7 *Time-Oriented Data*

Most of the data we are dealing with is related to time. It is an important data dimension that is common to many application domains such as medicine, engineering, science, or business. Understanding time-oriented data enables us to learn from the past in order to predict, plan, and shape the future.

Due to the distinct characteristics of the dimension of time, appropriate visual and automated methods are required to explore and analyze them. In contrast to other quantitative data dimensions, time has an inherent semantic structure which increases its complexity dramatically. The hierarchical structure of granularities in time, as for example minutes, hours, days, weeks, months, is unlike most other quantitative dimensions. Specifically, time comprises different forms of divisions (e.g., 60 minutes resemble one hour while 24 hours resemble one day) and granularities are combined to form calendar systems (e.g., Gregorian, Julian, business, or academic calendars). Time also contains natural cycles and re-occurrences, as for example seasons, as well as social (often irregular) cycles, like holidays or school breaks. Moreover, time-oriented data can be given either for a time point (instant) or a time

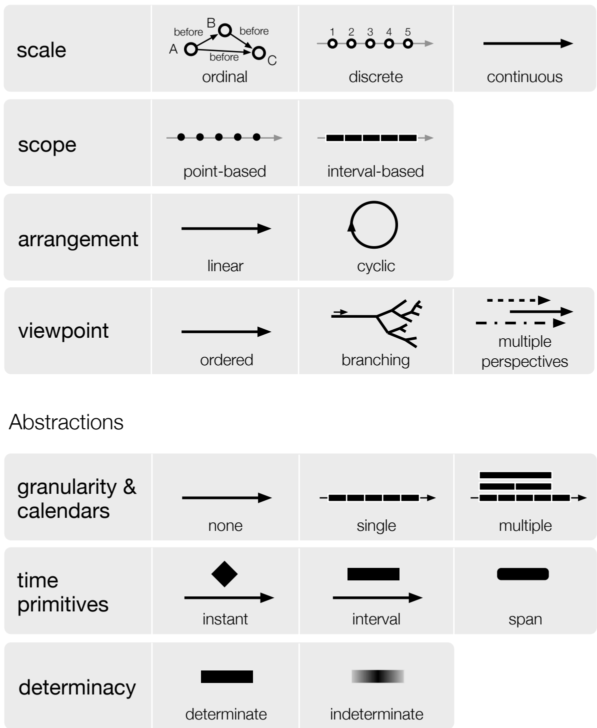


FIGURE 1.4 – **Characteristics of time** [Aigner et al., 2011b].

interval. While intervals can easily be modeled by two time points, they add complexity of 13 different qualitative temporal relationships [Allen, 1983].

An overview of the characteristics of time is given in Figure 1.4 and concern [Aigner et al., 2011b]

- the scale (ordinal vs. discrete vs. continuous),
- the scope (point-based vs. interval-based),
- the arrangement (linear vs. cyclic), and
- the viewpoint (linear vs. branching vs. multiple perspectives) of time-oriented data, as well as
- hierarchically structured calendar systems to accommodate different levels of temporal granularity (e.g., seconds, minutes, hours),
- different time primitives (instant, interval, span), and their
- determinacy (determinate vs. indeterminate).

In order to adequately address these specific characteristics of time, specialized visualization, interaction, and automated methods are needed. As a matter of fact, the particular characteristics of time and data as well as the specific user tasks largely influence the design of visualization solutions. Visualization has been applied to present, explore, and analyze this kind of data for a long time; indeed, early representations even trace back to the 11th century [Tufte, 1983]. In general, the dimension of time can be represented using the display space (i.e., static representation) or physical time (i.e., dynamic representation). Most visualization approaches that use display space employ one display dimension as the time axis. Classic examples for this are charts where time is mapped to the horizontal x-axis (e.g., line plot).

Visualizing time-oriented data is not an easy task. Even though many approaches to this task have been published in recent years, most of them are specific to only a particular analysis problem [Aigner et al., 2011b]. The reason why most methods are highly customized is simple: It is enormously difficult to consider all aspects involved when visualizing time-oriented data. Time itself has many theoretical and practical aspects. For instance, time points and time intervals use different sets of temporal relations. It also matters if we interpret time as a linearly ordered set of temporal primitives, or if we assume the temporal primitives to recur cyclically. The data that tie to the time axis are another decisive concern. Do we have a single variable per temporal primitive or are there multiple variables we have to consider? Moreover, data can be abstract or can be bound to a spatial frame of reference. These and several other data-related questions have to be considered when designing visual analysis methods. Only if the characteristics of the data are taken into account is it possible to generate expressive visual representations. Finally, visual representations themselves imply the need to think about representational and perceptual issues. However, the specifics of time are often not reflected appropriately by treating time just like any other quantitative dimension, as emphasized by Silva and Catarci [2000, p. 318]: "It is now recognized that the initial approaches, just considering the time as an ordinal dimension in a 2D or 3D visualizations, are inadequate to capture the many characteristics of time-dependent information. More sophisticated and effective proposals have been recently presented. However, none of them aims at providing the user with a complete framework for visually managing time-related information."

All these aspects are important when applying or developing visual methods for analyzing data that are connected to time. A main problem is that the diversity of the involved aspects makes it difficult for practitioners to find appropriate solutions for their task at hand, and difficult for researchers to identify directions for future work to bring forward the visualization of time-oriented data.

1.1.8 *Research Challenges*

In the foundational research and development agenda, Thomas and Cook [2005] define *enabling profound insights* as the grand challenge of Visual Analytics. This involves the analysis of huge amounts of heterogeneous, possibly incomplete, conflicting, inconsistent, and dynamic information. For this, human judgement is required to deal with ill-defined problem solving, synthesize knowledge, and make decisions based on complex data. More recently, a more structured view on the challenges of Visual Analytics has been developed by a broad consortium of Visual Analytics stakeholders in the VisMaster consortium[2] [Keim et al., 2010]. The main challenges of Visual Analytics can be grouped into the four themes *data*, *users*, *design*, and *technology*.

Data. *Data quality* is an important issue and there is hardly any real-world dataset that does not contain wrong or missing data. Visual Analytics methods need to deal with such incomplete or conflicting data appropriately. Apart from issues concerning data quality, the fast growing quantities of data also pose a considerable challenge for the *scalability* of Visual Analytics methods. Further, exploring *heterogeneous data* is a difficult task. Such datasets comprise multiple variables of different data types that often stem from different sources and are sampled irregularly and independently from each other. This poses difficulties for many commonly used data modeling and visualization techniques. Moreover, *specialized data types* such as spatio-temporal data or non-numerical data types such as text need to be managed and analyzed. Specifically, it is acknowledged that *time* is a unique data dimension with distinct characteristics. Many visualization, analysis, and simulation systems deal with time-related aspects. However, these systems could be improved by considering the specifics of time and their implications in a broader sense.

Users. As Visual Analytics is inherently user-driven, *meeting users' needs* is vital. In order to achieve this, a thorough understanding of users' needs, tasks, goals, and their work environment is needed. However, this is not easy to accomplish, since typical Visual Analytics problems are not well defined, often ad-hoc, and far from routine tasks. In order to allow for a seamless interplay between the human analyst and visual artifacts, a *high degree of interactivity* is needed. Moreover, evaluating highly interactive visualization artifacts is a challenging and thorny task because visualization usually aims at supporting ill-defined problems and tasks based on large amounts of complex data [Elmqvist and Yi, 2012; Lam et al., 2012]. Particular challenges are [Plaisant, 2004]: the need to work with data over longer periods of time and from different perspectives; the exploratory nature of visual analysis where users may pose questions and get insights they did not know they will have prior to looking at the visualization; and the fact that important discoveries occur rarely and maybe not at all.

Design. Designing usable, useful, and effective Visual Analytics solutions is a complex endeavor. In order to be successful, researchers and engineers need *guidance* on how to design and develop such systems. However, research on such guidance is sparse [Keim et al., 2010]. Although Visual Analytics can build on a large body of research from HCI, the special characteristics of

[2] http://vismaster.eu, accessed at March 30, 2013.

Visual Analytics, namely its focus on data and automated analysis, call for refined approaches.

Technology. Building and providing reusable *infrastructure* is a challenge seen as particularly important for the growth of the discipline. Designing such infrastructure is a formidable research challenge, not to mention the effort required to actually implement the necessarily diverse functionality.

Each of the themes presented offers various opportunities for further work; the fruitful and seamless integration of all of the components is an especially big challenge. More powerful and more usable data exploration and knowledge discovery methods are the goal of these efforts. Nevertheless, powerful, human-centered knowledge discovery environments, obtained from the integration of the best of visual methods and automated methods are not only very challenging – they also appear to be a very promising approach for enabling profound insights.

1.1.9 *Discussion and Overview*

This section has provided a concise introduction to the field of Visual Analytics and its most important research challenges. However, not all relevant topics in Visual Analytics have been accounted for in this brief overview of the field. Instead, the main intention was to provide context and background information necessary to follow the contents of the remainder of this thesis: Chapter 2 presents an overview of the issues of visualization and automated methods for analyzing time-oriented data. It explains why time demands specialized methods and emphasizes the importance of adhering to the characteristics of the data. Moreover, it advocates a focus on the user, while placing emphasis on task-specific methods. Chapter 3 provides a more detailed view of the characteristics of time-oriented data and introduces a framework for designers of visualization methods for time-oriented data that considers the three main aspects time, data, and representation. In the following chapter, the knowledge built up in Chapters 2 and 3 is applied to the problem of collaborative resource scheduling. It presents the human-centered design and development of Visual Analytics methods that closely integrate an interactive visual method with an optimization algorithm. In Chapter 5, the Visual Analytics challenge of scalability is the main topic. A concept proposed earlier is improved in terms of visual, interactive, and computational methods for the analysis of asymmetrically large contingency tables. For the remaining chapters of the thesis, the main focus is on evaluation. This includes the presentation of software infrastructure in the form of an extensible and reusable evaluation library in Chapter 6 as well as two controlled experiments that provide empirical evidence on the effectiveness and efficiency of different visualization methods for time-oriented data in Chapters 7 and 8.

1.2 SELECTED PUBLICATIONS

This thesis contains the following publications. These publications have been chosen to provide a representative selection of the author's work in the field of Visual Analytics and to present a coherent view of contributions that address major challenges of the discipline.[3]

- [Aigner et al., 2008b] Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2008). Visual Methods for Analyzing Time-Oriented Data, *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60. [MODEL/COMMENTARY]
- [Aigner et al., 2007c] Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2007). Visualizing Time-Oriented Data – A Systematic View, *Computers & Graphics*, 31(3):401–409. [MODEL/TAXONOMY]
- [Schneider and Aigner, 2011] Schneider, T., and Aigner, W. (2011). A-Plan: Integrating Interactive Visualization With Automated Planning for Cooperative Resource Scheduling, In *Proceedings of International Conference on Knowledge Management and Knowledge Technologies (I-KNOW), Special Track on Theory and Applications of Visual Analytics (TAVA)*, pages 44:1–44:8. ACM. [DESIGN STUDY]
- [Alsallakh et al., 2012] Alsallakh, B., Aigner, W., Miksch, S., and Gröller, E. (2012). Reinventing the Contingency Wheel: Scalable Visual Analytics of Large Categorical Data, *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2849–2858. Best paper honorable mention. [TECHNIQUE]
- [Aigner et al., 2013] Aigner, W., Hoffmann, S., and Rind, A. (2013). Eval-Bench: Design and Implementation of a Software Library for Evaluation in Visualization, *Computer Graphics Forum*. 32(3). forthcoming. [SYSTEMS]
- [Aigner et al., 2011a] Aigner, W., Kainz, C., Ma, R., and Miksch, S. (2011). Bertin was Right: An Empirical Evaluation of Indexing to Compare Multivariate Time-Series Data Using Line Plots, *Computer Graphics Forum*, 30(1):215–228. [EVALUATION]
- [Aigner et al., 2012] Aigner, W., Rind, A., and Hoffmann, S. (2012). Comparative Evaluation of an Interactive Time-Series Visualization that Combines Quantitative Data with Qualitative Abstractions, *Computer Graphics Forum*, 31(3):995–1004. [EVALUATION]

The discipline is characterized by a high degree of collaboration that includes interdisciplinary cooperations between research groups, domain experts, PhD and master students, and senior researchers, which is necessary to tackle the complex and multi-faceted problems in Visual Analytics. This also explains why (with the exception of a short paper and one paper under review) none of the peer-reviewed publications of the author are single-author papers. As a matter of fact, single-author publications are exceptions rather than the rule in Visual Analytics research.

---

[3] The paper type following the categorization of Munzner [2008] is provided in square brackets and SMALL CAPITALS for each article. Please refer to the end of Section 1.3 for a brief explanation of these paper types.

With respect to the publications listed above, the author had a leading role in all of them and provided significant contributions. This is reflected by being the first-author in all but two of them. More details about the specific contributions are given in the following section.

The publications in the following chapters are included unmodified as they were originally published and no changes to the text have been made. Only the typesetting has been adapted to conform to the overall style of this thesis. Furthermore, in order to be self-contained, each chapter has its own references section.

1.3 CONTRIBUTIONS

Over the years the author has successfully made significant contributions to the research concerned with a number of the main challenges of Visual Analytics. Overall, the author's publication list comprises 60 peer-reviewed papers at the time of writing.[4]

Apart from the articles listed in the previous section, a major contribution of the author is the book "Visualization of Time-Oriented Data" recently published by Springer [Aigner et al., 2011b]. It is devoted to a systematic discussion of this topic and represents the first of its kind. Its main contributions to the Visual Analytics research challenges listed above can be considered in terms of *data* via its special focus on time and time-oriented data and in in terms of *design* by providing a framework and reference for scientists and practitioners seeking information on how their time-oriented data can best be visualized in order to gain valuable insights. Moreover, the book advocates a human-centered design approach (*user*) and a conceptual framework for Visual Analytics systems for time-oriented data is proposed in order to provide guidance for future Visual Analytics system design (*technology*).

In the following, an overview of all research contributions will be given with regard to data, users, design, and technology as introduced in Section 1.1. After that, more detailed descriptions of the articles that make up the body of this thesis will be provided.

Data. The main interest of the author's work centers on Visual Analytics for *time and time-oriented data*. Apart from the already mentioned works, contributions include novel methods for interactive visualization dealing with temporal uncertainties [Aigner et al., 2005] and time granularities [Lammarsch et al., 2009], as well as novel methods for the automated analysis of time-oriented data such as pattern mining that takes time intervals into account [Bertone et al., 2010]. Moreover, theoretical work has been carried out on the modeling of the Visual Analytics process with respect to the special

[4] A full list of publications and open access pre-/post-prints of most publications are available from http://www.ifs.tuwien.ac.at/~aigner/publications

characteristics of time [Lammarsch et al., 2011] and on the development of an extended task framework for time-oriented data [Lammarsch et al., 2012]. From an application perspective, the *medical domain* is another focal point of the author's work. More specifically this includes interactive visual exploration methods for heterogeneous medical data, mainly in terms of clinical guidelines and protocols (CGPs) and electronic health records (EHR) [Aigner and Miksch, 2004, 2006; Gschwandtner et al., 2011a,b; Pohl et al., 2011; Rind et al., 2011a,b, 2010]. Moreover, overview articles on both topic areas have been published [Aigner et al., 2008a; Rind et al., 2013]. More recently, the author contributed to work on *dynamic networks*, i.e., (social) network data that changes over time. Apart from work on interactive visual exploration methods for this kind of data [Federico et al., 2012a, 2011; Smuc et al., 2013; Windhager et al., 2012], contributions were made to a paper proposing network metrics specific to the dynamics of networks [Federico et al., 2012b]. Another prominent Visual Analytics challenge is *scalability*, where work has been conducted to visually analyze large contingency tables [Alsallakh et al., 2012]. Last, but not least, the author conducted research on *data quality* by participating in the development of a taxonomy for dirty time-oriented data [Gschwandtner et al., 2012].

Users. As already mentioned, a centerpiece of the author's work is a strong emphasis on human-centered methods. Publications that report on *design studies* are concerned with visual exploration of dynamic networks [Smuc et al., 2013], of electronic health records [Gschwandtner et al., 2011a,b; Pohl et al., 2011; Rind et al., 2011a,b, 2010], as well as Visual Analytics methods for resource scheduling [Schneider and Aigner, 2011]. Furthermore, qualitative as well as quantitative *evaluation studies* have been carried out to provide empirical evidence on the effectiveness, usability, and utility of the designed methods. These works contain controlled experiments to compare different visualization methods [Aigner et al., 2011a, 2012; Biffl et al., 2005] and qualitative evaluations [Pohl et al., 2011; Rind et al., 2011b]. Moreover, the author participated in conceptual works by analyzing insight-based evaluation results [Smuc et al., 2009] and by structuring the problem solving strategies of analysts [Mayr et al., 2010]. A particular focus of the author is on the concepts of *interaction and interactivity*. In this regard, the author participated in research on highly interactive visual exploration methods [Hinum et al., 2005] and the development of specialized animated transitions that aid users in interacting with visualizations [Federico et al., 2012a]. Furthermore, the author has also undertaken theoretical work aiming at the better integration of cognitive models and theories [Aigner, 2011] while also making contributions to work on analyzing interaction logs [Pohl et al., 2012].

Design. As visualization and Visual Analytics are relatively young research disciplines, foundational theories and methodologies are relatively rare. Nevertheless, it is important to guide the designers and engineers of Visual Analytics systems in order to put the results of research into practice. The work of the author includes *systematic views* (overview papers) on the visualization of time-oriented data [Aigner et al., 2008b, 2011b] and specifically on the visualization of clinical guidelines and protocols [Aigner et al., 2008a] as well as on the visualization of electronic health records [Rind et al., 2013]. Moreover, contributions have been made to *conceptual frameworks* for Visual Analytics systems [Aigner et al., 2007b, 2011b] and process models [Lammarsch et al., 2011] for time-oriented data. Apart from that, a task framework has been proposed for interactive visualization with time-oriented data in general [Lammarsch et al., 2012] as well as a taxonomy for data quality problems of time-oriented data [Gschwandtner et al., 2012]. These works aid researchers and designers in systematically approaching the design space and enhancing comparability between methods. Finally, systematic views on the design of visualizations for time-oriented data have been set forth in [Aigner et al., 2007c, 2011b].

Technology. In terms of technology, the author has participated in research *comparing technological possibilities* for web browser based interactive visualizations [Lammarsch et al., 2008] as well as *comparing applications and tools* with a focus on time-oriented data [Wohlfart et al., 2008]. Recently, work has been published on the design and implementation of a *software library* to aid visualization evaluation [Aigner et al., 2013].

In addition to his contributions to work tackling the major Visual Analytics challenge areas described above, the author is an active member of the *scientific community* and has provided useful services and resources for the Visual Analytics community. The author was involved in giving a tutorial on Visual Analytics [Aigner et al., 2007a] and was co-organizer of a workshop on Visual Analytics education at IEEE VisWeek 2008 [Schratt et al., 2008]. Moreover, he is the founder and coordinator of the *InfoVis:Wiki* (www.infovis-wiki.net), co-creator of the *The TimeViz Browser* (survey.timeviz.net), an interactive survey of visualization techniques for time-oriented data, and leader of the development team of *EvalBench* (www.evalbench.org), an open source software library for visualization evaluation.

1.3.1 *Contributions of selected papers*

The author's research covers all of the above mentioned areas and deals with intricate problems that are recognized by the Visual Analytics community as important research challenges. In the following, the papers that constitute the body of this thesis will be briefly outlined along with explanations of the author's role in carrying out the research and writing the papers.

Chapter 2: Visual Methods for Analyzing Time-Oriented Data. This publication is an overview paper that investigates the role of time in the context of visually driven analysis. It covers the aspects of visualization, automated analysis, and user-centered methods. The main challenges that arise when visualizing time-oriented data are identified and possible solutions are discussed. First, concerning visual methods, it points out the importance of choosing and parameterizing visualization and interaction techniques properly with respect to the characteristics of time. Second, the usefulness of combining visual and automated analysis methods is illustrated by the examples of temporal data abstraction, principal component analysis, and clustering. Third, it is shown why user support or guidance is helpful and how such support can be realized by an approach that emphasizes relevant information, called event-based visualization. With this paper the authors aimed to improve the integration of visual, automated, and user-centered methods.

Wolfgang Aigner was the lead author and developed the main idea of the article. He wrote the core sections of the paper with feedback by Heidrun Schumann and Wolfgang Müller. The sections on analytical methods and the event-based approach were written together with Silvia Miksch, Wolfgang Müller and Christian Tominski.

This article was published in the prestigious IEEE Transactions on Visualization and Computer Graphics (TVCG) journal. It can be considered a reference work as it has received interest from researchers inside and outside of the community and has been cited more than 100 times.

Chapter 3: Visualizing Time-Oriented Data – A Systematic View. Over the years, a wide repertoire of interactive techniques for visualizing datasets with temporal dependencies has been accumulated. This variety makes it difficult for prospective designers and users to select methods or tools that are suited for their particular task at hand. This paper presents a systematic view of the visualization of time-oriented data. It proposes a systematic way of looking at the methods for visually analyzing time-oriented data based on three main criteria: time (What are the characteristics of the time axis?), data (What is analyzed?), and representation (How is it represented?). The article explains the basics of visualizing time-oriented data and discusses why time is important and why it deserves special consideration in the context of visual analysis methods. The systematic view is used to categorize visualization approaches and is illustrated with examples from literature.

Wolfgang Aigner was the lead author of the paper together with Christian Tominski. He developed the core idea and the categorization scheme. Together with Christian Tominski, the approach was refined and the main parts of the article were written with feedback from Silvia Miksch, Heidrun Schumann, and Wolfgang Müller.

This article was published in the Computers & Graphics journal and is a precursor to the book mentioned previously. It received great interest, was cited more than 100 times, and was awarded a Top Cited Article 2005-2010 by Pergamon/Elsevier.

Chapter 4: A-Plan: Integrating Interactive Visualization with Automated Planning for Cooperative Resource Scheduling. Assigning staff to work tasks is a complex problem that involves a large number of factors and requires a lot of expertise. This paper reports on a design study of a Visual Analytics tool for scheduling technicians for gas device maintenance and is a self-contained example of a human-centered design approach. The design approach consists of three main phases. First, the user and the task analysis, which was based on a combination of contextual observation and task demonstration, are described. Second, the user-centered visualization and interaction design approach based on personas and scenarios is presented. The designed system integrates interactive visualizations with automated planning using an optimization algorithm. Special consideration has been given to supporting the collaborative working style of the target user group. Third, a qualitative evaluation of the implemented prototype with domain experts was carried out. For this, the methods of prototype demonstration, thinking aloud, and semi-structured interviews were employed. The results were analyzed along a set of heuristics specific to information visualization. Overall, the work demonstrates the successful application of a Visual Analytics approach that closely integrates interactive visualization and automated methods with human users in a collaborative environment in the context of resource scheduling.

Wolfgang Aigner was the lead author of the paper and led the human-centered design process. He guided and supervised the prototypical implementation that was performed by Thomas Schneider (master student). Wolfgang Aigner developed the experiment designs and the experiments were conducted by Thomas Schneider. Wolfgang participated in the analysis of experimental results and wrote the article with feedback from Thomas Schneider.

This paper was presented at the Special Track on Theory and Applications of Visual Analytics (TAVA) of the International Conference on Knowledge Management and Knowledge Technologies (I-KNOW) which is a relatively recent venue but has become an integral part of the main conference this year. The proceedings are published by ACM.

Chapter 5: Reinventing the Contingency Wheel: Scalable Visual Analytics of Large Categorical Data. Many problems in scientific domains require the analysis of associations between categorical variables. Contingency tables (also known as crosstabs) are a common way to summarize categorical data as a first step of analysis. Several visualization methods have been developed to analyze associated categories in contingency tables. However, these methods are designed to handle rather small tables having few categories. For real-world datasets, often much larger contingency tables need to be analyzed, which poses a problem for these methods. In this paper, a Visual Analytics method is presented that overcomes the major shortcomings when dealing with large and dense contingency tables. This is carried out in three major areas. First, improved automated methods are presented that are based on Pearson's residuals. Second, a frequency-based abstraction is employed to provide aggregated views effectively, eliminating overplotting problems while preserving the key features of the data. Third, a visual multi level overview+detail interface enables the effective visual exploration of individual data items that are aggregated in the visualization or in the table along with their attributes. Finally, a usage scenario demonstrates how these methods can be used to find nontrivial patterns in large categorical data and how further attributes can be analyzed.

The author contributed core ideas for scalability (mainly for visualization and interaction methods) and supervised the parts of the implementation performed by Bilal Alsallakh (PhD candidate). Wolfgang Aigner performed the usage scenario research and wrote significant parts of the paper with feedback from Silvia Miksch and Eduard Gröller.

This recent paper is one of only 10 selected papers to be part of a special TVCG journal issue on the IEEE Visual Analytics Science and Technology (VAST) 2012 conference. Moreover, it received a best paper honorable mention at the top-level venue IEEE VisWeek 2012.

Chapter 6: EvalBench: A Software Library for Visualization Evaluation. As the field of visualization is maturing, evaluation has become an increasingly important part of research. However, the difficulty of conducting these evaluations remains an important issue. Specifically, evaluating highly interactive visualization artifacts is a challenging and thorny task because visualization usually aims at supporting ill-defined problems and tasks based on large amounts of complex data. It is widely acknowledged that there is a need for a solid evaluation infrastructure to encourage visualization researchers to carry out evaluations of their methods and tools. To support them in their efforts, the paper presents the open source software library EvalBench with the goals of being easy to set up and use, customizable, and as independent and loosely coupled to the visualization artifact to be evaluated as possible. The main contributions of the work are threefold. First, an open source software library specifically suited to the requirements of evaluating visualizations has been developed and made available to the community. Second, the architectural choices and basic abstractions of the library are discussed independently of specific programming languages. Third, the practical utility of the library is shown using a number of case studies including both quantitative and qualitative methods.

Wolfgang Aigner was the lead author of the article. He led the development of the core concept and architecture of the library. The system design was refined together with Alexander Rind (PhD candidate) and Stephan Hoffmann (master student). He guided and supervised the development of the library performed mainly by Stephan Hoffmann and Alexander Rind.

This paper has been accepted for presentation at Eurographics/IEEE Conference on Visualization (EuroVis) 2013 and will be published in the Computer Graphics Forum journal by Wiley/Blackwell. EuroVis and VisWeek are the two most competitive venues for visualization researchers. For this reason and due to the two-stage reviewing process, the papers are published as journal articles.

Chapter 7: Bertin was Right: An Empirical Evaluation of Indexing to Compare Multivariate Time-Series Data Using Line Plots. Line plots are very well suited for visually representing time-series. However, several difficulties arise when multivariate heterogeneous time-series data is displayed and compared visually. The three main challenges in this regard are largely different value domains, the fact that percent changes are not represented accordingly, and heterogeneous data. To mitigate this, visualization pioneer Jacques Bertin presented a method called "indexing", which transforms data into comparable units for visual representation. Although the indexing method was introduced by Bertin more than 40 years ago, there was no empirical evidence on its effectiveness and efficiency. The main contribution of this paper is a comparative empirical study that assesses the indexing method. Three visualization types for the display of multivariate time series were examined in a series of tasks with 24 subjects. Five hypotheses were tested based on the two measured dependent variables, task completion time and task correctness. For evaluating the visualization techniques, realistic stimuli were used in the form of tasks related to stock market data that were based on a specific task taxonomy for spatio-temporal data. The experiment was conducted in a within-subjects design and the gathered data was analyzed using statistical hypotheses tests. The main results include clear evidence that using indexing in general yields a higher correctness rate than the two other visualization types linear scale with juxtaposition and log scale with superimposition. With regard to task completion times, the results are less clear and only slight advantages for indexing were found.

Wolfgang Aigner was the lead author of the article and contributed the idea and concept. He guided and supervised the prototypical implementation performed by Rui Ma (master student). Furthermore, he contributed the experiment designs. The experiments were conducted mainly by Christian Kainz (master student). Wolfgang performed the complete analysis of the quantitative experimental results and wrote the complete article with feedback from Rui Ma, Christian Kainz, and Silvia Miksch.

This article appeared in a regular issue of the Computer Graphics Forum journal. The fact that it represents a significant contribution to the literature is underlined by the selection of the paper as one of only two invited talks at EuroVis 2012.

Chapter 8: Comparative Evaluation of an Interactive Time-Series Visualization that Combines Quantitative Data with Qualitative Abstractions. In many application areas, analysts have to make sense of large volumes of multivariate time-series data. Especially in the medical domain, there is a growing awareness that it is important to support decision-making in real-time environments like intensive care units. It can be difficult for the clinicians to make accurate decisions, particularly when the decisions are based on multiple clinical parameters. Temporal data abstraction reduces data complexity by deriving qualitative statements that reflect domain-specific key characteristics. In prior research an interactive visualization technique was presented that combines colored qualitative representations with more detailed quantitative representations (STZ). This technique eases the recognition of critical periods or concrete fluctuations in the data even if the visualization has a small height. However, until now, no empirical evidence on its effectiveness and efficiency was available. In order to fill this gap, this paper reports on a controlled experiment that compares this technique with another visualization method used in the well-known KNAVE-II framework. The results show that a combined visualization of quantitative and qualitative data using different visual encodings (STZ) performs at least equally as well as comparable techniques (KNAVE) and excels especially at more complex tasks. Furthermore, the quantitative evaluation and connected hypotheses are based on the perceptual theory of the "proximity compatibility principle" [Wickens and Carswell, 1995] as well as the ranking of visual variables as proposed by Mackinlay [1986]. This paper presented important empirical evidence that confirmed the proximity compatibility principle proposed more than 15 years ago and proposes a task-oriented adaption of visual variable rankings.

Wolfgang Aigner was the lead author of the article and contributed the idea and concept. He guided and supervised the prototypical implementation performed by Stephan Hoffmann (master student). The study was conducted together with Alexander Rind (PhD candidate) and Stephan Hoffmann (master student), and the analysis of experimental results were guided by Wolfgang Aigner.

This article was presented at the IEEE/Eurographics Conference on Visualization (IEEE) 2012 and was published in the Computer Graphics Forum journal.

In the visualization community, publications are usually categorized into one of five different paper types proposed by Munzner [2008]: model, design study, technique, systems, and evaluation. *Model papers* focus on formalisms and abstractions with the purpose of helping other researchers think about their own work. Three specific subcategories mentioned by Munzner are *taxonomy* (propose categories for understanding the design space of a topic), *formalism* (new models, definitions, or terminology), and *commentary* (critical reflections of a topic). *Design study papers* present novel solutions for a particular domain problem (explain the domain problem, state design requirements, present the applied interactive visualization and analysis methods, and results that back up the claims made). *Technique papers* focus on presenting novel algorithms and implementations are expected. *Systems papers* present and discuss architectural choices made in the design of an infrastructure, framework, or toolkit. And finally, *evaluation papers* focus on validation and empirical results of assessing a system or technique and do not typically introduce novel techniques.

The author covers all of the described paper types and examples for each type of paper are contained in this thesis (Chapter 2 can be considered a model/commentary paper, Chapter 3 a model/taxonomy paper, Chapter 4 is a design study paper, Chapter 5 a technique paper, Chapter 6 a systems paper, and Chapters 7 as well as 8 are evaluation papers). This is also an indicator of the broad contributions made by the author in the field.

REFERENCES

Aigner, W. (2011). Understanding the Role and Value of Interaction: First Steps. In Miksch, S. and Santucci, G., editors, *Proceedings of the International Workshop on Visual Analytics (EuroVA 2011)*, pages 17–20. The Eurographics Association.

Aigner, W., Bertone, A., and Miksch, S. (2007a). Tutorial: Introduction to Visual Analytics. In Holzinger, A., editor, *Proceedings of 3rd Symposium Usability & HCI for Medicine and Healthcare (USAB 07)*, LNCS 4799, page 457–460. Springer-Verlag Berlin Heidelberg.

Aigner, W., Bertone, A., Miksch, S., Schumann, H., and Tominski, C. (2007b). Towards a Conceptual Framework for Visual Analytics of Time and Time-Oriented Data. In Henderson, S. G., Biller, B., Hsieh, M., Shortle, J., Tew, J. D., and Barton, R. R., editors, *Proceedings of the 2007 Winter Simulation Conference*, page 721–729. Invited paper.

Aigner, W., Hoffmann, S., and Rind, A. (2013). EvalBench: A Software Library for Visualization Evaluation. *Computer Graphics Forum*, 32(3). forthcoming.

Aigner, W., Kainz, C., Ma, R., and Miksch, S. (2011a). Bertin was Right: An Empirical Evaluation of Indexing to Compare Multivariate Time-Series Data Using Line Plots. *Computer Graphics Forum*, 30(1):215–228.

Aigner, W., Kaiser, K., and Miksch, S. (2008a). Visualization Techniques to Support Authoring, Execution, and Maintenance of Clinical Guidelines. In ten Teije, A., Lucas, P., and Miksch, S., editors, *Computer-Based Medical Guidelines and Protocols: A Primer and Current Trends*, page 140–159. IOS Press, Health Technology and Informatics.

Aigner, W. and Miksch, S. (2004). Communicating the Logic of a Treatment Plan Formulated in Asbru to Domain Experts. In Kaiser, K., Miksch, S., and Tu, S., editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101 of *Studies in Health Technology and Informatics*, page 1–15. IOS Press.

Aigner, W. and Miksch, S. (2006). CareVis: Integrated Visualization of Computerized Protocols and Temporal Patient Data. *Artifical Intelligence in Medicine (AIIM)*, 37(3):203–218.

Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2007c). Visualizing Time-Oriented Data – A Systematic View. *Computers & Graphics*, 31(3):401–409.

Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2008b). Visual Methods for Analyzing Time-Oriented Data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60.

Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011b). *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer Verlag, London, UK.

Aigner, W., Miksch, S., Thurnher, B., and Biffl, S. (2005). PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 457–463, Los Alamitos, CA, USA. IEEE Computer Society.

Aigner, W., Rind, A., and Hoffmann, S. (2012). Comparative Evaluation of an Interactive Time-Series Visualization that Combines Quantitative Data with Qualitative Abstractions. *Computer Graphics Forum*, 31(3):995–1004. The definitive version is available at http://diglib.eg.org/.

Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.

Alsallakh, B., Aigner, W., Miksch, S., and Gröller, E. (2012). Reinventing the Contingency Wheel: Scalable Visual Analytics of Large Categorical Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2849–2858. Best paper honorable mention.

Bertone, A., Lammarsch, T., Turic, T., Aigner, W., Miksch, S., and Gärtner, J. (2010). MuTIny: A Multi-Time Interval Pattern Discovery Approach To Preserve The Temporal Information In Between. In *European Conference on Data Mining (ECDM'10)*, page 101–106.

Biffl, S., Thurnher, B., Goluch, G., Winkler, D., Aigner, W., and Miksch, S. (2005). An Empirical Investigation on the Visualization of Temporal Uncertainties in Software Engineering Project Planning. In *Proceedings of the 4th International Symposium on Empirical Software Engineering (ISESE'05)*, pages 437–446, Los Alamitos, CA, USA. IEEE Computer Society.

Card, S., Mackinlay, J., and Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA, USA.

Carpendale, S. (2008). Evaluating Information Visualizations. In Kerren, A., Stasko, J. T., Fekete, J.-D., and North, C., editors, *Information Visualization*, pages 19–45. Springer, Berlin.

Elmqvist, N., Moere, A. V., Jetter, H.-C., Cernea, D., Reiterer, H., and Jankun-Kelly, T. J. (2011). Fluid Interaction for Information Visualization. *Information Visualization*, 10(4):327–340.

Elmqvist, N. and Yi, J. S. (2012). Patterns for Visualization Evaluation. In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, pages 12:1–12:8, New York, NY, USA. ACM Press.

Fayyad, U., Grinstein, G. G., and Wierse, A. (2001). *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, San Francisco, CA, USA.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3):37–54.

Federico, P., Aigner, W., Miksch, S., Windhager, F., and Smuc, M. (2012a). Vertigo Zoom: Combining Relational and Temporal Perspectives on Dynamic Networks. In *Proceedings of the 11th International Working Conference on Advanced Visual Interfaces (AVI2012)*, pages 437–440. ACM Press.

Federico, P., Aigner, W., Miksch, S., Windhager, F., and Zenk, L. (2011). A Visual Analytics Approach to Dynamic Social Networks. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies (i-KNOW), Special Track on Theory and Applications of Visual Analytics (TAVA)*, page 47:1–47:8. ACM Press.

Federico, P., Pfeffer, J., Aigner, W., Miksch, S., and Zenk, L. (2012b). Visual Analysis of Dynamic Networks using Change Centrality. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 179–183. IEEE Computer Society.

Gschwandtner, T., Aigner, W., Kaiser, K., Miksch, S., and Seyfang, A. (2011a). CareCruiser: Exploring and Visualizing Plans, Events, and Effects Interactively. In *Proceedings of the 4th IEEE Pacific Visualization Symposium (PacificVis)*, pages 43–50.

Gschwandtner, T., Aigner, W., Kaiser, K., Miksch, S., and Seyfang, A. (2011b). Design and Evaluation of an Interactive Visualization of Therapy Plans and Patient Data. In *Proceedings of the BCS HCI Conference*.

Gschwandtner, T., Gärtner, J., Aigner, W., and Miksch, S. (2012). A Taxonomy of Dirty Time-Oriented Data. In Quirchmayr, G., Basl, J., You, I., Xu, L., and Weippl, E., editors, *Lecture Notes in Computer Science (LNCS 7465): Multidisciplinary Research and Practice for Information Systems (Proceedings of the CD-ARES 2012)*, pages 58 – 72. Springer, Berlin / Heidelberg.

Han, J. and Kamber, M. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, CA, USA.

Heer, J. and Shneiderman, B. (2012). Interactive Dynamics for Visual Analysis. *Communications of the ACM*, 55(4):45–54.

Hinum, K., Miksch, S., Aigner, W., Ohmann, S., Popow, C., Pohl, M., and Rester, M. (2005). Gravi++: Interactive Information Visualization to Explore Highly Structured Temporal Data. *Journal of Universal Computer Science*, 11(11):1792–1805.

Keim, D. (2005). Scaling Visual Analytics to Very Large Data Sets. Workshop on Visual Analytics, Darmstadt, June, 2005.

Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F., editors (2010). *Mastering the Information Age – Solving Problems with Visual Analytics*. Eurographics Association, Geneve, Switzerland.

Kerren, A., Ebert, A., and Meyer, J., editors (2007). *Human-Centered Visualization Environments*, volume 4417 of *Lecture Notes in Computer Science*. Springer, Berlin, Germany.

Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. (2012). Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536.

Lammarsch, T., Aigner, W., Bertone, A., Gärtner, J., Mayr, E., Miksch, S., and Smuc, M. (2009). Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-based Visualizations. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 44–49, Los Alamitos, CA, USA. IEEE Computer Society.

Lammarsch, T., Aigner, W., Bertone, A., Gärtner, J., Miksch, S., and Turic, T. (2008). A Comparison of Programming Platforms for Interactive Visualization in Web Browser Based Applications. In *Proceedings of 12th International Conference on Information Visualisation (IV08)*, page 194–199. IEEE Computer Society.

Lammarsch, T., Aigner, W., Bertone, A., Miksch, S., and Rind, A. (2011). Towards a Concept how the Structure of Time Can Support the Visual Analytics Process. In Miksch, S. and Santucci, G., editors, *Proceedings of the International Workshop on Visual Analytics (EuroVA 2011)*, pages 9–12.

Lammarsch, T., Rind, A., Aigner, W., and Miksch, S. (2012). Developing an Extended Task Framework for Exploratory Data Analysis Along the Structure of Time. In Matkovic, K. and Santucci, G., editors, *Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA 2012)*. Eurographics, Eurographics.

Mackinlay, J. (1986). Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2):110–141.

Mayr, E., Smuc, M., Risku, H., Aigner, W., Bertone, A., Lammarsch, T., and Miksch, S. (2010). Mapping the Users' Problem Solving Strategies in the Participatory Design of Visual Analytics Methods. In *Proceedings of 6th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB 2010)*.

Munzner, T. (2008). Process and Pitfalls in Writing Information Visualization Research Papers. In Kerren, A., Stasko, J., Fekete, J.-D., and North, C., editors, *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*, pages 134–153. Springer Berlin / Heidelberg.

Munzner, T. (2009). A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928.

Nielsen, J. (1994). Usability Inspection Methods. In *Conference Companion Human Factors in Computing Systems, CHI*, pages 413–414. ACM Press.

Pike, W. A., Stasko, J., Chang, R., and O'Connell, T. A. (2009). The Science of Interaction. *Information Visualization*, 8(4):263–274.

Pirolli, P. and Card, S. (2005). The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of International Conference on Intelligence Analysis*.

Plaisant, C. (2004). The Challenge of Information Visualization Evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 106–119, New York, NY, USA. ACM Press.

Pohl, M., Wiltner, S., Miksch, S., Aigner, W., and Rind, A. (2012). Analysing Interactivity in Information Visualisation. *KI - Künstliche Intelligenz*, 26:151–159.

Pohl, M., Wiltner, S., Rind, A., Aigner, W., Miksch, S., Turic, T., and Drexler, F. (2011). Patient Development at a Glance: An Evaluation of a Medical Data Visualization. In Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., and Winckler, M., editors, *Proceedings of 13th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT 2011), Part IV*, page 292–299. Springer.

Rind, A., Aigner, W., Miksch, S., Wiltner, S., Pohl, M., Drexler, F., Neubauer, B., and Suchy, N. (2011a). Visually Exploring Multivariate Trends in Patient Cohorts Using Animated Scatter Plots. In Robertson, M. M., editor, *Ergonomics and Health Aspects of Work with Computers, Proceedings of the International Conference held as part of HCI International 2011*, page 139–148. Springer.

Rind, A., Aigner, W., Miksch, S., Wiltner, S., Pohl, M., Turic, T., and Drexler, F. (2011b). Visual Exploration of Time-oriented Patient Data for Chronic Diseases: Design Study and Evaluation. In Holzinger, A. and Simonic, K.-M., editors, *Proceedings of USAB 2011: Information Quality in e-Health*, page 301–320. Springer.

Rind, A., Miksch, S., Aigner, W., Turic, T., and Pohl, M. (2010). VisuExplore: Gaining New Medical Insights from Visual Exploration. In Hayes, G. R. and Tan, D. S., editors, *Proceedings of the 1st International Workshop on Interactive Systems in Healthcare (WISH@CHI2010)*, pages 149–152.

Rind, A., Wang, T. D., Aigner, W., Miksch, S., Wongsuphasawat, K., Plaisant, C., and Shneiderman, B. (2013). Interactive Information Visualization to Explore and Query Electronic Health Records. *Foundations and Trends in Human-Computer Interaction*, 5:207–298.

Saraiya, P., North, C., Lam, V., and Duca, K. A. (2006). An Insight-Based Longitudinal Study of Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1511–1522.

Schneider, T. and Aigner, W. (2011). A-Plan: Integrating Interactive Visualization With Automated Planning for Cooperative Resource Scheduling. In *Proceedings of International Conference on Knowledge Management and Knowledge Technologies (I-KNOW), Special Track on Theory and Applications of Visual Analytics (TAVA)*, pages 44:1–44:8. ACM Press.

Schratt, A., Aigner, W., Bertone, A., and Miksch, S. (2008). Body of Knowledge for Visual Analytics Education. In *Workshop at VisWeek 2008*. IEEE Computer Society. Refereed Abstract.

Sedig, K., Parsons, P., and Babanski, A. (2012). Towards a Characterization of Interactivity in Visual Analytics. *Journal of Multimedia Processing and Technologies, Special issue on Theory and Application of Visual Analytics*, 3:12–28.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, Los Alamitos, CA, USA. IEEE Computer Society.

Shneiderman, B. and Plaisant, C. (2004). *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley.

Shneiderman, B. and Plaisant, C. (2006). Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. In *Proc. 2006 AVI Workshop BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV '06)*, pages 38–43. ACM Press.

Silva, S. F. and Catarci, T. (2000). Visualization of Linear Time-Oriented Data: A Survey. In *Proceedings of the International Conference on Web Information Systems Engineering (WISE)*, pages 310–319, Los Alamitos, CA, USA. IEEE Computer Society.

Smuc, M., Federico, P., Windhager, F., Aigner, W., Zenk, L., and Miksch, S. (2013). How do you Connect Moving Dots? Insights From User Studies on Dynamic Network Visualizations. In Huang, W., editor, *Human Centric Visualizations: Theories, Methodologies and Case Studies*. Springer. forthcoming.

Smuc, M., Mayr, E., Lammarsch, T., Aigner, W., Miksch, S., and Gärtner, J. (2009). To Score or Not to Score? Tripling Insights for Participatory Design. *IEEE Computer Graphics and Applications*, 29(3):29–38.

Spence, R. (2007). *Information Visualization: Design for Interaction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2nd edition.

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, Los Alamitos, CA, USA.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

Van Wijk, J. J. (2006). Views on Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):421–433.

Wainer, H. (1997). *Visual Revelations: Graphical Tales of Fate and Deception from Napoleon Bonaparte to Ross Perot*. Copernicus, New York, NY, USA.

Ware, C. (2000). *Information Visualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA, USA.

Wegner, P. (1997). Why Interaction Is More Powerful Than Algorithms. *Communications of the ACM*, 40(5):80–91.

Wertheimer, M. (1938). Laws of Organization in Perceptual Forms. In Ellis, W. D., editor, *A Sourcebook of Gestalt Psychology*, pages 71–88. Routledge and Kegan Paul.

Wickens, C. D. and Carswell, C. M. (1995). The Proximity Compatibility Principle: Its Psychological Foundation and Relevance to Display Design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(3):473–494.

Windhager, F., Smuc, M., Zenk, L., Federico, P., Pfeffer, J., Aigner, W., and Miksch, S. (2012). Visual knowledge networks analytics. In Liebowitz, J., editor, *Knowledge Management Handbook*, page 187–206. CRC Press.

Wohlfart, E., Aigner, W., Bertone, A., and Miksch, S. (2008). Comparing Information Visualization Tools Focusing on the Temporal Dimensions. In *Proceedings of 12th International Conference on Information Visualisation (IV08)*, page 69–74.

CHAPTER 2

Visual Methods for Analyzing Time-Oriented Data

Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski

Abstract · *Providing appropriate methods to facilitate the analysis of time-oriented data is a key issue in many application domains. In this paper, we focus on the unique role of the parameter time in the context of visually driven data analysis. We will discuss three major aspects – visualization, analysis, and the user. It will be illustrated that it is necessary to consider the characteristics of time when generating visual representations. For that purpose we take a look at different types of time and present visual examples. Integrating visual and analytical methods has become an increasingly important issue. Therefore, we present our experiences in temporal data abstraction, principal component analysis, and clustering of larger volumes of time-oriented data. The third main aspect we discuss is supporting user-centered visual analysis. We describe event-based visualization as a promising means to adapt the visualization pipeline to needs and tasks of users.*

Keywords · Time-oriented data, visualization, analysis, user.

This article originally appeared as [Aigner et al., 2008]:

Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2008). Visual Methods for Analyzing Time-Oriented Data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60.

2.1 INTRODUCTION & MOTIVATION

CONSIDERING the characteristics of data is vital when designing visual representations. A salient characteristic is whether or not data are related to time. That time is an outstanding dimension is reflected by Shneiderman's Task by Data Type Taxonomy [Shneiderman, 1996], where temporal data are identified as one of seven basic data types. Nowadays, time-oriented data are ubiquitous in many application domains as for example in business, medicine, history, planning, or project management. For a long time visual methods have been successfully applied to analyze such data. A wide repertoire of interactive techniques for visualizing datasets with temporal dependencies is available. However, many current visualization frameworks have not yet considered time as a special dimension, but rather as a common quantitative parameter. According to Thomas and Cook [2006] it is in general a problem that *"Most visualization software is developed with incomplete information about the data and tasks. (...) New methods are needed for constructing visually based systems that simplify the development process and result in better targeted applications."*

In this paper, we point out challenges that arise when visualizing time-oriented data, and take a look at possible solutions to these challenges. To find solutions, it is absolutely mandatory to take into account the following three major aspects:

- Visualization,
- Analysis, and the
- User.

In Section 2.2, we focus on visualization methods for time-oriented data. We will show that the term *time-oriented data* comprises several types of data with different meanings and applications. Designing or applying visual representations can only be successful if one is aware of these different types. This will be demonstrated with several examples of visualization techniques that stem from our own work or are available in literature.

Usually, time-oriented data are large – not only in terms of the number of data items, but also in terms of the number of observed attributes. Ordinary visualizations of such data can lead to overcrowded and cluttered displays, and are therefore of limited use. Data abstractions can help to gain insight even into larger datasets. This is the point where analytical methods come into play. In Section 2.3, we will illustrate (again by examples) the usefulness of combining visual and analytical methods particularly related to time-oriented data.

In order to achieve better targeted applications, users and their tasks and needs must not be neglected, as it is still often the case in today's visualization tools. Apparently, interaction is a key to adapting visual and analytical methods to the user's task at hand. However, not all parameters are intuitive and easy to set. Particularly in cases where complex visual analysis processes

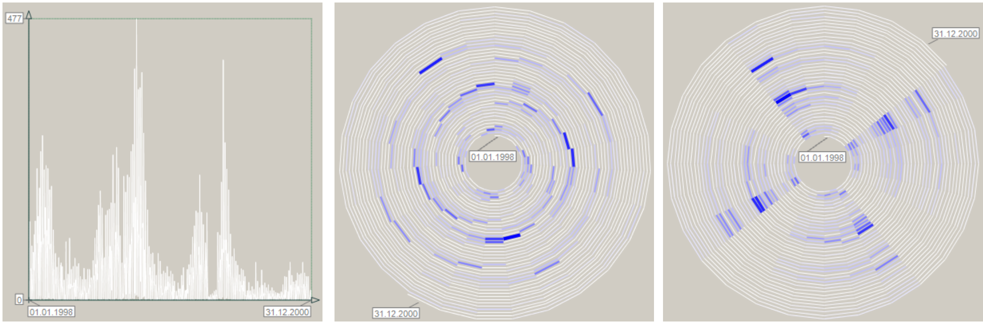


Figure 2.1 – **Different visual representations of a time-oriented dataset describing the number of influenza cases over a period of three years** – left: Time series plot (periodic pattern is difficult to discern), center: SpiralGraph encoding 27 days per cycle (improperly parameterized – periodic pattern is hard to see), right: SpiralGraph encoding 28 days per cycle (properly parameterized – periodic pattern stands out).

have to be steered, having some form of user support or guidance turns out to be helpful. Section 2.4 discusses how such a support can be realized. The basic idea is to find events in the data and to trigger automatic parameter adjustments.

In the last section (Section 2.5), we will briefly recapitulate our discussions and derive possible directions for future work on visual analysis of time-oriented data.

2.2 VISUALIZING TIME-ORIENTED DATA

When we speak of time-oriented data, we basically mean data that are somehow connected to time. Certainly, this vague description is not sufficient when users have to choose or developers have to devise appropriate visualization methods. An essential requirement for achieving expressive and effective visualization is to consider the characteristics of the data to be presented, which in our case are particularly related to the dimension time. A lot of work has been done to formulate the notion of time in many areas of computer science, including artificial intelligence, data mining, simulation, modeling, databases, and more. A theoretical overview along with many references to fundamental publications is provided by Hajnicz [1996]. However, as she points out, the terminology is not consistent across the different fields [Hajnicz, 1996], and hence, does not integrate well with visualization. Therefore, we adapted the work of Frank [1998], where he presents principal orthogonal design dimensions to characterize different types of times. The most important criteria from a visualization point of view are the following:

- *Linear time vs. cyclic time:* Linear time assumes a starting point and defines a linear time domain with data elements from past to future. On the other hand, many natural processes are cyclic, e.g., the cycle of the seasons. To represent such phenomena, a cyclic time domain can be applied. The ordering of points in a strictly cyclic time domain is meaningless with respect to a cycle, e.g., winter comes before summer, but winter also succeeds summer.
- *Time points vs. time intervals:* Discrete time points describe time as abstractions comparable to discrete Euclidean points in space. Time points have no duration. In contrast to that, interval time uses an interval scaled time domain like days, months, or years. In this case, data elements are defined for a duration, delimited by two time points. Both time points and time intervals are called temporal primitives.
- *Ordered time vs. branching time vs. time with multiple perspectives:* Ordered time domains consider things that happen one after the other. For branching time, multiple strands of time branch out, which facilitates description and comparison of alternative scenarios (e.g., for project planning). This type of time supports decision making processes where only one alternative will actually happen. Time with multiple perspectives allows more than one point of view at observed facts (e.g., eye-witness reports).

Since it is difficult to consider all of the mentioned aspects in a single visualization technique, the majority of available visualization methods address specific cases only – mostly the visualization of linear time dependencies. The approaches known in literature can basically be differentiated into techniques that visualize time-oriented data and techniques that visualize time per se. In the first case, the focus is set on representing data. Mostly quantitative, but also qualitative time-oriented attributes are represented with respect to a rather simple time axis (e.g., multivariate data represented with respect to linear time). The second case focuses on representing characteristics of the time domain and its temporal primitives, while only rather simple data representations are considered (e.g., Gantt charts to represent relations between time intervals).

It must be stressed that techniques developed for a particular time characteristic should not be applied to visualize data that exhibit different characteristics. Doing so can result in inexpressive or ineffective visual representations, and can lead to misunderstandings and false interpretations. To support the data analysis process via adequate visualization methods, it is therefore crucial to analyze the time characteristics of the dataset under investigation.

In what follows, we will illustrate the importance of choosing and parameterizing a visualization method properly with respect to given time characteristics. We will also give examples of visualization techniques that are suitable for different instances of Frank's taxonomy of types of times as presented before. Note that the considered time characteristics are used for illustrative purposes and cannot cover all aspects of the complexity of the dimension time. Frank's taxonomy encompasses more features and besides that, other taxonomies for characterizing time and visualization techniques for time-oriented data exist [Aigner, 2006; Goralwalla et al., 1998; Müller and Schumann, 2003; Silva and Catarci, 2002]. We do not intend to provide a comprehensive overview on all aspects of the dimension time, but instead focus on the importance of considering the characteristics of time for an integrated visually driven data analysis.

Linear time vs. cyclic time First, we point out the crucial influence of linear vs. cyclic time characteristic on the expressiveness of a visualization. Fig. 2.1 shows three different visual representations of the same time-oriented dataset, which contains the daily number of cases of influenza that occurred in the northern part of Germany during a period of three years. In the leftmost figure, a simple time series plot is used. Although peaks in time can be easily recognized when examining this representation, cyclic behavior of the data can only be guessed and it is hard to discern whether repetitive quantitative patterns in fact do exist. The representation is not particularly helpful in analyzing data with respect to cyclic temporal patterns.

The *Spiral Graph* [Weber et al., 2001] (see also [Carlis and Konstan, 1998; Hewagamage et al., 1999]) is a visualization technique that focuses on cyclic characteristics of time-oriented data by using a spirally shaped time axis (see Fig. 2.1 center and right). The main purpose of this technique is the detection of previously unknown periodic behavior of the data. This requires appropriate parameterization of the visualization method. The representation in the center of Fig. 2.1 is suited for cyclic time-oriented data, but it is improperly parameterized with a cycle length of 27 days; a pattern is not clearly visible. In contrast to that, the rightmost representation in Fig. 2.1 is adequately parameterized with a cycle length of 28 days, and immediately reveals a periodic pattern present in the analyzed data. The continuous differences of the number of cases between Sundays and Mondays are quite obvious. Apparently, that pattern will also be visible if the cycle length is set to 7 or 14 days.

Usually, it is difficult to find suitable parameter settings for unknown datasets. Therefore, it makes sense to support the detection of patterns either by applying analytical methods (see Section 2.3) or by animating smoothly through possible parameter settings (i.e., different cycle lengths). In the latter case, periodic behavior of the data becomes immediately apparent by the emergence of a pattern. When such a pattern is spotted, the user stops the animation and an interesting cycle length has been found.

This discussion shows that not only selecting an appropriate technique is decisive for successful visualization, but also the proper parameterization of the chosen technique. This also implies that interaction facilities are needed to

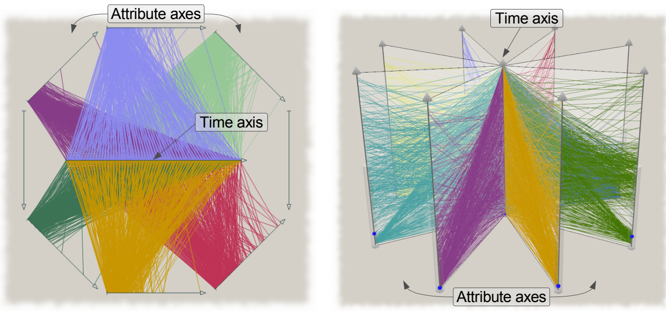


FIGURE 2.2 – **TimeWheel** – Multivariate time-oriented data represented using a TimeWheel – left: a 2D TimeWheel [Tominski et al., 2004], right: the 3D analog [Tominski et al., 2005].

allow users to re-parameterize visualization methods according to their task at hand. Only then, visualization can take full advantage of the capabilities of the human perceptual system, e.g., in recognizing patterns and motion.

Time points vs. time intervals Whether temporal attributes are conceptually modeled as time points or time intervals, is another important characteristic that influences the appropriateness of visualization methods.

Most of the known visualization techniques that represent time-oriented data consider time points. An example for a technique particularly suited for point-based time is the *TimeWheel* technique [Tominski et al., 2004]. The TimeWheel is a multi-axes representation for visualizing multivariate data over time (see Fig. 2.2). This is achieved by putting a time axis to a prominent position in the center of the display. A set of axes that encode time-dependent attributes is circularly arranged around the central time axis. For each time point in the considered data, lines descend from the time axis to the corresponding points on each of the attribute axes. The TimeWheel can be rotated to bring different attributes into the focus. Furthermore, each axis can be equipped with a slider to zoom into value ranges of interest, and in particular, to navigate the time axis. Interactive labels can be activated on demand to facilitate the identification of data values. Since the TimeWheel uses lines to represent data for each point in time, it is useful only for multivariate data that are related to time points; data based on time intervals cannot be represented.

So far, we have mentioned techniques that visualize quantitative data values related to time points. Other approaches focus on representing temporal primitives and relations among them (e.g., *LifeLines* [Plaisant et al., 1996] to visualize personal histories, or the new metaphors for visualizing temporal queries introduced by Chittaro and Combi [2003]). A technique particularly suited to visualize temporal intervals (here used to model activities) and their

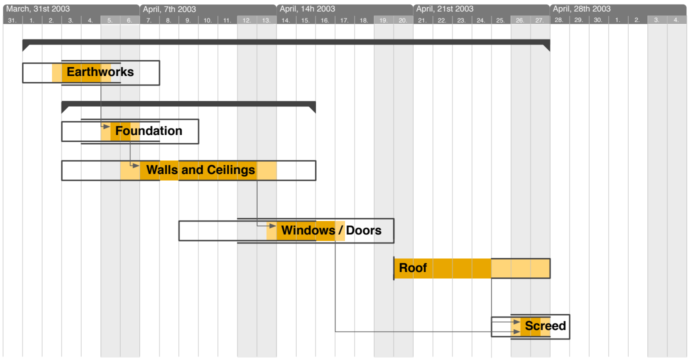


FIGURE 2.3 – **PlanningLines** [Aigner et al., 2005] – Project plan represented using PlanningLines, which allow depiction of temporal uncertainties via special glyphs.

uncertainties at a high level of detail are the *PlanningLines* [Aigner et al., 2005]. PlanningLines consist of two encapsulated bars that represent minimum and maximum duration and are bounded by two caps representing start and end intervals (see Fig. 2.3). Apart from allowing the representation of possible distributions of start, end, and duration of an activity, a second important issue is addressed by PlanningLines – temporal uncertainty. Uncertainty might be introduced by explicit specification usually connected with future planning (e.g., "The meeting will start at 11 a.m. and will take approximately one hour" – which means that it is not quite clear when the meeting will be over) or is implicitly present in cases where data are given with respect to different temporal granularities (e.g., days vs. hours). PlanningLines support interactive zooming and panning, which is particularly useful for fine-grain plans with large time scales.

Ordered time vs. branching time vs. time with multiple perspectives Although Frank's taxonomy [Frank, 1998] lists branching time and time with multiple perspectives as relevant types of time, most techniques for visualizing time-oriented data consider ordered time only.

An example of a visualization technique that assumes an ordered collection of time points is the *ThemeRiver* [Havre et al., 2002]. It represents the number of occurrences of particular news topics in print media. Each topic is displayed as a colored current that changes its width continuously as it flows through time. The overall image is a river that comprises all considered topics (see Fig. 2.4). The ThemeRiver provides an overview on what topics were important at certain points in time. Even though the ThemeRiver was originally invented to visualize thematic changes in document collections, it is also suitable to represent other quantitative data. In such cases, it is important to provide interaction techniques to rearrange the horizontal position of

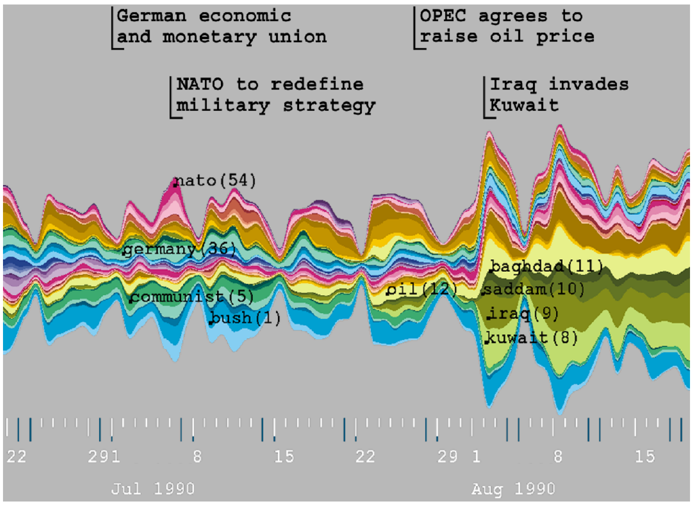


FIGURE 2.4 – **The ThemeRiver** [Havre et al., 2002] – The visual representation uses the metaphor of a river that flows through time. Currents within the river represent thematic changes in a document collection.

variables within the river. This is necessary because variables in the center of the river are perceptually emphasized, whereas variables represented at the rims of the river diminish in perceptibility.

The ThemeRiver as well as most visualization techniques known in literature are not suited to represent branching time or time with multiple perspectives. The few techniques for representing these types of time are capable of depicting only univariate qualitative data (e.g., Decision Chart [Harris, 1999] or PlanningLines [Aigner et al., 2005]), or even visualize temporal primitives only; they cannot represent multiple time-oriented variables. Here, we see the need for advanced techniques to effectively visualize multivariate data exhibiting these specific time characteristics. This is an interesting direction for future work.

The bottom line of our discussion is that the characteristics of the parameter time have to be considered when creating visual representations of time-oriented data. We also indicated that integrating appropriate interaction methods is a key concern. Interaction is mandatory to allow users to re-parameterize a visual representation, and interaction is a must to facilitate different user tasks including navigation in time, directed and undirected search, comparison, and manipulation. Similar to what we said about visualization methods, interaction facilities also need to be user- and task-specific. For example, if the main task of a user is to compare multiple time-dependent variables, it makes sense to provide interaction techniques that allow navigating the time axis or brushing certain data values in different views (e.g., [Doleisch et al., 2007; Hochheiser, 2003]). In conclusion, only an adequately chosen and properly parameterized visualization technique in combination with user- and task-specific interaction methods can fully support the development of insight into time-oriented data.

2.3 ANALYZING TIME-ORIENTED DATA

In the preceding section, we have indicated that choosing appropriate techniques, parameterizing them correctly, and incorporating useful interaction methods are essential requirements to achieve expressive and effective visual representations. When dealing with large volumes of data, additional analytical methods have to be included to derive higher levels of abstraction of the data. A large variety of time-series mining techniques have been developed in recent years[1]. Applying these techniques facilitates the interactive exploration of even huge datasets by starting with a compact overview image, which avoids overlapping of data, and then adding more details interactively [Lin et al., 2005].

From a visualizer's perspective, this fundamental procedure is expressed in Keim's *Visual Analytics Mantra* [Keim, 2005]: *"Analyze First - Show the Important - Zoom and Filter, and Analyze Further - Details on Demand."* Indeed, developing methods that fully adhere to this mantra (i.e., tightly integrate time-series mining and visualization) is a challenging task for future research.

In what follows, we describe our experiences in integrating visual and analytical methods. We will illustrate the usefulness of Keim's mantra by three examples: the concept of *temporal data abstraction*, *principal component analysis*, and *clustering*. These concepts address different concerns. Temporal data abstraction reduces value ranges from quantitative values to qualitative values, which are much easier to understand. PCA reduces the number of variables by switching the focus to major trends in the data. Clustering methods reduce the number of data tuples by finding expressive representatives for groups of tuples.

2.3.1 *Temporal Data Abstraction*

Temporal attributes are an important aspect in high-frequency domains or domains where heterogeneous data are present (e.g., the medical domain, observing human activities and behavior, or environmental monitoring). The big question is how huge volumes of continuously assessed data can be analyzed to ease further decision making. On the one hand, the data are too large to be interpreted all at once. On the other hand, the data are more erroneous than usually expected and some data are missing too. One possibility to tackle these problems is to apply knowledge-based techniques to derive *qualitative*

[1] A review of the vast body of work in time-series mining is beyond the scope of this paper. A valuable source for more information is http://www.cs.ucr.edu/~eamonn/TSDMA/ (accessed March 2007).

values or patterns of current and past situations, called *data abstraction* – a term originally introduced by Clancey in his classical proposal on heuristic classification [Clancey, 1985]. The objective of data abstraction in general is *"to create an abstraction that conveys key ideas while suppressing irrelevant details"* [Thomas and Cook, 2005]. The basic idea is to use qualitative values or patterns, rather than raw data, for further analysis or visualization processes [Lin et al., 2003]. This helps in coping with the complexity of these processes. To compute data abstractions, several tasks must be conducted (e.g., selecting relevant information, filtering out unneeded information, performing calculations, sorting, and clustering). The consequent next step is to provide techniques to visualize data abstractions in a user- and task-specific manner.

Temporal data abstraction represents an important subgroup where the processed data are time-oriented. We distinguish *basic temporal abstraction* methods (e.g., state, gradient, and rate) and more *complex temporal abstraction* methods. The basic abstraction *state* corresponds to a classification (or computational transformation) of data values. *Gradient* corresponds to the sign of the derivative of a data value, and *rate* complies with the magnitude of the derivative during an interval (e.g., abstractions: *high*, *decreasing*, and *fast* for a temperature variable). Basic temporal data abstractions alone are not always sufficient to deal with time-oriented data, because these abstractions are unable to tackle shifting contexts, different expectations concerning the development of variables, or detection of more complex patterns. Higher-order temporal abstraction methods are needed to derive unified qualitative values and patterns. Therefore, we have investigated methods of complex temporal abstraction.

VIE-VENT [Miksch et al., 1996] addresses context-sensitive and expectation-guided temporal abstraction methods in a medical application domain. The developed methods incorporate knowledge about data points, data intervals, and expected qualitative trend patterns to arrive at unified qualitative descriptions. They are based on context-aware schemata for data point transformation (see Fig. 2.5) and curve fitting to express the dynamics of and the reaction to different degrees data abnormalities. Smoothing and adjustment mechanisms are used to keep qualitative descriptions stable in case of shifting contexts or data oscillating near thresholds. For example, during intermittent positive pressure ventilation (IPPV), the transformation of the quantitative value $P_{tc}CO_2 = 56mmHg$ results in a qualitative $P_{tc}CO_2$ value of "substantially above target range"[2]. During intermittent mandatory ventilation (IMV) however, $56mmHg$ represent the "target value". Qualitative $P_{tc}CO_2$ values and schemata of curve fitting are subsequently used to decide if the value progression happens too fast, at normal rate, or too slow (see Fig. 2.6).

Qualitative descriptions and patterns as derived by temporal abstraction methods are heavily data dependent. The methods developed in the

[2] $P_{tc}CO_2$ = transcutaneous partial pressure of carbon dioxide

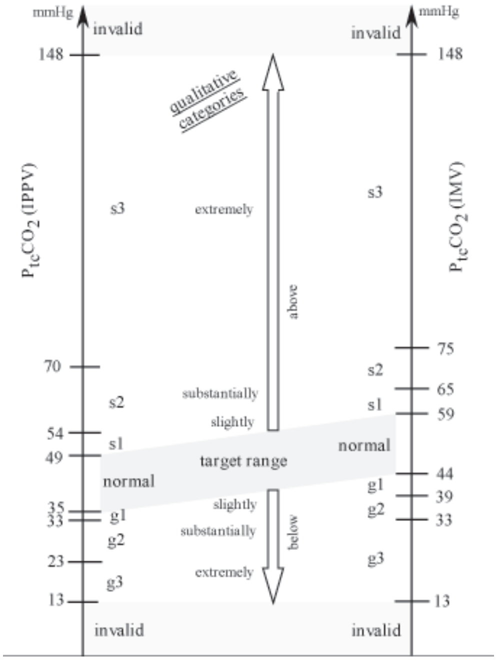


Figure 2.5 – **VIE-VENT's schemata for data-point transformation** [Miksch et al., 1996] of $P_{tc}CO_2$ during intermittent positive pressure ventilation (IPPV, left) and intermittent mandatory ventilation (IMV, right). The qualitative data point categories are given in the middle column. For example, a $P_{tc}CO_2$ value of 60 will be transformed to "substantially above target range (s2)" during IPPV and to "slightly above target range (s1)" during IMV.

VIE-VENT system are one way to deal with cases of oscillating data where abstractions and hence interpretations are frequently changing. Another solution is presented in the *The Spread* [Miksch et al., 1999]. It implements a time-oriented data abstraction method to derive steady qualitative descriptions from oscillating high-frequency data. We distinguish the following steps of processing and abstracting the data:

1. *Eliminating data errors.* Sometimes up to 40% of the input data are obviously erroneous, i.e., exceed the limits of plausible values.
2. *Clarifying the curve.* Transform the still noisy data into a steady curve with some additional information about the distribution of the data along that curve.
3. *Qualifying the curve.* Abstract quantitative values to qualitative values like "normal" or "high" and join data points with equal qualitative values to time intervals.

The Spread provides parameters to adjust the abstraction process (e.g., length of time window, permitted gaps, or points of changing the qualitative value). As an example, consider a physician who is observing continuously

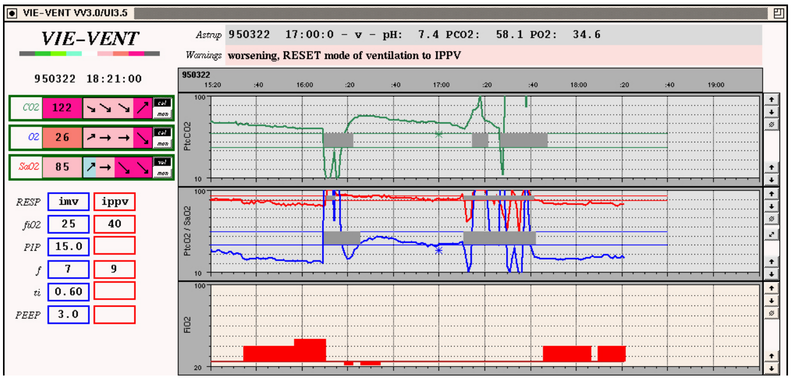


Figure 2.6 – **VIE-VENT** [Miksch et al., 1996] – The user interface of VIE-VENT. The left-hand side region shows the blood gas measurements, their corresponding qualitative temporal abstraction on the top and the actual and recommended ventilator settings below. The right-hand side region gives plots of the most important variables over the last four hours (e.g., transcutaneously assessed blood gas measurements and ventilator settings).

assessed measurements and wants to find time intervals of different qualitative regions like “*$P_{tc}CO_2$ is high for 5 minutes*”. When looking at the raw data, which typically oscillate, the physician will certainly have difficulties in finding reasonably long time spans with stable values. The Spread is able to support the physician in making qualitative assessments of the time intervals she is interested in (see Fig. 2.7).

Temporal abstraction methods as provided in VIE-VENT and The Spread are generic methods that can be used for different purposes. In the *Midgaard* project [Bade et al., 2004] these methods have been extended by several

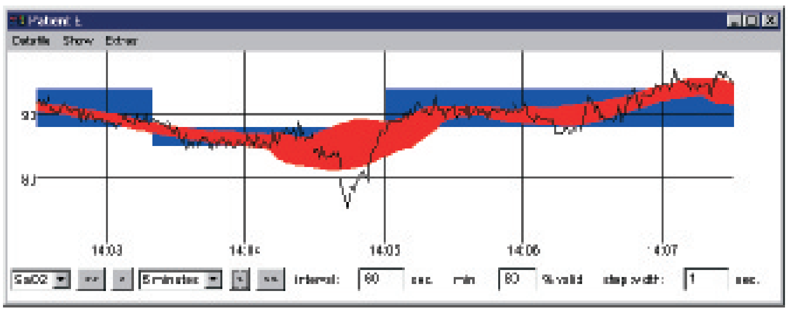


Figure 2.7 – **The Spread** [Miksch et al., 1999] – The thin line shows the raw data. The red area depicts the *Spread*, the blue rectangles represent the derived temporal intervals of steady qualitative values. Increased oscillation leads to increased width of the spread, but not to a change of the qualitative value. The lower part of the figure shows the used parameter settings.

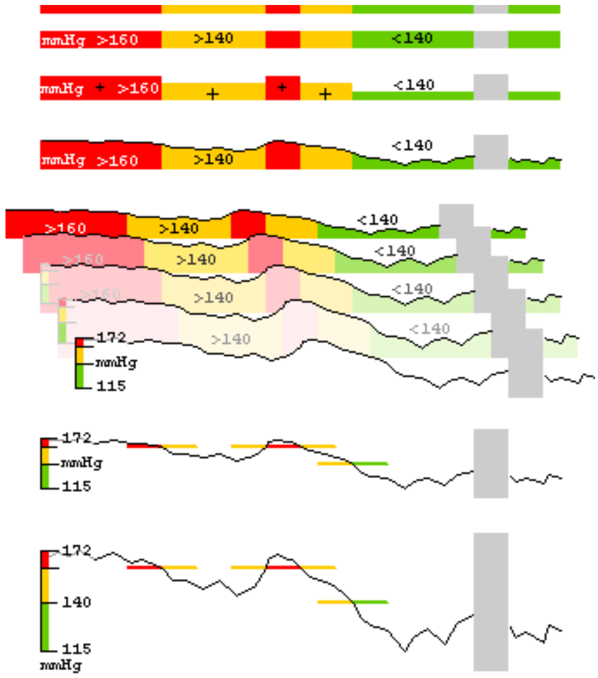


Figure 2.8 – **Midgaard** [Bade et al., 2004] – Steps of resizing/zooming the representation of a data stream from a broad overview with qualitative values to the fine structure with quantitative details (top to bottom).

visualization techniques to enhance the understanding of qualitative and quantitative characteristics of a given time-oriented dataset. The challenges were not only to support the user in exploring the data with different tasks in mind, but also to capture as much temporal information as possible on a limited display space without loss of overview and details. We provide different levels of abstractions for time-oriented data. Switching between these levels results in a smoothly integrated semantic zoom functionality (see Fig. 2.8 and left-hand side of Fig. 2.9). Our methods were designed to allow users to interact with data and time (e.g., browsing and searching the time axis). The visualization of temporal aspects comprises three linked time axes (see Fig. 2.9). The first one (bottom) provides a fixed overview of the underlying data and their full temporal range. Selecting a subrange in that time axis defines the temporal bounds for the second (middle) and the third (top) time axis. By interactively adjusting the subrange users can easily *zoom and pan* in time.

The described basic and complex temporal abstraction methods are very useful in tackling the complexity of analyzing and interpreting huge volumes of time-oriented data. We have explored the usefulness of our methods by cooperating with medical experts, who found it easy to capture severe or

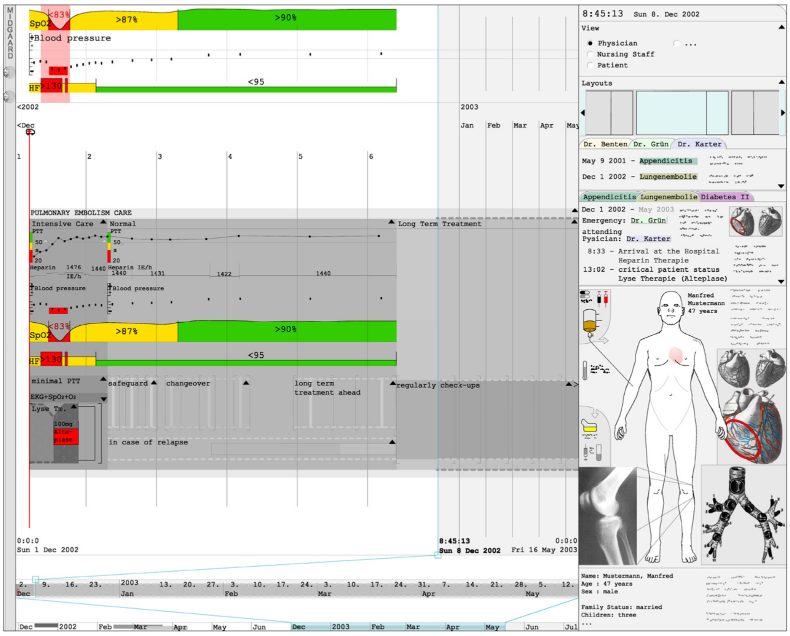


FIGURE 2.9 – **The user interface of Midgaard** [Bade et al., 2004] – The upper left part shows different measurements (e.g., blood gas measurements, blood pressure) and their corresponding temporal abstractions. The right part explains additional patient's information and the lower left part explains the time axis interaction: the selected subrange at the bottom time axis can be moved and rescaled to pan+zoom the time range shown in the middle and top time axes.

stable health conditions of patients. Moreover, these abstractions can be used for further reasoning or in guideline-based care for a simplified representation of treatment plans.

Using data abstraction is more than ever a current research topic [Lin et al., 2007]. The advantage of abstract descriptions or patterns is their unified applicability in various applications scenarios, regardless of the origin of the data to be visualized.

2.3.2 *Principal Component-Based Analysis*

As already mentioned, time-oriented data are often of multivariate nature. *Principal component analysis* (PCA) [Jolliffe, 2002] is a technique frequently applied to reduce the number of variables and to detect structure in multivariate datasets [dos Santos and Brodlie, 2004]. As such, PCA represents another approach to data abstraction. Different to the previously discussed approaches, which work on the original data space to derive qualitative data abstractions, PCA results in a transformation of the original data space into a different

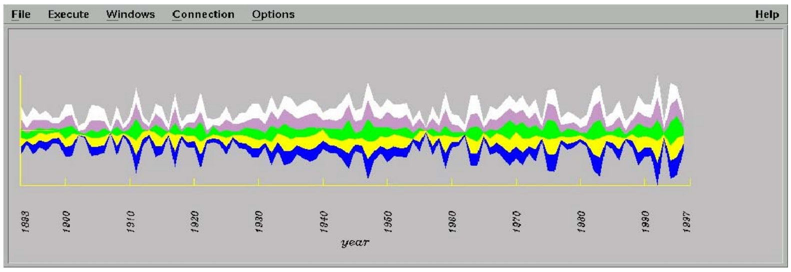


Figure 2.10 – **Visualization of a climate dataset using a ThemeRiver [Havre et al., 2000] approach.** The graph depicts five time-dependent variables: summer warmth (blue), summer days (violet), hot days (green), summer mean temperature (yellow), and mean of extreme (white) for a period of more than 100 years.

domain – the *principal component space*. The goal of this transformation is to make important trends in the data directly accessible.

The extraction of *principal components* (PCs) amounts to a variance-maximizing rotation of the original variable space. That is, the original data space is transformed in such a way that the first PC resembles most of the original dataset's variance, the second PC most of the remaining variance, and so on. Identifying these factors leads to a more compressed description of correlations in the data, and thus, to a better understanding of underlying features and trends. Since the PCA provides PCs ordered by their significance, it also offers an excellent basis for dimension reduction in case of multidimensional data. Less relevant factors can be omitted leading to a lossy, but more compact representation.

In principle, PCA does not distinguish between independent and dependent variables in this process: all variables are weighted and handled equally. As mentioned before, this often raises problems in the context of time-oriented data. In particular, the temporal context gets lost and the interpretation gets hampered. Therefore, it is preferable to exclude the independent variable "time" from PCA. Time and computed PCs should be rejoined to restore the temporal context afterwards.

To demonstrate the strengths of combining PCA with visualization we will take a look at a simple example. The data we consider is related to climate research. The basis of the example is a meteorological dataset that contains daily observations of temperature, precipitation, wind, air pressure, and others for a period of more than 36,500 days (100 years). To analyze the development of global warming over the last century, we cooperated with climate researchers to derive a dataset that focuses on summer weather conditions only [Nocke et al., 2004]. That condensed dataset is on a yearly basis and comprises five variables: summer warmth (sum of max temperatures for days with $T_{max} \geq 20^oC$), summer days (number of days with $T_{max} \geq 25^oC$), hot days (number of days with $T_{max} \geq 30^oC$), summer mean temperature (mean of daily average temperature T_{avg}), and mean of extreme (mean of daily max temperatures T_{max}). All five are quantitative variables that either count days with specific weather conditions or contain aggregated temperature information; their strong correlation has been intended by the climate researchers involved.

The condensed dataset can be visualized with a *ThemeRiver* (see Fig. 2.10). In this graph, constrictions in the river stand for low data values, which indicate particularly cold summers. Broad flow snapshots characterize particularly hot summers. On first impression, a general overview and important characteristics of the dataset are depicted well.

We will now show how PCA and an additional simple bar chart representation can help to derive further information from the data. To find major trends in terms of climate change, PCA was applied to the condensed dataset. Time was excluded from the analysis to retain the temporal context. The bar graph in Fig. 2.11 depicts the first PC only (i.e., PC0), to which all variables contribute. Bars above the time axis represent hot summers, whereas bars below the time axis stand for colder summers. Additionally, a color-coding of PC frequencies was added to enhance expressiveness: Orange bars represent outliers, whereas blue bars represent more common conditions (the colors are not related to temperature). The combination of PCA and simple visualization succeeds in presenting major trends in the data very clearly: Average warm summers dominate the first third of the century, containing also the coldest summers (orange bars below the time axis). Hot outlier summers cumulating at the end of the century can also be detected very easily (orange bars above the time axis). Moreover, two additional converse trends can be identified: Hot summers occur followed by colder summers in the end of this period. In the last third hot summers preponderate, with the warmest summers at all. The PC visualization in Fig. 2.11 depicts corresponding trends very well. This demonstrates the value of PC-based temporal abstractions in the visual analysis of time-dependent data. Nonetheless, one should recall that our condensed climate dataset represents a special case where all variables are strongly correlated. That correlation is the reason why PC0 separates warm and cold summers so well. When analyzing arbitrary temporal datasets, further PCs may be necessary to describe all trends. In such cases, not only more responsibility of the user is required, but also flexible mechanisms and controls are needed to determine variables that should be considered for PCA and to select PCs that should be visualized. This calls for an integration of analytical analysis and visualization in a single tool.

As mentioned above, PCA represents an almost completely automatic approach for temporal data abstraction. The advantage is that a user can get an abstracted view on the data very easily. Nonetheless, it is sometimes hard to relate patterns visible in PC space to original data variables and the

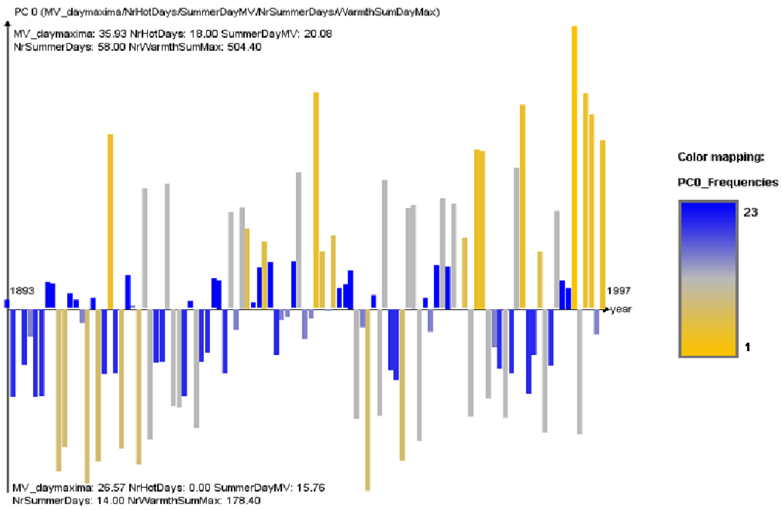


FIGURE 2.11 – **Bar chart visualization of PC0 over time for the dataset from Fig. 2.10.** Upward bars represent warmer conditions, whereas downward bars stand for colder summers (but not necessarily negative temperatures). Frequencies of data values are mapped onto color to further distinguish typical (blue) and outlier (orange) years. Major trends are clearly visible: The first third of the time line is dominated by average warm summers mixed with the coldest summers; hot summers occur followed by cold summers in the end of this period; in general, outlier summers cumulate at the end of the time line.

abstracted views are not always easy to interpret. What can help in these cases are approaches to enhance readability of PC-based diagrams by incorporating additional information or interactive means to support relating PCs to original data [Müller et al., 2006]. Still, there is room to improve the expressiveness of PC-based visualization in further research.

2.3.3 *Clustering*

After discussing temporal data abstraction and dimension reduction with PCA, we now want to take a closer look on data aggregation. Clustering methods provide a basis for this purpose. Clustering relates to partitioning a dataset into subsets exhibiting a certain similarity. The clustering process also provides an abstraction of the data. Concentrating on the *clusters*, rather than on individual data values allows for an analysis of datasets with a much larger number of tuples. Appropriate *distance or similarity measures* lay the ground for clustering. Distance and similarity measures are profoundly application dependent. This has lead to a large number of different measures and clustering algorithms [Jain et al., 1999]. Selecting appropriate algorithms is typically difficult. Careful adjustment of parameters and regular validation of the results are also essential tasks in the process of clustering. Different

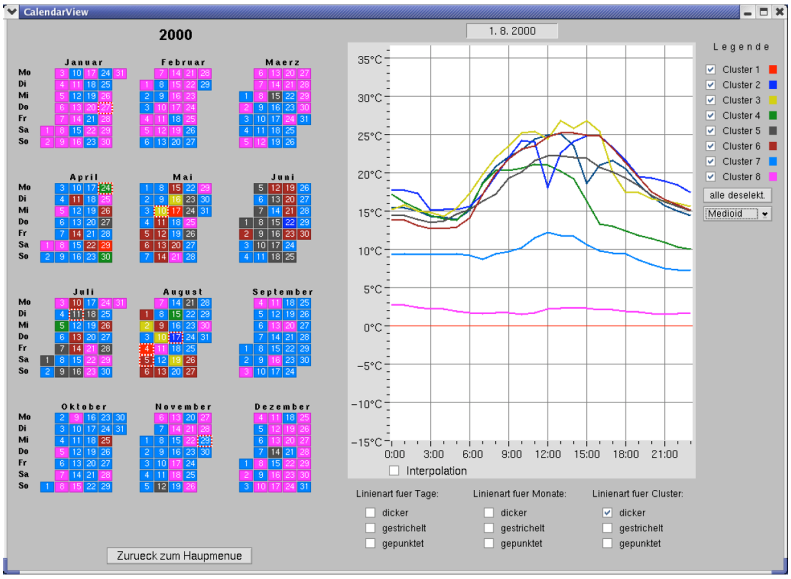


FIGURE 2.12 – **Cluster Calendar View** [Nocke et al., 2003; Van Wijk and Van Selow, 1999] – The plots show cluster representatives as daily temperature profiles. The calendar view illustrates by color which days belong to which cluster, i.e., show similar profiles.

to PCA, the variable "time" is typically included in the clustering process to reveal clustering with respect to temporal aspects. The resulting clustering may also lead to a temporal data abstraction.

Visualization has been frequently applied to validate and guide the clustering process. Different mining tools provide cluster algorithms and techniques to visualize the clustering results. However, most of the techniques for visualizing clusters neglect the temporal context, thus making it difficult to analyze data with respect to fundamental time-oriented tasks (e.g., to associate data values and clusters with particular time steps).

A technique specifically designed for the analysis of clustered time-oriented data is the *Cluster Calendar View* [Van Wijk and Van Selow, 1999] (see Fig. 2.12). It applies a calendar metaphor to represent the temporal context. Cluster affiliation is presented indirectly by color-coding. A line plot presents details on trends subsumed in selected clusters. Fig. 2.12 shows an example in the context of meteorological data. In this example, clusters 7 (light blue) and 8 (magenta) represent typical daily temperature curves and hence dominate the calendar. All the other clusters are more or less atypical and represent outliers. Furthermore, the color-coded calendar allows to reveal fast changes in cluster sequences for example in the first part of August. Brushing techniques provide additional support in the exploration process. For instance, we can highlight clusters that are similar to a selected cluster. The Cluster

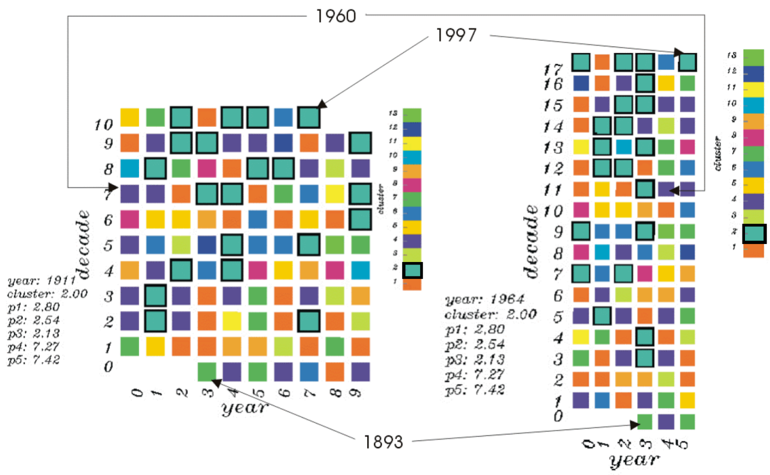


FIGURE 2.13 – **Rectangular View** – Visualization of a temporal clustering of meteorological time-oriented data from the Potsdam observation station. Thirteen clusters of yearly temperature curves have been extracted from the data. In this example, changing the periodicity (denoted as decade) from 10 years (left) to 6 years (right) helps in identifying a temporal pattern for cluster 2 (see [Nocke et al., 2003]).

Calendar View facilitates comparison of cluster representatives (overview), exploration of the values of a single cluster representative (abstract detail), and exploration of daily and monthly values of interest (specific details).

In contrast to the Cluster Calendar View, the *Rectangular View* [Nocke et al., 2003] depicts cluster information directly, thus allowing for the display of data for much larger time frames. The Rectangular View utilizes a tablet-like layout to present clusters as well as cluster centroids. Each cluster is visualized as a color-coded square. Clusters are positioned on the tablet from the lower left to the upper right with respect to their temporal location. Various interaction techniques extend the functionality. Temporal brushing allows to focus on specific time steps. Interactive modification of the cluster arrangement helps in detecting and understanding temporal patterns. In Fig. 2.13, for instance, a certain periodicity of cluster 2 can be observed when placing 6 years per row instead of 10. While this cluster appears frequently in columns 1 to 3, it is less existent in all other columns (0, 4, 5). The implication of a quasi-6-year cycle leads to new explanations and models on the transition from stable climatic states to new ones for the previously introduced meteorological dataset.

In this section, we demonstrated the usefulness of analytical methods to gain insight into larger volumes of time-oriented data. Temporal data abstraction aims at gaining qualitative high-level insights. Principal component analysis and clustering help in handling larger numbers of variables respectively tuples in time-oriented data. All three methods applied to large time-oriented datasets provide different levels of abstraction and help to reveal major trends in the data.

Many more time-series analysis methods are known in literature. The information gained by these methods can be utilized to further support different steps in the analysis and visualization process to provide additional guidance to users. For instance, Seo and Shneiderman [2005] and Müller et al. [2006] present interactive techniques for data selection and attribute mapping based on information from clustering and PCA; Keogh et al. [2002] integrate mining methods to drive interactive visual exploration of time-series.

2.4 USER-CENTERED ANALYSIS VIA EVENTS

The methods presented in the previous sections are useful tools to facilitate visualization and analysis of time-oriented data. We already indicated that this is true only if the methods are parameterized according to the users' needs and tasks. This brings us to the third major point of our discussion – the user. User interaction is a way to manually parameterize the described visualization and analysis tools. Many tools provide an interactive graphical user interface to adjust the parameters of analytical methods (e.g., via sliders or check boxes). Visualization views can usually be adjusted via common view navigation (zoom, pan, rotation) [Henriksen et al., 2004], dynamic queries [Keogh et al., 2002], and brushing [Doleisch et al., 2007].

However, it is not always easy for users to find parameter values that suit the analysis task at hand. Particularly analytical methods often have parameters that are not self-explanatory, and hence, are not easy to set. Moreover, the increasing complexity of visualization methods makes it more difficult for users to parameterize the visualization properly. What is needed is some form of support that helps in steering the visual analysis. A promising concept that addresses the automatic parameterization of visual representations is *event-based visualization* [Tominski, 2006]. The thought behind this concept is to gain benefit from incorporating visualization and event methodology. Commonly, events are considered happenings of interest that trigger some automatic actions. This concept is prevalent in various application fields, including active databases, software engineering, and software visualization.

In our understanding, events occur if user-defined conditions, which are expressed with respect to entities of a dataset, become true. The basic idea of event-based visualization is to let users specify their interests as *event types* (i.e., encapsulations of conditions), to determine if and where these interests match in the data (i.e., detect *event instances*), and to consider detected event instances when generating the visual representation. This basic procedure requires three main steps - *1) event specification*, *2) event detection*, and *3) event representation*. We will give detailed descriptions on

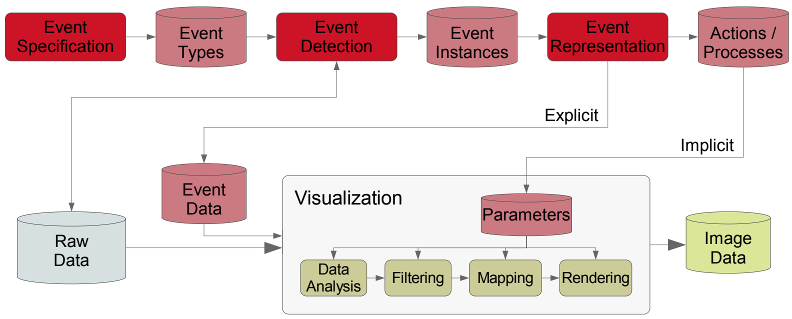


FIGURE 2.14 – **Model of event-based visualization** – The figure shows the major steps of event-based visualization (event specification, event detection, and event representation) attached to the well-known visualization pipeline.

each of these steps in the next paragraphs. Fig. 2.14 illustrates how event-related components can be attached to the visualization pipeline (see [dos Santos and Brodlie, 2004]), which internally comprises data analysis, filtering, mapping, and rendering. Data analysis and filtering can be realized by the methods presented in Section 2.3. How time-oriented data can be mapped (and rendered) to graphical representation was shown in Section 2.2.

Describing user interests

The *event specification* is the step where users describe their interests. To be able to find actual matches of user interests in the data, the event specification must be based on formal descriptions. For this purpose, event formulas have been developed. These formulas make use of elements of predicate logic, including variables, predicates, functions, aggregate functions, logical operators, and quantifiers. The elements may be used in a well defined way to create valid event formulas. We consider different variants of event types to facilitate the specification of interests with respect to relational datasets. Tuple event types can be used to detect interesting data tuples (e.g., tuples that show an exceeded threshold) and attribute event types are useful for finding attributes of interest (e.g., attribute with the highest average value). For an analysis of time-oriented data, this alone is not sufficient. Therefore, sequence event types are also supported. They enable users to specify conditions of interest regarding temporally ordered sequences of tuples (e.g., sequence of days with rising stocks). Sequence event types extend the existing event formulas with sequence-related notations (inspired by Sadri et al. [2004]). A combination of event types to composite event types is also possible. They are realized via set operators. Because we rely on extended predicate logic and set theory, the expressiveness of the introduced event types is limited to these formalisms. However, the model of event-based visualization is not limited to certain fixed event types, but can be extended with event types as required

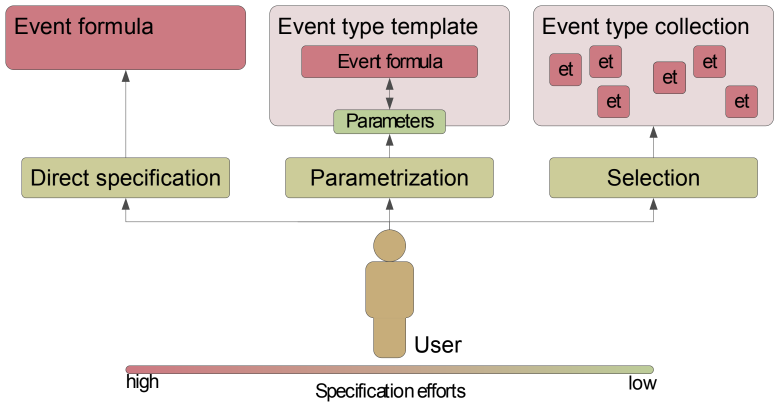


FIGURE 2.15 – **User-centered event specification model** – Event types can be specified by using event formulas directly (left), by parameterizing event type templates (middle), or by selecting from a predefined application-specific collection of event types (right). The effort required for event specification decreases in the same order.

for particular application contexts.

To give a simple example of a sequence event type, we will formulate the following interest: *"Find three successive days with increases of more than 15% in the number of influenza infections."* This interest is expressed as $\{(x,y,z)_{date} \mid z.flu \geq y.flu * 1.15 \wedge y.flu \geq x.flu * 1.15\}$. The first part of the formula defines three variables $(x,y,z)_{date}$ that are sequenced by date. To express the condition of interest, these three variables are set into relation using predicates, functions, and logical connectors.

Certainly, common users will have difficulties in describing their interests by using event formulas directly. To facilitate the specification of interests as formal event types, we developed a model for user-centered event specification. This model provides expert, common, and less-experienced visualization users with different specification methods. The three different levels of the model are *direct specification*, *specification by parameterization*, and *specification by selection* (see Fig. 2.15).

Although the model is based on the described event formulas, the complete functionality of these formulas is available only to expert users at the level of direct specification.

To ease the event specification for common users, so called event type templates are provided. Basically, the idea was to hide the complexity of event formulas from users. Event type templates use an internal event formula that cannot be changed directly, but can be adjusted to the users' needs via easy to set parameters. An example of an event type template is a threshold template where the two parameters "threshold" and "variable" can be set by users. An event instance is detected once the chosen variable exceeds the set threshold. Templates are particularly useful to encapsulate sequence event types. Interests like an increase of a "variable" over a certain "period of time" can be easily adjusted to the task at hand without typing entirely new event formulas.

The third level of event specification is based on simple selection. The event specification by selection addresses not only less-experienced visualization users, but also users (e.g., managers) who seek quick access to relevant information contained in the data to be analyzed. The idea is to provide a collection of expert-defined event types that are particularly tailored to the application context. In the case of time-oriented data, visualization tasks like identification of certain values in time or detection of behavioral patterns (e.g., the afore-mentioned increase in cases of influenza) could be formulated by domain experts. Predefined event types must be assigned with expressive labels and descriptions, so that users can easily select the event types they are interested in. It is also helpful to enhance the event collection with a semantic structure (e.g., by grouping the collection with respect to different user tasks). Again, to devise such a semantic structure and to describe it expressively is a task for domain experts.

Finding relevant data portions

The *event detection* step determines whether the interests defined as event types are present in the dataset under consideration. Conducting the event detection results in a set of event instances, which describe where in the data interesting information is located. That is, entities that comply with user interest are marked as event instances. For event detection, the variables used in event formulas are substituted with concrete entities of the dataset (tuples, attributes, or sequences of tuples). In a second step, predicates, functions, and logical connections are evaluated, so that the event formula as a whole can be evaluated to either true or false. Since this procedure is very costly in terms of computation time, efficient methods must be utilized for the event detection. For detecting interesting tuples and attributes, capabilities of relational database management systems can be utilized. The detection of sequence events makes use of the OPS algorithm [Sadri et al., 2004], which has proved to be efficient for querying sequenced data. If dynamic data (i.e., data that change over time) have to be considered, detection efficiency becomes crucial. Here, incremental detection methods can help. Such methods operate on a differential dataset, rather than on the whole data. However, incremental methods also impose restrictions on possible event types.

Considering user interests in visual representations

The last important step of event-based visualization is the *event representation*. The goal of this step is to incorporate detected event instances (which reflect the interests of the user) into visual representations. We identified three requirements that have to be accomplished in this regard:

1. Communicate the fact that something interesting has been found.
2. Emphasize interesting data among the rest of the data.
3. Convey what makes the data interesting.

The most important requirement is that the visual representation must reflect that something interesting is contained in the data. This is essential for event-based visualization of time-oriented data. To meet this requirement, easy to perceive visual cues (e.g., a red frame around the visual representation, exclamation marks, or annotations) are used. Alpha blending can be applied to fade out past events. The second requirement aims at emphasizing those parts of the visual representation that are of interest. Additionally, the visualization should communicate what makes the highlighted parts interesting (i.e., what is the particular event type). However, facing arbitrarily definable event formulas, this last requirement is difficult to accomplish.

We distinguish two basic possibilities for representing events. On the one hand, it makes sense to visualize event instances, rather than the whole dataset. In this way, the focus is set exclusively on the interests of the user. Since the number of events is usually smaller than the number of data items, even large datasets can be analyzed (certainly, the same holds true for principal components and clusters as presented in Section 2.3). This way of representing events is referred to as *explicit event representation*. On the other hand, adjusting the parameters of visual representations according to occurred event instances is a promising alternative. By pursuing what we call *implicit event representation*, we can automatically set visualization parameters according to interests detected in the data. If we assume that user interests are related to user tasks and vice versa, implicit event representation can help to achieve better targeted visual representations. The big challenge is to meet the above stated requirements merely by adapting visualization parameters. Apparently, availability of adequate visualization parameters is a prerequisite for implicit event representation.

To illustrate the potential of event-based visualization, we will discuss an example. We assume a user who has to search time-dependent human health data for uncommonly high numbers of cases of influenza. The task at hand is to detect where in time these situations have occurred. A possible way to accomplish this task is to use the TimeWheel technique [Tominski et al., 2004]. However, without event integration the user will be provided with a TimeWheel that uses a standard parameterization (see Fig. 2.16(a)). The standard view shows influenza on the upper left axis (light green), time is represented on the central axis. Alpha-blending has been applied by default to reduce visual clutter. From the TimeWheel in Fig. 2.16(a) one can only guess from the labels of the axis showing influenza that there are higher numbers of cases; the alpha-blending made the particular lines almost invisible (see question mark). Several interaction steps are necessary to re-parameterize the TimeWheel to accomplish the task at hand.

In contrast to that, in an event-based visualization environment, the user can specify the interest *"Find days with a high number of cases of influenza."* as an event type ($\{x \mid x.flu \geq 300\}$) to be considered for the current analysis task. The event type can be stored and may be reused in further visualization sessions or by other users. If a new dataset is opened or if new tuples are added dynamically to a time-oriented dataset, the event detection is run to determine whether or not the data conform to the condition expressed in the event type. If this is the case, event instances are created for those data portions that fulfill the condition. To reflect the interest of the data analyst, i.e., to provide an individually adjusted TimeWheel, the parameters of the visual representation have to be altered. Parameter changes can be implemented either as instantaneous *actions* or gradual *processes* (e.g., smooth animation). In our particular example, we use an action that switches color and transparency of line segments representing event instances. Days with high numbers of influenza cases are excluded from alpha-blending and are drawn in white color. Additionally, the TimeWheel is rotated (as a whole) such that the axis representing influenza is moved gradually to an exposed position. The application of a gradual process is important in this case to support users in maintaining their mental map of the visual representation. The result of applying parameter changes as response to event instances is depicted in Fig. 2.16(b). This figure illustrates that event-based visualization eases the visual analysis of time-oriented data significantly, since the visual representation is adapted to the current visualization task. In the example, the identification of days with higher numbers of influenza infections is easy.

As the previous example indicates, considering user interests helps to achieve better targeted visual representations. By combining event-based methodology with visualization approaches, we give users the opportunity to describe their interests. The described event types address not only tuples and attributes of relational data, but also sequences of tuples, which are important when dealing with time-oriented data. By using predicate logic, a high level of flexibility is achieved; a wide range of concrete event types can be imagined. It must also be mentioned that our approach has been developed to support directed search, i.e., users know what they are looking for. Being aware of what users are interested in, we are able to automatically generate visualizations that are potentially more helpful for the users' task at hand than standard representations. By focusing on relevant parts of the data, we also achieve another level of data abstraction.

Until now, event-based visualization is not suited to automatically mine potential events in time-oriented data, i.e., to support undirected search, where users have no hypotheses about the data. With a tighter integration of visual and analytical methods, it should be possible to alleviate this concern. A second challenge for future work is to find general guidelines on how to

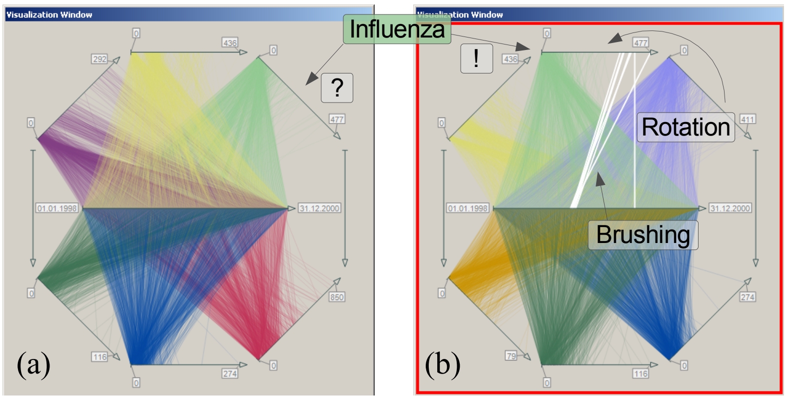


FIGURE 2.16 – **Standard vs. automatic parameterization of a TimeWheel** – (a) TimeWheel representing a time-dependent health dataset; the interests of the user are not considered in the standard parameterization, which aims at showing main trends. (b) TimeWheel representing the same data; the user's interests were recognized in the data and have been emphasized via highlighted lines and automatic rotation; the presentation is better targeted for the user's task at hand.

realize parameter changes that indeed highlight event instances. Because the parameter space of visualization methods is usually very large and contains many interdependencies, we have to apply sophisticated methods (e.g., as suggested by House et al. [2006]) to find and test appropriate parameter settings.

2.5 CONCLUSION

In this paper, we have investigated the role of time-oriented data in the context of visually driven data analysis. We have elaborated on the importance of choosing and parameterizing visualization techniques and interaction functionality properly with respect to characteristics of the time domain present in the data. However, in the light of huge datasets, visualizing all data in a comprehensible manner without burying possibly important information becomes more and more challenging. This challenge can be dealt with by conducting additional data analysis steps; many time-series analysis approaches are known in literature. By the examples of temporal data abstraction, PCA, and clustering, we have illustrated that analytical methods support the identification of the important in time-oriented datasets. The third question we addressed concerns the integration of the user into the visual analysis process. We detailed on an approach to emphasize relevant information, called event-based visualization. This approach is mainly task-driven and aims at generating better targeted visual representations of time-oriented data (e.g., by automatic highlighting of relevant data as well as hiding of less relevant data).

Nonetheless, much more work has to be conducted in the future to support a comprehensive visual analysis. This includes the development of expressive visualization techniques for all kinds of time-oriented data. Especially multivariate data in the context of non-linear time domains as well as interval-based data and temporal uncertainties have to be considered to an increasing degree. A particularly challenging problem is to find new ways of describing tasks of the visual exploration process and to automatically adapt the whole analysis procedure according to the tasks at hand. This also includes specific interaction functions for investigating time-dependencies. For example, Doleisch et al. [2007] introduce different brushing functions that could be useful in this regard. Finally, studying tighter combinations of analysis steps and event-based visualization (e.g., to detect events on temporal data abstractions) could result in new powerful means for the visual analysis of time-oriented data.

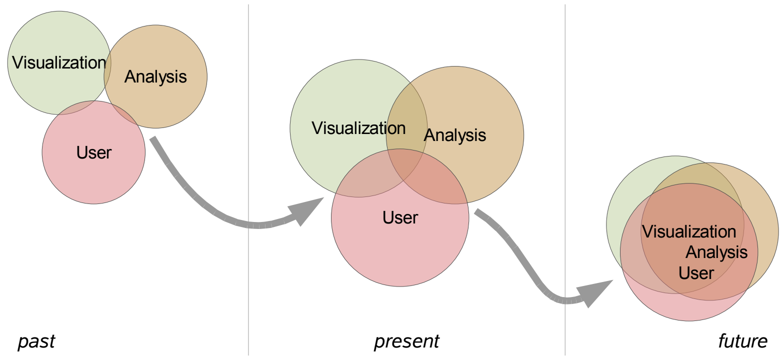


FIGURE 2.17 – To further advance a visually driven analysis of time-oriented data, it is necessary to integrate visual, analytical, and user-centered methods more tightly.

As a conclusion of our paper we would like to take a look at Fig. 2.17. Each distinct research field shown in the figure has yielded many powerful approaches. With this paper we tried to make a point on a better integration of visual, analytical, and user-centered methods. We suggest that these aspects are further advanced in a direction that leads to convergence of user-centered, visually driven analysis methods for time-oriented data.

ACKNOWLEDGMENTS

We would like to thank all people who have contributed to the many examples we present in this paper. The anonymous reviewers deserve special thanks for their valuable comments. The work presented here was partially supported by the DFG (German Research Foundation) and by the “Fonds zur Förderung der wissenschaftlichen Forschung - FWF” (Austrian Science Fund), grant P15467-INF.

REFERENCES

Aigner, W. (2006). *Visualization of Time and Time-Oriented Information: Challenges and Conceptual Design*. PhD thesis, Vienna University of Technology.

Aigner, W., Miksch, S., Thurnher, B., and Biffl, S. (2005). PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proc. of the 9th Intl. Conf. on Information Visualisation (IV05)*. IEEE Press.

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proc. of the 2004 Conf. on Human Factors in Computing Systems (CHI04)*, pages 105–112. ACM Press.

Carlis, J. V. and Konstan, J. A. (1998). Interactive Visualization of Serial Periodic Data. In *Proc. of Symposium on User Interface Software and Technology (UIST)*.

Chittaro, L. and Combi, C. (2003). Visualizing Queries on Databases of Temporal Histories: New Metaphors and their Evaluation. *Data and Knowledge Engineering*, 44(2):239–264.

Clancey, W. J. (1985). Heuristic Classification. *Artificial Intelligence*, 27:289–350.

Doleisch, H., Hauser, H., Gasser, M., and Kosara, R. (to appear 2007). Interactive Focus+Context Analysis of Large, Time-Dependent Flow Simulation Data. *Transactions of the Society for Modeling and Simulation International*.

dos Santos, S. and Brodlie, K. (2004). Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28:311–325.

Frank, A. U. (1998). Different Types of "Times" in GIS. In Egenhofer, M. J. and Golledge, R. G., editors, *Spatial and Temporal Reasoning in Geographic Information Systems*. Oxford University Press, New York.

Goralwalla, I. A., Özsu, M. T., and Szafron, D. (1998). An Object-Oriented Framework for Temporal Data Models. In et al., E., editor, *Temporal Databases: Research and Practice*, pages 1–35. Springer.

Hajnicz, E. (1996). *Time Structures: Formal Description and Algorithmic Representation*. Number 1047 in Lecture Notes in Computer Science. Springer-Verlag, Berlin.

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press.

Havre, S., Hetzler, E., and Nowell, L. (2000). ThemeRiver: Visualizing Theme Changes Over Time. In *Proc. IEEE Symp. on Information Visualization (InfoVis'00)*, pages 115–123, Salt Lake City, USA.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. (2002). ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20.

Henriksen, K., Sporring, J., and Hornbaek, K. (2004). Virtual Trackballs Revisited. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):206–216.

Hewagamage, K. P., Hirakawa, M., and Ichikawa, T. (1999). Interactive Visualization of Spatiotemporal Patterns Using Spirals on a Geographical Map. In *Proceedings of Symposium on Visual Languages (VL)*, Tokyo, Japan.

Hochheiser, H. (2003). *Interactive Graphical Querying of Time Series and Linear Sequence Data Sets*. PhD thesis, University of Maryland.

House, D. H., Bair, A. S., and Ware, C. (2006). An Approach to the Perceptual Optimization of Complex Visualizations. *IEEE Transactions on Visulaization and Computer Graphics*, 12(4):509–521.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer Verlag, New York, 2nd edition.

Keim, D. (2005). Scaling Visual Analytics to Very Large Data Sets. Workshop on Visual Analytics, Darmstadt, June 2005.

Keogh, E., Hochheiser, H., and Shneiderman, B. (2002). An Augmented Visual Query Mechanism for Finding Patterns in Time Series Data. In *Proc. Fifth International Conference on Flexible Query Answering Systems*. Springer-Verlag.

Lin, J., Keogh, E., and Lonardi, S. (2005). Visualizing and Discovering Non-Trivial Patterns in Large Time Series Databases. *Information Visualization*, 4(2):61–82.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proc. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. ACM Press.

Lin, J., Keogh, E., Wei, L., and Lonardi, S. (2007). Experiencing SAX: a Novel Symbolic Representation of Time Series. *Data Mining and Knowledge Discovery*. to appear.

Miksch, S., Horn, W., Popow, C., and Paky, F. (1996). Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. *AI in Medicine*, 8(6):543–576.

Miksch, S., Seyfang, A., Horn, W., and Popow, C. (1999). Abstracting Steady Qualitative Descriptions over Time from Noisy, High-Frequency Data. In *Proc. of the Joint European Conf. on AI in Medicine and Med. Decision Making (AIMDM'99)*, pages 281–290. Springer, Berlin.

Müller, W. and Schumann, H. (2003). Visualization Methods for Time-dependent Data - an Overview. In *Proc. of Winter Simulation 2003*, New Orleans, USA.

Müller, W., Nocke, T., and Schumann, H. (2006). Enhancing the Visualization Process with Principal Component Analysis to Support the Exploration of Trends. In *Proc. of APVIS'06*.

Nocke, T., Schumann, H., and Böhm, U. (2004). Methods for the Visualization of Clustered Climate Data. *Computational Statistics*, 19(1):75–94.

Nocke, T., Schumann, H., Böhm, U., and Flechsig, M. (2003). Information Visualization Supporting Modeling and Evaluation Tasks for Climate Models. In *Proc. of Winter Simulation 2003*, New Orleans, USA.

Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. (1996). LifeLines: Visualizing Personal Histories. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press.

Sadri, R., Zaniolo, C., Zarkesh, A., and Adibi, J. (2004). Expressing and Optimizing Sequence Queries in Database Systems. *ACM Transactions on Database Systems*, 29(2):282–318.

Seo, J. and Shneiderman, B. (2005). A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization*, 4(2):99–113.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. of the IEEE Symp. on Visual Languages*, pages 336–343. IEEE Press.

Silva, S. F. and Catarci, T. (2002). Visualization of Linear Time-Oriented Data: a Survey (Extended version). *Journal of Applied System Studies*, 3(2).

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press.

Thomas, J. J. and Cook, K. A. (2006). A Visual Analytics Agenda. *IEEE Computer Graphics and Applications*, 26(1):10–13.

Tominski, C. (2006). *Event-Based Visualization for User-Centered Visual Analysis*. PhD thesis, University of Rostock.

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proc. of ACM Symp. on Applied Computing*, pages 1242–1247. ACM Press.

Tominski, C., Abello, J., and Schumann, H. (2005). Interactive Poster: 3D Axes-Based Visualizations for Time Series Data. In *Poster Compendium of IEEE Symp. on Information Visualization (InfoVis'05)*, Minneapolis, USA.

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proc. of the IEEE Symp. on Information Visualization 1999 (InfoVis'99)*, pages 4–9.

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing Time-Series on Spirals. In *Proc. of the IEEE Symp. on Information Visualization 2001 (InfoVis01)*, pages 7–14.

CHAPTER 3

Visualizing Time-Oriented Data – A Systematic View

Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski

Abstract · *The analysis of time-oriented data is an important task in many application scenarios. In recent years, a variety of techniques for visualizing such data have been published. This variety makes it difficult for prospective users to select methods or tools that are useful for their particular task at hand. In this article, we develop and discuss a systematic view on the diversity of methods for visualizing time-oriented data. With the proposed categorization we try to untangle the visualization of time-oriented data, which is such an important concern in Visual Analytics. The categorization is not only helpful for users, but also for researchers to identify future tasks in Visual Analytics.*

Keywords · Visual analytics, visualization, time-oriented data.

This article originally appeared as [Aigner et al., 2007]:

Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2007). Visualizing Time-Oriented Data – A Systematic View. *Computers & Graphics*, 31(3):401–409.

3.1 INTRODUCTION

TIME is an outstanding dimension. For ages, scientists have been thinking about meaning and implications of time. Understanding temporal relations enables us to learn from the past to predict, plan, and build the future. This rationale can be found throughout sciences. Hence, it is no surprise that time is also a key concern in Visual Analytics, where the goal is to support the knowledge crystallization process with appropriate analytical and visual methods [Thomas and Cook, 2006].

Visualizing time-oriented data, which is the focus of this paper, is not an easy business. Even though many approaches to this task have been published in recent years, most of them are specific to only a particular analysis problem. The reason why most methods are highly customized is simple: It is enormously difficult to consider all aspects involved when visualizing time-oriented data. Time itself has many theoretical and practical aspects. For instance, time points and time intervals use different sets of temporal relations. It also matters if we interpret time as a linearly ordered set of temporal primitives, or if we assume the temporal primitives to recur cyclically. The data that tie to the time axis are another decisive concern. Do we have a single variable per temporal primitive or are there multiple variables we have to consider? Moreover, data can be abstract or can be bound to a spatial frame of reference. Many more data-related questions have to be thought of when designing visual analysis methods. Only if the characteristics of the data are taken into account is it possible to generate expressive visual representations. Finally, visual representations themselves imply the need of thinking about representational and perceptual issues.

All these aspects are important when applying or developing visual methods for analyzing data that are connected to time. The problem is that the diversity of the involved aspects makes it difficult for practitioners to find appropriate solutions for their task at hand, and difficult for researchers to identify directions for future work to bring forward the visualization of time-oriented data.

In this paper, we develop a systematic view on the visualization of time-oriented data. We are aware that this is not an easy endeavor. Our categorization must be specific to be useful for others. A too general view would not be of much help in alleviating the addressed problem. A very fine-grain categorization is not desirable because categories would hardly be distinctive. What we aim for with this article is to initiate categorization of visual concepts for analyzing time-oriented data.

In Section 3.2, we will explain the basics of visualizing time-oriented data. We describe why time is important and what makes time worth special consideration in the context of visual analysis methods. Our attempt to categorize approaches for visualizing time-oriented data is presented in Section 3.3. The categorization will be illustrated with examples from visualization literature;

it is not our intent to provide a comprehensive state-of-the art overview. A discussion of the proposed categorization and its implication is provided in Section 3.4. Our paper summarizes the made statements and gives an outlook on future work in Section 3.5.

3.2 BASIC CONSIDERATIONS

When analyzing time-oriented data, users are commonly interested in the evolution of their data over time. To achieve this goal, the users' primary task is to compare data located at different positions of the time axis. Detecting trends and pattern are second order goals that lead to insight, and to understanding the data. In giving this coarse description of analysis goals we do not neglect that there is certainly an interplay of further basic visualization tasks (e.g., as described in [Andrienko and Andrienko, 2006; MacEachren, 1995]: check for existence of data elements, locate data elements in time, determine rates of change).

Many different types of data are related to time. One can think of stock exchange data, census data, simulation data, and much more. But also news articles, photo collections, or project plans can contain temporal information. From a theoretical point of view, all these data are time-oriented and should be representable with one and the same approach. From a practical point of view, each of these types of data requires a dedicated visualization. For instance, stock exchange data can be visualized with Flocking Boids [Moere, 2004], census data can be represented as described in [Shanbhag et al., 2005], SimVis [Doleisch et al., 2004b] is efficient for visualizing simulation data. News articles (or contained keywords) can be analyzed with ThemeRiver [Havre et al., 2002], photo collections can be mapped via MyLifeBits [Gemmell et al., 2006], and project plans can be made comprehensible with PlanningLines [Aigner et al., 2005]. Apparently, this list of techniques is not exhaustive. The aforementioned approaches are examples out of many that recognize the special role of the dimension time.

Time-oriented data can also be visualized using generic approaches. Since time is mostly seen as a quantitative dimension (or at least can be mapped to a quantitative domain), common visualization frameworks like the XmdvTool [Ward, 1994] or Visage [Kolojejchick et al., 1997], standard visualization techniques like Parallel Coordinates[1] [Inselberg, 1998], or more or less sophisticated diagrams and charts [Harris, 1999] have their eligibility for visualizing time-oriented data. For simple data and basic analysis tasks, these approaches outperform specialized techniques, because they are easy to learn and understand (e.g., common time diagram). However, in many cases, time is treated

[1] Interestingly, the class *ParallelCoordinatesVisualization* of the InfoVis Toolkit [Fekete, 2004] derives from a super class *TimeSeriesVisualization*, which gives evidence of the importance of time in a visualization environment.

as one quantitative variable among many others, as for instance in Parallel Coordinates – not more, not less. Therefore, generic approaches usually do not support establishing a direct visual connection between multiple variables and the time axis, and they are limited in their capabilities to enable direct interactive exploration and browsing of time-oriented data, which is essential for a successful visual analysis.

Interaction is indeed crucial for the analysis process. To allow users to explore their data, direct manipulation (as already suggested in [Shneiderman, 1983]) and brushing are the means of choice in many interactive visualization tools (e.g., [Hauser et al., 2002; Hochheiser, 2003; Kosara et al., 2004]). Particularly, browsing the time axis and switching between different levels of temporal aggregation (e.g., daily, weekly, or monthly data) are important. Such interactions are rather uncommon for other quantitative variables, and hence, are uncommon in generic visualization frameworks.

The bottom line is that time must be especially considered in Visual Analytics. Different types of time-oriented data need to be visualized with dedicated methods. Additionally, visualization tools must provide a high degree of interactivity. As the cited examples suggest, a variety of concepts for analyzing time-oriented data are known in literature [Aigner, 2006; Müller and Schumann, 2003; Silva and Catarci, 2000]. This variety makes it difficult to assess the current state of the art in visualization of time-oriented data. What is required is a systematic view. In the next section, we will present a categorization schema that is intended to help in untangling this important subarea of Information Visualization and Visual Analytics.

3.3 CATEGORIZATION OF TECHNIQUES FOR VISUALIZING TIME-ORIENTED DATA

As indicated earlier, devising a categorization that is broadly applicable is not an easy task. We decided to develop a systematic view that is geared to three practical questions, so that prospective users and researchers find an easy entry to the ideas behind it:

1. What are the characteristics of the time axis?
2. What is analyzed?
3. How is it represented?

These three questions correspond to the categorization criteria: *time*, *data*, and *representation*. The criterion time addresses the time axis itself. Question (2) considers the data that tie to the time axis. How the data are represented is covered by the third criterion. The following sections will provide detailed explanations of each criterion, including sub-criteria and respective categories.

3.3.1 *Criterion: Time*

Self-evidently, the temporal dimension itself is an interesting aspect for any approach to temporal Visual Analytics. It is virtually impossible to design effective analysis methods, without knowledge about the characteristics of the time axis. From a theoretical perspective, different models and algorithmic representations of time have been studied well in literature [Hajnicz, 1996]. However, we adhere to a more practical categorization of different types of time as presented in [Frank, 1998]. From Frank's taxonomy, the following two sub-criteria are worth discussing.

Temporal primitives: time points vs. time intervals

This first differentiation addresses the temporal primitives that make up the time axis. A time axis can be composed of time points or of time intervals. A *time point* can be considered an instant in time. In contrast to that, a *time interval* is a temporal primitive with an extent. It can be specified by two time points or by a time point plus a duration.

When reasoning about time, the question of whether time points or time intervals are considered is decisive. As described in [Hajnicz, 1996], different relations are possible among time points and among time intervals (see Fig. 3.1). Accordingly, different analysis tasks or goals can be accomplished depending on the addressed temporal primitives.

One might think that the distinction between time points and time intervals is of minor relevance for visualization. However, when considering the validity of data, which is an important concern in Visual Analytics, it becomes clear that this is not true. If data are given on a time axis that is composed of time points, then particular data values are valid only at certain points in time; there is nothing said about how the data look like between adjacent time points. This fact should be reflected in the visual representation to avoid misinterpretations. On the other hand, it is necessary to visualize the range in which interval-based data are valid.

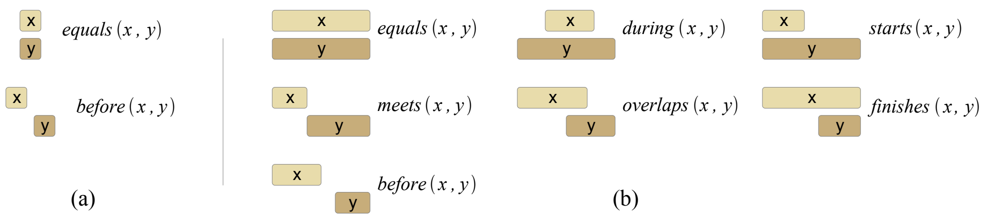


FIGURE 3.1 – **Temporal relations:** (a) Relations between time points; (b) Relations between time intervals [Allen, 1983]; respective inverse relations are possible as well.

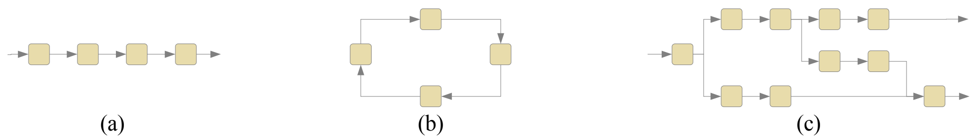


FIGURE 3.2 – **Structure of time:** (a) Linear time; (b) Cyclic time; (c) Branching time.

Structure of time: linear vs. cyclic vs. branching

Now that time points and time intervals have been described as basic temporal primitives, we approach the question of the possible structure of a time axis. We will distinguish three different structures: linear, cyclic, and branching time (see Fig. 3.2). *Linear time* corresponds to our natural perception of time as being a (totally or partially) ordered collection of temporal primitives, i.e., time proceeds from the past to the future. A *cyclic time* axis is composed of a finite set of recurring temporal primitives (e.g., the seasons of the year). On a cyclic time axis, any temporal primitive A is proceeded and succeeded at the same time by any other temporal primitive B (e.g., winter comes before summer, but winter also succeeds summer). In practical applications it is often useful to unroll a cyclic time axis to a linear time axis. *Branching time* axes are modeled as graphs. Temporal primitives are the vertices of the graph. Directed edges describe temporal order. Vertices with more than one outgoing edge indicate a split of the time axis into alternative scenarios, which is particularly relevant for planning or prediction. Apparently, linear time and cyclic time can be seen as special cases of branching time where the graph obeys certain constraints (i.e., for linear time, every vertex has no more than one outgoing edge; for cyclic time, the graph is a circle).

Linear and cyclic time are covered well by existing visual and analytical approaches. However, methods for analyzing branching time are still rare. Here we see a potential task for future work in Visual Analytics.

The decision to which category a time-oriented dataset belongs is not always fully determined, but can depend on the interpretation of the user, on the task, or on the application. If for instance a user seeks to find a general trend in the data, a linear interpretation of the time axis makes sense. On the other hand, detecting seasonal effects in the data can be easier if a cyclic time axis is assumed. Similarly, it is a question of interpretation whether a date is considered as a time point (a day) or a time interval (a period of 24 hours or 86,400 seconds). This dependence on interpretation implies a need for highly flexible Visual Analytics methods.

We also point out that branching time is important in Visual Analytics, because data analysts commonly have to assess alternative scenarios from a whole bunch of facts from heterogenous data sources. In this context it is worth noting that Frank suggests a further category - *multiple perspectives* [Frank, 1998]. In contrast to branching time where only one path through time will actually happen (e.g., in planning applications), multiple perspectives facilitate simultaneous (even contrary) views on time (as for instance required to structure eyewitness reports). Both branching time and multiple perspectives introduce the need for taking care of probability (or uncertainty), to convey, for instance, which path through time will most likely be taken, or which evidence is believable.

3.3.2 *Criterion: Data*

We will now take a look at the data that ties to the time axis. Like the time axis, also the data have major impact on analytical and visual approaches. As indicated in Section 3.2, time-oriented data can be manifold. To answer the question "What is analyzed?", we suggest categorization based on the following sub-criteria.

Frame of reference: abstract vs. spatial

To categorize time-oriented data, it makes sense to consider their context (or frame of reference). Without going into too much detail, we distinguish *abstract* and *spatial* data. By abstract data we mean data that have been collected in a non-spatial context, i.e., data that are not per se connected to some spatial layout. In contrast to that, spatial data contain an inherent spatial layout, which can be conditioned by natural circumstances or modeled realities.

The distinction between abstract and spatial data reflects the crystallization of different subfields of visualization research in the last decade. Information visualization, graph visualization, or software visualization are more concerned with abstract data, whereas spatial data are addressed by scientific visualization (flow visualization, volume visualization) or geographic information systems. Each field handles time-oriented data differently, despite the fact that a unified view would be more desirable.

However, the main reason for a distinction between abstract and spatial data is the way of how data are processed in Visual Analytics. For spatial data, the inherent spatial information can be exploited to find a suitable mapping of data to screen. The representation of time has to be incorporated into that mapping, where it is not always easy to achieve an emphasis of the time domain. For abstract data, no a priori spatial mapping is given. On the one hand, that implies, it is first of all necessary to contrive an expressive spatial layout. This requires creative thinking and experience. On the other hand, screen dimensions can be used almost exclusively to expose the time domain.

Number of variables: univariate vs. multivariate

The second data-related categorization criterion concerns the number of time-dependent variables. When speaking of variables, we do not limit our consideration to basic data types like integers, real numbers, or categorical enumerations. We also consider a vector, a matrix, or a news article as possible data variables if this is required by the application at hand. Obviously, it makes a difference if we have to represent data where each temporal primitive is associated with a single data value (i.e., *univariate data*) or if multiple data values (i.e., *multivariate data*) must be considered. With the latter case, an additional visualization goal – the detection of correlations – is introduced.

Approaches for single-valued data have been around for a long time. There are also various techniques that allow the visualization of two or three data values (which are literally already multivariate). However, the big challenge in Visual Analytics is to handle larger numbers of variables. This is where analytical methods come into play. Usually, it is necessary to apply dimension reduction methods (e.g., principle component analysis) to derive major temporal trends.

Level of abstraction: data vs. data abstractions

"Above all else, show the data" is what Tufte claims in [Tufte, 2001]. The majority of visual methods follow that claim. Visualizing *data* is useful in many application scenarios. However, if larger data sets must be analyzed, Tufte's postulation is hard to fulfill without introducing new problems like overcrowded and cluttered displays. In such cases, it makes sense to melt down the data to condensed form, i.e., to derive *data abstractions* (see [Roddick and Spiliopoulou, 2002] for a survey) that reflect interests and needs of users. Calculating aggregated data values [López et al., 2005] is one example for deriving abstractions, which is particularly useful to drive overview+detail interfaces [Shneiderman, 1996]. Feature visualization also follows the idea of computing data abstractions. *Features* are data portions that obey certain user-defined constraints [Silver, 1997]. In the context of time-oriented data a third derivable information unit must be mentioned – *events*. Events are special situations in the development of time-oriented data. Events can be user-defined or found by methods of Artificial Intelligence. Focusing on events lifts the data analysis to yet a higher level of abstraction [Fails et al., 2006; Reinders et al., 2001; Tominski, 2006].

The essence of this categorization criterion is that visually driven analysis of time-oriented data should not be limited to a mere representation of data. Visual Analytics methods have to consider task- and user-centered higher order data abstractions specifically designed for time-oriented data. To communicate such data abstractions efficiently, a better integration of analytical and visual methods is required [Thomas and Cook, 2006].

These three criteria for data-centric aspects are intentionally settled at a quite high level of abstraction. We are aware that the data aspect is certainly worth further discussion. However, we do not want to overemphasize data aspects, but refer the interested reader to Shneiderman's "Task by Data Type Taxonomy" [Shneiderman, 1996] and Wilkinson's "The Grammar of Graphics" [Wilkinson, 2005], which are widely accepted references already available in literature.

3.3.3 *Criterion: Representation*

Finally, this last criterion addresses the visual representation of time-oriented data. We do not try to investigate subtle details of the variety of visual approaches available, but concentrate on two fundamental sub-criteria that concern the time dependency and the dimensionality of the presentation space.

Time dependency: static vs. dynamic

Static representations visualize time-oriented data in still images (i.e., representations that do not change automatically over time). In contrast to that, *dynamic* representations utilize the physical dimension time to convey the time dependency of the data (i.e., representations that change automatically over time such as slide shows or animations). The presence or absences of interaction facilities has no influence on whether a visualization approach is categorized as static or dynamic.

Distinguishing between static and dynamic representations is crucial for Visual Analytics, because different tasks and goals are supported. Dynamic representations are well suited to convey the general development of the analyzed data over time. However, there are also critical voices on animation (e.g., [Simons and Rensink, 2005; Tversky et al., 2002]). Especially when longer multivariate time series have to be visualized, animation-based approaches reach their limits. Users simply cannot follow all changes in the visual representation and the animation takes too long for the user to remember its course. Static representations show all information on one screen, which is advantageous to fully concentrate on the data and to compare different parts of the time axis. However, in contrast to animations, static representations require screen real estate to represent the time axis itself. Therefore, it is challenging to develop representations that avoid visual clutter. Again larger data sets aggravate this problem.

Dimensionality: 2D vs. 3D

This sub-criterion simply distinguishes between 2D and 3D presentation spaces. The question of whether or not it makes sense to exploit three dimensions for visualization is discussed heavily in the community. One camp of researchers argues that two dimensions are sufficient for effective data analysis. In their thinking the third dimension involves unnecessary difficulties like occlusion and lost information on back faces. The other camp of researches see the third dimension as a possibility to encode further information. Undoubtedly, certain types of data (e.g., flow data or volume data) even require the third dimension for expressive data visualization. The mentioned disadvantages of a three dimensional presentation space are tackled with advanced interaction techniques or additional visual cues. We will not take either position, but think that both options are required depending on task and data at hand.

3.3.4 *Examples*

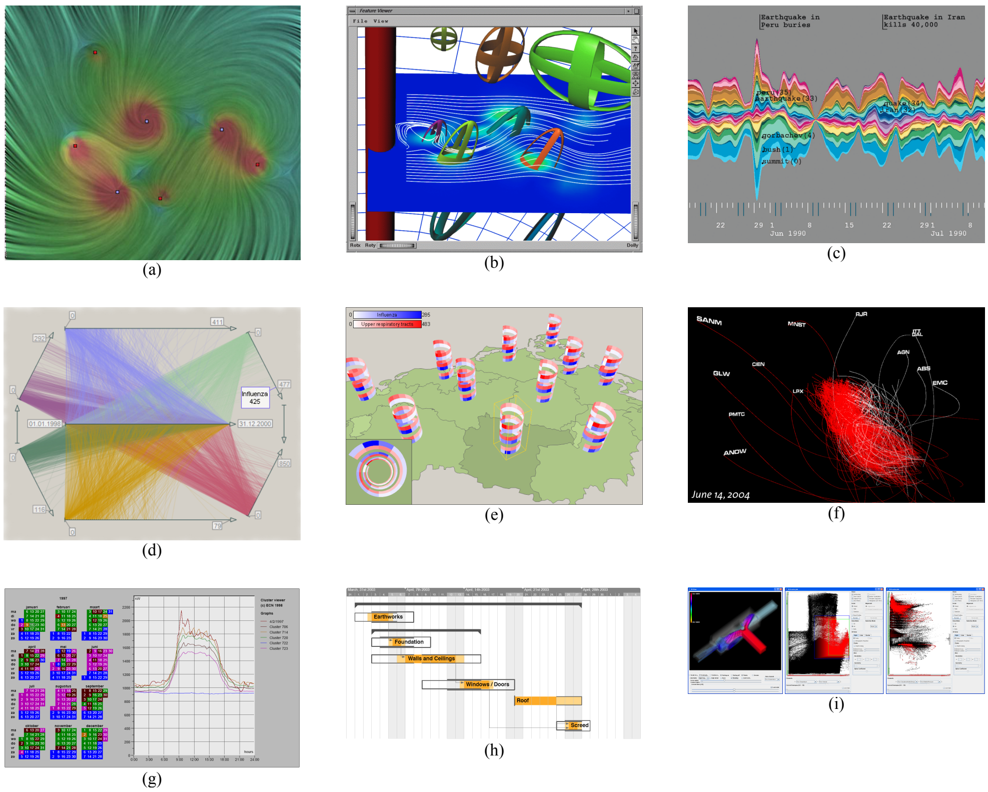


Figure 3.3 – **Examples** of techniques for visualizing time-oriented data.

We will now give some examples of visual methods for analyzing time-oriented data. Some of the examples stem from our own work on visualization of time-oriented data, further examples are taken from literature. This selection of techniques does not claim any completeness; a comprehensive overview can be found in [Aigner, 2006]. Our goal is to demonstrate the applicability of the developed classification scheme. We will not provide introductions to the examples, but refer the interested reader to the original publications for detailed explanations. Fig. 3.3 shows, on the one hand, the following methods and techniques:

(a) *Animated flow visualization* [Van Wijk, 2002]: Smooth animations created from streamline images,
(b) *Feature and event based flow visualization* [Reinders et al., 2001]: Animated visualization based on data abstraction and iconic representations,
(c) *ThemeRiver* [Havre et al., 2002]: Static representation of thematic changes in document collections,
(d) *TimeWheel* [Tominski et al., 2004]: Axes-based visualization of multivariate data with focus on temporal dependencies,
(e) *Helix glyphs on maps* [Tominski et al., 2005]: Emphasis of cyclic patterns in spatio-temporal human health data,
(f) *Flocking boids* [Moere, 2004]: Stock market visualization based on simulation and animation of flocking behavior,
(g) *Cluster and calendar based visualization* [Van Wijk and Van Selow, 1999]: Visualization of univariate time series on different levels of aggregation,
(h) *PlanningLines* [Aigner et al., 2005]: Visualization of project plans with temporal uncertainty,

and, on the other hand, also screenshots of a larger visualization system:

(i) *SimVis* [Doleisch et al., 2004b]: Larger system that combines several views to facilitate flow visualization.

The systematic view along with a categorization of the aforementioned examples is presented in Table 3.1.

3.4 DISCUSSION

In the previous section, we have elaborated on a categorization of visual methods for analyzing time-oriented data. In this section, we will discuss findings, implications, and limitations of our systematic view. We will use them as starting point to derive open problems and future work in Visual Analytics of time-oriented data.

Preliminary remark: First of all, it must be mentioned that we have considered only top-level criteria. Indeed, one can easily figure out more criteria with several further categories (e.g., representation method: pixel-based, map-based, glyph-based, etc.). However, we think that such and other criteria should not be added to an initial categorization of the field

Time	**Temporal primitives**	time points (a) (b) (c) (d) (e) (f) (g) (i)		time intervals (g) (h)
	Structure of time	linear (a) (b) (c) (d) (f) (g) (h) (i)	cyclic (e)	branching (h)
Data	**Frame of reference**	abstract (c) (d) (f) (g) (h) (i)		spatial (a) (b) (e) (i)
	Number of variables	univariate (a) (b) (f) (g) (h)		multivariate (c) (d) (e) (i)
	Level of abstraction	data (a) (b) (c) (d) (e) (f) (g) (h) (i)		data abstractions (b) (g) (i)
Representation	**Time dependency**	static (c) (d) (e) (g) (h) (i)		dynamic (a) (b) (f) (i)
	Dimensionality	2D (a) (c) (d) (g) (h) (i)		3D (b) (e) (f) (i)

Table 3.1 – **Categorization schema** for visual methods for analyzing time-oriented data.

by default, but only on demand. The reason is that some aspects are not relevant in certain specialized areas (e.g., distinguishing pixel-based, map-based, and glyph-based techniques makes no sense for volume visualization). Nonetheless, identifying further general categories may turn out helpful once future development in Visual Analytics yields first methodological results.

Observation 1 – Multiple Views: We noticed that the visual methods currently available stand separate, i.e., are suitable only for particular categories of time and data characteristics. To our knowledge, there exists no visualization framework that can handle all types of times and data, or provides a broader selection of possible representations. We think that an open framework fed with pluggable visual and analytical components for analyzing time-oriented data is useful. Such a framework will be able to support multiple analysis tasks and data characteristics, which is a goal of Visual Analytics.

Unfortunately, there is no ad hoc way of combining or linking the available methods. However, from the example of SimVis (see Fig. 3.3 (i)) and from current research on coordinated multiple views (e.g., [Shimabukuro et al., 2004], [Aigner and Miksch, 2004]), we see that linking several views together can extend the applicability and usefulness of visual methods. Multiple views are particularly helpful in analyzing time-oriented data. Therefore, we underline the need for a flexible system that offers various methods to support visual analysis and decision making. The goal is to provide views that are dedicated to different analysis aspects, are helpful in conveying different levels of temporal granularity and data abstraction, or are used to represent different parts of the time axis. The categorization developed in this paper can be used to identify mandatory and optional views to be developed (depending on the types of time and the data at hand). Representational preferences as well as tasks and goals of users must also be considered. For example, using an animation to analyze data can be difficult (goal: analysis), but using an animated view to present analysis results might impress the director (goal: presentation). What this example suggests is that different visual representations are needed to fully support the analysis of time-oriented data and the communication of analysis results. A similar statement was already made by Bertin in 1981, although he used different words:

"A graphic is not drawn once and for all; it is constructed and reconstructed until it reveals all the relationships constituted by the interplay of the data." [Bertin, 1981]

Observation 2 – Interaction: It is apparent that interaction is a must particularly for analyzing time-oriented data. All presented examples provide some level of interactivity. However, scientific papers often discuss visual representations only; interaction is not always in the focus. Navigating in time and switching between different levels of temporal granularity are prominent examples of interacting with time-oriented data. Note that such interactions are rather uncommon for abstract quantitative dimensions. Even though direct manipulation (direct interaction with the visual representation, rather than with buttons or sliders) or advanced brushing techniques are known in literature (e.g. [Doleisch et al., 2004a; Hauser et al., 2002; Hochheiser, 2003; Shneiderman, 1983]), they are only rarely considered to drive the visual analysis of time-oriented data.

Therefore, it makes sense to put more effort in investigating the potential of interaction in Visual Analytics. We need evaluated and accepted interaction techniques that allow intuitive exploration and analysis of time-oriented data. At the same time, we have to take care not to overload the user with functionality (e.g., hold shift and control key then click and hold right mouse button and move the mouse). This means, we need not only visual methods that suit the task at hand, but also interaction that is adapted to it.

A further aspect that infers from interaction and multiple views is coordination, i.e., the propagation of interaction originated from one view to all other views (that are coordinated). To ease the use of multiple views, coordination methods are commonly applied. To facilitate reasoning about time-oriented data, coordination can be targeted in accordance with the categories of our systematic view. A particular challenge in temporal Visual Analytics is to coordinate visual *and* analytical methods that not necessarily share common parameters. For instance, how can a view that shows derived principal components be coordinated with a view that shows predicted future trends, or is it impossible to coordinate such views at all?

Observation 3 – Analytical Methods: When looking at our categorization, we see the following situation: Most examples visualize time-oriented data, only few of our examples support temporal data abstractions [Doleisch et al., 2004b; Reinders et al., 2001; Van Wijk and Van Selow, 1999]. That is, many methods focus on representing time-oriented data, and neglect the analytical component.

To fully support the knowledge discovery process, visual methods for analyzing time-oriented data should take Keim's Visual Analytics mantra into account:

"*Analyse first, Show the Important, Zoom, filter and analyse further, Details on demand.*" [Keim, 2005]

Keim's mantra demands for a better integration of visual and analytical methods. With ever increasing volumes of data, temporal abstractions become more and more indispensable. Only if analytical methods (e.g., segmentation, clustering, detection of events) are applied to compute expressive abstractions is it possible to analyze larger data sets efficiently. Moreover, data abstractions are necessary for interactivity to prevail.

When speaking of analytical methods, a further aspect must be taken into account: Time-oriented data often involves uncertainty [Griethe and Schumann, 2006]. Analytical methods (e.g., prediction of trends) also might compute vague information. It is mandatory to notify users of this circumstance, so that they can adjust their confidence in the generated analysis results.

3.5 CONCLUSION

In this paper, we proposed a systematic view on methods for visually analyzing time-oriented data. Our view is based on three main criteria: *time*, *data*, and *representation*. We presented examples and discussed implications of our proposal in the context of Visual Analytics.

We see quite a lot methods available in literature [Aigner, 2006]. Most of them support only certain parts of our categorization. As a conclusion, we suggested the development of an open framework for Visual Analytics of time-oriented data. We identified the following directions for future research on this aspect:

- Multiple views for different data aspects, different levels of temporal aggregation/abstraction, and different parts of the time axis,
- Sophisticated and adaptable interaction and coordination facilities particularly suited for time-oriented data, and
- Tighter integration of visual and analytical methods.

An important issue that concerns all previous points is task-orientation. This means that Visual Analytics systems should automatically suggest and parameterize visual, analytical, and interaction methods based on the users' task at hand. Recently, an interesting analysis of possible visualization tasks has been published in [Andrienko and Andrienko, 2006]. That list of tasks can be used as a basis for future research on task-oriented Visual Analytics. In that regard, perceptual issues must be further investigated. Empirical tests have to be conducted to judge which forms of presentation (2D or 3D, static or dynamic, etc.) are best suited for particular analysis tasks.

ACKNOWLEDGEMENTS

This work was partly supported by the "Fonds zur Förderung der wissenschaftlichen Forschung - FWF" (Austrian Science Fund), grant P15467-INF. We gratefully acknowledge support by the German Research Foundation.

REFERENCES

Aigner, W. (2006). *Visualizing Time and Time-Oriented Information: Challenges and Conceptual Design*. PhD thesis, Vienna University of Technology, Vienna, Austria.

Aigner, W. and Miksch, S. (2004). Supporting Protocol-Based Care in Medicine via Multiple Coordinated Views. In *Proceedings of International Conference on Coordinated & Multiple Views in Exploratory Visualization*, London, UK.

Aigner, W., Miksch, S., Thurnher, B., and Biffl, S. (2005). PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proceedings of International Conference on Information Visualisation*, London, UK.

Allen, J. F. (1983). Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, 26(11):832–843.

Andrienko, N. and Andrienko, G. (2006). *Exploratory Analysis of Spatial and Temporal Data*. Springer, Berlin, Germany.

Bertin, J. (1981). *Graphics and Graphic Information Processing*. Walter de Gruyter, Berlin, Germany.

Doleisch, H., Hauser, H., Gasser, M., and Kosara, R. (2004a). Interactive Focus+Context Analysis of Large, Time-Dependent Flow Simulation Data. Technical Report TR-VRVis-2004-024, currently in submission, VRVis Research Center, Vienna, Austria.

Doleisch, H., Mayer, M., Gasser, M., Wanker, R., and Hauser, H. (2004b). Case Study: Visual Analysis of Complex, Time-Dependent Simulation Results of a Diesel Exhaust System. In *Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization*, Konstanz, Germany.

Fails, J. A., Karlson, A., Shahamat, L., and Shneiderman, B. (2006). A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events across Multiple Histories. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, Baltimore, USA.

Fekete, J.-D. (2004). The InfoVis Toolkit. In *Proceedings of IEEE Symposium on Information Visualization*, Austin, USA.

Frank, A. U. (1998). Different Types of "Times" in GIS. In Egenhofer, M. J. and Golledge, R. G., editors, *Spatial and Temporal Reasoning in Geographic Information Systems*, chapter 3, pages 40–62. Oxford University Press, New York, USA.

Gemmell, J., Bell, G., and Lueder, R. (2006). MyLifeBits: A Personal Database for Everything. *Communications of the ACM*, 49(1):89–95.

Griethe, H. and Schumann, H. (2006). The Visualization of Uncertain Data: Methods and Problems. In *Proceedings of Simulation and Visualization*, Magdeburg, Germany.

Hajnicz, E. (1996). *Time Structures: Formal Description and Algorithmic Representation*. Number 1047 in Lecture Notes in Computer Science. Springer, Berlin, Germany.

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Management Graphics, Atlanta, USA.

Hauser, H., Ledermann, F., and Doleisch, H. (2002). Angular Brushing of Extended Parallel Coordinates. In *Proceedings of IEEE Symposium on Information Visualization*, Boston, USA.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. (2002). ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20.

Hochheiser, H. (2003). *Interactive Graphical Querying of Time Series and Linear Sequence Data Sets*. PhD thesis, University of Maryland, College Park, USA.

Inselberg, A. (1998). A Survey of Parallel Coordinates. In Hege, H.-C. and Polthier, K., editors, *Mathematical Visualization*, chapter 3, pages 167–179. Springer, Berlin, Germany.

Keim, D. (2005). Scaling Visual Analytics to Very Large Data Sets. Presentation at Workshop on Visual Analytics, Darmstadt, Germany, June 4th, 2005.

Kolojejchick, J., Roth, S. F., and Lucas, P. (1997). Information Appliances and Tools in Visage. *IEEE Computer Graphics and Applications*, 17(4):32–41.

Kosara, R., Bendix, F., and Hauser, H. (2004). TimeHistograms for Large, Time-Dependent Data. In *Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization*, Konstanz, Germany.

López, I. F. V., Snodgrass, R. T., and Moon, B. (2005). Spatiotemporal Aggregate Computation: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):271–286.

MacEachren, A. M. (1995). *How Maps Work: Representation, Visualization, and Design*. Guilford Press, New York, USA.

Moere, A. V. (2004). Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of IEEE Symposium on Information Visualization*, Austin, USA.

Müller, W. and Schumann, H. (2003). Visualization Methods for Time-dependent Data - An Overview. In *Proceedings of Winter Simulation Conference*, New Orleans, USA.

Reinders, F., Post, F. H., and Spoelder, H. J. W. (2001). Visualization of time-dependent data with feature tracking and event detection. *The Visual Computer*, 17(1):55–71.

Roddick, J. F. and Spiliopoulou, M. (2002). A Survey of Temporal Knowledge Discovery Paradigms and Methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767.

Shanbhag, P., Rheingans, P., and desJardins, M. (2005). Temporal Visualization of Planning Polygons for Effcient Partitioning of Geo-Spatial Data. In *Proceedings of IEEE Symposium on Information Visualization*, Minneapolis, USA.

Shimabukuro, M. H., Flores, E. F., de Oliveira, M. C. F., and Levkowitz, H. (2004). Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. In *Proceedings of International Conference on Coordinated & Multiple Views in Exploratory Visualization*, London, UK.

Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer*, 16(8):57–69.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, Boulder, USA.

Silva, S. F. and Catarci, T. (2000). Visualization of Linear Time-Oriented Data: A Survey. In *Proceedings of International Conference on Web Information Systems Engineering*, Hong Kong, China.

Silver, D. (1997). Feature Visualization. In Nielson, G. M., Hagen, H., and Müller, H., editors, *Scientific Visualization, Overviews, Methodologies, and Techniques*, chapter 3, pages 279–293. IEEE Press, Los Alamitos, USA.

Simons, D. J. and Rensink, R. A. (2005). Change Blindness: Past, Present, and Future. *Trends in Cognitive Sciences*, 9(1):16–20.

Thomas, J. J. and Cook, K. A. (2006). A Visual Analytics Agenda. *IEEE Computer Graphics and Applications*, 26(1):10–13.

Tominski, C. (2006). *Event-Based Visualization for User-Centered Visual Analysis*. PhD thesis, University of Rostock, Rostock, Germany.

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of ACM Symposium on Applied Computing*, Nicosia, Cyprus.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of International Conference on Information Visualisation*, London, UK.

Tufte, E. R. (2001). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, USA, second edition.

Tversky, B., Morrison, J. B., and Betrancourt, M. (2002). Animation: Can It Facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262.

Van Wijk, J. J. (2002). Image Based Flow Visualization. In *Proceedings of SIGGRAPH*, San Antonio, USA.

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of IEEE Symposium on Information Visualization*, San Francisco, USA.

Ward, M. O. (1994). XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of IEEE Visualization*, Washington, USA.

Wilkinson, L. (2005). *The Grammar of Graphics*. Springer, New York, USA, 2 edition.

CHAPTER 4

A-Plan: Integrating Interactive Visualization with Automated Planning for Cooperative Resource Scheduling

Thomas Schneider and Wolfgang Aigner

Abstract · *Assignment of staff to work tasks is a complex problem that involves a large number of factors and requires a lot of expertise. Long term as well as short term requirements need to be met which demands flexible solutions. Software tools can aid planners in reaching optimal dispatching plans but currently available solutions provide only incomplete support. This paper describes the design, development, and evaluation of a prototype for semi-automated assignment planning called A-Plan. We have carried out this work in the context of a gas device maintenance provider. In A-Plan, assignments of service technicians to customers are displayed visually and can be modified by direct manipulation. Smooth cooperative work is possible and an optimization algorithm has been integrated that facilitates semi-automatic planning. A qualitative evaluation with potential users and IT professionals provided encouraging feedback on the proposed integration of automated methods and interactive visual interfaces.*

Keywords · Visual analytics, resource scheduling, optimization.

This article originally appeared as [Schneider and Aigner, 2011]:

Schneider, T. and Aigner, W. (2011). A-Plan: Integrating Interactive Visualization With Automated Planning for Cooperative Resource Scheduling. In *Proceedings of International Conference on Knowledge Management and Knowledge Technologies (I-KNOW), Special Track on Theory and Applications of Visual Analytics (TAVA)*, pages 44:1–44:8. ACM Press.

4.1 INTRODUCTION

PROPER resource utilization (e.g., staff, machines, rooms, vehicles) is one of the most pressing cost factors in many economic areas. For example a service provider for gas devices needs to maintain technical equipment within certain maintenance intervals. In addition to those regular maintenances that can be planned well in advance, sudden defects might occur and have to be repaired promptly. This demands adaptive and manipulable scheduling. Apart from that, also different skill sets and levels are needed for certain kinds of maintenances or defects which increases the complexity of the problem. On the one hand, personnel needs to be scheduled according to these constraints in order to keep downtime as low as possible. On the other hand, the amount of needed employees should be kept as low as possible and their utilization should be as optimal as possible. Adding to that, different regulations like for example laws on working time need to be followed. Other examples are airlines that have to maintain their aircraft or mobile nursing care companies that have to dispatch their staff to the patients depending on their condition. Especially in the latter case, it is also very important to minimize the distances between the assignments because the necessary travel between the patients causes costs for the company.

In our work, we collaborated with a gas device (e.g., heaters, stoves) maintenance provider that has to dispatch its service technicians to their customers. Currently, the planners schedule assignments for the technicians manually for the next weeks. A major challenge is to optimize the distances between the customers. They also have to be aware of absences of the technicians and the number of assignments per day is also restricted, depending of the time of the year (e.g., in autumn failures of gas devices are more likely than in summer). Moreover, some of the technicians have special skills for special gas devices. All these constraints have to be taken into consideration by the users and are not supported by the currently used software.

The described setting is an example for an optimization task. More precisely it is a linear programming task that can be described by the following model:

$$Min!/Max! : c_1x_1 + c_2x_2 + c_3x_3 + ... + c_nx_n \tag{4.1}$$

with

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n &<= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... + a_{2n}x_n &<= b_2 \\ &... \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + ... + a_{mn}x_n &<= b_m \end{aligned}$$

The first function is the linear function to be maximized or minimized (in our case we want the minimum distance). The other functions are constraints.

$x_1 \ldots x_n$ are the variables to calculate (when calculating a route plan, the variable is 1 when the customer will be visited, and 0 when not), $c_1 \ldots c_n$ are the cost parameters (distances between the customers in our case). Examples for constraints are that the number of assignments per day is limited or that there are certain skills of technicians needed. Another constraint of the model is that $x_1 \ldots x_n$ can only be integer numbers. Optimization problems of this type are called combinatorial problems or integer programming. The specialty of a combinatorial problem is the exponential growth of possibilities for a larger number of variables. There are for example 3,628,800 possibilities to bring 10 objects in sequence, for 20 objects 20! is a number with 19 digits.

The given examples illustrate the relevance of this problem and the need for tool support. Better scheduling can help to reduce costs and increase the quality of service for the customers while reducing the administrative work as well as travel and vacancy times. Particularly, a Visual Analytics (VA) approach that integrates automatic methods and supports humans via interactive visual interfaces [Thomas and Cook, 2005] seems to be a perfect fit for this kind of problem complexity. In this paper we present the design, implementation, and evaluation of a VA prototype that combines automatic optimization and interactive visual interfaces to aid employees of a gas device maintenance provider to schedule their service technicians. The three main challenges in this context are to (1) design interactive visualization methods; (2) integrate automated planning functionality; and (3) support synchronous collaboration when handling customer requests. Our main aim is to aid personnel in their complex planning and scheduling tasks while keeping an optimal balance between automatic algorithms and user interaction.

First, we will present some related research work in the areas of interactive visualization and optimization as well as some widely-used commercial systems in this field. After that, we will report on our user and task analysis in Section 4.3 that has been conducted in order to inform and guide the design and implementation of our prototype which will be presented in Sections 4.4 and 4.5. To assess the utility and usability of our approach, an evaluation has been conducted as described in Section 4.6. Finally, we will summarize and discuss our findings in Section 4.7.

4.2 RELATED WORK

In the following, we will discuss related work along the lines of Information Visualization (InfoVis), optimization, and commercial scheduling systems. For scheduling tasks, visual representations need to be considered that are able to represent time intervals (rather than time instants). An overview of visualization techniques for time interval data is given by Aigner et al. [2011]. Timelines are a simple and widely-used representation of events or time intervals. Karam [1994] describes timelines as linear graphical visualization of events over time. In general the time axis is presented along one display dimension (mostly the horizontal axis) and category along the other axis. Events are shown as lines or bars whereas their length represents the temporal extent of an event. Plaisant et al. [1996] introduced LifeLines as a further development of timelines. In LifeLines the thickness and color of the bars can be used to encode additional information. Plaisant et al. also included interactive features, e.g., for overview+detail and zooming. GANTT charts are a special form of timelines invented by Henry L. Gantt (1919) which are ubiquitously used in several project management products today. Modern project management software also shows milestones and uses hierarchies of tasks to keep the complexity at a manageable level. The disadvantage of the aforementioned visualization techniques is their high space consumption as inactive intervals need to be represented as blank areas. Therefore, it is difficult to simultaneously investigate many categories on whether they are running in parallel. Luz and Masoodian [2007] try to decrease the space consumption in their Temporal Mosaic technique by visually aggregating related interval bars into compound rectangles. Color is used to distinguish between different categories of events. However, moving assignments in temporal mosaics via direct manipulation is difficult which renders them inappropriate for scheduling tasks.

Because of the exponential growth of possibilities for combinatorial problems, only approximation algorithms can be used. For the special problem described in the previous section a large amount of literature can be found. The multiple traveling salesman (M-TSP) is a model where some salesmen have to travel to customers [Bektas, 2006]. The goal is to travel to all customers with minimal costs. The generalized formulation of the problem is the vehicle routing problem (VRP). The VRP is an M-TSP with cargo capacity restrictions. Cargo capacity can be interpreted as a restriction for the number of customers that can be visited by a technician per day. Several approaches exist for VRP. Genetic algorithms are adaptive heuristic search methods based on population genetics [Bräysy, 2004]. A dynamic genetic algorithm can be found in [Hanshar, 2007] where dynamic means that new customer orders can be dispatched after the planning has completed (in our example a failure assignment). A similar solution with stochastic customers is described in [Cheung et al., 2007]. When developing algorithms it is important that the calculation is fast (nearly in realtime) for a huge number of customers (about 20,000). The result of the algorithm is a suggestion for the plan that can be adapted by the user. Two examples are the sweep algorithm and the savings algorithm. The sweep algorithm [Gillett and Miller, 1974] assumes that the locations of customers and the headquarter are given by coordinates (x_i, y_i) and the dispatching point is in the origin of the coordinate system. The distances between the locations are determined as Eucleadian distances. The savings algorithm [Clarke and Wright, 1964] is the most widely known heuristic for the VRP [Toth and Vigo, 2002]. The idea behind it is that savings can be obtained by joining two routes into one route.

Due to the relevance of resource scheduling problems in industry, also a large number of commercial and open source software products are available. Examples are SAP CRM, Microsoft CRM, Service Ledger[1], ORS Online Resource Scheduler[2], phpScheduleIt [3], Flight Schedule Pro[4] , Schedule Pro[5], and Titanium Schedule[6]. The main focus of SAP CRM and Microsoft CRM is customer relationship management, but they also provide service modules and are widely used in companies. More details on the usage and issues with SAP CRM will be presented in Section 4.3. An interesting feature of Service Ledger is its MapPoint integration that shows routes on a map with the approximate travel times. Online Resource Scheduler and phpScheduleIt are open source web applications that offer planning functionality for any resource. Flight Schedule Pro is a specialized software for scheduling of aircraft for flight schools, universities, and flying clubs. For the visualization of the scheduling data, timelines and calendars are used in the mentioned products. The multiple views of Service Ledger provide a calendar overview and details when selecting an entry. Flight Schedule Pro delivers more information when pointing on an entry via tooltips. Schedule Pro distributes all changes to the clients, so all clients always have the actual data which is necessary for cooperative work. Titanium Schedule uses many colors and icons on the scheduling plan to inform the user about the different assignments. Due to this fact the representation is rather cluttered and confusing. Schedule Pro uses a plan representation where only one day is shown in full detail and the other days are shown in a compressed form similar to DateLens [Bederson et al., 2004].

After discussing related work we will now present the user and task analysis we conducted.

4.3 USER & TASK ANALYSIS

At the beginning of our project qualitative research was conducted to analyze current work practices in order to understand behaviors and attitudes of users as well as technical, business, and environmental contexts (the domain) of the tool to be designed. Furthermore, vocabulary and other social aspects regarding how existing products are used are important to understand the domain in question.

[1] http://www.serviceledger.com
[2] http://ors.sourceforge.net
[3] http://www.php.brickhost.com
[4] http://www.flightschedulepro.com
[5] http://www.invisionwfm.com
[6] http://www.titaniumschedule.com
(all URLs accessed at January 20, 2011)

4.3.1 *Method*

Kulyk et al. [2007] present several methods for user and task analysis: *contextual observation* is a method where the designer observes the user's working environment in practice. Observation is very useful but also has to deal with several problems. The observations can be misinterpreted, e.g., when the observer does not know the context of the actions. An observation can also disturb the work of the observed group and so the observed work can differ from normal work situations. In *interviews* subjects will be asked about their work and the use of the software and artifacts. Bartlett and Toms [2005] notice that *"a drawback to the interview approach is that it relies on recall, rather than directly capturing the activity of interest, and is thus vulnerable to missing details that were either forgotten, or not considered relevant." Task demonstration* is similar to observation but in contrast the observer may ask questions and the demonstrator may explain some tasks in more detail. A disadvantage is that the task is described by the user, so the feedback may be very limited and problems may not become visible, since most experienced users are not aware of these problems [Shneiderman and Plaisant, 2004]. To mitigate the disadvantages of the methods we used a combination of task demonstration, contextual observation, and interviews for the analysis. We started with a task demonstration to get the big picture about the currently used application (SAP CRM) and the process. In the contextual observation we found out how the work is really done. Finally, interviews were a possibility to get more information about the users and the problems with the current application. These contained questions like: What experience with IT systems do the people have? What software do they use for their work? What should an optimal system for the desired tasks look like?

4.3.2 *Results*

To illustrate the problem, all tasks and currently used software were investigated. In the analyzed company 35 technicians do the service work and seven employees do the planning and dispatching in the office. Three main use cases were identified:

- *Regular assignment for customers with maintenance contract*: All customer with a maintenance contract periodically receive a service assignment for their gas device. This assignments have to be planned for service technicians whereas the distances between the customers should be as short as possible.
- *Failure of gas devices*: Customers announce failures of their gas devices to the call center of the company. The agent tries to fix the problem on the telephone; if the problem can't be solved, she makes an entry in the failure list. Later she tries to contact a service technician near the customer to solve the problem.

- *Refusal or deferral of assignments for maintenance*: When a customer wishes to cancel or change the date of her assignment she also calls in. The dispatcher writes the corrected date of the customer into the printed assignment plan.

When analyzing the current work practice, we found many usability issues in the used software system (SAP CRM). This includes for example that information about the assignments is abbreviated in an unreadable way; the behavior for drag-and-drop of assignments is not consistent (for moving of assignments direct manipulation is used whereas for the insertion of new assignments this is not possible); the usage of colors is confusing because contrary to expectation, color does not describe the current status of an assignment; withdrawing of operations is not possible in the whole system because undo does not exist in the entire application. Moreover, the user interface is overloaded and cluttered, some information is redundant, and the important information is hard to find. Finally, the user has to handle three different applications for conducting the aforementioned use cases.

During contextual observation we encountered that in many cases the dispatcher does not use the software at all. She has to answer questions and confirm dates of customers on the phone within seconds which is not possible with the current system (it takes for example about four minutes to generate a new assignment). Therefore, the users came up with a workaround that makes rapid answering on the phone possible. Rather than retrieving information from the electronic system, two paper-based artifacts are used:

- *A book calendar* that includes all appointment requests from customers. Thus, the customers' data can quickly be entered during the phone call. In the evening all data is recorded in the system, thus the long waiting times of the electronic system to read the data will occur only once for all assignments of the day.
- *A folder* with separator sheets for each technician and a list of assignments per day per technician. The lists are always printed from the SAP system some days in advance. When a customer calls the dispatcher and reports short-notice cancellations or schedule changes, the information will be recorded in this folder on paper.

4.3.3 *Personas & Scenarios*

The goal for this user & task analysis besides gaining more knowledge about the domain, user requirements, and desires, was the creation of representative user profiles, their goals, and the construction of interaction scenarios based on this user model. Following the user-centered design approach by Cooper [2004], this lead to the creation of scenarios and personas that aid design and evaluation. Personas are a created cast of characters representing real persons along with both their knowledge in the computer and the application domain. These persons have certain goals they want to achieve when using a product. A scenario is basically a detailed story about a person performing a certain task to achieve her goals. In our case, we identified two personas (Erich Gruber, a 50-year-old technician and Julia Steiner, a 30-year-old more business-oriented dispatcher) and we created four scenarios that capture the main use cases: *Appointment Request with Assignment*, *Appointment Request without Assignment*, *Cancellation*, and *Failure*. These personas and scenarios mostly cover the different kinds of employees and use cases in the company.

Based on the results of the user & task analysis as well as the created personas & scenarios, we designed a prototype that will be described next.

4.4 DESIGN

Two guiding lines of the design of A-Plan were to fulfill user requirements and to avoid reported problems and issues of current work practice. To support and ease the workflow of users, an automated planning function should be integrated into the software. Furthermore, multiple users should be able to work simultaneously with the data while being aware of each others' actions. Following Cooper et al.'s recommendation [Cooper et al., 2007] we eliminated save buttons and avoided OK buttons. Instead, every action should be saved automatically and an undo function should be available for the user to take back unwanted operations. Following that, every transaction should be saved immediately and distributed to the other clients.

Figure 4.1(a) shows the basic screen layout of A-Plan. The screen is divided into three areas: (1) The *planning area* is the place where the user can view, insert, move and delete assignments. (2) *Details* are shown on the right side of the window. In this location the user can also start actions like searching for a customer and planning of assignments. (3) *Collaboration:* In the lower area of the window the user is supported in the cooperation with other users.

4.4.1 *Planning Area: Visualization & Interaction Design*

We decided to use traditional timelines for the visualization of the assignment plan because of its widespread use and ability to display the data characteristics at hand. Furthermore, timelines are well-suited to be used and manipulated interactively and cooperatively by more than one user. In our case, assignments are displayed as boxes showing the most important information about the assignment directly as text (city, customer, time). The timelines are arranged in horizontal lanes whereas a single lane corresponds to a specific service technician. The shape and the color of the box also gives information about the assignment: When the box has rounded corners, only the date of the assignment is fixed but not the exact time within the day whereas when the corners are angular the time is also fixed. The color of the assignment gives quick information about the type of the assignment (maintenance or failure). When hovering over an assignment a tooltip shows

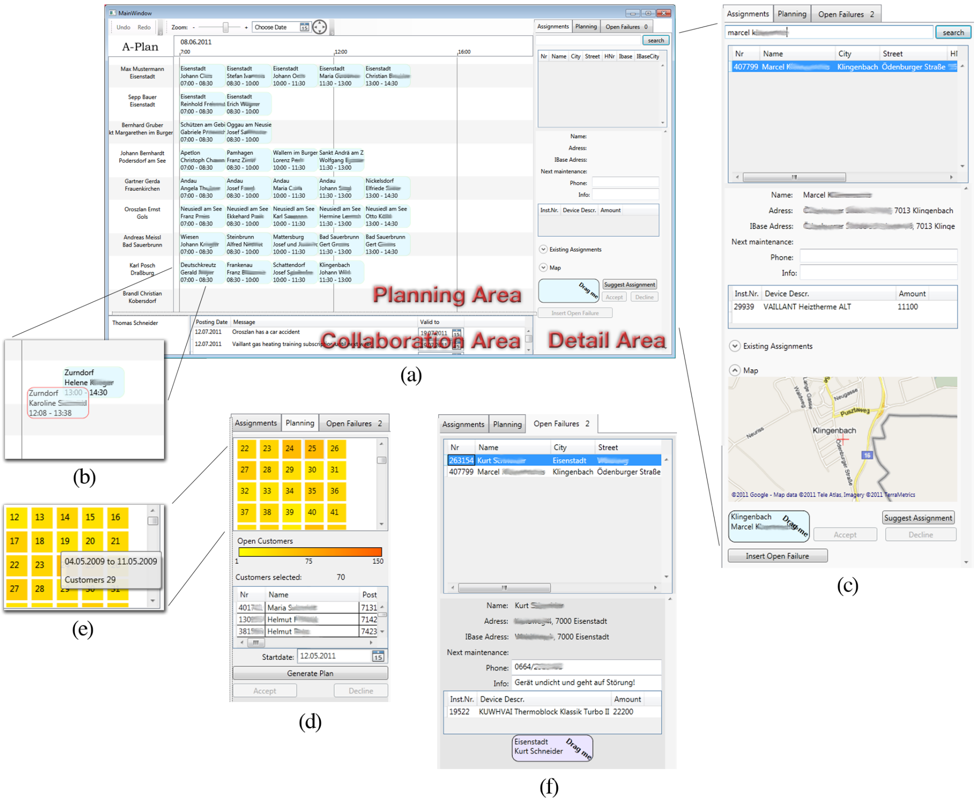


FIGURE 4.1 – **A-Plan:** (a) basic screen layout (top: toolbar including undo/redo buttons, zoom slider, date chooser; left: planning area with interactive timeline visualization; right: detail area; bottom: collaboration area including user list and messages) (b) assignment is shown semi-transparent while dragging; (c) assignment tab showing details with map view; (d) planning tab; (e) heatmap view; (f) open failures tab.

more details and when clicking on an assignment all information about the customer is shown in the detail area to the right. This resembles an overview first and detail-on-demand approach, where overview and detail information are displayed simultaneously in a distinct presentation space [Cockburn et al., 2008].

Different interactions for smooth panning and zooming allow for navigation in time. I.e., zooming can be performed by using the mouse wheel, a slider in the toolbar, or two buttons next to the slider. Panning is done by dragging the background of the plan or by using a navigation element in the toolbar. Furthermore, users might navigate by directly selecting a day of choice using a date chooser widget. Changes to the plan can be performed by direct manipulation. The user might drag-and-drop assignments in the plan.

This form of interaction can also be used to insert new assignments. In this case the user can drag a surrogate assignment from the detail area into the planning area. All changes are distributed to all clients immediately. While one user is dragging an assignment all other users can follow the movement of the assignment live. Until the assignment is dropped the rectangle is rendered transparent on the plan (see Fig. 4.1(b)). This movement is also distributed to all other clients, so all users can see the movement of the assignment. On the other clients the assignment is also rendered transparent until the assignment is dropped.

The application provides unlimited undo/redo functionality. All changes on assignments are saved in a database and can be made undone by using the undo button. The undo/redo function is user-specific which means that change histories are stored separately for each user.

4.4.2 *Detail Area*

The detail area not only shows the details of selected items in the planning area but the user can perform a wide range of interactions: search for customers, insert new assignments, insert new open failures, and plan customers with due maintenance contracts. It is organized using three tabs: The assignment tab, the planning tab, and the open failures tab.

Assignment Tab The assignment tab (see Fig. 4.1(c)) provides an interface for four activities: search, detail information, edit, and insert. Customers can be searched by entering data about the customers in the search box. The tool performs a full text search over all data fields of the customers. For example a search for 'Schneider Eisenstadt' will provide all customers with 'Schneider' in the name and living in 'Eisenstadt', furthermore the result will also contain customers with 'Schneider' in the name and living in the street 'Eisenstadt street' for example. This way to search is similar to internet search engines, which is familiar to most users. This is an advantage over conventional search functionality where the user has to specify the search criteria for each database field explicitly. When a single customer is selected in the search result, the newest existing assignment is shown in the planning area, so the user does not have to pan — the application does this automatically.

Below the search panel, all detail information about a selected assignment is shown including customer data, information about installed devices, and existing assignments. Assignment data and customer data like the phone number can be edited in place. Below that, a map view is integrated, where the location of the customer will be displayed. To insert an assignment the user has two alternatives: drag-and-drop the surrogate assignment below the map to the plan manually or let the system suggest a date for an assignment using an automated algorithm. Furthermore, open failures can be inserted using a button which are then displayed in the 'Open Failures' tab (see Fig. 4.1(f)).

Planning Tab The planning tab (see Fig. 4.1(d)) offers the functionality to do automatic planning of due contracts. The number of open customers are visualized using a heatmap (see Fig. 4.1(e)) where each square shows the data of one week, the number of customers is shown via color intensity, and the week number is displayed as text. The optimization procedure is initiated by the user upon selection of a number of weeks to plan. For the selected customers a plan will be generated using the savings algorithm [Clarke and Wright, 1964], as this algorithm is used in many applications in practice [Domschke, 1990] and delivered good results in our own tests. The algorithm optimizes the distances between the customers. In the first step, a route plan from the existing assignments is generated. Afterwards, the algorithm creates pending tours between each customer and the technician with the minimal distance to the customer. In the last step, savings by joining two tours are calculated: the algorithm starts with the biggest saving and joins two tours into one under compliance with the restriction (maximum number of tours per day per technician) until no tours can be joined anymore.

Upon completion of the computation, the assignments proposed by the automatic algorithm are shown in the planning area as semi-transparent assignments. The user can now review the suggested assignments and accept or deny the proposed plan. Using this human-in-the-loop approach, automatic planning and human judgment are closely coupled using interactive visual interfaces. If scheduled appointments need to be changed or cancelled, no automatic re-planning is performed. This is necessary because customers already got notified about their appointments and further changes might only be made upon human intervention.

Open Failures Tab A list of open failures (see Fig. 4.1(f)) is displayed on the third tab. This list can be seen as to-do list and the dispatcher has to search for suitable technicians and assign them to the open failures. The layout of this tab is very similar to the assignments tab, with the only difference that open failure assignments cannot be inserted automatically: the call center agent has to find a free technician by calling them on the phone.

4.4.3 *Collaboration area*

A-Plan supports synchronous and asynchronous collaboration as well as task-oriented and social awareness as suggested by Prinz [1999]. In the collaboration area (see Fig. 4.1(a), bottom) two lists are shown: the active users (left; for synchronous collaboration) and social awareness and common messages (right; for asynchronous collaboration). To facilitate the awareness of presence, all users currently working on the system are shown in the active users list. As soon as they close the application they disappear on all clients. The list of common messages can be used to notify the other users about important news as for example recalls of a gas device provider or vacations of technicians. Apart from that, activities like editing assignments are synchronized across clients and users for seamless coordination between dispatchers.

Overall, three different interactive visualizations are used in A-Plan that are tightly integrated: the planning area using timelines, the map view showing the location of selected assignments, and the heatmap to display amount of open assignments to plan on a weekly granularity. Moreover, an optimization algorithm is used for supporting the semi-automatic planning of assignments. Other more or less standard UI elements frame these core components and form a coherent tool design.

4.5 IMPLEMENTATION

As proof-of-concept for the described design, we implemented a prototype using C# in Visual Studio 2010 with .net 4.0. For the user interface parts, the WPF[7] framework was used and the communication between the server and the clients is implemented with WCF[8]. In the implementation the MVVM (Model-View-Viewmodel) pattern was used. The goal of MVVM is to keep the code as maintainable as possible by separating the user interface from the logic as strong as possible. The freely available MVVM light toolkit[9] was used to facilitate the work with the MVVM pattern.

From an architectural point of view, we followed a client-server approach for coordination. We used the net.tcp binding of WCF which is restricted to WCF applications only, but offers better performance. The interoperability provided by other bindings was not necessary for our application. In the application all calls between clients and the server are made asynchronously. If the call would be synchronously the processing of the application would be suspended until the call of the service is terminated. As these waiting times would cause interruptions for users, we decided to develop the communication between server and client with asynchronous calls. At the end of each function on the server the client is called with the result as parameter.

4.6 EVALUATION

In order to assess the usability and utility of our design and implemented prototype, a qualitative evaluation with domain experts was conducted. The goal of this evaluation was less focused on validating the correct behavior of the prototype, but on investigating to what extent A-Plan meets the presented requirements.

---

[7] WPF = windows presentation foundation

[8] WCF = windows communication foundation

[9] http://www.galasoft.ch/mvvm/getstarted (April 24, 2011)

4.6.1 *Method & Participants*

Our evaluation was structured into three parts: First, the prototype was demonstrated to the participants explaining the interface and basic functionality. After that, user testing took place where subjects had to carry out a set of given tasks. During that the thinking-aloud method was applied [Van Someren et al., 1994], i.e., participants were encouraged to verbalize their thoughts. Finally, semi-structured interviews were conducted in order to reflect on the design and usability of the prototype as well as gathering input on perceived advantages and disadvantages over the current work practice and suggestions for future work.

Five persons participated in the evaluation of A-Plan. Three of them were female and two male with an age between 26 and 49 years. Four out of them were domain experts with 2, 4, 15, and 20 years of experience in that area. Two of the domain experts work in customer care, one is a customer service technician, and one is an IT expert. One of the subjects was a master student in computer science who was not familiar with the domain and had no experience with planning software. The main focus was on the potential users of the system (the customer care agents). The reason to test the prototype with a service technician is that in times where the work load is very high for customer care agents, a service technician has to help in the back office. With the two IT experts we wanted to get more critical feedback about the usability of the prototype.

4.6.2 *Material & Analysis Approach*

For the test a PC and a laptop were used to demonstrate the possibilities of collaboration. During this process we made audio recordings and the activities of the system were recorded by a screen recorder software. In addition, written notes were taken by the study facilitator recording the activities of the testers. Through this multiple logging approach we wanted to avoid that interesting aspects will not be included in the analysis of the evaluation.

For the user testing, users had to carry out seven different tasks. These tasks were developed in order to cover the most important use cases identified in the user & task analysis as well as evaluating the novel interactive features and automatic planning functionality. Examples for the posed tasks are "The customer 'Kurt Schn..., Eisenstadt' reports a device failure, record this case", or "Move this assignment to the next day and set the assignment as fixed".

In the semi-structured interviews questions about the application, about the visualization of the plan, about the detail area (assignments and open failures), and about the long term planning functionality of A-Plan were used in the interview guideline.

The written notes and audio recordings of both the user testing and interview phases were analyzed along the proposed categories by Forsell and

Johansson [2010]. These are heuristics specifically developed for evaluating information visualizations to assess common and important usability problems.

4.6.3 *Results & Discussion*

The general feedback of study participants was very positive and a number of shortcomings and future improvements could be identified. When analyzing the found problems and issues based on the heuristics of Forsell and Johansson [2010], most of them were of the categories *B7 Orientation and help* and *B5 Information Coding* (both 10 times), followed by *E7 Minimal actions* (8 times). In the following we will provide more details on the gathered data.

Three persons (the customer care agents and the service technician) noted the speed in which the required tasks can be solved with A-Plan. For example the scheduling of a customer for maintenance in the current system can be estimated with four minutes effort, in A-Plan this can be done within a few seconds. The search function received much praise as well. As the way to search in A-Plan is familiar to all users from search engines it was conceived as being a very easy and fast way to find the desired data. The overall screen layout was clear to all testers. For three testers it was a little overloaded in some areas, but they also mentioned that they have no concrete suggestions for improvements. For one customer care agent the simple way to modify the data was unfamiliar, she would prefer an additional confirmation action for some operations. This might be attributed to the fact that this is the way how to work in the current system. One tester would prefer to use a save button instead of the automatic save with undo/redo capability. The program crashed in some situations. This was not a problem for the testers, as we educated them about the early development and test state of the system but for a productive system bugs should be corrected.

The visualization of the plan using timelines was clear for all testers. The functionality to show details by tooltips has not been recognized by the users. Only after we gave them a hint, they used it but found the function useful. The way to move assignments by drag-and-drop was no problem for the testers. The way to get details of assignments earned positive feedback too. However, of course also several problems were identified, some of which occurred for several or even all of the participants: The most common problem for the users during the prototype evaluation occurred when shifting assignments. The task was to move an assignment to the next day and all users tried to do this by dragging the timeline of the assignment to the next day. The problem occurred when they reached the border of the view which would not pan automatically and they could not move the assignment anymore. For the test persons the distinction between fixed and not fixed assignments using shape only was not strong enough. They would prefer to use different colors or a border around the fixed assignments. For three testers the way to zoom and pan was very unfamiliar, they would prefer a fixed view where no free panning and zooming is possible. One test person mentioned that as the movement of an assignment by drag-and-drop can happen unintentionally, she would prefer to move assignments with the right mouse button. Another problem occurred two times at the insertion of an open failure: two testers clicked the button to insert an new open failure more than once, which created two entries in the list that is hidden on another tab. Here, better system feedback should be provided making the user aware that an open failure was successfully recorded.

An important functionality that is missing, is the activity history of assignments. For example if an agent deletes an assignment, it is hard to comprehend what has been done, by whom, and why because the data does not exist anymore. This has been planned in the conceptual design but was not implemented due to time constraints.

The semi-automatic planning functionality earned very positive feedback. In the currently used system, planning has to be done completely manually which is a very time consuming operation. The computed plan of A-Plan was perceived as acceptable to the study participants and the handling of the planning process was found to be easy and clear. Only one tester had problems to understand the heatmap representation which shows open maintenances to plan.

4.7 conclusion & future work

In this work we reported on the design, prototypical implementation, and evaluation of a VA tool for scheduling of technicians for gas device maintenance. In the beginning we investigated the field of work in such a company. It became apparent that the standard software used does not support the needed tasks well. Interestingly, paper-based workarounds have been developed by the employees of the investigated company to mitigate these problems. We were astonished to encounter this sophisticated system of paper-based artifacts to reach a more or less smooth working environment. This provided very valuable insights for our own development.

Based on the information gained via contextual observation and interviews, we designed and implemented a prototype called A-Plan that integrates interactive visualizations with automated planning and supports collaborative work. A-Plan uses an interactive visualization for presenting planned assignments that is based on timelines. A detail view of assignments includes an interactive map view for localizing customers. Furthermore, an automated planning functionality based on the savings algorithm has been integrated which allows for bulk planning of recurring service contracts and is supported by a heatmap visualization. The algorithm accounts for a complex set of constraints like geographic areas and timings and is suggesting an automatically optimized set of assignments. The suggested plan can be reviewed and altered by the user via the interactive visual interface.

We evaluated the implemented prototype with three users of the current system and two IT experts. Testing the prototype with people who do their daily work in this field yielded much interesting feedback. Some issues arose from the fact that the users who did the evaluation were not familiar with techniques like direct manipulation. Overall, we received encouraging feedback and were able to identify shortcomings of the design and functionality of A-Plan. All test subjects would prefer to use A-Plan instead of the existing system.

Some general lessons learned for future developments in this area are that timelines are an easily understandable visual representation and allow for intuitive user interaction. Furthermore, combining automatic and visual methods in a semi-automatic fashion is a well-suited approach for this problem area. Especially, using a visual representation to display the suggestions of the optimization algorithm and make them manipulable was praised by users. Fully automating the process might be doable in theory but in that case users might no longer have the feeling of being in control and might not be able to create a mental model of the inner workings of the system. We believe that human reasoning capabilities add value to the planning process and that the taken approach is superior to both, purely manual and fully automatic methods. From a user's point of view, main challenges are to support synchronous collaboration in real-time and that needed information can be found quickly.

As A-Plan is currently in the prototype stage, a number of directions for future work remain. Apart from fixing a number of software bugs the issues that surfaced in our evaluation should be addressed, as for example the movement of an assignment outside the current view, the visual distinction of fixed and variable assignments, and improvements in direct manipulation. Other than that, introducing a semantic zoom functionality could increase and optimize the displayed information.

The main contribution of our work is that we have demonstrated the successful application of a Visual Analytics approach in the context of resource scheduling. We have shown an effective integration of automatic methods and interactive visualizations based on a user-centric development approach. The integration of the strength of both the human and the computer enables the creation a powerful environment for a set of non-trivial and complex tasks.

ACKNOWLEDGMENTS

This work was supported by the Centre for Visual Analytics Science and Technology CVAST (funded by the Austrian Federal Ministry of Economy, Family and Youth in the Laura Bassi Centres of Excellence initiative).

REFERENCES

Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of Time-Oriented Data*. Springer.

Bartlett, J. C. and Toms, E. G. (2005). Developing a Protocol for Bioinformatics Analysis: An Integrated Information Behavior and Task Analysis Approach. *Am. Soc. Inf. Sci. Technol.*, 56(5):469–482.

Bederson, B. B., Clamage, A., Czerwinski, M. P., and Robertson, G. G. (2004). DateLens: A Fisheye Calendar Interface for PDAs. *ACM TOCHI*, 11(1):90–119.

Bektas, T. (2006). The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures. *Omega*, 34(3):209–219.

Bräysy, O. (2004). Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. *Heuristics*, 10(6):587–611.

Cheung, R., Xu, D., and Guan, Y. (2007). A Solution Method for a Two-dispatch Delivery Problem with Stochastic Customers. *Math. Model. Algorithm*, 6:87–107.

Clarke, G. and Wright, J. V. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Oper. Res.*, 12(4):568–581.

Cockburn, A., Karlson, A., and Bederson, B. B. (2008). A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, 41(1):1–31.

Cooper, A. (2004). *The Inmates Are Running The Asylum*. Sams Publishing.

Cooper, A., Reimann, R., and Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design*. Wiley Publishing.

Domschke, W. (1990). *Logistik, Rundreisen und Touren*. Oldenbourg, 3rd edition.

Forsell, C. and Johansson, J. (2010). An Heuristic Set for Evaluation in Information Visualization. In *Proc. of Int. Conf. on Adv. Visual Interfaces (AVI)*, pages 199–206. ACM.

Gillett, B. and Miller, L. (1974). A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Oper. Res.*, 22(2):340–349.

Hanshar, F. (2007). Dynamic Vehicle Routing Using Genetic Algorithms. *Appl. Intell.*, 27(1):89–99.

Karam, G. M. (1994). Visualization Using Timelines. *Proc. of Int. Symp. on SW Testing and Analysis (ISSTA '94)*, pages 125–137.

Kulyk, O., Kosara, R., Urquiza, J., and Wassink, I. (2007). Human-Centered Aspects. In Kerren, A., Ebert, A., and Meyer, J., editors, *Human-Centered Visualization Environments*, pages 13–75. Springer.

Luz, S. and Masoodian, M. (2007). Visualisation of Parallel Data Streams with Temporal Mosaics. In *Int. Conf. Information Visualization (IV '07)*, pages 197–202. IEEE.

Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. (1996). LifeLines: Visualizing Personal Histories. In *Proc. of Conf. on Human Factors in Computing Systems (CHI96)*, pages 221–227. ACM.

Prinz, W. (1999). NESSIE: An Awareness Environment for Cooperative Settings. In *Proc. of the Europ. Conf. on Comp. Supported Cooperative Work (ECSCW'99)*, pages 391–410. Kluwer Academic Publishers.

Shneiderman, B. and Plaisant, C. (2004). *Designing the User Interface*. Addison Wesley, 4th edition.

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE.

Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*. SIAM.

Van Someren, M. W., Barnard, Y. F., and Sandberg, J. A. (1994). *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Academic Press Limited.

CHAPTER 5

Reinventing the Contingency Wheel: Scalable Visual Analytics of Large Categorical Data

Bilal Alsallakh, Wolfgang Aigner, Silvia Miksch, and Eduard Gröller

Abstract · *Contingency tables summarize the relations between categorical variables and arise in both scientific and business domains. Asymmetrically large two-way contingency tables pose a problem for common visualization methods. The Contingency Wheel has been recently proposed as an interactive visual method to explore and analyze such tables. However, the scalability and readability of this method are limited when dealing with large and dense tables. In this paper we present Contingency Wheel++, new visual analytics methods that overcome these major shortcomings: (1) regarding automated methods, a measure of association based on Pearson's residuals alleviates the bias of the raw residuals originally used, (2) regarding visualization methods, a frequency-based abstraction of the visual elements eliminates overlapping and makes analyzing both positive and negative associations possible, and (3) regarding the interactive exploration environment, a multi-level overview+detail interface enables exploring individual data items that are aggregated in the visualization or in the table using coordinated views. We illustrate the applicability of these new methods with a use case and show how they enable discovering and analyzing nontrivial patterns and associations in large categorical data.*

Keywords · Large categorical data, contingency table analysis, information interfaces and representation, visual analytics.

This article originally appeared as [Alsallakh et al., 2012]:

Alsallakh, B., Aigner, W., Miksch, S., and Gröller, E. (2012). Reinventing the Contingency Wheel: Scalable Visual Analytics of Large Categorical Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2849–2858. Best paper honorable mention.

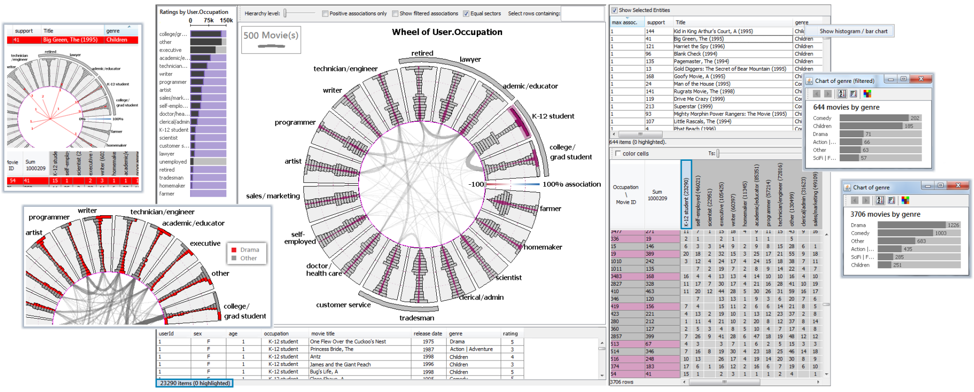


Figure 5.1 – **Contingency Wheel++** uses complementing visual representations and a multi-level overview+detail user interface to enable rich exploratory analysis of large categorical data. The example above shows information about 1 million user ratings on 3706 movies.

5.1 INTRODUCTION

Many problems in scientific domains such as medicine, biology and pharmacology, as well as in business domains such as retail and logistics require analyzing associations between categorical variables. For example, a movie retailer might be interested in associations between movies and users based on sales data with the goal of optimizing marketing strategies. The discrete nature of categorical data and their lack of an inherent similarity measure pose significant challenges to the fields of information visualization [Bendix et al., 2005] and data mining [Xiong et al., 2009]. Contingency tables (also known as crosstabs) are a common way to summarize categorical data as a first step of analysis. A two-way contingency table is an $n \times m$ matrix that records the frequency of observations f_{ij} for each combination of categories of two categorical variables. Many data analysis frameworks such as KNIME [Berthold et al., 2008], WEKA [Hall et al., 2009] and R [R Development Core Team, 2009] offer possibilities to create and analyze contingency tables. One of the best-known statistical tests for the overall association (or independence) between two categorical variables is Pearson's χ^2 test [Rao and Scott, 1981]. It assesses the significance of associations between the categories of the two variables. However, it does not provide information about how single pairs of categories are associated.

Several visualization methods were developed to analyze associated categories in contingency tables. As we discuss in Sect. 5.5, these methods are designed to handle rather small tables having few categories. However, often much larger contingency tables need to be analyzed, which poses a problem to these methods. Fig. 5.2a shows large categorical data from the MovieLens

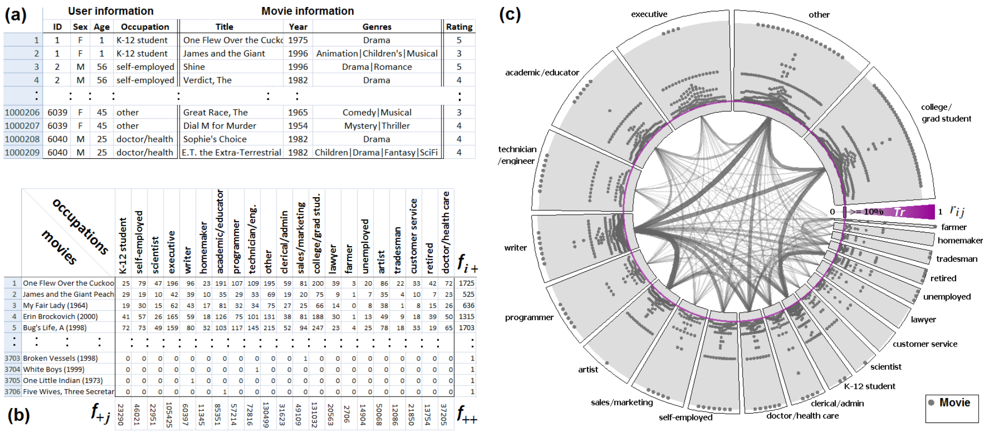


Figure 5.2 – (a) Categorical variables of the MovieLens data set [GroupLens, 2012] showing about one million user ratings on 3706 movies, (b) the contingency table of the variables "movie title" and "user occupation", (c) the Contingency Wheel of the table in (b): Sectors represent occupations and dots represent movies positively associated with them. Thicker arcs show which occupations share more movies highly associated with both of them.

data set [GroupLens, 2012]. It contains about one million user ratings on movies. For each user, it provides his or her occupation, sex, and age group, and for each movie, its release date and genres. Examples for tables extracted from this data set are:

- A 3706 × 21 table which counts for each movie, how many times it was rated from users of each occupation (Fig. 5.2b).
- A 6040 × 17 table which counts for each user, how many times he/she rated movies from each genre (Fig. 5.6d).

The Contingency Wheel [Alsallakh et al., 2011] has been introduced as an interactive visual method for exploring positive associations in asymmetrically large tables. The column categories are visualized as sectors of a ring chart and the table cells are depicted as dots in these sectors (Fig. 5.2c). The dot for cell (i, j) is placed in sector i at a radial distance from the ring's inner circle proportional to the strength of association r_{ij} between row i and column j. A layout algorithm calculates the angular positions of the dots in each sector to reduce occlusion. It copes with a large number of rows by visualizing only the cells that represent significant associations r_{ij}, determined by adjustable thresholds. An arc is drawn between two sectors if one or more rows have dots in both sectors. This arc is thicker if more such rows exist and if their dots represent higher associations with both sectors. User interaction enables analyzing different types of associations in large tables.

Scalability is one of the major challenges visual analytics aims to address [Thomas and Cook, 2005]. The wheel metaphor explained above has several shortcomings which degrade its readability and scalability, especially with large and dense tables (Sect. 5.2). In this paper we propose Contingency Wheel++: new visual analytics methods that tackle the issues of the original wheel. Our methods (described in Sect. 5.3) address its computational component, visual representation and interactive interface, and intertwine these three components to enable scalable analysis of categorical data. The new methods encompass:

- *Automated methods*: a new association measure results in a better distribution of the dots to sectors of different sizes. This is important when analyzing large tables that often exhibit high skewness in the distribution of their frequencies.
- *Visualization methods*: a frequency-based abstraction of the dots eliminates overlapping which allows showing all the cells, instead of just small subsets thereof. This enables analyzing and querying both positive and negative associations.
- *Interactive exploration environment*: an overview+detail interface allows exploring individual items aggregated in the visualization or in the table, and analyzing their attributes.

In Sect. 5.4 we present a use case to illustrate how our new methods can be used to explore the MovieLens data set. We show how nontrivial patterns and associations in the data can be discovered. In Sect. 5.5 we compare our approach with other methods for visualizing categorical data and elaborate on its scalability.

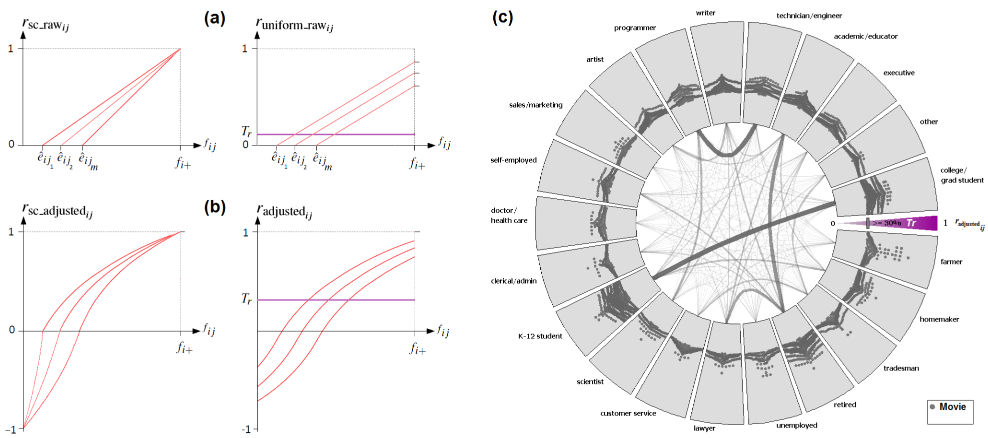


FIGURE 5.3 – (a) raw residuals and (b) adjusted residuals plotted as a function of f_{ij} for different values of f_{+j} with both nonuniform- (left) and uniform scaling (right), (c) the same data plotted in Fig. 5.2c using uniformly-scaled adjusted residuals instead of raw residuals (with $T_r = 30\%$ and $T_s = 1$).

5.2 LIMITATIONS OF THE CONTINGENCY WHEEL

Based on a pilot evaluation study of the Contingency Wheel [Kriglstein et al., 2012], we identified several issues that limit its readability and scalability. In particular, we focus on issues related to the conceptual design of the wheel and its interpretability rather than usability issues:

Data mapping: The Contingency Wheel visualizes association values r_{ij} that represent deviations from expected values (Sect. 5.3.1) rather than absolute frequencies f_{ij}. Many users did not have sufficient background on statistical association measures to interpret that correctly.

Visual mapping: Users agreed that the visualization provides a quick overview of the distribution of dots within sectors as compared with a tabular view. However, they found it difficult to accurately interpret the meaning of these dots at the beginning. They expected absolute frequencies f_{ij} rather than association values. It was confusing that the dot size and its radial position convey the same information. The angular position of the dots was even more confusing since it bears no meaning. It was also confusing that dots in different sectors can represent the same entities. Though arcs are intended to clarify this fact, users realized it only after selecting a dot (which also highlights all dots in other sectors that represent the same row).

Interaction: Dots closer to the center were often too small and overlapping, which made them difficult to identify. The same issue applies to arcs between small sectors. Also, filtering the dots by moving a slider became clear only after the users understood the data representation. Some users forgot that parts of the dots were filtered out and drew wrong conclusions about the data.

Most of the above-mentioned readability issues are related to dots. Dots as representations of individual table cells suffer inherently from limited scalability: Only a few hundred dots can be shown at once without overlapping. The Contingency Wheel reduces the large number of dots by filtering out cells (i, j) with $r_{ij} \leq T_r$ (where T_r is the association threshold) and by filtering out entire rows with $f_{i+} < T_s$ (where T_s is the support threshold) [Alsallakh et al., 2011]. However, filtering limits the ability to gain insights into the whole dataset and it does not work well for dense tables with large f_{ij} values.

Contingency Wheel++ [1] improves both on the readability and on the scalability issues mentioned above by employing visual analytics methods as presented in the next section.

[1] A prototype implementation of Contingency Wheel++ is available at http://www.cvast.tuwien.ac.at/wheel

5.3 CONTINGENCY WHEEL++

In the following, f_{ij} denotes the frequency in cell (i, j), $f_{i+} = \sum_{j=1}^{m} f_{ij}$ and $f_{+j} = \sum_{i=1}^{n} f_{ij}$ are the marginal row- and column frequencies, and f_{++} is the sum of all table frequencies (Fig. 5.2b). We first address the data mapping employed by Contingency Wheel++ (Sect. 5.3.1). Then we propose a frequency-based visual representation which abstracts the dots (Sect. 5.3.2). In Sect. 5.3.3 we show how an interactive visual interface integrates additional table views to bridge the gap between the data representation and the visual representations and to support a flexible visual exploration process.

5.3.1 *Mapping Frequencies to Associations*

The main goal of Contingency Wheel++ is to reveal how the row categories of a contingency table are associated with its column categories. For this purpose, it uses a statistical measure r_{ij} that computes the association between row i and column j based on f_{ij} and takes value in the range $[-1, 1]$. This measure is usually based on statistical residuals between the actual frequency f_{ij} and expected frequencies $\hat{e}_{ij}$. The frequency in cell (i, j) predicted under the null hypothesis H_0, i.e., assuming no association, is [Simonoff, 2003]:

$$\hat{e}_{ij} = \frac{f_{i+} \cdot f_{+j}}{f_{++}} \tag{5.1}$$

If $f_{ij} = \hat{e}_{ij}$ holds for cell (i, j), its share f_{ij}/f_{i+} of the marginal row frequency is equal to the column's share f_{+j}/f_{++} of all table frequencies. This means that row i is neither positively nor negatively associated with column j, and corresponds to a zero association value $r_{ij} = 0$. Cells with $f_{ij} > \hat{e}_{ij}$ indicate a positive association between row i and column j. Statistical residuals r_{ij} can be used to quantify this association. They can be designed to incorporate a priori information about the data and their distribution. In the following we describe the originally-used residuals and our improvements on them.

Raw residuals The association measure used originally by the Contingency Wheel is based on raw residuals $(f_{ij} - \hat{e}_{ij})$ [Alsallakh et al., 2011]. To generate association values $r_{ij} \leq 1$, the raw residual for cell (i, j) is divided by the maximum value it can take $(f_{i+} - \hat{e}_{ij})$:

$$r_{\text{sc\_raw}_{ij}} = \frac{f_{ij} - \hat{e}_{ij}}{f_{i+} - \hat{e}_{ij}} \tag{5.2}$$

This measure maps frequencies linearly to association values (Fig. 5.3a-left). The maximum association $r_{ij} = 1$ is reached when all cells of row i have zero frequencies except for cell (i, j). For such a row, only one dot is created on the outer boundary of sector j. A cell with $r_{ij} = 0$ creates a dot on the inner boundary of sector j (assuming no thresholds). Cells with negative associations are ignored. The above-mentioned normalization is not uniform with respect to the columns: For row i, different scaling factors are used in different columns, because the expected frequency $\hat{e}_{ij}$ is larger for columns with larger f_{+j}. This makes better use of the sector area for revealing the distribution of dots along the radial dimension. Also, rows i that are fully associated with column j ($f_{ij} = f_{i+}$) can be easily found as dots at the outer boundary. However, the different scaling factors result in a bias especially when f_{+j} varies largely between sectors. This impacts the comparison of associations between different sectors and reduces the expressivity of the arcs. A uniform scaling factor for all columns can be used instead:

$$r_{\text{uniform\_raw}_{ij}} = \frac{f_{ij} - \hat{e}_{ij}}{f_{i+}} \tag{5.3}$$

Fig. 3a-right shows how this scaling maps frequencies to associations. For cells with $f_{ij} = f_{i+}$, Eq. 5.3 evaluates to $1 - f_{+j}/f_{++}$ which is independent of i. Such cells are hence mapped to the same radial distance within a sector (Fig. 5.2c). The sectors are scaled by their marginal frequencies. Sectors with larger f_{+j} values attract more dots than sectors with smaller f_{+j} values, due to an inherent statistical bias that raw residuals suffer from (even with uniform scaling).

Adjusted residuals Standardized Pearson residuals [Simonoff, 2003] avoid the bias of raw residuals by adjusting the variance of the r_{ij} values to $N(0,1)$:

$$r_{\text{pearson}_{ij}} = \frac{f_{ij} - \hat{e}_{ij}}{\sqrt{\hat{e}_{ij} \cdot (1 - f_{i+}/f_{++}) \cdot \left(1 - f_{+j}/f_{++}\right)}} \tag{5.4}$$

We use a logarithmic scale for the visual mapping of these residuals to better reveal their distribution along the radial dimension (where cte is a constant computed from the table to ensure $-1 \le r_{ij} \le 1$):

$$r_{\text{adjusted}_{ij}} = \frac{\operatorname{sgn}(r_{\text{pearson}_{ij}})}{cte} \cdot \ln\left(1 + \left|r_{\text{pearson}_{ij}}\right|\right) \tag{5.5}$$

Fig. 5.3b-right, shows how this measure maps frequencies to associations. Fig. 5.3c shows the same data as in Fig. 5.2c using $r_{ij} = r_{\text{adjusted}_{ij}}$ with $T_r = 30\%$ and with equal sectors. The dots are distributed more uniformly among the sectors. This results in arcs that suggest other similarities between occupations. The logarithmic scale amplifies smaller raw residuals, giving them more visual prominence. This potentially generates more dots, and hence a higher value for T_r is needed to filter out insignificant associations. Cells with $f_{ij} = f_{i+}$ are mapped to different radial distances in sector j, depending on f_{i+}. This makes the arcs more robust to changes in T_s since rows with smaller f_{i+} values contribute less to the arcs. On the other hand, these cells are somewhat difficult to locate. The following nonuniform scaling stretches r_{ij} to the range $[-1, 1]$:

$$r_{\text{sc\_adjusted}_{ij}} = \frac{r_{\text{adjusted}_{ij}}}{\max\left(s_{ij} \cdot r_{\text{adjusted}_{ij}}|_{f_{ij}=f_{i+}}, s_{ij} \cdot r_{\text{adjusted}_{ij}}|_{f_{ij}=0}\right)} \tag{5.6}$$

where $s_{ij} = \operatorname{sgn}(r_{\text{adjusted}_{ij}})$ and $r_{\text{adjusted}_{ij}}|_{f_{ij}=x}$ is the value $r_{\text{adjusted}_{ij}}$ would take if $f_{ij} = x$. Fig. 5.3b-left depicts how this scaling maps frequencies. As can be seen, rows with $f_{ij} = f_{i+}$ are always mapped to the largest radial distance. Also, if the visualization can include negative associations, rows with $f_{ij} = 0$ are always mapped to the lowest radial distance. Nonuniform scaling, however, re-introduces a small bias in the associations, toward columns with larger f_{+j}.

5.3.2 *Visualizing the Contingency Table*

The visualization aims to reveal how the row categories of a contingency table are associated with its column categories, based on the association measure used. Our new visual representation makes use of the advantages of uniformly adjusted residuals (Sect. 5.3.1). It provides a clearer and more intuitive visualization of the table, as compared to the original wheel design [Alsallakh et al., 2011]. Moreover, depending on the user's choice, it enables showing all associations or positive associations only as we describe in the following subsections.

Visualizing columns Like in the original wheel metaphor, columns are drawn as sectors of a ring chart. The main difference is that they are drawn with equal size. This has several advantages: First, this is in accordance with the fact that adjusted residuals result in a more uniform distribution of the cells to the sectors. Second, by using a frequency-based representation (Sect. 5.3.2), the distribution of the associations can be compared between different sectors. Third, the arcs become evenly distributed in the central area, unlike the arcs in Fig. 5.2c which overlap more near small sectors. Finally, column categories are treated equally from a visual point of view, in the same way as the dimensions of a star plot [Harris, 1999]. This simplifies the visualization and eliminates confusion about the meaning of different sector sizes. The information of different column marginal frequencies f_{+j} is conveyed in a linked bar chart (Sect. 5.3.3). Incorporating it in the wheel representation would not contribute to the goal of Contingency Wheel++, i.e., to explore associations.

Visualizing row-column associations The radial dimension of the ring chart linearly encodes the association values r_{ij} computed by one of the association measures. The outer boundary corresponds to $r_{ij} = 1$. The inner boundary corresponds to $r_{ij} = -1$ if showing all associations, and to

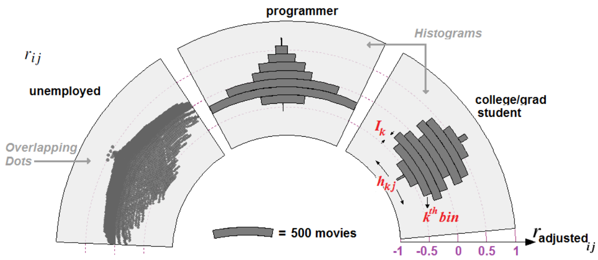


FIGURE 5.4 – **Dot vs. histogram representation of row-column associations.** The dimensions of a histogram bin are annotated (Eq. 5.8).

$r_{ij} = 0$ if showing positive associations only. Instead of the dot representation originally used, we suggest a frequency-based representation to visualize the row-column associations. A histogram H_j is created in each sector j to show the distribution of the associations r_{ij} along the radial dimension. An adjustable number b of equal bins is used for all histograms, initially determined by Scott's normal reference rule [Scott, 1979]. Each bin k in sector j aggregates the rows i having associations in the interval $I_k = [l_k, l_{k+1}[$. The interval boundaries l_k are equally spaced between $[-1, 1]$:

$$l_k = \frac{2(k-1) - b}{b} \tag{5.7}$$

A closed interval $I_b = [l_b, 1]$ is used for the last bin to account for $r_{ij} = 1$. Hence, the number of items h_{kj} in the k^{th} bin of sector j is:

$$h_{kj} = \left| \left\{ 1 \leq i \leq n : f_{i+} \geq T_s \wedge r_{ij} \in I_k \right\} \right| \tag{5.8}$$

Each bin k of histogram H_j is visualized as a track in sector j. This track occupies the radial position which corresponds to its interval I_k. The length of this track is proportional to h_{kj}. A uniform or sector-specific scaling factor ensures that all tracks fit in their sectors. Tracks are centered in their sectors, following the Gestalt principle of symmetry [Wertheimer, 1938]. This avoids artificial asymmetry along the angular dimension in different sectors and makes it easier to compare their histograms. Fig. 5.4 shows both dot and histogram representations for some sectors of Fig. 5.3c. The histograms show how 3706 movies are associated with two occupations. Both positive and negative associations are included.

Rows whose associations with sector j lie in a specific interval can be inspected individually along with the attributes of their entities, as explained in Sect. 5.3.3. The distribution of a numerical or categorical attribute of these entities can be shown by coloring the histograms instead of coloring individual dots. This provides a clearer understanding of the attribute distribution at different radial distances. Fig. 5.5a shows the release-date distribution of movies positively associated with specific occupation categories. Fig. 5.5b shows the genres of the movies. Movies highly associated with the "Retired" category tend to be old. The opposite holds for the "K-12 student" category which also tends to be highly associated with "Children" movies. Movies highly associated with "Technician/Engineer" are more likely to have "Sci-Fi / Fantasy" genres. These tendencies seem stronger as compared to the distribution of both attributes among all movies (Fig. 5.5c).

The frequency-based representation has several advantages over the dot representation: First, the angular dimension now has a clear meaning (frequency of associations at different radial distances in the sectors). Second, the artifacts and overlaps caused by showing separate dots are eliminated. Third, histograms are familiar visualizations that are easy to interpret. They better emphasize that the visualization is showing a distribution of the row associations in each sector, and not individual entities. This avoids the confusion due to multiple dots representing the same row. Finally, the redundancy of double-coding the association using both dot size and dot location is also eliminated.

Bended histograms embedded in a ring chart suffer from visual illusions in perceiving different arc lengths at different radial distances. This effect can be accounted for computationally and is minimized when arcs are short that are perceptually flattened [Robinson, 1998] (Fig. 5.6).

Visualizing column similarities We compute similarities between the columns of the contingency table based on their row associations. A similarity value $rc_{j_1 j_2}$ is computed for every pair of columns (j_1, j_2), to assess how similar

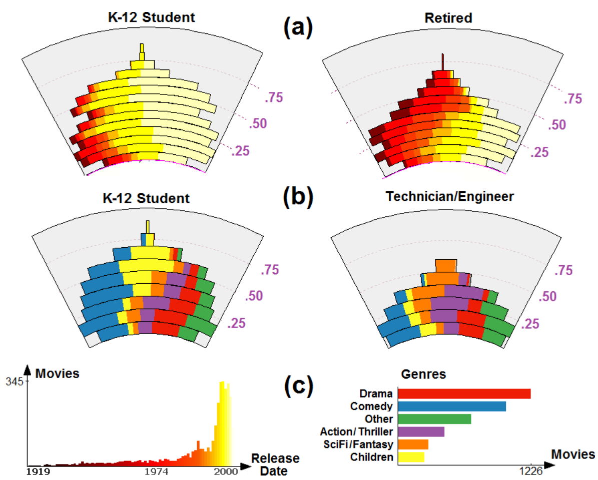


FIGURE 5.5 – **Distributions** of (a) a numerical attribute (release date) or, (b) a categorical attribute (genre) of the movies in the histograms. (c) The global distributions of release date and genre among all movies.

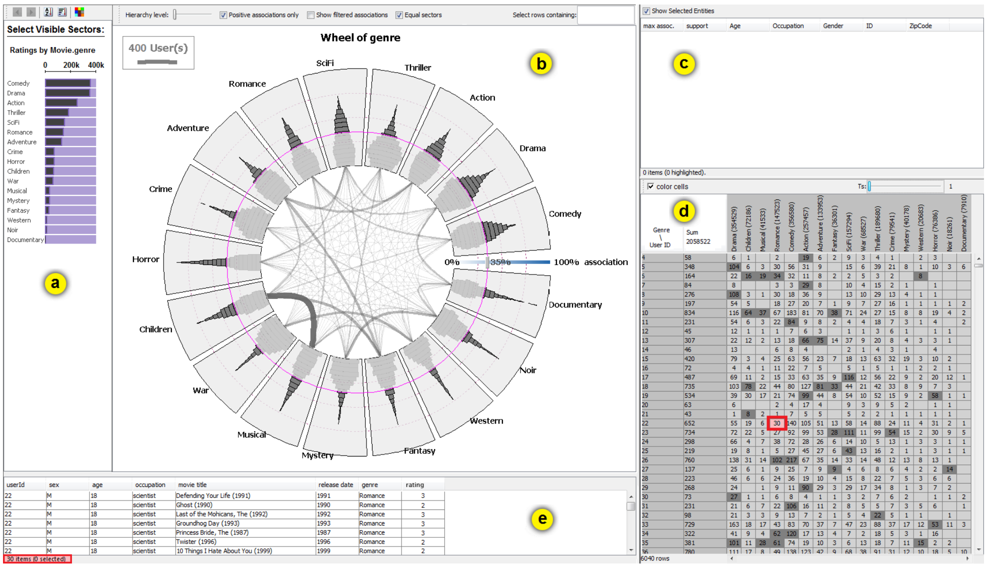


FIGURE 5.6 – **Five levels of abstraction to explore the user-genre table and the underlying information:** (a) a bar chart of the column categories (genres), (b) the wheel view showing sectors for the items selected in (a), (c) detail view for items selected in (b) (currently empty), (d) the contingency table with cells in active parts in (a) colored in dark gray, (e) the categorical data summarized in the cell highlighted in red in (d).

the two distributions r_{ij_1} and r_{ij_2} are. Only active rows in both sectors are included in the computation. Active rows in sector j have sufficient support f_{i+} and associations r_{ij} higher than T_r, and are defined as follows:

$$A_j = \left\{1 \leq i \leq n : r_{ij} \geq T_r \wedge f_{i+} \geq T_s\right\} \tag{5.9}$$

Active rows in each sector are depicted in dark gray in the respective histogram (Fig. 5.6b). The column similarities are computed as follows:

$$rc_{j_1 j_2} = \frac{1}{\left|A_{j_1}\right| + \left|A_{j_2}\right|} \cdot \sum_{i \in A_{j_1} \cap A_{j_2}} r_{ij_1} \cdot r_{ij_2} \tag{5.10}$$

Between each pair of sectors (j_1, j_2), an arc is drawn whose thickness and opacity are determined by $rc_{j_1 j_2}$. A thick arc means that the active rows in both sectors tend to have similar associations with the two columns j_1 and j_2. Changing the T_r value results in smaller or larger active parts, and hence influences the thicknesses. By checking the arcs with different T_r values, the user can examine in which ranges and to which extent the column similarities hold.

Arcs showing column similarities based on row associations is a unique feature of the Contingency Wheel and one of the main reasons of adopting a circular layout for the visualization. This layout provides a compact representation to show and compare column similarities. Furthermore, arcs are useful in creating a user-controlled hierarchical grouping of the column categories based on their similarities: A right-click on an arc merges the two affected sectors into one sector. The resulting wheel is built from the contingency table that results by merging the corresponding columns into one column, by summing up the frequencies cell-by-cell. The new sector is inserted at its appropriate position according to the sector ordering scheme currently in use (alphabetical, by size, or user-defined sector ordering). Successively merging pairs of sectors connected by thick arcs enables abstracting the visualization by reducing the number of visual items. Moreover, it enables analyzing similarities between groups of similar columns and not only between pairs of columns, as the use case shows (Sect. 5.4).

Visual aids We provide several visual aids to facilitate understanding. Three association levels evenly spaced between the inner and the outer sector boundaries are shown to allow an easier interpretation of the radial distances. An additional circle in pink shows the current value of the association threshold T_r, which can be adjusted using the slider embedded in the ring chart. Inactive parts of the histograms (Eq. 5.9) are visually de-emphasized. A color gradient is shown in the background of the T_r slider to reflect the association levels. It uses either a diverging or a sequential color scale [Harrower and Brewer, 2003], depending on whether negative associations are included or not. Arcs outside the ring chart indicate sector groups (Fig. 5.1). Finally, a legend shows the scale used in the histograms by depicting an arc of average length.

5.3.3 *Interactive Exploration Environment*

The original Contingency Wheel may result in a cluttered visualization especially for large data because it creates dots for individual row entities. These dots need to be selected individually to obtain details about the corresponding entities [Alsallakh et al., 2011]. To improve on these shortcomings, our new methods follow Shneiderman's visual information-seeking mantra [Shneiderman, 1996]: The visualization first shows an overview of the data using histograms. The user can filter the data interactively and select entities she is interested in exploring. Then, details about these entities can be obtained in linked views. Contingency Wheel++ offers an overview visualization of an asymmetrically-sized contingency table. Likewise, the contingency table offers a summarization of a larger data set by cross-tabulating two categorical dimensions. We designed the user interface to enable exploring the data at these multiple levels of abstraction as explained in the following.

Multiple Views Whenever we explain Contingency Wheel++ to new users, our first step is to show the underlying contingency table. This allows explaining the basic concepts like row- and column marginal frequencies, actual- and expected frequencies (Eq. 5.1), and row-column associations (Eq. 5.2-5.5). We are thus showing both the wheel visualization and the underlying table side-by-side in one interface. This combination bridges the gap between the visual representations and the data representation (i.e., association values) computed by the automatic methods. The main user interface (UI) of our prototype is divided into five coordinated views:

A *bar chart* shows the column categories and their marginal frequencies f_{+j} (Fig. 5.6a). Columns selected in this view define the sectors of the wheel view. The user can thus focus on selected columns. Also, if the number of columns exceeds the limits for the wheel, smaller subsets of columns can still be visualized.

The *wheel view* is the central part of the interface (Fig. 5.6b). It provides an overview of the data and existing associations within. Several interactions are possible to find interesting patterns in the data and select specific row entities for further analysis. The association threshold T_r can be adjusted interactively via the slider embedded in the ring chart. Also, this view enables setting several parameters by means of its toolbar and context menu.

A *list view* shows details about the row entities selected in the wheel view (Fig. 5.6c and Fig. 5.7d). Beside the attributes of these entities (available from the data set), their marginal row frequencies f_i+ and associations r_{ij} with a specific column j are listed. The entities can be sorted according to one of the columns, and histograms or bar charts can be created for a specific column in the list.

A *tabular view* shows the contingency table and the marginal frequencies (Fig. 5.6d). By hovering the mouse pointer over a cell (i, j), a tooltip shows the expected frequency $\hat{e}_{ij}$ and the association value r_{ij} according to the measure used. If cell coloring is enabled via a checkbox, the cell is shown in dark gray if it corresponds to an active part in the visualization (i.e., $i \in A_j$). Also, the support threshold T_s can be adjusted via a slider to filter out entire rows i with $f_{i+} < T_s$.

A *second list view* shows details about selected items from the tabular view (Fig. 5.6e). By double-clicking on a cell, a row, or a column in the tabular view, cross-tabulated data items are shown in this list view along with their attributes. The items can be sorted and the distributions of the values in a specific column can be explored using a histogram or a bar chart.

These views make it easier to explain to new users how the data are visualized in Contingency Wheel++. Even more importantly, they constitute a multi-level overview+detail exploration interface. This allows experienced users to perform elaborate analysis workflows by having quick access to all information available in the data. Hence, associations can be detected and investigated further in relation to other attributes. The incorporation of analytical methods in the visual interface enables a visual analytics process following Keim's mantra [Keim et al., 2008]: Analyze first – show the important – zoom, filter and analyze further – details on demand. After computing the row-column associations (Sect. 5.3.1) and the columns similarities (Sect. 5.3.2), the visualization shows the important results, i.e., strong associations or high similarities. Using different interactions, the user can change the thresholds T_r and T_s, merge columns, or set a different association measure. This causes the analytical methods to recompute the associations and similarities which are then visualized interactively. Details on selected items in the wheel or in the tabular view can be obtained on demand.

Linking and Brushing Contingency Wheel++ offers multiple ways to brush the visualized row categories. One way is by clicking on a bar in the histograms, which selects the rows it aggregates (Sect. 5.3.2). Another way is using the sector marquee tool to define a radial interval I in a sector j using the mouse (Fig. 5.1). This selects the rows i with $r_{ij} \in I$. Clicking on sector j selects the rows A_j that are currently active (Sect. 5.3.2). Also, clicking on an arc selects rows active in both sectors it connects (the items that define this arc). Rows i with $r_{ij} \leq T_r$ for all columns j can be selected using a menu command. When T_r is positive but small, this command selects rows that do not exhibit a high association with any column. Finally, rows can be selected using an external query, like the instant search box at the top of the view. This box selects row categories containing a specific text.

The top-right list view (Fig. 5.7d) shows the selected rows defined either by filtering, brushing, selection, or the search box query. When an item in this list is clicked, the tabular view scrolls to and highlights the corresponding row i which shows the frequencies f_{ij} (Fig. 5.10e). Also, a star graph [Harris, 1999] of the associations r_{ij} can be shown in the wheel view, labeled with these frequencies. Selected row categories are highlighted in the histograms of all sectors. The original histograms become desaturated and new sub-histograms are drawn centered on top of them showing the selected portion using color (Fig. 5.7c). Likewise, the original arcs are desaturated and the parts corresponding to the selected items are highlighted. Three modes are offered for performing brushing operations in the wheel, depending on keyboard modifiers:

- Set union: the new selection is added to an existing selection.
- Set intersection: the new selection is intersected with the existing selection. This enables creating nested queries on the data. For example, in the wheel showing the movies-occupation table, the user can select movies highly associated with the categories "programmer" and "scientist" but negatively associated with the category "executive". This is done by drawing ranges at the corresponding radial distances in each sector while the CTRL key is pressed. TimeSearcher uses a similar brushing technique for time-series data by means of timebox widgets [Hochheiser and Shneiderman, 2004].

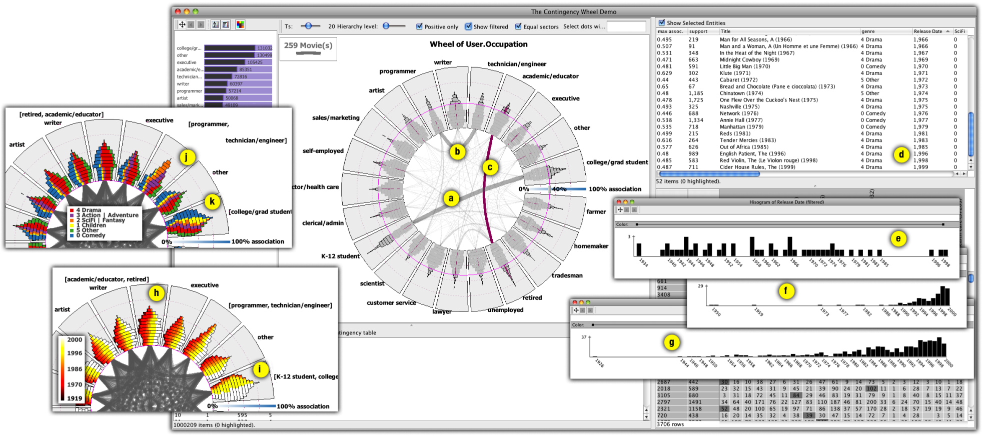


FIGURE 5.7 – **Visual exploration of movies associated to user occupations:** (a-c) major overlaps between user groups, (d) details of selected items in (c), (e-g) histograms of movie release date for different subsets of movies, (h, i) wheel view colored by movie release dates to reveal its relation with different user groups, (j, k) wheel view colored by movie genre to reveal dominant genres in the movie preferences of different user groups.

- If no modifier is defined or if brushing is performed using an external query, the active selection is replaced by the new one.

5.4 USE CASE

To demonstrate the applicability of our approach, we present a use case along the fictitious character Jane, who is an analyst at a large movie rental service. The use case is based on 10 analysis sessions conducted over the course of a week and added up to a total of 8 hours time. For the analysis, the MovieLens data set [GroupLens, 2012] has been used as introduced in Sect. 1. Jane's goal is to get insights into the massive amount of data they have collected about their customers who rented, watched, and rated movies using their service. Based on the insights gained from this analysis she plans to make decisions and take actions related to the ongoing marketing strategies and recommender algorithms they have in place. Jane uses mainly two different tables for her analysis: first, *occupations* (movies × user occupations) and second, *genres* (users × movie genres). By exploring the interactive wheels and the associated views and diagrams, Jane gains a number of insights, some of which were expected but also some surprising ones. Before Jane starts the analysis, she asks herself about the semantics of the data – what do associations between the entities user and movie actually mean? A user and a movie are associated if a user has entered a rating for a movie in the system which in turn implies that he or she has watched the movie. However, an association does not express how much they liked a movie.

5.4.1 *Categories & Characteristics*

As a first step, Jane aims for getting an overview of the categories to get a feeling for the data and overall (dis)similarities, to explore characteristics of single categories as well as to possibly simplify the data by merging categories.

User occupations Jane starts her analysis by creating a wheel based on a contingency table that displays the different occupations (i.e., jobs) of the users as sectors and the related movies as histogram bars within these sectors (Fig. 5.7). After opening the initial wheel, she adjusts the association slider to >40% in order to focus on higher associations and similarities between sectors. By studying the thickness of the connections inside the wheel she notices that there is a lot of overlap (same movies rated) between "K-12 student" and "college/grad student " (Fig. 5.7a) as well as between "programmer" and "technician/engineer" (Fig. 5.7b) which seems plausible to her. However, a more surprising aspect is that there is also a higher degree of overlap between "academic/educator" and "retired" (Fig. 5.7c). To investigate this connection in more detail, Jane selects the arc (Fig. 5.7c) and takes a look at the selected entities in the list view in the upper right of the UI (Fig. 5.7d). She sorts the movies by release date by clicking on the respective table header and discovers that they seem to be mostly older movies. Only three out of 52 movies are from the 90's, the rest are older movies. Based on the mentioned similarities, she merges "K-12 student" with "college/grad student", "programmer" with "technician/engineer", and "academic/educator" with "retired" by right-clicking on the corresponding arcs in order to simplify the further analysis.

Then, Jane continues her exploration of release dates of movies highly associated to the group [academic/educator, retired]. For this, she brings up a histogram using the context menu of the release-date column-header (Fig. 5.7e). This provides details about the distribution of movies over time. For comparison, she also brings up release-date histograms for [college/grad student, K-12 student] (Fig. 5.7f), as well as for all movies that were rated (Fig. 5.7g). This confirms that the distribution concerning [academic/educator, retired] is quite different. Moreover, Jane finds out that there seems to be a peak of watched and rated movies in the mid 80s followed by a valley at the beginning of the 90s and another peak at the end of the 90s. To get a further overview of release dates of categories, she colors the wheel by release date which also clearly shows that [academic/educator, retired] more often watch older movies than others (larger portion of dark parts than average, Fig. 5.7h). [K-12 student, college/grad student] watch more often newer movies than others (larger portion of bright parts than average, Fig. 5.7i).

Jane concludes her exploration by coloring the occupation wheel by genre. This reveals that [programmer, technician/engineer] contain a much larger portion of highly associated "SciFi/Fantasy" movies (orange, Fig. 5.7j) and [college/grad student, K-12 student] have a much larger portion of highly associated "Children" and "Comedy" movies (blue and yellow, Fig. 5.7k) than on average.

Movie genres Jane switches to the genre wheel and finds high overlaps of "Musical" and "Children", "Action" and "Adventure", and "War" and "Western" which seem to be reasonable to her. More surprisingly, she finds no particularly high overlap between [War, Western] and "Crime" which she would have suspected. Besides, she observes that "Horror" seems inversely related to many other genres. Based on her observations she merges genres into [Children, Musical, and Fantasy], [Noir, Mystery, Thriller], [Adventure, Action, SciFi], and [War, Western] (Fig. 5.8a-b). Jane colors the wheel by age using a diverging color scheme (Fig. 5.8a). Looking at this, she finds it surprising that the age distribution of users watching [Musical, Children, Fantasy] is not very different from others. Overall, age-group distributions seem to be quite similar in all genres.

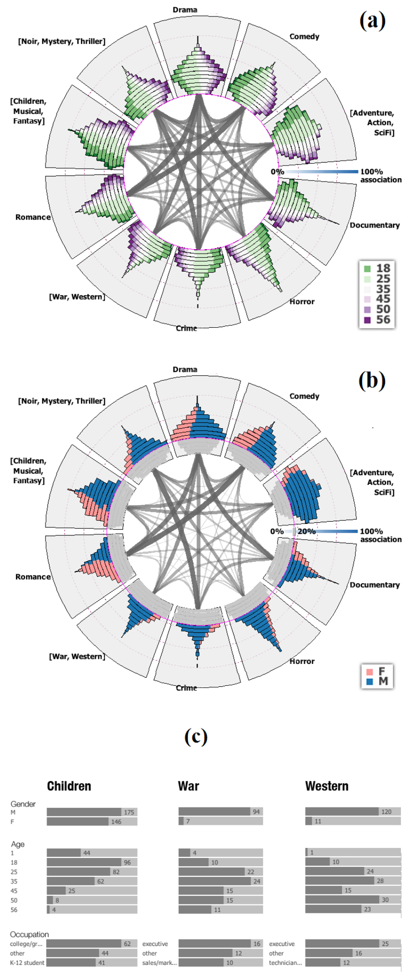


Figure 5.8 – **Associations between users and movie genres** (a) colored by age, (b) colored by gender. (c) Details about selected genres.

Further, Jane wants to inspect possible gender differences and turns on coloring by gender in the genre wheel (Fig. 5.8b). As a general observation, she recognizes that there are a lot more men than women rating movies. As anticipated, Jane finds the genre "Romance" as an exception where the most highly associated users are female. Surprisingly "Horror" does not show less women than other genres such as "Documentary", [Adventure, Action, SciFi], [Noir, Mystery, Thriller], or "Crime". After that, Jane takes a closer look on different genres using histograms of gender, age, and occupations (Fig. 5.8c). For "Children" movies she notices that there is an almost equal distribution between male and female viewers, and that most viewers are in the age group of 18–24 years. Particularly, the last fact is somewhat surprising to Jane, since she thought that the majority of users watching "Children" films would be younger. Having a look at "War" movies she spots that there are by far more men present which are often executives and in the age group of 35–44 years. "Western" movies show a quite similar picture, except that even older age groups watch and rate these movies.

5.4.2 *Single Movies*

After her top-down exploration of occupations and genre categories, Jane has a couple of movies in mind she wants to inspect further for potentially interesting findings in a bottom-up manner.

Mainstream erotic films She takes a look at the two movies Basic Instinct (1992) and Nine 1/2 Weeks (1986) and compares the star plots in the wheel views. Interestingly, both movies are highly associated to "technician/engineer" but negatively associated to "programmer" (Fig. 5.9a-d) which are otherwise quite similar as she had found out earlier.

The Godfather trilogy Next, Jane remembers that The Godfather movies (1972, 1974, 1990) were quite big hits at her movie rental service in the last years. She uses the search box (Fig. 5.10a) to find them. The movies matching the query are shown in the detail list in the upper right of the UI. She selects the first movie of the trilogy in the list (Fig. 5.10b) which brings up a star plot in the center of the wheel view showing individual associations for the different occupations. She can see that the movie has the highest associations with the occupations "executive" and "lawyer" (Fig. 5.10c,d). When she selects the second movie, the picture is quite similar, however, the third movie is somewhat different. Jane sees that it is highly associated to "executive" again but that it is negatively associated to "lawyer" and highly associated to "sales/marketing". Further, Jane would like to inspect how the three movies were rated among executives. For this, she double clicks on the "executive" column in the table (Fig. 5.10e) which brings up the ratings in the list view on the lower left of the UI (Fig. 5.10f). Via a context menu, she displays the rating histograms (Fig. 5.10g-i) and spots that they are quite positive and similar for the first two but much lower for the third movie.

5.4.3 *Hypotheses and Specific Questions*

During the visual exploration, Jane generated some hypotheses and specific questions she is trying to answer subsequently.

Association vs. rating behavior One hypothesis Jane had in mind is to check whether it is true that very high associations of movies correspond to more positive ratings. Using the occupation wheel, she is probing rating histograms of highly associated movies (>75%) with "college/grad student", such as Transformers (1986, good ratings) and Teenage Mutant Turtles II (1991, bad ratings). As a result, she finds evidence that her hypothesis does not hold.

Rating Another question Jane wants to inspect is whether there are differences in the general rating behavior of different user occupations, i.e., are particular groups more or less critical than others in general? By using the occupation wheel and comparing the grading histograms of selected sectors, Jane observes that the rating behavior is strikingly similar among groups. She can only spot subtle differences such as that unemployed persons tend to give lower ratings whereas retired persons do not tend to give many low ratings.

5.4.4 *Decisions and Actions Planned*

Visually exploring the vast data collection of her movie rental service helped Jane to better understand her customers and unearth commonalities as well as differences between groups of users and movies. Based on the gained insights, decisions are taken and actions are planned that are intended to make her business more successful: The merging of some user categories and movie genres can simplify the internal recommender engine. New SciFi &

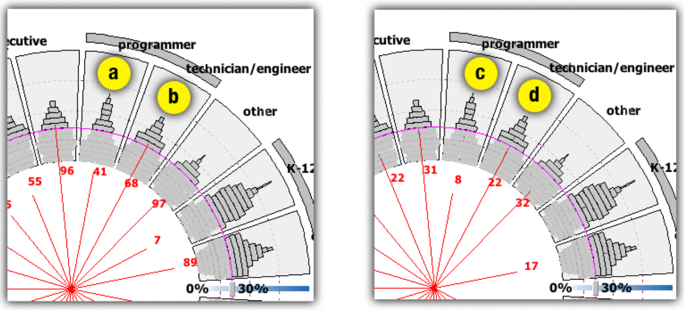


FIGURE 5.9 – **Associations of mainstream erotic films to "programmer" (a, c) and "technician/engineer" (b, d)** – left: Basic Instinct (1992), right: Nine 1/2 Weeks (1986)..

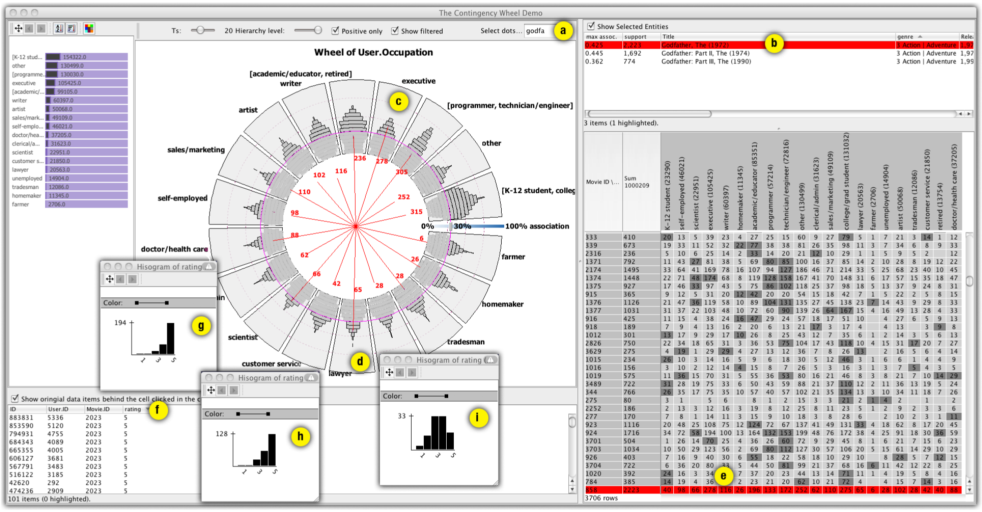


FIGURE 5.10 – **Visual exploration of the Godfather trilogy:** (a) search box, (b) search result, (c, d) star plot of associations of selected item to user occupations, (e) row of selected movie in contingency table, (f) raw data of user ratings, (g-i) rating histograms for the three movies of the trilogy.

Fantasy releases will be presented particularly to programmers and technicians/engineers. As there were more movies watched and rated from the mid 1980s, there will be a campaign highlighting some of these. Suggestions concerning Children movies will no longer be focused on younger customers but concentrate on the age group of 18–24 years. War and Western movies will be recommended more intensively to male executives older than 35 years. Finally, Horror movies will be suggested to users who already watched those without restricting suggestions to men.

5.4.5 *Improvements of Contingency Wheel++*

Jane benefited from the improvements in the new design and gained insights that would not have been possible using the original Contingency Wheel. Due to the fact that dots have been replaced by histograms, she was able to represent all movies without filtering steps which would have been necessary to avoid overlaps. The distribution of an attribute of the movies (e.g., release date) can now be inspected using colored histograms which allowed for complex insights involving multiple data attributes. Because of the multi-level overview+detail exploration environment, Jane had easy on-demand access to all available data, such as movie details, contingency table, and raw data. This allowed for drilling down to clarify and check findings from an aggregate level. Further, she was able to create bar charts and histograms of selected elements from different attributes, such as movie release dates or ratings, and compare the results with the global distributions. On top of that, the ability to merge sectors enabled the detection of patterns between groups of sectors that could not be detected when looking at single columns.

5.5 STATE OF THE ART

Heat maps can be used as a generic method for visualizing large matrix data as a colored image [Bertin, 1983]. While they can provide an overview of the data distribution, they are limited in terms of exploring associations. Methods dedicated for visualizing contingency tables are usually designed to handle a small number of categories. Based on what the visual representations depict, they can be classified into three types:

Frequency representations These methods map the table frequencies f_{ij} to visual elements of proportional size. Mosaic Displays [Hartigan and Kleiner, 1981] and their variations use tiles to represent the frequencies (similar to Treemaps [Shneiderman, 1992]). Parallel Sets [Bendix et al., 2005] and their variations, such as Circos [Krzywinski et al., 2009], represent frequencies as stripes or ribbons between visual elements that depict the categories. These approaches offer an intuitive visual representation that can be divided further to accommodate additional dimensions. However, they can handle only a relatively small number of categories (≤ 30 for Parallel Sets). With larger tables, the clutter increases in Parallel Sets, and the increased skewness and number of zeros in the table values make it difficult to identify and compare the tiles in Mosaic Displays [Unwin et al., 2006].

Deviation representations Association Plots [Meyer et al., 2003] use bar charts to show the deviations between the actual frequencies f_{ij} and the expected frequencies $\hat{e}_{ij}$ (Eq. 5.1). Sieve Diagrams [Friendly, 1992] plot f_{ij} as sieves in Mosaic Displays of $\hat{e}_{ij}$ to show how both deviate from each other. Sieves with smaller holes represent higher associations. A recent approach was proposed for exploring proportions in multivariate categorical data [Piringer and Buchetics, 2011]. It adopts the layout of Parallel Sets, but depicts the proportionality of relationships between the categories instead of f_{ij}.

Intermediate representations Correspondence Analysis [Benzécri, 1990] (CA) projects the categories to points in a 2D space, spanned by the two most contributing factors of the χ^2 statistic, in a way similar to Principal Component Analysis [Jolliffe, 2002]. A higher association between categories of the same class positions their points closer together, in a way similar to multidimensional scaling [Kruskal and Wish, 1978]. The approach can also accommodate additional categorical dimensions [Greenacre and Blasius, 2006].

However, with a growing number of categories, the plot becomes more difficult to read. It lacks an intuitive structure as its axes bear no interpretable semantics. Johansson et al. [2008] and Rosario et al. [2004] proposed methods for quantifying categorical data based on CA. The quantified data can then be visualized using scatter plots or parallel coordinates. However, the latent numerical variables used for the quantification are not always easy to interpret.

The dot-based Contingency Wheel uses deviation representations for the cells as dots along the radial dimension. Like many approaches for dealing with large data [Dix and Ellis, 2002; Rafiei and Curial, 2005; Stone et al., 1994] it uses data reduction to handle tables having a large number of rows. Also, it employs alpha blending to reveal overlapping, as done by other approaches for dealing with similar issues [Fekete and Plaisant, 2002; Johansson et al., 2006; Kosara et al., 2002]. In contrast, Contingency Wheel++ employs a frequency-based approach to abstract large data, as used by many other techniques [Hauser et al., 2002; Kosara et al., 2004; Rodrigues et al., 2003]. Also, it makes use of interactive visual analytics techniques to enable the exploration of individual data items. As the use case illustrates, asymmetrically-sized tables with a small number of columns (≤ 30) and thousands of rows can be handled efficiently by Contingency Wheel++ without filtering the data.

5.6 CONCLUSION

Contingency Wheel++ employs novel visual analytics methods that address the major shortcomings of the original dot-based wheel for visualizing and discovering patterns in large categorical data. It improves on the computational component by introducing an association measure based on Pearson's residuals to alleviate the bias in the association measure originally used. It eliminates the scalability and readability limitations caused by overlapping dots, by using a frequency-based abstraction that shows distributions rather than individual entities. Finally, it offers a multi-level overview+detail interface to explore individual entities that are aggregated in the visualization or in the table along with their attributes. The use case demonstrates how these methods can be used to find nontrivial patterns in large categorical data, and how further attributes can be analyzed in separate views or by coloring the histograms in the wheel visualization.

Future work aims to conduct comparative user studies to assess the effectiveness and efficiency of Contingency Wheel++, and to apply it to different real-world domains. Also, we are exploring further measures of associations and column similarities. Finally, we are investigating the applicability of our approach to other problems having similar data structures, such as point-set memberships or the class probabilities computed by a fuzzy classifier for a large number of samples.

ACKNOWLEDGEMENTS

We thank Peter Filzmoser for statistical help, Margit Pohl, Simone Kriglstein, and Florian Scholz for insightful ideas and evaluation, Georg Beischlager and Josef Tiefenbacher for implementation help, and Hannes Obweger and Martin Suntinger from UC4 Software for reviews. This work was supported by the Austrian Federal Ministry of Economy, Family and Youth via CVAST, a Laura Bassi Centre of Excellence (No. 822746), and by the Austrian Science Fund (FWF) through the ViMaL project (No. P21695).

REFERENCES

Alsallakh, B., Gröller, E., Miksch, S., and Suntinger, M. (2011). Contingency Wheel: Visual Analysis of Large Contingency Tables. In *EuroVA 2011: International Workshop on Visual Analytics*, pages 53–56, Bergen, Norway. Eurographics Association.

Bendix, F., Kosara, R., and Hauser, H. (2005). Parallel sets: visual analysis of categorical data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 133–140.

Benzécri, J. P. (1990). *Correspondence Analysis Handbook*. Marcel Dekker, New York.

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., and Wiswedel, B. (2008). KNIME: The Konstanz information miner. In *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 319–326. Springer Berlin Heidelberg.

Bertin, J. (1983). *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press, Madison, Wisconsin, USA.

Dix, A. and Ellis, G. (2002). By chance: enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pages 167–176, New York, NY, USA. ACM.

Fekete, J.-D. and Plaisant, C. (2002). Interactive information visualization of a million items. In *IEEE Symposium on Information Visualization, 2002.*, pages 117–124.

Friendly, M. (1992). Graphical methods for categorical data. In *SAS User Group International Conference Proceeding*, volume 17, pages 190–200.

Greenacre, M. J. and Blasius, J. (2006). *Multiple correspondence analysis and related methods*. Chapman & Hall/CRC.

GroupLens (2012). MovieLens data sets. http://www.grouplens.org/node/73. Accessed: August 2012.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, Inc., New York, NY, USA.

Harrower, M. and Brewer, C. (2003). ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, pages 27–37.

Hartigan, J. A. and Kleiner, B. (1981). Mosaics for contingency tables. In *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, pages 268–273. Springer-Verlag.

Hauser, H., Ledermann, F., and Doleisch, H. (2002). Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization*, pages 127 – 130.

Hochheiser, H. and Shneiderman, B. (2004). Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18.

Johansson, J., Ljung, P., Jern, M., and Cooper, M. (2006). Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization*, 5(2):125–136.

Johansson, S., Jern, M., and Johansson, J. (2008). Interactive quantification of categorical variables in mixed data sets. In *Proceedings of the 12th International Conference on Information Visualisation*, pages 3–10, Washington, DC, USA. IEEE Computer Society.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, second edition.

Keim, D., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H. (2008). Visual analytics: Scope and challenges. In *Visual Data Mining*, volume 4404 of *Lecture Notes in Computer Science*, pages 76–90. Springer Berlin / Heidelberg.

Kosara, R., Bendix, F., and Hauser, H. (2004). Timehistograms for large, time-dependent data. In Deussen, O., Hansen, C., Keim, D., and Saupe, D., editors, *Symposium on Visualization (VisSym)*, pages 45–54, 340. Eurographics Association.

Kosara, R., Miksch, S., and Hauser, H. (2002). Focus+context taken literally. *IEEE Computer Graphics and Applications*, 22:22–29.

Kriglstein, S., Scholz, F., Pohl, M., Alsallakh, B., and Miksch, S. (2012). Contingency wheel evaluation: Results from an interview study. Technical Report CVAST-2012-2, Vienna University of Technology, Vienna, Austria.

Kruskal, J. B. and Wish, M. (1978). Multidimensional scaling. *Methods*, 116(2):463–504.

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S. J., and Marra, M. A. (2009). Circos: an information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645.

Meyer, D., Zeileis, A., and Hornik, K. (2003). Visualizing independence using extended association plots. In Hornik, K., Leisch, F., and Zeileis, A., editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria*.

Piringer, H. and Buchetics, M. (2011). Exploring proportions: Comparative visualization of categorical data. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 295 –296.

R Development Core Team (2009). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rafiei, D. and Curial, S. (2005). Effectively visualizing large networks through sampling. *Visualization Conference, IEEE*, pages 375 – 382.

Rao, J. N. K. and Scott, A. J. (1981). The analysis of categorical data from complex sample surveys: Chi-squared tests for goodness of fit and independence in two-way tables. *The Journal of the American Statistical Association*, 76:221–230.

Robinson, J. O. (1998). *The Psychology of Visual Illusion*. Dover Publications, Inc.

Rodrigues, J.F., J., Traina, A., and Traina, C., J. (2003). Frequency plot and relevance plot to enhance visual data exploration. In *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)*, pages 117–124.

Rosario, G. E., Rundensteiner, E. A., Brown, D. C., Ward, M. O., and Huang, S. (2004). Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95.

Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika*, 66(3):605–610.

Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics (TOG)*, 11(1):92–99.

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, pages 336–343.

Simonoff, J. S. (2003). *Analyzing Categorical Data*. Springer-Verlag, New York, USA, 2nd edition.

Stone, M. C., Fishkin, K., and Bier, E. A. (1994). The movable filter as a user interface tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 306–312, New York, NY, USA. ACM.

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society.

Unwin, A., Theus, M., and Hofmann, H. (2006). *Graphics of Large Datasets: Visualizing a Million*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Wertheimer, M. (1938). Laws of organization in perceptual forms. In Ellis, W. D., editor, *A sourcebook of Gestalt psychology*, pages 71–88. Routledge and Kegan Paul.

Xiong, T., Wang, S., Mayers, A., and Monga, E. (2009). A new MCA-based divisive hierarchical algorithm for clustering categorical data. In *Proceedings of IEEE International Conference on Data Mining*, pages 1058–1063. IEEE Computer Society.

CHAPTER 6

EvalBench: A Software Library for Visualization Evaluation

Wolfgang Aigner, Stephan Hoffmann, and Alexander Rind

Abstract · *It is generally acknowledged in visualization research that it is necessary to evaluate visualization artifacts in order to provide empirical evidence on their effectiveness and efficiency as well as their usability and utility. However, the difficulties of conducting such evaluations still remain an issue. Apart from the required know-how to appropriately design and conduct user studies, the necessary implementation effort for evaluation features in visualization software is a considerable obstacle. To mitigate this, we present EvalBench, an easy-to-use, flexible, and reusable software library for visualization evaluation written in Java. We describe its design choices and basic abstractions of our conceptual architecture and demonstrate its applicability by a number of case studies. EvalBench reduces implementation effort for evaluation features and makes conducting user studies easier. It can be used and integrated with third-party visualization prototypes that need to be evaluated via loose coupling. EvalBench supports both, quantitative and qualitative evaluation methods such as controlled experiments, interaction logging, laboratory questionnaires, heuristic evaluations, and insight diaries.*

Keywords · Evaluation/methodology.

This article originally appeared as [Aigner et al., 2013]:

Aigner, W., Hoffmann, S., and Rind, A. (2013). EvalBench: A Software Library for Visualization Evaluation. *Computer Graphics Forum*, 32(3). forthcoming.

6.1 INTRODUCTION & MOTIVATION

As the field of visualization is maturing, evaluation has become an increasingly important part of research. The relevance of evaluation for visualization research is documented by its prominent role in virtually all currently published research agendas and future challenge discussions, such as [Thomas and Cook, 2005] and [Keim et al., 2010], the organization of workshops and conference sessions dedicated to visualization evaluation, such as BELIV (www.beliv.org), as well as special issues of scientific journals. It is agreed that evaluation is a key element in human-centered visualization design and that it is necessary to provide empirical evidence on the effectiveness and efficiency as well as usability and utility of new methods [Plaisant, 2004]. Yet, the difficulty of conducting these evaluations remains a common topic [Lam et al., 2012]. There is a need for a solid evaluation infrastructure to encourage visualization researchers to carry out evaluations of their methods and tools [Keim et al., 2010; Thomas and Cook, 2005]. Researchers need solutions how to integrate different methods for evaluation into their prototypes and how to collect and measure the data generated by the users participating in a study. To support them in their efforts, we set out to develop the open source software library *EvalBench* with the goals of being easy-to-setup and use, customizable, and as independent and loosely coupled to the visualization artifact to evaluate as possible. In addition, this paper's design section presents the architectural choices and basic abstractions of our library independent from specific programming languages.

Next, we give an overview of the challenges of visualization evaluation followed by a discussion of related work. In Sect. 6.4, we present the conceptual design of our library. Then, we describe its implementation in Sect. 6.5 and demonstrate its utility by case studies in Sect. 6.6. Finally, we provide a discussion of the benefits and limitations in Sect. 6.7 and conclude with pointers to future work in Sect. 6.8.

6.2 EVALUATION IN VISUALIZATION

Evaluating highly interactive visualization artifacts (techniques or tools) is a challenging and thorny task because visualization usually aims for supporting ill-defined problems and tasks based on large amounts of complex data [Elmqvist and Yi, 2012; Lam et al., 2012]. Particular challenges are [Plaisant, 2004]: the need to work with data over longer periods of time and from different perspectives; the exploratory nature of visual analysis where users may pose questions and get insights they did not know they will have prior to looking at the visualization; and that important discoveries occur rarely and maybe not at all. To support the multi-faceted challenge of visualization evaluation, a variety of different evaluation methods have been applied in different scenarios and at different stages of visualization research

[Carpendale, 2008; Lam et al., 2012; Lazar et al., 2010; Plaisant, 2004]: field observations, interviews, case studies, laboratory observations, controlled experiments, logging, heuristic evaluation (e.g., usability inspection), informal evaluation (expert reviews), usability tests, laboratory questionnaires, visualization quality assessments, and algorithmic performance analysis. Our goal is to support a wide range of common evaluation methods applied in visualization.

Broadly, evaluation methods can be divided into quantitative and qualitative methods, whereas the first group emphasizes measurable outcomes and the second one emphasizes interpretative analysis of collected material. In a quantitative study, a controlled setting is created and empirical evidence in the form of measurements (usually time and error) is collected under different conditions. The results are analyzed with statistical hypotheses tests. Controlled experiments are an important evaluation method because precise results can be collected. However, a lot of effort is required to perform such studies. Apart from the design and prototypical implementation of the studied visualization and interaction artifacts, a number of additional steps are needed [Forsell, 2010]: (1) preparation of a study design, (2) defining tasks and providing data, (3) participant recruitment and assignment to conditions, (4) conducting the study and collecting data, (5) analysis of results, and (6) reporting the findings. Qualitative evaluation methods often provide a different and complementing view on visualization artifacts. They can be conducted in more realistic settings and yield deeper understanding of human reasoning processes. Qualitative studies follow the same steps presented above, though some steps apply different methods (e.g. coding transcripts) and other steps might leave more freedom to participants (e.g., choosing tasks and data).

Each of these six steps is challenging in itself. While there are guidelines (e.g., [Carpendale, 2008; Lazar et al., 2010]) and supportive systems available for various steps (e.g., *SAS JMP* or *Touchstone* [Mackay et al., 2007] for study design, task taxonomies, dataset repositories, *Amazon Mechanical Turk* for participant recruitment [Heer and Bostock, 2010], or scripts for data preprocessing and statistical analysis), most of the steps mentioned above need to be highly customized for different evaluation scenarios and are hard to reuse (e.g., writing effective task lists, recruiting domain experts, coding qualitative results). However, the step of conducting the study and collecting data using evaluation features built into visualization software appears to have the inherent potential for reusable components. This involves guiding participants through an evaluation session, collecting results through evaluation features in the software (i.e., instrumentation), and providing means for linking transcripts of insight studies or usability inspections to the state of a visualization artifact and preceding interactions. There are a number of experimentation frameworks mainly from the HCI realm. However, those are generally more tightly coupled and rigid in terms of prescribing a certain structure that visualization artifacts need to follow. Thus, more realistic evaluation settings such as long-term and insight-based evaluations can hardly be achieved. To develop interactive visualizations, a number of libraries and toolkits exist such as prefuse [Heer et al., 2005], Obvious [Fekete et al., 2011], or D3 [Bostock et al., 2011] that provide reusable components (e.g., range slider) and generalizable concepts (e.g., view management for brushing and linking). But, to the best of our knowledge, there are no such libraries that provide components for *visualization evaluation* to be added to interactive visualization artifacts. To fill this gap, we present our library *EvalBench* after reviewing related work in more detail.

6.3 RELATED WORK

Above we presented the range of evaluation methods at the disposal of visualization researchers and a generic six-step evaluation process. In this work we focus on the step of conducting the study. This step demands rigorous and consistent execution of the study design. Thus, it is evident that many researchers automate parts of their studies with evaluation features. Next, we present some examples of evaluation features and evaluation systems that influenced our work.

Quantitative Evaluation. The *Hierarchical Visualisation Testing Environment (HVTE)* [Andrews and Kasanicka, 2007] launches predefined tasks and system configurations, and automatically records user answers and task completion times. However, it is tightly integrated with a visualization system for hierarchically structured information. *Touchstone* [Mackay et al., 2007] is a bundle of platforms to design, run, and analyze HCI experiments on pointing interaction techniques. The run platform launches the experiment as specified on the design platform, supporting different input devices, and collecting performance data and interaction logs. For this, all experiment components (e.g., visualization techniques) need to extend base classes of the framework. For graphical output, it includes a simple zoomable scene graph viewer. Furthermore, it is tailored for tasks that are completed by interaction gestures or item selection, but not for answering questions (e.g., input of a numeric value). Thus, it needs far-fetched adaption to evaluate anything other than pointing interaction techniques. The *Generalized Fitts' Law Model Builder* [Soukoreff and MacKenzie, 1995] is an earlier system specialized on pointing experiments. Likewise, behavioral experiment systems such as *OpenSesame* [Mathôt et al., 2012] or *Presentation* from Neurobehavioral Systems intend to produce visual stimuli through their particular framework. Heer and Bostock [2010] identified limitations of *Amazon Mechanical Turk* as a platform to conduct graphical perception experiments and they recommend to use it only for recruitment and participant management but launch a evaluation/ visualization system via Flash (e.g., [Heer and Bostock, 2010]), Java Web Start (e.g., [Jin and Szekely, 2010]), or JavaScript. *USEMATE* [Ahmad et al., 2010] is a system for administrating and conducting usability experiments. With it, the facilitator can fill out usability questionnaires and record execution times. There is, however, no synchronization with the studied artifact.

Qualitative Evaluation. Rester et al. [2007] developed an insight collection system for a study that compared a visualization prototype to machine learning and exploratory data analysis methods. Subjects of their study reported insight as free text with a confidence rating via a web interface that was independent from the visualization prototype. They were also asked to upload a representative screenshot.

Logging. Recording system states or user interactions is common practice in visualization research. These logs can either be used to investigate reasoning processes in insight studies (e.g., [Dou et al., 2009]) or as supplemental material in controlled experiments (e.g., [Jin and Szekely, 2010]). *Glassbox* [Cowley et al., 2006] records keyboard events, window events, file events, screenshots once per second, etc., as well as logging through an application programming interface (API). In *aspect-oriented programming* logging can be a separate aspect from the evaluated visualization artifact [Henry et al., 2008]. Through tight integration with a *P-Set Model* of the visualization artifact, it is possible to log the effects of interactions on the visualization artifact rather than logging the interactions themselves. *TRUE* [Kim et al., 2008] logs event sequences with contextual information in complex environments (e.g., video games). *Tome* [Gomez and Laidlaw, 2012] builds keystroke-level models from interaction logs, so that routine task performance of alternative user interfaces can be predicted. Moreover topically related are history tracking mechanisms such as in *VisTrails* [Callahan et al., 2006], a prototypical Tableau extension [Heer et al., 2008], or *Aruvi* [Shrinivasan and van Wijk, 2008].

Summary. On the one hand, evaluation features are often developed ad-hoc and tightly integrated with the studied visualization systems (e.g., HVTE). On the other hand, the evaluation systems we are aware of provide a framework and the studied artifacts have to be integrated within their architecture (e.g., Touchstone). While it is possible to adapt them for visualization prototypes, it enforces more structure on the evaluated artifact than our approach. This limits their applicability for complex systems especially if evaluation of an already existing system is intended. Furthermore, many systems support only one evaluation method. Considering that, a lack of an easy-to-use, flexible, and reusable software library providing evaluation features can be observed.

6.4 CONCEPTUAL DESIGN

This section describes the architecture and basic abstractions of the library independent of specific programming languages. Our intention is to facilitate the implementation of evaluation functionality to be combined with existing visualization artifacts. To provide a clear structure and allow for extensibility, we propose our conceptual design partially based on the software design patterns by Gamma et al. [1995]. The diagrams that are shown in the following use the extended object-modeling technique notation of [Gamma et al., 1995] to depict classes and their relations (e.g., a circle at the end of an arrow indicates a 1-to-n relationship and a diamond at base of a relation denotes aggregation). First, the overall structure of the library is presented.

6.4.1 *Evaluation Manager & Delegate*

Fig. 6.1 illustrates the abstract high-level structure for the evaluation library. The EvaluationManager is the central component and responsible for managing the evaluation process. Moreover, the manager functions as a FACADE object providing a simple interface to the underlying subcomponents in order to make the library easier accessible and reduce dependencies to the subcomponents. Because only one instance of the manager is needed, it can be implemented following the SINGLETON pattern to make it globally available and enable easy access. The rationale behind this is that the manager is often needed by subcomponents of the library but also by the evaluated visualization artifact, e.g., to setup the study design or to log user interactions.

The EvaluationDelegate is the interface to the visualization artifact. It is re-

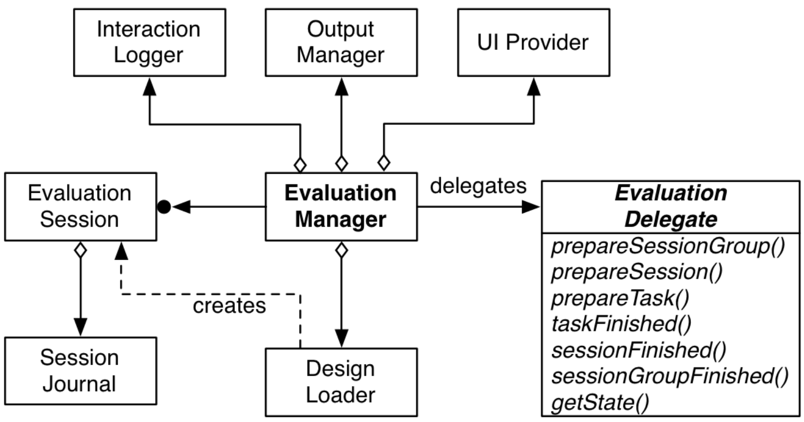


FIGURE 6.1 – **High-level structure of evaluation library.** The EvaluationManager controls the evaluation. The abstract EvaluationDelegate needs to be implemented to connect the library to the visualization artifact.

sponsible for controlling the visualization and also for providing information about the current state of the visualization artifact. The EvaluationDelegate can be part of a visualization artifact's source code and control the visualization directly or it might have indirect access to the investigated visualization through an interface (e.g., an API). The DELEGATION pattern was selected in order to decouple the evaluation process from the visualization and is intended to facilitate the integration of the evaluation functionality into existing software. The EvaluationManager forwards important events in the evaluation procedure to the EvaluationDelegate to enable adaption of the visualization according to the current evaluation state. Such adaptions could be to change the dataset that is currently displayed or to use a different visual encoding. The EvaluationDelegate is also in charge of presenting a possibly necessary evaluation user interface (UI) to the test persons (e.g., instructions and input textbox). Since the EvaluationManager is agnostic to the visualization artifact to be evaluated, the EvaluationDelegate implementation acts as link. *Design-time attributes* (such as the dataset to be visualized, session structure, task definitions, and instructions) can be read from an external data source by the DesignLoader to create groups of evaluation sessions that have to be accomplished by a study participant. The SessionJournal and InteractionLogger record the *run-time attributes* that arise during the execution of an evaluation session (such as task completion times, error rates, insights, interactions), which are the subject of analysis after the evaluation process. The OutputManager is responsible for configuring the output destination of the run-time attributes.

6.4.2 *Quantitative Evaluation*

In a controlled experiment, participants usually have to perform a predefined set of tasks in one or more sessions reflecting different treatments formed by varying experiment factors [Lazar et al., 2010]. We developed a data structure that intends to reflect this common structure (see Fig. 6.2). A task usually contains one question; the response to such a question could be to identify a certain data value from a visual element in the visualized dataset or select a certain area in the visualization. But in some cases it might also be necessary to ask several questions that pertain to one task (e.g., *Which element is larger? By how much?*). Therefore, a Task contains at least one and possibly multiple Questions. A quantitative evaluation can also include several sessions (or blocks) that are bundled in a session group. A practical example would be if an experimenter wants the test persons to perform a training session before advancing to an actual experiment session. Each session group may contain a list of sessions or even session groups again to allow multiple levels of aggregation. This is necessary if a divergent execution order of sessions (e.g., Latin square) is intended but sub-sessions are required to stay in sequential order (e.g., training and actual session). In order to realize this behavior, the

Composite pattern enables the composition of Sessions and SessionGroups into a multilevel tree structure, where the nodes can be treated uniformly.

A task contains several predefined design-time attributes like *category* to classify tasks or *instructions* that are displayed to the test persons. A task might also contain a *repeat* attribute that allows the test person to receive feedback on correctness and duration, and decide if they want to repeat the task. This is particularly useful for training tasks. To enable the assignment of various experiment factors to tasks and sessions (e.g., dataset or visualization type), the data structure of the *configuration* attribute needs be capable of storing multiple key/value pairs (e.g., a map or dictionary). The run-time attribute *completion time* is stored in the data structure of the task rather than for each question individually. This decision was taken because we assume that if multiple questions are assigned to a task, each of them can be answered based on a single discovery or insight gained from a visualization artifact. The abstract class Question provides the base class for each ConcreteQuestion and defines the method that is invoked to assess the correctness. This aims to facilitate flexible extension of the library with new question types. Because each question type needs to store and process different data types and may have different correctness criteria, each ConcreteQuestion needs to host a data structure that is capable of storing the correct answer (e.g., a numerical value), possible restrictions for the answer (e.g., an allowed value range), and the provided answer (e.g., numerical value given by the test person).

Execution of an evaluation. To start an evaluation, the EvaluationManager has to be initialized with a study design. The design can be defined manually

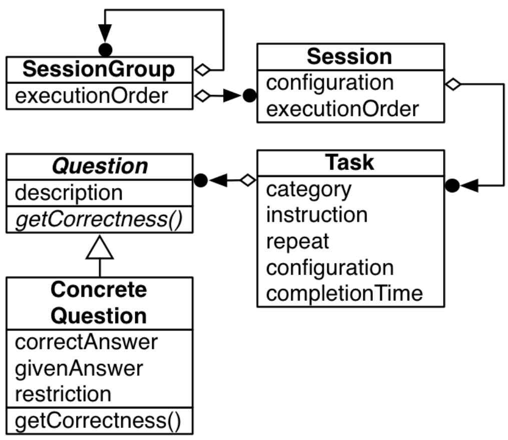


Figure 6.2 – **Data model for quantitative experiments.** ConcreteQuestions need to host and process different data types according to the question type (e.g., multiple choice).

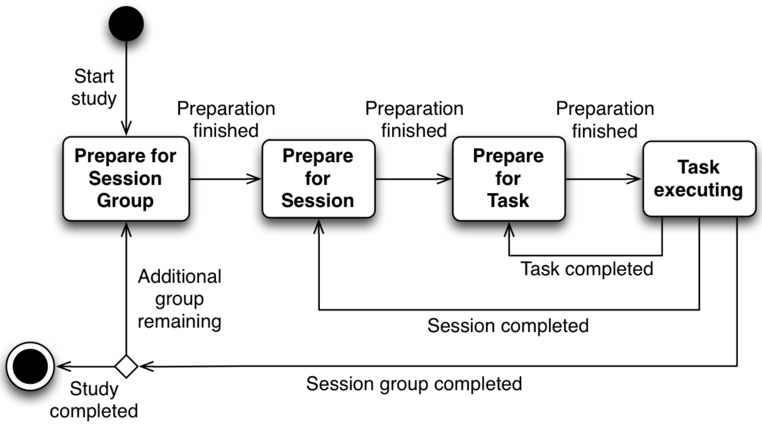


FIGURE 6.3 – **Evaluation process.** Simplified state machine that is run by the EvaluationManager during an evaluation.

in the source code or by passing an externally stored study design to a DesignLoader implementation. Subsequently, the EvaluationManager runs a state machine defined by the study design (see Fig. 6.3). The events are triggered by the EvaluationDelegate and the test persons. The EvaluationDelegate is responsible for preparing the visualization for each experiment state and the test persons interactively trigger the completion of tasks.

User Interface Provider. In order to accomplish a task during an experiment, test persons need to be provided with a UI showing instructions and giving them a possibility to specify the answers to the questions that are

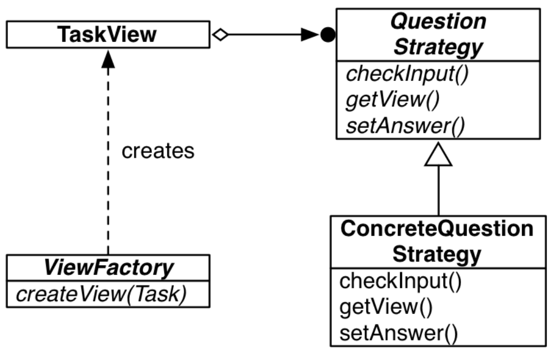


FIGURE 6.4 – **ViewFactory.** Creates a view for a given task, whereas the QuestionStrategies provide a subview, set the answers, and check for sufficient input for each question.

assigned to a task. The ViewFactory (see Fig. 6.4) is responsible for creating an appropriate UI for a given task; the created view should provide interactive means to answer its assigned questions. The factory creates a TaskView using a strategy for each question assigned to the given task. A strategy is responsible to check for sufficient input, store the given answer, and provide the necessary UI. The Strategy pattern ensures flexibility for providing individual UIs for existing or new question types by extending the factory with new strategies. If the task requires some interaction with the visualized data (e.g., selecting an item in the visualization), one can implement a special question strategy that integrates with the UI of the evaluated visualization artifact (e.g., an Observer).

Session Journal & Interaction Logging. It is essential to record a protocol of the experiment sessions for further analysis. A common practice in evaluations of user performance is to record task accuracy and task completion time. For this reason, each EvaluationSession holds an instance of SessionJournal (see Fig. 6.1) which is in charge of saving all relevant data for each task after its completion. In addition to the journal, a separate interaction log is saved for each session. The visualization artifact notifies the EvaluationManager of interaction events that need to be logged. These will be forwarded to an InteractionLogger (see Fig. 6.1). The creation and naming of the files and directories for sessions and session groups is performed by the OutputManager to ensure that session journals and interaction logs are saved in the same directories and can be configured in one place. The OutputManager may also be used to save screenshots of the visualization tool, record audio and video logs, or save the current state of a visualization artifact. To make the evaluation system independent of the implementation of the SessionJournal (e.g., CSV or XML output) an Abstract Factory pattern is applied for creating a concrete journal (see Fig. 6.5).

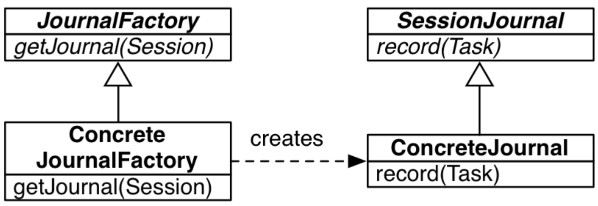


Figure 6.5 – **JournalFactory.** Enables the configuration of various journal implementations.

6.4.3 *Qualitative Evaluation*

The presented design of the library is also capable to support the collection of qualitative data as with insight diaries, questionnaires, heuristic evaluations, and interaction logs.

Insight Diaries. Insight-based methodologies [North, 2006] evaluate visualization systems in real-world data analysis scenarios and are usually less guided than quantitative methodologies. The test persons are requested to keep a diary of the insights gained while using one or more visualization artifacts. These diary entries usually consist of text describing the found insight in narrative form but can also be structured in several ways (for example by adding Likert scales to classify the level of relevance or certainty). In our concept, insight diary entries can be modeled similar to tasks in quantitative evaluations. To provide the UI for an insight diary entry, special Question types can be created (e.g., Likert scale question or free-text question) that are assembled using a tailored ViewFactory. To represent predefined templates of diary entries, different tasks representing such entries can be collected in an evaluation session. Instead of letting the actual evaluation session automatically choose the next task, it can be left to the test persons to choose the task (i.e., insight type) they want to record using the EvaluationDelegate. The Session class thus needs an additional *repeat* property that is recognized by the EvaluationManager to allow the repetition of the session that consists of diary templates. A session execution order of *free choice* indicates that the upcoming task (i.e., diary entry) is required to be selected by the test person. The EvaluationManager needs to be configured to run an extended version of the state machine (see Fig. 6.3) with a *Pause* state between the *Task executing* and *Prepare for Task* states, and an event that triggers the termination of a repeating session.

Heuristic Evaluations. Heuristic evaluations and inspections aim at finding, for example, usability problems and are performed by evaluators that examine the visualization tool [Nielsen, 1994]. In terms of data collection and execution of the study, heuristic evaluations are similar to insight-based evaluations and differ essentially only in their intention. Instead of collecting insight diary entries, observed problems with the investigated tool are collected, often using a set of heuristics to focus on important aspects of the tool.

Questionnaires. In short, questionnaires are a well-defined and well-written set of questions to which an individual is asked to respond [Lazar et al., 2010]. Questionnaires are frequently used in addition to other evaluation methods in order to collect demographical information, feedback, or opinions from the test persons. Questionnaires can be modeled as a set of Tasks that contain a list of Questions. These tasks need to be aggregated in Sessions, similar to quantitative evaluations.

Interaction Logging. Logging screenshots and/or application states is important to make sense of the recorded insights or usability issues by relating them to what actually happened on the screen. Moreover, interaction logs can be used to find patterns of interactivity that users produce to draw conclusions about the exploration process [Pohl et al., 2012]. The evaluation progress (i.e., start and completion of tasks and sessions) is automatically added to the log and the state of the investigated visualization artifact is added by querying the EvaluationDelegate (see Fig. 6.1). The conceptual design of our library allows for logging of interactions via the globally available EvaluationManager that forwards the actual logging to an InteractionLogger instance. As this can introduce a potentially unwanted dependency between the visualization artifact and the evaluation library, also a logger framework of choice can be used directly. In both cases, the output location can be configured for the various stages of an evaluation (see Fig. 6.3) by the EvaluationDelegate using the OutputManager.

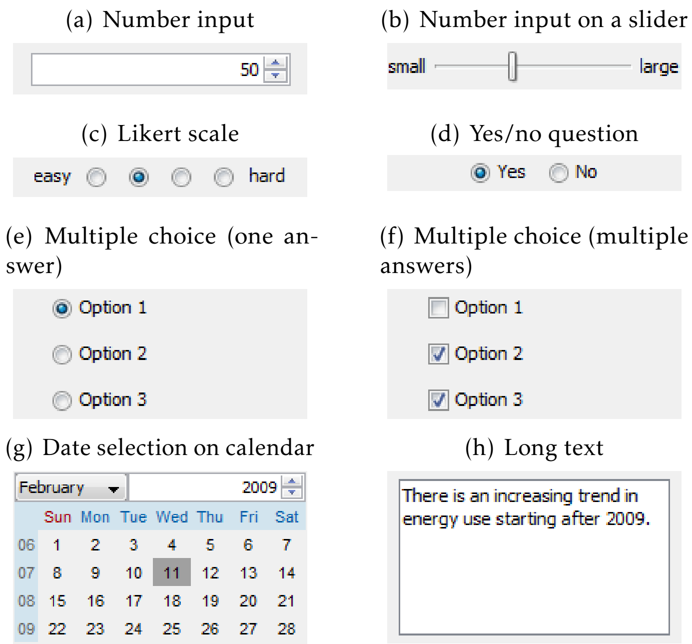
Figure 6.6 – **Implementation of EvaluationDelegate.** Fragmentary example of an evaluation system.

Figure 6.7 – **Task definition in XML.** Illustrative example with one task comprised of a multiple-choice question and a yes/no question.

6.5 EVALBENCH LIBRARY

Based on the conceptual design presented above, we developed EvalBench as a software library to support evaluation studies in visualization. EvalBench is written in Java 1.6 and uses the libraries Apache Commons Lang 3.0, Apache log4j 1.2, and JCalendar 1.4 by Kai Toedter.

For an evaluation study, the studied visualization artifact needs to implement the EvaluationDelegate interface (see Fig. 6.6) and, thus, is notified of the progress in the evaluation process. The setup of session groups and assignment of subjects needs to be specified in the source code. EvalBench can load the task list for each session from an XML file (see Fig. 6.7). It provides subclasses of Question and related UIs for multiple choice questions, free text, numbers as text or on a slider, date selection from a calendar, agreement on a Likert scale, and yes/no questions (see Fig. 6.8). The timing of task executions and answers to questions are collected by a SessionJournal, which includes also the participant id and design-time attributes (e.g., task description) in



(a) Number input (b) Number input on a slider (c) Likert scale (d) Yes/no question (e) Multiple choice (one answer) (f) Multiple choice (multiple answers) (g) Date selection on calendar (h) Long text

FIGURE 6.8 – **Gallery of question-answering controls provided by EvalBench.**

order to be self-contained. The journals can be saved either in CSV format for import in statistics software or in XML format. Interaction logging is handled by the library log4j. EvalBench ensures that log4j creates one log file per session and that they are stored at the same location as the evaluation journal. It also provides convenience dialogues to display intermediate messages and hide the visualization artifact.

Overall, the EvalBench library is comprised of about 3,000 lines of code (LoC). In order to allow widespread use by the research community as well as practitioners, we make it available as open-source software under a BSD 2-Clause license. The project is available at www.evalbench.org and GitHub. We invite the community to share extensions via this platform.

6.6 CASE STUDIES

We developed EvalBench in the course of evaluation studies for our visualization research projects (such as [Aigner et al., 2011, 2012; Neubauer, 2012; Pohl et al., 2012]). Three of these evaluations will serve as case studies to show how EvalBench can be applied.

Comparative Evaluation with Facilitator. We conducted a controlled experiment to compare task completion times and error rates of two visualiza-

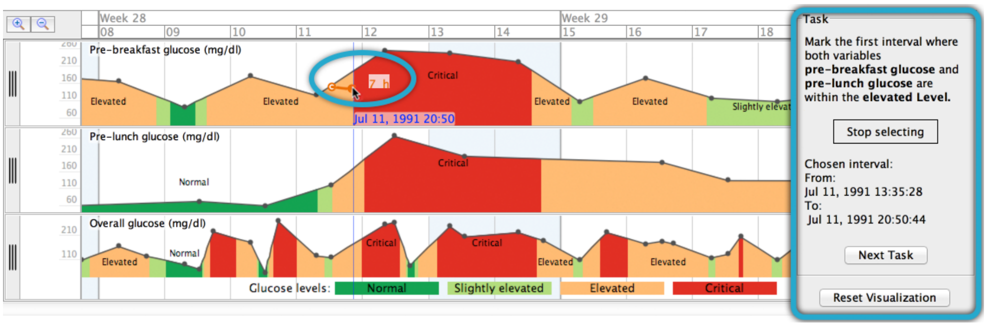


FIGURE 6.9 – **User interface for comparative evaluation in [Aigner et al., 2012].** The visualization technique takes the larger part of the screen with the task view on the right (blue rectangle). The screenshot demonstrates an interval selection question, which the participants answer by brushing a time interval in the visualization (blue oval).

tion techniques that combine quantitative time-series data with qualitative abstractions [Aigner et al., 2012]. We recruited 20 participants and tested both techniques with each participant. Since the study was designed within-subjects, we provided two similar datasets and two task lists for each dataset (three training tasks and 24 evaluation tasks). The experiment was conducted with a facilitator who manually set the sequence of techniques and dataset using a Latin square. Within each evaluation session the tasks were presented in random order. The session journals in CSV format were analyzed with R using boxplots and statistical hypothesis tests. We also logged the available interaction techniques to compare the performed interactions.

We implemented both visualization techniques and the evaluation features in the same Java application. The main reason to do this was to avoid bias from different user interfaces and interaction modes, and it also made it easier to execute the study. Fig. 6.9 shows the complete user interface with the evaluation features on the right and one of the tested visualization techniques occupying the rest of the screen. The screenshot also demonstrates the flexibility of EvalBench. It can be extended with individual UIs for questions and make direct use of interactions with the visualization artifact for data input. Here, an interval selection question is answered by brushing a time interval directly in the visualization. The implementation effort for the evaluation features was about 750 LoC and focused on putting together sessions, task lists, data files, and the user interface for the facilitator to start a session. Additional 200 LoC were needed for the interval selection question strategy. In comparison, a similar study [Aigner et al., 2011] we conducted without using EvalBench required about twice as many LoC.

Comparative Evaluation with Java Web Start. In this study we experimentally compared three visualization techniques to explore bivariate patterns across time [Neubauer, 2012]. The study design was similar to the one described above but featured some notable differences: First, we built the evaluation system self-contained and self-explanatory enough, so that the participants could work on their own computers without a facilitator. This allowed us to test with a larger number of participants and keep administrative overhead low. For this, we made the evaluation system runnable via Java Web Start. It began with a dialog to ask for the participant id and then executed all six sessions after each other. Before each evaluation session (22 tasks) there was a training session (five tasks) and the three compared techniques were tested in random order. Between sessions, a dialog asked the participants to take a break if needed. At the end, the participants were presented a short questionnaire to determine subjective feedback on the understandability and usability of the visualization techniques. The session journal, interaction logs, and questionnaire results were saved to a directory on the participants' computers and they were asked to upload the complete directory. Second, the evaluated visualization system offered a wider range of interaction techniques. We collected interaction logs as an additional data source for our study, and we also checked whether participants had manipulated the log files. Furthermore, each task needed to start in a well-defined state of the visualization techniques. Thus, the EvaluationDelegate was used to load the visualization states (stored in an external file) and apply them using an API.

Even though the evaluation was completely automated, the implementation effort of about 800 LoC was comparable to the previously presented case study. Here, additional 750 LoC were required for the questionnaire, but this method is now supported natively by EvalBench.

Interaction Logging during a Usability Study. In this qualitative study of a medical visualization system [Pohl et al., 2011] the participants were guided by three high-level tasks but were not constrained in their usage of the visualization system. On advice of the social scientist in the team, the tasks and subsequent interviews were provided by a human facilitator in order to use the limited time of domain experts more effectively. Thus, we did not include any visible evaluation features in the system and only performed interaction logging with EvalBench. To analyze the log files we coded the interactions by the user intent categorization of Yi et al. [2007] and investigated interaction sequences and transition probabilities [Pohl et al., 2012]. The implementation effort amounted to about 150 LoC for logging, in particular to log the activation of tooltips.

6.7 DISCUSSION

EvalBench grew out of the experiences we made in carrying out evaluations of our research prototypes and was driven by the practical requirements imposed from different evaluation methods. Developed initially for internal use, we generalized our concepts and have now reached a stage with an adequate amount of functionality to support different evaluation scenarios and mature enough to make it available for the visualization community as open source library. Thus, visualization researchers can take advantage of it and also contribute to the improvement and further development of EvalBench.

Benefits. The implementation is based on an extensible architecture and integrates multiple software design patterns [Gamma et al., 1995] to ensure flexibility and pave the way for future experimenters to reuse and adopt the library for their special needs. This has been demonstrated in one of the case studies, where custom data input was added in the form of direct time interval selection. In contrast to related work for supporting evaluations in HCI, EvalBench takes a different approach. Through loose coupling, the library can be integrated into existing visualization solutions without the need for major changes in the architecture of the artifact to examine. The flexibility of the library is shown by its support for a variety of different evaluation methods that are commonly used in visualization such as controlled experiments, laboratory questionnaires, interaction logging, insight diaries, heuristic evaluations, as well as combinations thereof. EvalBench allows more reliable and precise evaluation than approaches relying on manual methods of data gathering. Studies such as [Aigner et al., 2005; Martins et al., 2008; Ordóñez et al., 2010] report problems because of imprecise and false measurements due to manual time recording. These can be mitigated by using automated methods for data recording. Moreover, a study performed with EvalBench can be reused and reproduced. The exact tasks and the study design are structured in a modular way and can be reused in subsequent studies. Additionally, the setup of an experiment with EvalBench is relatively easy and does not require the developer to engage deeply with the inner working principles of the library as long as the default implementations of the components are sufficient. This is underlined by the fact that several master students as well as high school interns were quickly able to use and extend EvalBench in the past. The library has already shown to be a valuable asset when conducting user studies as documented in Sect. 6.6 and we have also demonstrated that it decreases implementation effort considerably.

Limitations. Although EvalBench already has a good set of functionality, there are a number of limitations to consider. First, it only supports a subset of the portfolio of evaluation methods applicable to visualizations. Second, even though EvalBench was designed as minimally invasive as possible for the visualization to evaluate, implementation effort is still necessary. Black-box evaluation is possible but not in a straightforward manner. For example, to perform a study with Tableau, the delegate can launch it with command line parameters or control it through an API while leading through the study in its own window. Third, loading study designs from files is not yet implemented in EvalBench. Another limitation is that currently the library does not support remote controlling of experiment sessions. In a computer lab setting for example it would be desirable to perform a number of evaluation sessions simultaneously with a group of test persons. This approach would allow equally distributed factors among subjects and also ensure equal conditions for all test persons, since the instructions that test persons receive play a crucial role and physical and social environmental factors may introduce systematic errors [Lazar et al., 2010].

6.8 conclusion & future work

Our main contributions are threefold: First, we have developed an easy-to-use, flexible, and reusable software library specifically suited to the requirements of evaluating visualizations. EvalBench helps in carrying out commonly used evaluation methods in the field and puts attention not only to support controlled laboratory studies but allowing for a higher degree of realism such as with long-term and insight-based methods. Second, we have discussed the design choices and basic abstractions of our conceptual architecture independent of specific programming languages. Third, we have shown the practical utility of applying EvalBench in a number of case studies. They cover a range of popular evaluation scenarios including both, quantitative and qualitative methods.

Future Work. Although EvalBench was designed on the basis of a number of different user studies, not every aspect of possible experiments is covered. Therefore, the library currently constitutes a primal structure and will hopefully be adapted and further developed by future experimenters for additional applications in human-centered visualization design and development. There are already a number of ideas on how to extend the library. Despite the fact that in many cases it is sufficient to edit an XML file and write a few lines of code to setup an experiment, a visual editor would be desirable to ease experiment design for users. Another possibility to support this would be to establish interoperability with the design platform of Touchstone and to extend EvalBench to load these externally created experiment designs. As mentioned in the limitations, it would make sense to have centralized administration of remote experiments. Doing so, we could, for example, make sure that subjects are assigned evenly to experiment groups or that measurements and results are collected on a single place on a server. Another useful extension that would make an even broader usage possible is to integrate further modalities of data collection. This could include audio for thinking aloud, video in terms of screen recording and/or videotaping of subjects, eye tracking, or functional magnetic resonance imaging (fMRI). Moreover, EvalBench allows for recording data in a structured format but does not offer any functionality for analyzing the data. In this sense, it could be extended to include generic statistical processing functions or interface directly with statistics software such as R. It is also expected that future evaluation experiments will make it necessary to extend the question types and corresponding answering possibilities. This is why the structure of the library was designed to facilitate the extension with new question types. Finally, the library can be ported to other object-oriented programming languages, following the conceptual design presented in Sect. 6.4.

Apart from the ways to possibly extend EvalBench, other organizational measures could help to increase the quality of evaluations. Specifically, it would be very helpful to have an online repository that hosts benchmark data sets, analysis scripts, examples for study designs, tasks, questionnaires (e.g., a standardized demographic questionnaire) possibly already in a form that can be directly reused in EvalBench. Our aim is to provide useful infrastructure in order to leverage evaluation in visualization as a service to the community.

Acknowledgements. This work was supported by the Laura Bassi Centre of Excellence initiative via CVAST (#822746), by the Austrian Science Fund (FWF) via the HypoVis project (#P22883), and by the European Commission via the MobiGuide project (#FP7-287811). Many thanks to David Bauer, Barbara Neubauer, and Thomas Turic for their implementation contributions, as well as Paolo Federico and Silvia Miksch for feedback to our manuscript.

REFERENCES

Ahmad, W. F. W., Sulaiman, S., and Johari, F. S. (2010). Usability management system (USE-MATE): A web-based automated system for managing usability testing systematically. In *Proc. 2010 Int. Conf. User Science and Engineering, i-USEr*, pages 110–115.

Aigner, W., Kainz, C., Ma, R., and Miksch, S. (2011). Bertin was right: An empirical evaluation of indexing to compare multivariate time-series data using line plots. *Comp. Graphics Forum*, 30(1):215–228.

Aigner, W., Miksch, S., Thurnher, B., and Biffl, S. (2005). PlanningLines: novel glyphs for representing temporal uncertainties and their evaluation. In *Proc. of Int. Conf. on Information Visualisation, IV*, pages 457–463. IEEE.

Aigner, W., Rind, A., and Hoffmann, S. (2012). Comparative evaluation of an interactive time-series visualization that combines quantitative data with qualitative abstractions. *Comp. Graphics Forum*, 31(3):995–1004.

Andrews, K. and Kasanicka, J. (2007). A comparative study of four hierarchy browsers using the hierarchical visualisation testing environment (HVTE). In *Proc. Int. Conf. Information Visualization, IV*, pages 81–86. IEEE.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3: Data-driven documents. *IEEE Trans. Visualization and Computer Graphics*, 17(12):2301–2309.

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. (2006). VisTrails: visualization meets data management. In *Proc. 2006 ACM SIGMOD Int. Conf. Management of Data*, pages 745–747.

Carpendale, S. (2008). Evaluating information visualizations. In Kerren, A., Stasko, J. T., Fekete, J.-D., and North, C., editors, *Information Visualization*, pages 19–45. Springer, Berlin.

Cowley, P., Haack, J., Littlefield, R., and Hampson, E. (2006). Glass box: capturing, archiving, and retrieving workstation activities. In *Proc. Workshop Continuous Archival and Retrieval of Personal Experiences*, pages 13–18. ACM.

Dou, W., Jeong, D. H., Stukes, F., Ribarsky, W., Lipford, H. R., and Chang, R. (2009). Recovering reasoning processes from user interactions. *IEEE Comp. Graphics and Applications*, 29(3):52–61.

Elmqvist, N. and Yi, J. S. (2012). Patterns for visualization evaluation. In *Proc. Workshop BEyond time and errors: novel evaLuation methods for Information Visualization, BELIV*, pages 12:1–12:8. ACM.

Fekete, J.-D., Hemery, P.-L., Baudel, T., and Wood, J. (2011). Obvious: A meta-toolkit to encapsulate information visualization toolkits—one toolkit to bind them all. In *Proc. IEEE Conf. Visual Analytics Science and Technology*, pages 91–100.

Forsell, C. (2010). A guide to scientific evaluation in information visualization. In *Proc. 2010 14th Int. Conf. Information Visualisation, IV*, pages 162–169. IEEE.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman.

Gomez, S. and Laidlaw, D. (2012). Modeling task performance for a crowd of users from interaction histories. In *Proc. SIGCHI Conf. Human Factors in Computing Systems, CHI '12*, pages 2465–2468. ACM.

Heer, J. and Bostock, M. (2010). Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proc. Conf. on Human Factors in Computing Systems, CHI*, pages 203–212. ACM.

Heer, J., Card, S. K., and Landay, J. A. (2005). prefuse: a toolkit for interactive information visualization. In *Proc. Conf. on Human Factors in Computing Systems, CHI*, pages 421–430. ACM.

Heer, J., Mackinlay, J. D., Stolte, C., and Agrawala, M. (2008). Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans. Visualization and Computer Graphics*, 14:1189–1196.

Henry, N., Elmqvist, N., and Fekete, J.-D. (2008). A methodological note on setting-up logging and replay mechanisms in InfoVis systems. In *Proc. Workshop BEyond time and errors: novel evaLuation methods for Information Visualization at ACM CHI, BELIV*.

Jin, J. and Szekely, P. (2010). Interactive querying of temporal data using a comic strip metaphor. In *Proc. IEEE Symp. Visual Analytics Science and Technology*, pages 163–170.

Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F., editors (2010). *Mastering The Information Age – Solving Problems with Visual Analytics*. Eurographics.

Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., and Wixon, D. (2008). Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. In *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems, CHI '08*, pages 443–452.

Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. (2012). Empirical studies in information visualization: Seven scenarios. *IEEE Trans. Visualization and Computer Graphics*, 18(9):1520–1536.

Lazar, J., Feng, J. H., and Hochheiser, H. (2010). *Research Methods in Human-Computer Interaction*. John Wiley & Sons.

Mackay, W. E., Appert, C., Beaudouin-Lafon, M., Chapuis, O., Du, Y., Fekete, J.-D., and Guiard, Y. (2007). Touchstone: exploratory design of experiments. In *Proc. Conf. on Human Factors in Computing Systems, CHI*, pages 1425–1434. ACM.

Martins, S. B., Shahar, Y., Goren-Bar, D., Galperin, M., Kaizer, H., Basso, L. V., McNaughton, D., and Goldstein, M. K. (2008). Evaluation of an architecture for intelligent query and exploration of time-oriented clinical data. *Artificial Intelligence in Medicine*, 43:17–34.

Mathôt, S., Schreij, D., and Theeuwes, J. (2012). OpenSesame: an open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44:314–324.

Neubauer, B. (2012). A comparison of static and dynamic visualizations for time-oriented data. Master's thesis, Vienna University of Technology.

Nielsen, J. (1994). Usability inspection methods. In *Conf. Companion on Human Factors in Computing Systems, CHI '94*, pages 413–414. ACM.

North, C. (2006). Toward measuring visualization insight. *IEEE Comp. Graphics and Applications*, 26(3):6–9.

Ordóñez, P., desJardins, M., Lombardi, M., Lehmann, C. U., and Fackler, J. (2010). An animated multivariate visualization for physiological and clinical data in the ICU. In *Proc. of Int. Health Informatics Symp.*, pages 771–779. ACM.

Plaisant, C. (2004). The challenge of information visualization evaluation. In *Proc. Conf. on Advanced Visual Interfaces (AVI)*, pages 109–116. ACM.

Pohl, M., Wiltner, S., Miksch, S., Aigner, W., and Rind, A. (2012). Analysing interactivity in information visualisation. *KI – Künstliche Intelligenz*, 26:151–159.

Pohl, M., Wiltner, S., Rind, A., Aigner, W., Miksch, S., Turic, T., and Drexler, F. (2011). Patient development at a glance: An evaluation of a medical data visualization. In Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., and Winckler, M., editors, *Proc. IFIP Human-Computer Interaction, INTERACT, Part IV*, LNCS 6949, pages 292–299. Springer.

Rester, M., Pohl, M., Wiltner, S., Hinum, K., Miksch, S., Popow, C., and Ohmann, S. (2007). Evaluating an InfoVis technique using insight reports. In *Proc. Conf. Information Visualization, IV*, pages 693–700. IEEE.

Shrinivasan, Y. B. and van Wijk, J. J. (2008). Supporting the analytical reasoning process in information visualization. In *Proc. Conf. on Human Factors in Computing Systems, CHI*, pages 1237–1246.

Soukoreff, R. W. and MacKenzie, I. S. (1995). Generalized fitts' law model builder. In *Conference Companion on Human Factors in Computing Systems, CHI '95*, pages 113–114.

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE.

Yi, J. S., Kang, Y. A., Stasko, J. T., and Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Visualization and Computer Graphics*, 13(6):1224–1231.

CHAPTER 7

Bertin was Right: An Empirical Evaluation of Indexing to Compare Multivariate Time-Series Data Using Line Plots

Wolfgang Aigner, Christian Kainz, Rui Ma, and Silvia Miksch

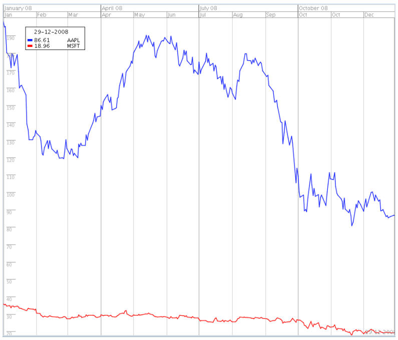
Abstract · *Line plots are very well suited for visually representing time series. However, several difficulties arise when multivariate heterogeneous time-series data is displayed and compared visually. Especially, if the developments and trends of time-series of different units or value ranges need to be compared, a straight forward overlay could be visually misleading. To mitigate this, visualization pioneer Jacques Bertin presented a method called indexing that transforms data into comparable units for visual representation. In this paper, we want to provide empirical evidence for this method and present a comparative study of the three visual comparison methods linear scale with juxtaposition, log scale with superimposition, and indexing. While for task completion times, indexing only shows slight advantages, the results support the assumption that the indexing method enables the user to perform comparison tasks with a significantly lower error rate. Furthermore, a post-test questionnaire showed that the majority of the participants favor the indexing method over the two other comparison methods.*

Keywords · Information visualization, line plot, time-series, indexing, empirical evaluation.

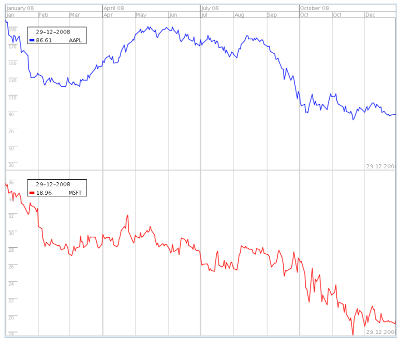
This article originally appeared as [Aigner et al., 2011]:

Aigner, W., Kainz, C., Ma, R., and Miksch, S. (2011). Bertin was Right: An Empirical Evaluation of Indexing to Compare Multivariate Time-Series Data Using Line Plots. *Computer Graphics Forum*, 30(1):215–228.

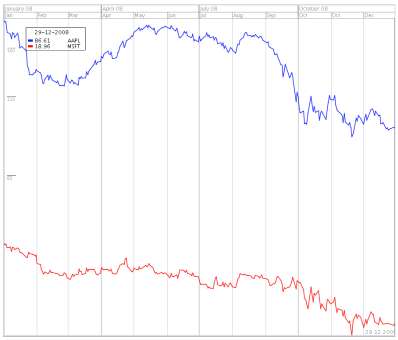
(a) Superimposition on linear scale.



(b) Juxtaposition on linear scale (Li+J).



(c) Superimposition on log scale (Lo+S).



(d) Indexing (I).

FIGURE 7.1 – **Different configurations for multivariate time-series comparison.** Closing prices of the two stocks of Microsoft (MSFT, red) and Apple (AAPL, blue) are shown.

7.1 INTRODUCTION & MOTIVATION: WHY CARE?

TIME-SERIES are one of the most common forms of data that can be found in diverse application areas such as finance, natural sciences, engineering disciplines, and many more. A time-series is a collection of observations made sequentially over time [Lee et al., 1998]. The most often used visual representations for such kind of data are line plots [Tufte, 1983]. Due to their simplicity and well known form, they are understood easily and no learning is required. Line plots employ position encoding in a Cartesian coordinate system that map time typically on position on the x-axis and the corresponding value on position on the y-axis. Subsequent data points are connected by lines and the slope of the line encodes the rate of change. The resulting polyline emphasizes the development over time rather than individual values. By using the most exact visual variable, line plots are particularly efficient, i.e., fast and exact to interpret by the human visual system [Cleveland and McGill, 1984; Mackinlay, 1986].

In the simplest case we are dealing with univariate time-series that contain data about one variable over time which can be represented in a straight forward manner. But hardly ever analysts have to deal with a single variable only. More often, developments need to be compared in order to gain insights on relationships, correlations, and patterns between several variables. However, several difficulties arise when multivariate heterogeneous time-series data is displayed and compared visually. In the following, we will discuss three of these problems together with possible solutions. First, we want to look at multivariate homogeneous data, i.e., data that contain variables of the same type and unit. How can these be visualized in order to allow for visual comparisons? The simplest case is to superimpose the different variables within a single coordinate system. This employs the major advantage that the individual lines are layed out close to each other and thus allow for an easy direct comparison.

Problem 1: Largely different value domains The superimposition approach stated above might be problematic if variables that have largely different value domains are involved. Fig. 7.1(a) shows an example to illustrate this. In this case, line plots of the closing prices of the two stocks of Microsoft (MSFT) and Apple (AAPL) are superimposed. MSFT, on the one hand, has a value domain in the range of 80 to 200 in the shown time interval while AAPL, on the other hand, has a value domain of 20 to 40. These largely different value domains lead to an underrepresentation of the dynamics of the smaller value domain and makes relative comparisons prone to errors.

A possible solution to this is juxtaposition, i.e., to display the different plots next to each other while adjusting the scale dynamically to make relative changes and the overall shape of variable development better comparable. Fig. 7.1(b) shows the same data as Fig. 7.1(a) by presenting the second variable underneath the first one on a synchronized time scale. Other arrangements are also possible and in its generalized form, this approach is related to *small multiples* [Tufte, 1983].

Problem 2: Percent changes are not represented accordingly Not only largely different value domains pose a challenge to line plots, but also the representation and comparison of percent changes (cmp. [Bissantz, 2008; Few, 2009]). On linear scales, constant percentual changes are displayed as exponentially increasing lines (see Fig. 7.2(a)). Furthermore, the same percentual changes are represented via lines of different slopes. E.g., an increase of 100% from a value of 10 to a value of 20 is represented by the same slope as an increase of only 10% from a value of 100 to a value of 110.

A possible solution to mitigate this is using logarithmic scales instead of linear ones. In this case, equal percentual changes are represented by equal

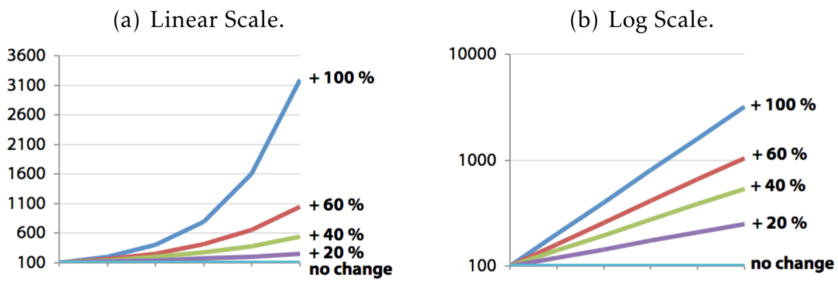


FIGURE 7.2 – **Influence of scales on visualization of constant percent changes.**

slopes (see Fig. 7.2(b)). This approach is shown in Fig. 7.1(c) where percentage changes of MSFT and AAPL stock prices can be compared visually directly and also the largely different value domains problem can be overcome by using log scales.

Problem 3: Heterogeneous data So far, we have focused on multivariate homogeneous data. In contrast to that, multivariate time-series are called heterogeneous in case different kinds of data or units are involved. The simplest solution is again to use juxtaposition as described before. A further, frequently applied approach is to use superimposition combined with multiple y-axes. However, this also introduces two main problems. First, it is limited to only very few heterogeneous variables (mostly not more than two). Second, and most important, the visual appearance and interrelationship of

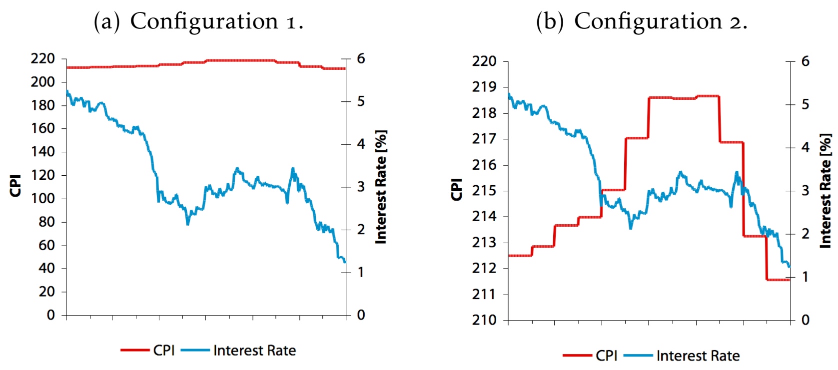


FIGURE 7.3 – **Superimposition with multiple y-axes:** The interrelationship of different variables is largely dependent on the selection of scales. Especially line crossings and vertical positions in relation to each other are mostly arbitrary.

different variables is largely dependent on the selection of the scales for the individual y-axes. These relationships (especially line crossings and vertical position in relation to each other) are mostly arbitrary as illustrated in Fig. 7.3. This problem is also very eloquently demonstrated by Wainer [1997] along an example of the relationship of smoking and lung cancer.

Visualization pioneer Bertin also dealt with this problem in his seminal work [Bertin, 1967] (English translation [Bertin, 1983]) and introduced *indexing* as possible solution. The indexing method avoids the problems mentioned before by using a simple transformation of the original values for each time-series. The result is a set of new values of a percent unit (see Fig. 7.1(d)). The heterogeneous time series are converted into homogeneous data, which can easily be compared by superimposition. Bertin defines the indexing method with the following formula:

$index_n = \frac{Q_n}{Q_i} * 100\ [\%]$

The new indexed value is calculated for every element in the original time series. The point i refers to the indexing point. This is a special point of the time series. It is the base point for all percent calculations. The index value for the point n is thus calculated via the formula described above. Q_i is the value of the indexing point and represents 100%. Q_n is the original value of the time series. By using this method all displayed time series values use the same percent dimension. Applying this, heterogeneous time-series are far easier to compare. For example the time-series can be drawn in superimposition without any arbitrary choice of scales and ranges of the different axes dimensions.

Although the indexing method was introduced by Bertin more than 40 years ago, there exists to our best knowledge no empirical evidence on its effectiveness and efficiency. To fill this gap, we conducted a comparative study to assess three different configurations of line plots with a particular focus on comparison tasks. In the upcoming section, we provide information on related work to our study. We based the experiment design on a well established task taxonomy [Andrienko and Andrienko, 2006] that is shortly explained in Section 7.3. After that, our hypotheses will be presented in Section 7.4 followed by a description of the experiment design in Section 7.5. Next, the results of the empirical study will be presented in Section 7.6 and discussed in Section 7.7. Finally, we will provide a conclusion and ideas on future work in Sections 7.8 and 7.9.

7.2 RELATED WORK

In their fundamental work on graphical perception, Cleveland and McGill [1984] conducted empirical experiments on different visualizations. In this regard, they identified a set of elementary tasks of perception (ordered from most to least accurate): 1) Position along a common scale, 2) Positions along

Nr.	Task Type	Question
1	Elementary Lookup	<stock 1>: On which day was the highest stock price in <year>?
2	Elementary Lookup	<stock 1>: On which day was the lowest stock price in <year>?
3	Elementary Comparison	Compare the values of <stock 1> and <stock 2> on the given <dates>. Which of the following statements are valid? On <date1>, <stock 1> was higher then <stock 2>. On <date 2>, <stock 1> was lower than <stock 2>.
4	Elementary Comparison	Please quantify the amount of price change for the given time periods in dollars for <stock 1> and <stock 2>.
5	Elementary Comparison	Please quantify the amount of price change for the given time periods in percent for <stock 1> and <stock 2>.
6	Elementary Comparison	Compare the values of <stock 1> and the <stock index> index on the given <dates>. On <date 1> was the value of <stock index> over <value 1>. On <date 2> was the value of <stock 2> under <value 2>.
7	Elementary Comparison	<stock index>: How much percent did the values change in <year>?
8	Elementary Relation-Seeking	<stock 1>: Which of the following months in <year> have a higher value than the value on <date>?
9	Synoptic Pattern Identification	<stock 1>: Which of the following months in <year> have a positive trend?
10	Synoptic Behavior Comparison	Which stock has a bigger percent increase from the beginning of <month> to the end of <month>?
11	Synoptic Behavior Comparison	Which stock has a lower percent loss in <year>?
12	Synoptic Behavior Comparison	In which months is the percent increase of <stock 1> greater than <stock index>?
13	Synoptic Behavior Comparison	Which stock or index has the highest volatility (relative variations) in September <year>?
14	Synoptic Relation-Seeking	In which year had <stock 1> the highest percent increase from beginning to the end of the year?

Table 7.1 – **14 user tasks of the experiment.** The second column (Task Type) refers to the task taxonomy in [Andrienko and Andrienko, 2006].

nonaligned scales, 3) Length, direction, and angle, 4) Area, 5) Volume, curvature, and 6) Shading, color saturation. Applying their theory, they propose using curve-difference plots instead of plotting individual variables for difference judgements. However, this is only possible for two homogeneous variables. Moreover, they propose that stacked charts should be avoided in case more than two variables are plotted. Instead, the individual variables should be plotted directly together with a line for their sum.

In a further work, Cleveland investigates the aspect ratio of line plots and demonstrates how it can impact graphical perception [Cleveland, 1993]. He proposes a method called *banking to 45°* that optimizes the aspect ratio of a line plot based on the average orientation of all line segments which should be 45 degrees. This technique has been revisited and extended by Heer and Agrawala [2006]. They present additional optimization criteria and a technique that includes spectral analysis called *multi-scale banking*. Related to that, Beattie and Jones investigated graph slope for change judgements of corporate financial performance reports in [Beattie and Jones, 2002]. They conducted an empirical study using realistic graphics of corporate reports finding that sub-optimal slope parameters do result in distorted judgements of visualizations. Again, the results confirm Cleveland's basic assumption that an average slope of 45° is optimal in terms of judgement accuracy. Recently, *horizon graphs*, a novel visualization technique for time series that seeks for optimal usage of space for large multivariate datasets has been investigated empirically [Heer et al., 2009]. Particularly, value comparison tasks have been studied and different configurations of chart height and number of bands were compared.

Considering the application domain of stock market data visualization, two prominent web-based examples are *Google Finance* (www.google.com/finance) and *Yahoo Finance* (finance.yahoo.com). Both applications apply indexing when multiple stocks are displayed for comparison. However, the position or value of the indexing point cannot be influenced by the user and is fixed to the first point in time displayed. Moreover, both Google and Yahoo allow for setting the value scale to linear or logarithmic. Other than that, indexing is applied in a number of stock market data visualization applications. But to the best of our knowledge, no empirical study exists that investigates *indexing* as comparison method for multivariate time-series data.

7.3 USER TASKS: COMPARING TIME-SERIES VISUALLY

The selection of the proper user tasks is critical for the relevance and also for the success of the evaluation. The structure of the tasks for the evaluation is based on the task taxonomy of Andrienko and Andrienko [2006] which is divided into two categories: elementary tasks and synoptic tasks.

Elementary tasks set their focus on single values or single points in time. Andrienko and Andrienko [2006] define three elementary task types: *lookup*, *comparison*, and *relation-seeking*. Elementary lookup tasks refer to seek a specific value of a single time series (e.g., find a value for a specific point in time). Elementary comparison tasks refer to tasks that involve a comparison of values at different points in time or different variables at the same point of time, and elementary relation-seeking tasks refer to patterns within a single time series (e.g., find the points in time which have a higher value than the value of a given point in time).

Synoptic tasks are centered on analyzing multiple configurations of characteristics corresponding to subsets of a time-series. Andrienko and Andrienko [2006] define the three following synoptic task types: *pattern identification*, *behavior (pattern) comparison* and *relation-seeking*. Synoptic pattern identification tasks refer to recognition of particular patterns in the given time-series data (e.g., do the values in a given month follow a positive or negative trend?). Synoptic behavior (pattern) comparison tasks refer to identifying and comparing patterns (e.g., which of two stocks has a higher volatility for a given time period?). Synoptic relation-seeking tasks refer to relation of patterns between multiple time-series (e.g., identify the intervals of two time-series that share the same trend). Because we are investigating visual methods for comparing time-series, we focus especially on different forms of comparison tasks rather than lookup tasks.

7.4 HYPOTHESES

Table 7.1 lists the 14 tasks used for the study along with their types according to the presented taxonomy. We are focusing on comparison and relation-seeking tasks with absolute (value) and relative (percent) comparisons.

A set of five hypotheses is the starting point of our study. Each of these hypotheses will be investigated by grouping relevant tasks of Table 7.1 into task sets accordingly. The relationship of tasks and hypotheses is shown in Table 7.2. The five hypotheses can be separated approximately into three groups. The first group (H1 and H2) focuses on questions about the visual comparison of percent changes in line plots. The second group (H3) is concerned with the visual recognition of the development of variables. The third group (H4, H5) contains questions on a more generic set of tasks also including lookup and pattern identification.

H1: Log scale is more appropriate for percent estimation than linear scale
A logarithmic scale represents percent changes of the displayed data directly and proportionally. It is predicted that estimations of percent changes are

Hypothesis	Task Set	Visualizations
H1	5, 7, 10-14	Li+J, Lo+S
H2	5, 7, 10-14	Lo+S, I
H3	5, 7-14	Li+J, Lo+S, I
H4	1-14	Li+J, Lo+S
H5	1-14	Li+J, Lo+S, I

TABLE 7.2 – **Hypotheses and associated task sets** (Li+J: juxtaposed line plot with linear scale, Lo+S: superimposed line plot with log scale, I: indexing).

more precise and faster when using a log scale compared to estimations of percent changes when using a linear scale.

H2: Indexing is better suited for percent estimation than log scale The indexing method transforms absolute values into percent changes based on the indexing point. This method should make visual comparisons of percent changes easier, i.e., reduce the needed time to estimate percent changes. It is predicted that the indexing method is more effective for estimation and comparison tasks of percent changes than logarithmic scaled line plots, which display absolute values.

H3: Indexing is more effective for trend comparison Indexing is useful for comparisons of time series trends. It is predicted that the subjects can make estimations and comparisons of trends more precise and faster.

H4: Superimposed, logarithmic scaled line plots are better than juxtaposed, linear scaled line plots for visual comparisons Superimposed, logarithmic scaled line plots represent percent changes directly and proportionally. Comparisons by superimposition should be easier than by juxtaposition. It is predicted that comparison of absolute values, comparison of percent changes, and comparisons of trends are faster and contain less errors than comparisons with juxtaposed, linear scaled line plots.

H5: Indexing is overall better for visual comparisons The indexing method leads to a direct display of percent changes and makes multivariate heterogeneous time series directly comparable. It is predicted that the indexing method makes comparisons of absolute values, relative values, and trends faster and comparison results have higher task correctness rates.

7.5 EXPERIMENT DESIGN: INTERACTIVE PROTOTYPE WITH EVALUATION FRAMEWORK & EXPERIMENT SETTING

Stock market data is a prototypical example for time-series and involve the difficulties of largely different value domains, percentage comparisons as well as heterogeneous data (e.g., stock indices or economic indicators such as

	Stock 1	*Stock 2*	*Stock Index*
Dataset 1	AAPL	IBM	NASDAQ
Dataset 2	AMZN	YAHOO	SP500
Dataset 3	MSFT	CHINA PETROLEUM	DJIA

TABLE 7.3 – **Datasets used for the experiment**.

consumer price index or interest rates). Particularly, comparisons of relative changes are often more important than absolute values. Due to the fact that most individuals are at least moderately familiar with stocks, this area seems to be well suited as application domain for an empirical study.

The type of visualization is the independent variable of the experiment and three different visualization types will be compared against each other. In order to provide a correct and fair comparison of multivariate heterogeneous data, the following three configurations of line plots are used. The first type is the juxtaposed line plot with linear scale (*Li+J*). The second type is the superimposed line plot with a logarithmic scaled y-axis (*Lo+S*). The third type is the line plot visualization based on the indexing method (*I*). These configurations have been deliberately chosen because they are recommended by well-known heuristics (see Section 7.1) to appropriately reflect the data visually for comparison tasks.

The two dependent variables of the user test are task completion time and task accuracy. In general, task accuracy will be interpreted as a binary value of true or false for each task and each subject. These values will subsequently be aggregated to an error rate per task set for each hypothesis.

The experiment was conducted using a within-subjects approach. This increased the output of the test results, because every test person evaluated all three visualization types. Each subject used the juxtaposed line plot, the superimposed line plot and the indexing plot instead of just one visualization type. This method implies the use of a Latin square variation of visualization types to counterbalance any learning and fatigue effects of the involved test users. Furthermore, the dataset for each task was randomly assigned during the test process to avoid any threats to validity because of differences between the datasets. Each subject had to complete 14 tasks for every visualization type focusing on visual comparison of time-series. Eight of these tasks were elementary tasks and the remaining six were synoptic tasks. Every task of the 14 tasks was defined for three datasets. The three datasets differed in their choice of stocks and stock index and are listed in Table 7.3. The comparison tasks were defined in such a way that three different combinations of stock market data were used. Homogenous data consisted of two stocks, heterogeneous data consisted of one stock and one stock index as well as combinations of two stocks and one stock index.

7.5.1 *Materials & Environment*

To compare and evaluate time-series comparison methods, an interactive prototype was developed in Java based on the visualization framework prefuse [Heer et al., 2005]. The prototype offers multiple line plot methods to compare data (comparison by *juxtaposition*, by *superimposition* and with the *indexing* method). To ensure comparability between different visualization types, the

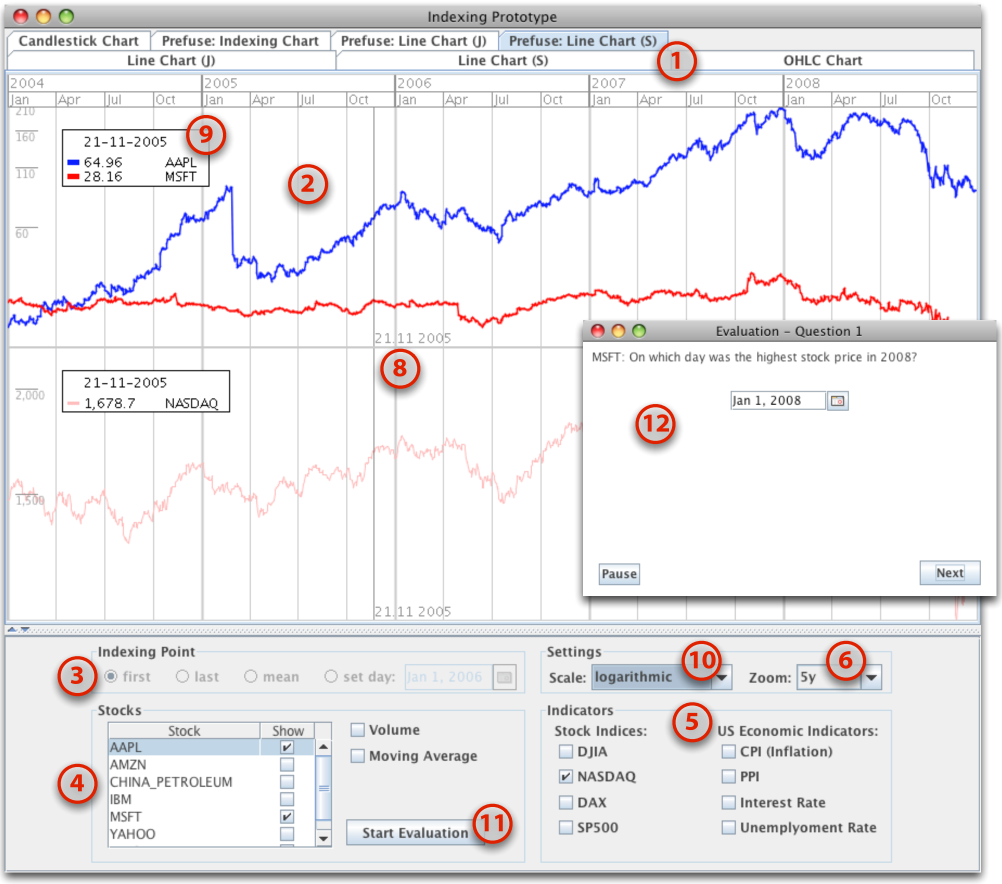


FIGURE 7.4 – **Interactive prototype** (1. Tab Bar, 2. Drawing Area, 3. Indexing Point, 4. Stock Selection, 5. Indicator Selection, 6. Zoom, 8. Mousetracker, 9. Dynamic Legend, 10. Scale Switching, 11. Evaluation Mode, 12. Evaluation Window).

same interaction possibilities are offered for all tested configurations. Fig. 7.4 shows a screenshot of the prototype, where the following areas are indicated.

1. **Tab Bar:** Switching between visualizations by selecting the respective tab.
2. **Drawing Area:** Display space for the visualizations.
3. **Indexing Point:** Indexing point can be set to first, last or a given date. It is also possible to define the indexing value based on the mean value. (Only available when the indexing plot is active.)
4. **Stock Selection:** Selection of stocks to display. Volume data and 20-days moving average can also be added.

5. **Indicator Selection:** Four major stock indices (DJIA, NASDAQ, DAX, and SP500) and four economic indicators (Consumer Price Index, Producer Price Index, interest rate and unemployment rate) of the USA are available.
6. **Zooming and Linking:** Available zoom ranges are five years, two years, one year, six months, three months, one month, and two weeks. More precise zooming is available using the mouse wheel. Juxtaposed plots are linked for visual comparisons between multiple time-series.
7. **Panning:** Via mouse clicking and dragging.
8. **Mousetracker:** Allows to gather information of the displayed time series according to the horizontal mouse position.
9. **Dynamic Legend:** Values for the legend are in accordance to the current mouse position on the horizontal time axis. This enables the user to quickly investigate the value for each displayed time series. To change the current date, the user has to move the mouse cursor to the desired location. This feature should improve the overall performance in visual comparison tasks. The user is able to get more precise information by looking up the actual values for a given day.
10. **Scale Switching:** Linear scale uses a constant ratio between a dimensional unit of the axis and the required space on the chart. The logarithmic scale can improve tasks where percent changes have to be compared.
11. **Evaluation Mode:** Button to start the evaluation mode.
12. **Evaluation Window:** Questions can be answered using different user interface widgets. The evaluation process can be paused any time by the user.

7.5.1.1 Evaluation framework

The evaluation itself was automated by integrating an evaluation framework into the prototype. It measures user performance by logging the needed time and accuracy of the results. The evaluation mode allows a simple execution of the user testing process. The application displays the tasks in a separate popup window that is displayed next to the main window (see Fig. 7.4). The evaluation framework guides the control of the tool, i.e., selection of stocks, indices, time range, and scale. Users need to work mainly based on the mousetracker, legend, and their perception and users of the indexing plot could additionally change the indexing point if they found it to be helpful. After the user answers the question the next task will be shown. Furthermore, the process can be paused by the user or the test supervisor in order to allow for breaks without distorting the recorded task completion times. In order to keep track of the test results the evaluation prototype automatically records the following set of parameters for each task of a user test in form of a CSV file: task number, visualization type, task completion time (ms), task correctness (yes/no), task description, correct answers, and given answers.

7.5.1.2 Environment

The test application was executed on the same laptop with the same computer mouse for all subjects. The laptop is fast enough to run the test application without any memory or processor problems. The hardware specifications of the used laptop are a 2 GHz Dual Core processor with 2 GB RAM and Windows XP SP3 as operating system. The graphics were displayed on a 15.4 inch LCD monitor with 1280 x 800 pixels resolution. A standard symmetrical shaped Logitech optical mouse was used as input device. Java Runtime version 1.6.0 was used to execute the evaluation application. All other programs were closed during the evaluation process. Otherwise some program might be interfering with the test application. The tests were conducted in a quiet environment with a relaxed and friendly atmosphere. Occurrences of external influences which would disturb the test user were minimized whenever possible.

7.5.2 *Subjects*

Twenty-four individuals participated in the comparative study. The age of the subjects is within the range of 20 to 30 years. Half of the test subjects are male and the other half are female. The education of all test users is at least a high school graduation and allows the owner to enroll at a university. Out of 24 test subjects 13 persons have a Bachelor's Degree and four people a Master's Degree. At the time of the test, 19 persons were studying at a university. One precondition for all participants was that they are used to work with a computer and a computer mouse as the ability to use the mouse as input device is essential for obtaining valid test results. The participants described themselves as more than average experienced with data analysis and line plots.

7.5.3 *Procedure*

Each test session involved a test supervisor and a test user. The duties of the test supervisor included setting up the test environment and ensuring that the test process was followed accordingly. The procedure is outlined in Table 7.4 and started with a greeting, introduction and orientation, and a pre-test questionnaire to gather basic personal information and previous experience with data analysis, stock analysis, and most common stock visualizations. After that, the prototype was demonstrated followed by the user test itself that consisted of 42 tasks for three visualizations. When the test was finished, a post-test questionnaire asked for the subjective visualization type preference. Finally, the test session ended with a debriefing.

Before the actual start of the user tests a pilot test was performed to find possible problems in the experiment design. The test process of the pilot test did correspond overall to the planned process. The estimation of the required time for the test process of 65 minutes was confirmed by the pilot test. The pilot test also showed that the set of 42 tasks is demanding a lot of concentration from the user. The required effort is relatively high but should nevertheless be reachable by most test users. After every block of 14 questions for one of the three visualizations a short break was made to ensure that the test user could remain concentrated for the remaining tasks.

To ensure repeatability, the used experiment material as well as the data collected on task completion times and error rates can be downloaded from http://ieg.ifs.tuwien.ac.at/research/bertin-was-right/.

7.5.4 *Analysis Approach*

In order to test the possible influence of using different datasets, a 2-way ANOVA was performed on both completion time and correctness rates of tasks using dataset and interface as factors. In terms of timing, the result yielded no significant influence based on the dataset ($F(2,999)=1.48$, $p=0.23$). Also for correctness rate, the influence of the dataset on the variance was not found to be significant ($F(2,117)=0.29$, $p=0.75$). Thus, for the remainder of analysis we compare the performance of different visualization methods regardless of the data set used.

The gathered data on completion times and correctness per task have been aggregated to task sets according to Table 7.2 whereas task completion times were summed up and error rates were calculated as ratio of errors to the overall number of task within a task set. The summary statistics for all task sets of the five hypotheses are presented in Table 7.5. Following this, completion times and error rates were tested for normal or log-normal distributions using the Shapiro-Wilk test. On the one hand, task set completion times follow a normal distribution for H1-H3 and a log-normal distribution for H4 and H5. On the

Activity	**Time [min]**
Greeting	2
Introduction and Orientation	5
Pre-Test Questionnaire	10
Demonstration of prototype	10
User Test	30
Post-Test Questionnaire	5
Debriefing	3
Total	65

Table 7.4 – **Overview of test procedure**.

time [s]	H1		H2		H3		H4		H5	
	mean	std.-dev.	mean	std.-dev.	mean	std.-dev.	mean	std.-dev.	mean	std.-dev
Li+J	223.79	36.23			282.51	52.83	425.83	82.87	425.83	82.87
Lo+S	184.39	46.58	184.39	46.58	240.43	66.69	391.45	98.84	391.45	98.84
I			142.82	33.62	226.28	58.19			406.91	111.29

error rate [%]	H1		H2		H3		H4		H5	
	mean	std.-dev.	mean	std.-dev.	mean	std.-dev.	mean	std.-dev.	mean	std.-dev
Li+J	0.55	0.19			0.50	0.18	0.36	0.14	0.36	0.14
Lo+S	0.49	0.14	0.49	0.14	0.44	0.12	0.31	0.10	0.31	0.10
I			0.14	0.20	0.17	0.20			0.16	0.16

TABLE 7.5 – **Summary statistics** of task set completion times (top) and task set error rates (bottom).

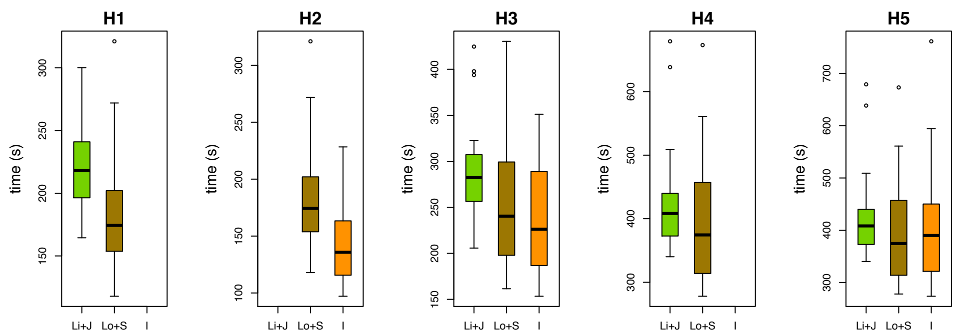


FIGURE 7.5 – **Boxplots for completion times** per visualization for each hypothesis.

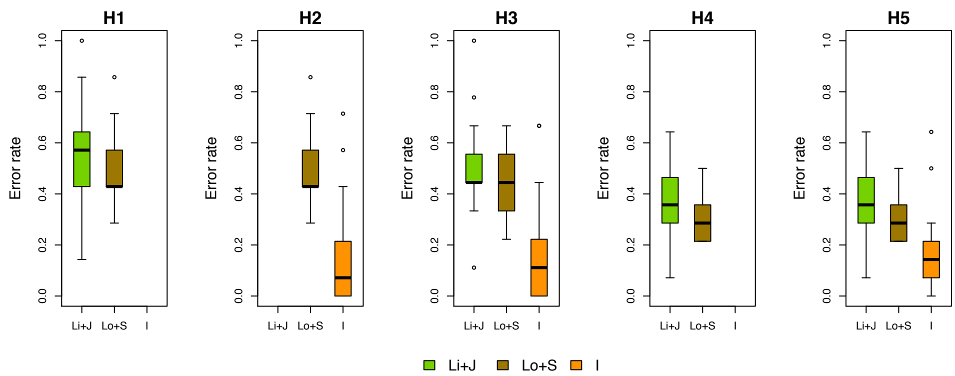


FIGURE 7.6 – **Boxplots for correctness rates** per visualization for each hypothesis.

other hand, error rates do not follow normal or log-normal distributions for any hypotheses. Additionally, F-tests on all task sets revealed that completion time and error rate data are having equal variances for hypotheses pairs and triples. Thus, the paired t-Test was used for testing completion times of H1, H2, and H4 and one-way repeated-measure ANOVA was applied for H3 and H5. For testing error rates, the non-parametric Wilcoxon signed-rank test was applied for H1, H2, and H4 and the Friedman test was used for H3 and H5. For post-hoc testing of H3 and H5, pairwise t-Tests as well as pairwise Wilcoxon tests with Bonferroni correction were applied. Furthermore, data on an individual task level was investigated in order to gather more detailed information on possible causes for test results.

The user preferences for the visualization for visual comparison tasks from the post-test questionnaires were analyzed using the chi-square test to determine if the user preferences are significantly different from an equal distribution.

7.6 RESULTS: INDEXING USERS HAVE HIGHER TASK CORRECTNESS RATES

As presented in Section 7.4, five hypotheses were tested in the empirical study. Testing was based on two measured dependent variables: task completion time and task correctness. Figures 7.5 and 7.6 show boxplots of the gathered data and Table 7.6 shows an overview of the statistical test results for the five hypotheses. All hypotheses, except the last, show a significant difference in the time needed for the execution of the task sets. In terms of error rate, H2, H3, and H5 show a significant difference.

H1: Log scale is more appropriate for percent estimation than linear scale With a paired t-test on completion times, we found a significant effect for visualization methods ($t(23) = 5.16$, $p < 0.001$) with Lo+S outperforming Li+J. The test results of the task completion time support the hypothesis that percent estimations in superimposed logarithmic scaled line plots are indeed significantly faster than in juxtaposed linear scaled line plots. However, a Wilcoxon signed-rank test shows that there is no significant effect of visualization method ($V=116$, $p>0.05$) for task correctness rates. In most tasks the correctness rates are close together. The values only diverge considerable in two tasks. One is task 5, which is an elementary comparison task with homogeneous data. The other is task 11, which is a synoptic behavior comparison task with homogeneous data. Visualization type Li+J has a 30 percent higher correctness in task 5, while visualization type Lo+S reaches a higher correctness rate of additional 40 percent in task 11.

H2: Indexing is better suited for percent estimation than log scale A paired t-test on completion times revealed a significant effect for visualization method ($t(23) = 5.70$, $p < 0.001$) with I outperforming Lo+S. Consistent with that, a Wilcoxon signed-rank test shows that there is a significant effect of visualization method on task correctness ($V=279$, $p<0.001$) with significantly lower error rates for I. Visualization type I has a higher correctness rate than visualization type Lo+S both overall and in each individual task involving percent estimation. This advantage should at least be partially based on the freely selectable indexing point. The user was able to set the time according to the needs which results in more correct answers. Interestingly, the task completion time is not increased although users had to additionally select a specific date.

hypotheses tests		H1	H2	H3	H4	H5
	visualizations	Li+J vs. Lo+S	Lo+S vs. I	Li+J vs. Lo+S vs. I	Li+J vs. Lo+S	Li+J vs. Lo+S vs. I
	task types	percent estimation	percent estimation	trend comparison	overall	overall
results	**time**	paired t-Test **t(23) = 5.16, p<0.001***	paired t-Test **t(23) = 5.70, p<0.001***	one-way repeated measures ANOVA **F(2,46)=12.27, p<0.001*** post-hoc: **Li+J vs. Lo+S (p<0.01)***, **Li+J vs. I (p<0.01)***	paired t-Test (log) **t(23) = 3.08, p<0.01***	one-way repeated measures ANOVA (log) F(2,46)=3.11, p>0.05
	error rate	Wilcoxon signed-rank test V=116, p>0.05	Wilcoxon signed-rank test **V=279, p<0.001***	Friedman test χ^2 = 21.57, **p<0.001*** post-hoc: **Li+J vs. Lo+S (p<0.01)***, **Li+J vs. I (p<0.01)***	Wilcoxon signed-rank test V=130, p>0.05	Friedman test χ^2 = 21.59, **p<0.001*** post-hoc: **Li+J vs. I (p<0.001)***, and **Lo+S vs. I (p<0.01)***

TABLE 7.6 – **Summary of hypotheses test results** (*...significant difference at an α-level of 0.05).

H3: Indexing is more effective for trend comparison With a one-way repeated-measure ANOVA, we found a significant effect of visualization method on completion time ($F(2,46)=12.27$, $p<0.001$). A post-hoc test on completion time using a pairwise t-Test with Bonferroni correction shows significant differences between Li+J and Lo+S ($p < 0.01$), and between Li+J and I ($p<0.01$). This means that indexing as well as logarithmic, superimposed visualization are outperforming the linear, juxtaposed visualization method but no significant difference was detected between I and Lo+S. A Friedman test also revealed a significant effect of visualization method on error rate ($\chi^2 = 21.57$, $p < 0.001$) and a post-hoc test with a pairwise Wilcoxon with Bonferroni correction showed significant differences between Li+J and I ($p < 0.001$), and between Lo+S and I ($p<0.001$). Hence, indexing outperforms both other visualization methods significantly for error rates. Again, the better results for the task correctness rates are probably partly based on the ability to select a user-defined indexing point. This hypothesis is generalizing the statement of the hypothesis H1 (Lo+S vs. Li+J - percent estimation) and H2 (I - percent estimation).

H4: Superimposed, logarithmic scaled line plots are better than juxtaposed, linear scaled line plots for visual comparisons With a paired t-test on log completion times, we found a significant effect for visualization methods ($t(23) = 3.08$, $p < 0.01$) with Lo+S outperforming Li+J. For error rates, a Wilcoxon signed-rank test shows that there is no significant effect of visualization method ($V=130$, $p>0.05$). These results are matching those of H1 for a broader set of tasks.

H5: Indexing is overall better for visual comparisons Task completion time is not significantly different between the three visualization types. This is shown with a one-way repeated-measure ANOVA, which could not find a significant effect of visualization method on log completion time ($F(2,46)=3.11$, $p>0.05$). But the task correctness rates of visualization type I are significantly higher compared to the other two visualization methods. A Friedman test revealed a significant effect of visualization method on error rate ($\chi^2 = 21.59$, $p < 0.001$). A post-hoc test with a pairwise Wilcoxon with Bonferroni correction revealed the significant differences between Li+J and I ($p < 0.001$), and between Lo+S and I ($p<0.01$) which means that indexing outperforms both other visualization methods for error rates in a broad set of tasks. This result is consistent with hypotheses H2 (I - percent estimation) and H3 (I - trend comparison). The visualization type I offers a higher correctness for similar or even faster task completion times. This is probably due to the advantage of the indexing plot to correctly superimpose homogeneous as well as heterogeneous time series.

7.6.1 *Subjective Preferences*

After a user completed the 14 tasks for each visualization type, one of the three visualization types had to be selected which was perceived as most useful. The visualization type I was chosen 19 times out of 24. Visualization type Li+J has been chosen only once and visualization type Lo+S has been chosen 4 times. A chi-square test revealed that the results are significantly different from a uniform distribution ($\chi^2 = 5.99$, $p < 0.001$).

7.7 DISCUSSION: SUPERIMPOSITION AND FLEXIBLE INDEXING POINT AS MAJOR FACTORS

Three visualization types for the display of multivariate time series were examined by a series of user tests with 24 subjects. Two dependent variables were measured to statistically compare the performance of the three visualization types linear scale, juxtaposition (Li+J), log scale, superimposition (Lo+S), and indexing (I). One dependent variable of the test was the task set completion time and the second dependent variable was task set correctness rate.

For percent estimation tasks, both dependent variables were consistently found to be significantly faster and less prone to errors using indexing compared to log scaled, superimposed line plots (H2). For the same set of tasks, a significant difference was found in task set completion time when comparing linear scaled, juxtaposed line plots with log scaled, superimposed line plots (H1). However, in terms of task set error rate, no significant difference was found between those two visualization methods. In other words, although subjects performed percent comparison tasks significantly faster using log scaled, superimposed line plots, error rates were not worse than with linear scaled, juxtaposed line plots. For trend comparison tasks, results are again consistent concerning an overall better performance of indexing over the two other visualization methods (H3). However, for task set completion time, indexing performed significantly better in contrast to linear scaled, juxtaposed line plots only and no significant difference to log scaled, superimposed line plots was found. Subjects using indexing were found to make significantly less errors than subjects using either linear scaled, juxtaposed line plots or log scaled, superimposed line plots. Concerning error rate, no significant differences were found between Li+J and Lo+S. For overall comparison tasks using linear scaled, juxtaposed and log scaled, superimposed line plots (H4), significant differences were found only concerning task set completion time favoring log scaled, superimposed line plots. When comparing all three visualization methods for overall comparison tasks (H5), no significant differences were found in terms of task set completion time. In contrast to that, subjects using indexing plots made significantly less errors as subjects using both Li+J or Lo+S. However, no significant difference could be detected between Li+J and Lo+S which is again consistent with H4. This means that although subjects using indexing plots were not significantly faster in comparison to the other two visualization methods, the error rate was significantly lower.

Task completion times of task 12 are standing out from completion times of the other tasks. The goal of the task is to visually compare the percentage increase between two time series each month of one year. The user has to identify which time series has the greater monthly percentage increase. This task is therefore consisting of twelve subtasks. This could explain a part of the higher task completion times. Although task 8 and 9 also consist of monthly comparisons, they are less complex and involve only one time series. Comparing task correctness rates between the three visualization types, they vary most of all for task 5, 7, 10, 11, and 12. Line plots with indexing have overall a higher correctness rate compared to the other two visualization types. Especially, percent estimation tasks are superior with indexing.

What can be said in general is that visualizations using log scales and indexing do not perform worse compared to linearly scaled line plots although they are not that widespread. The partly superior results of indexing could be a consequence of the ability to not only superimpose multivariate but also heterogeneous data. Any dimension is transformed into a percent dimension, which makes superimposition for any multivariate time series possible. The user can select an indexing point based on a specific point in time as start for the comparison. After that, all points on the chart represent relative changes in relation to the indexing point. This implies a significant increase in task correctness rates. Other than that, juxtaposed linear scaled line plots and superimposed logarithmic scaled line plots did mostly not have significant differences in their task correctness rate. So the test results give evidence that these two visualization types do not have a statistically significant effect on the correctness of the task results. Logarithmic scales enable the user to execute percent estimations faster than linear scales. The test results show that the scale has no significant effect on the task correctness. When performing a mixture of tasks the advantage of logarithmic scales disappears.

The results of the empirical study show that indexing is superior to the other two visualization types. Performance measures and test user's subjective opinions favor this visualization method.

7.7.1 *Limitations*

A main limitation of the study at hand is the relatively low number of subjects. Even though it is similar to comparable studies (e.g., [Heer et al., 2009]) and consists of a quite uniform and well-balanced group of subjects, a larger number of subjects would lead to more statistical power. Furthermore, one of the used tasks turned out to be an outlier in terms of task completion time and error rate (Task 12) as already discussed in the previous section. It would have been better to split this tasks into several more comparable tasks. Furthermore, many variations were introduced by slightly different task settings within one task type. Minimizing these variances and introducing more repetitions for tasks without changing any variable would have led to more statistical power. Apart from recording task completion times and given answers it would have been helpful to also log interactions performed by the user. This would for example allow to find out how many users did in fact change the indexing point during work on a task. Also, the randomized association of datasets for each task led to difficulties in analysis of the influence of the dataset because it can't be measured at hypotheses level accordingly due to aggregation. From a visualization design point of view, the visualization methods lack horizontal gridlines which might have lead to a disadvantage for the juxtaposed setting.

7.8 CONCLUSION: BERTIN WAS RIGHT

Line plots are very well suited for visually representing time-series. However, several difficulties arise when multivariate heterogeneous time-series data is displayed and compared visually. Especially, if the developments and trends of time-series of different units or value ranges need to be compared, a straight forward overlay could be visually misleading. To mitigate this, visualization pioneer Jacques Bertin presented a method called indexing that transforms data into comparable units for visual representation. The main contribution of this paper is an empirical study that assesses the indexing method as well as the design and implementation of an interactive visualization prototype including an evaluation framework.

Although the indexing method was proposed by Bertin more than 40 years ago, its effectiveness was not investigated empirically to date. Therefore, a comparative study with 24 subjects was conducted to examine differences in task completion times and task correctness rates for three line plot visualization variants. The three observed visualization types are juxtaposed linear scaled line plot, superimposed logarithmic scaled line plot and line plot with indexing. For evaluating the visualization techniques, realistic stimuli were used in form of tasks related to stock market data. The study consisted of 14 tasks for each visualization type and homogeneous as well as heterogeneous data. The used tasks were based on a specific task taxonomy [Andrienko and Andrienko, 2006] for spatiotemporal data. The focus of the test was set on both elementary and synoptic comparison tasks.

The test results give clear evidence that using indexing in general yields a higher correctness rate than the two other visualization types. For task completion times, the results are less clear but also show advantages in using indexing plots. One of the two main benefits of indexing is the ability to superimpose any data by transformation of values into a percent dimension. The other benefit is the user-defined setting of an indexing point. This makes comparisons more effective and precise. Moreover, subjective user preferences also support the indexing plot and 19 out of 24 users favor it for visual comparison tasks. In a broader sense, it can be inferred that data transformations into percent domains might be generalizable to other visualization techniques for comparison tasks of multivariate data. In fact, for horizon graphs, indexing is applied for comparing multiple variables. But in contrast to line plots, multivariate horizon graphs use juxtaposition rather than superimposition.

Apart from the empirical results presented, an evaluation framework was developed to automate and ease the process of empirical studies for interactive information visualization prototypes, particularly if they are built with *prefuse*.

7.9 FUTURE WORK: EMPIRICAL STUDIES & PROTOTYPE IMPROVEMENTS

Ideas for future work mainly concern the two areas of refined empirical studies as well as improving the visualization prototype. Considering the first area, future studies in this area should examine more interactive ways to set the indexing point dynamically. This could further increase the performance of the indexing plot. Particularly, questions like how the indexing point influences the test results and how setting the indexing point could be enhanced, should be answered. Moreover, the influence of different aspect ratios and slopes have not been considered in our study and should be examined in connection to that.

Regarding possible improvements of the prototype, a range of different measures might be taken. First, the process of selecting a specific date as indexing point is too time consuming and not interactive enough currently. Second, horizontal gridlines should be included. Third, the introduction of reference lines that represent important values of the y-axis might be advantageous for the indexing plot. A horizontal line at the indexing value of 100% would be an example for such a reference line. This principle can be further applied by additional reference lines for important values (e.g., +/- 150%, etc.). Third, line plots which use a logarithmic scale for the y-axis could display reference lines for certain percent values because a constant percental change has always the same gradient in the log scale. This would aid the user to better estimate percent changes on the plot. Apart from that, the evaluation framework that has been implemented to guide and automate the test procedure should be further generalized to be more flexible and easy to use. Particularly, other types of questions should be added and also qualitative results should be recordable.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers of our manuscript for their thoughtful comments and suggestions that were crucial to improve the quality of the paper. Furthermore, thanks to our internal reviewers Paolo Federico and Alexander Rind for their feedback and comments as well as Peter Filzmoser for his support regarding statistical analysis.

REFERENCES

Andrienko, N. and Andrienko, G. (2006). *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer.

Beattie, V. and Jones, M. J. (2002). The Impact of Graph Slope on Rate of Change Judgments in Corporate Reports. *Abacus*, 38(2):177–199.

Bertin, J. (1967). *Sémiologie graphique: Les diagrammes, les réseaux, les cartes*. Gauthier-Villars, Paris, France.

Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press.

Bissantz, N. (2008). Do managers have to ride rabid tigers? http://blog.bissantz.com/logarithmic-rabid-tiger-1. Created at: Sept 19, 2008, Accessed at: Dec 1, 2009.

Cleveland, W. S. (1993). *Visualizing Data*. Hobart Press, Summit, NJ.

Cleveland, W. S. and McGill, R. (1984). Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79:531–554.

Few, S. (2009). *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.

Heer, J. and Agrawala, M. (2006). Multi-Scale Banking to 45 Degrees. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):701–708.

Heer, J., Card, S. K., and Landay, J. A. (2005). prefuse: a toolkit for interactive information visualization. In *Proc. of Conference on Human Factors in Computing Systems (CHI '05)*, pages 421–430. ACM.

Heer, J., Kong, N., and Agrawala, M. (2009). Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations. In *Proc. of Conference on Human Factors in Computing Systems (CHI '09)*, pages 1303–1312. ACM.

Lee, J. Y., Elmasri, R., and Won, J. (1998). An Integrated Temporal Data Model Incorporating Time Series Concept. *Data and Knowledge Engineering*, 24(3):257–276.

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

Wainer, H. (1997). *Visual revelations: graphical tales of fate and deception from Napoleon Bonaparte to Ross Perot*. Copernicus, New York, NY.

CHAPTER 8

Comparative Evaluation of an Interactive Time-Series Visualization that Combines Quantitative Data with Qualitative Abstractions

Wolfgang Aigner, Alexander Rind, and Stephan Hoffmann

Abstract · *In many application areas, analysts have to make sense of large volumes of multivariate time-series data. Explorative analysis of this kind of data is often difficult and overwhelming at the level of raw data. Temporal data abstraction reduces data complexity by deriving qualitative statements that reflect domain-specific key characteristics. Visual representations of abstractions and raw data together with appropriate interaction methods can support analysts in making their data easier to understand. Such a visualization technique that applies smooth semantic zooming has been developed in the context of patient data analysis. However, no empirical evidence on its effectiveness and efficiency is available. In this paper, we aim to fill this gap by reporting on a controlled experiment that compares this technique with another visualization method used in the well-known KNAVE-II framework. Both methods integrate quantitative data with qualitative abstractions whereas the first one uses a composite representation with color-coding to display the qualitative data and spatial position coding for the quantitative data. The second technique uses juxtaposed representations for quantitative and qualitative data with spatial position coding for both. Results show that the test persons using the composite representation were generally faster, particularly for more complex tasks that involve quantitative values as well as qualitative abstractions.*

Keywords · Evaluation/methodology.

This article originally appeared as [Aigner et al., 2012]:

Aigner, W., Rind, A., and Hoffmann, S. (2012). Comparative Evaluation of an Interactive Time-Series Visualization that Combines Quantitative Data with Qualitative Abstractions. *Computer Graphics Forum*, 31(3):995–1004. The definitive version is available at http://diglib.eg.org/.

8.1 INTRODUCTION

Modern data collection systems produce huge amounts of quantitative data across different application domains such as medicine or finance. Especially in the medical domain there is awareness that it is important to support decision-making in real-time environments like intensive care units. It can be difficult for the clinicians to make accurate decisions, particularly when the decisions are based on multiple clinical parameters [Farrington, 2011]. The traditional monitoring of patients is a process where vital signs are measured with sensors and the raw quantitative values are shown on an electronic display or trigger an alarm in a severe condition. Line plots, scatter plots, or bar charts are typical representations to display time-oriented quantitative data. But these representations lack the possibility to display interpretations derived from a-priori or associated knowledge about the data to support the clinician in making quick decisions.

The term *data abstraction* was originally introduced by Clancey [1985] in his proposal on heuristic classification. In general, its objective is *"[...] to create an abstraction that conveys key ideas while suppressing irrelevant details"* [Thomas and Cook, 2005, p. 86] and to use qualitative values, classes, or concepts, rather than raw data, for further analysis or visualization processes [Combi et al., 2010; Lin et al., 2007]. This helps in coping with the amount and complexity of data. To arrive at suitable data abstractions, several tasks must be conducted, including selecting relevant information, filtering out unneeded information, performing calculations, sorting, and clustering. The abstraction of raw time-series data to a sequence of intervals of meaningful qualitative levels and its representation on a patient monitor can make interpretation of patient data faster and more reliable [Miksch et al., 1996].

In Bade et al. [2004] several interactive visualization techniques are presented that enable the users to view a large volume of time-oriented data at several levels of detail and abstraction, ranging from a broad overview to the fine structure. A major part of this work focused on a visualization method for qualitative abstractions and the associated quantitative time-oriented data, which we will refer to as "STZ" (SemanticTimeZoom) throughout the paper. To support the user in exploring the data and to capture as much qualitative and quantitative information as possible on a limited display space, different representation levels for abstractions of time-oriented data are provided (see Fig. 8.1): The lowest visual information resolution level only presents the qualitative abstractions of the underlying quantitative values as colored horizontal bars over a period of time (Fig. 8.1, top), similar to *LifeLines* [Plaisant et al., 1996]. The visual representation for the next level enhances the previous one by using different heights for the bars. The next step combines the qualitative representations with a more detailed quantitative representation (*hybrid representation*) using a line plot with color-coded areas under the curve. In the last step, the quantitative data is emphasized while qualitative abstractions are

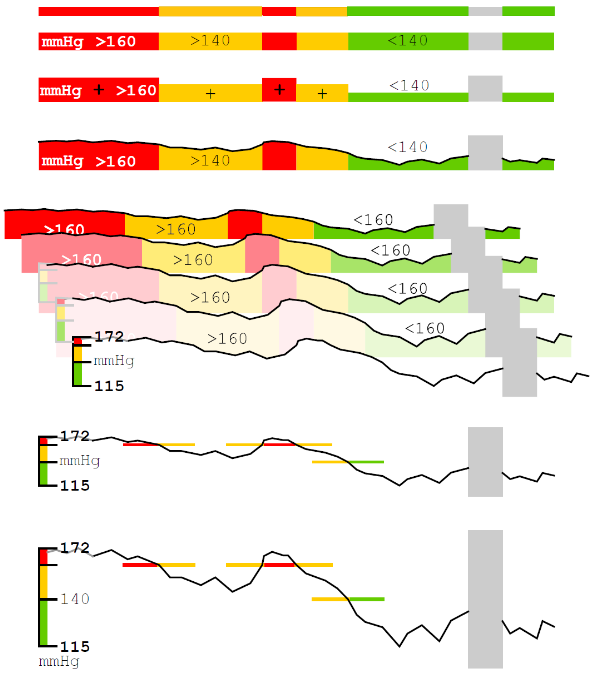


FIGURE 8.1 – **Smoothly integrated visualization of qualitative abstractions and quantitative data at different zoom levels** [Bade et al., 2004]. The representation depends on the available vertical display space, which is assigned interactively by the user.

shown by colored lines at level crossings (Fig. 8.1, bottom). Switching between these levels is achieved via a smoothly integrated semantic zoom functionality. Furthermore, semantic zooming concepts have also been introduced for the horizontal (time) axis using distortion and simplified boxplot representations. However, in the context of our work we focus on semantic zooming on the vertical (value) axis, which connects quantitative and qualitative data.

Although the concept of the STZ visualization technique appears very promising, it has not yet been evaluated. It has become crucial for researchers to present actionable evidence of measurable benefits to encourage widespread adoption of novel visualization techniques [Plaisant, 2004]. In other words, they need to show that the visualizations are fulfilling their proposed aims and meet the expectations and needs of users. To fill this gap we provide empirical evidence on the effectiveness and efficiency of STZ, which we collected in a comparative user study.

In the next section, we will present related visualization methods capable of representing qualitative abstractions together with quantitative data and evaluations conducted so far. Following that, the hypotheses and user tasks of our controlled experiment will be introduced in Section 8.3. In Section 8.4, the experiment design will be explained. We will report on the results of the experiment in Section 8.5 and discuss them in Section 8.6. Finally, we will provide a conclusion and give directions for future work in Section 8.7.

8.2 RELATED WORK

Visualization of time-series data is a prominent research area [Aigner et al., 2011; Shneiderman, 1996]. In this context, STZ tackles two research challenges: First, to convey meaningful information at higher abstraction levels and second, to show as many variables as possible on limited screen space. Next, we present a number of visualization methods that are related to these challenges.

For the *Graphical Summary of Patient Status* the axis of quantitative data is split to five severity ranges and scaled linearly within each range [Powsner and Tufte, 1994]. Thus, clinically significant displacements can easily be spotted and compared between heterogeneous variables. However, the distorted scale makes it hard to read quantitative trends and slopes. *KNAVE-II* (Knowledge-based Navigation of Abstractions for Visualization and Explanation) is a framework for interactive visualization, interpretation, and exploration of time-oriented clinical data [Shahar et al., 2006]. It supports on-the-fly interpretation of time-oriented clinical data using a distributed knowledge-based temporal abstraction mediator for the computation of qualitative abstractions. The main part of the KNAVE-II interface consists of the data-browsing panels, which either show raw quantitative data as line plots or qualitative abstractions that are the result of the temporal abstraction process represented as LifeLines [Plaisant et al., 1996] in different vertical positions. In addition, statistics for the data can be displayed on each panel. *LiveRAC* is a system for interactive visual exploration of large collections of network devices time-series data [McLachlan et al., 2008]. It provides a semantic zoom technique with different visual representations for the data at varying display space and user focus. But unlike STZ it presents data in a grid with rows representing network devices and columns presenting metrics or alarms of these devices. For each cell in the grid a qualitative severity level is abstracted from the raw data and is color-coded as the hue of the cell background. Quantitative data are shown as a line plot, which is reduced to a sparkline [Tufte, 2006] or faded out as the cell becomes smaller. It may also aggregate cells in order to show more network devices than pixels are available. Thus, LiveRAC differs from STZ primarily by showing only one qualitative abstraction for the complete observed time frame and no changes of qualitative abstractions over time.

Alternatively to qualitative abstractions, visualizations can also represent meaningful information about variables directly. For example, *MIVA* [Faiola and Newlon, 2011] and *TimeRider* [Rind et al., 2011] mark a variable's normal value range in the background as a colored area. However, these approaches only allow simple abstractions (e.g., a common threshold for all data items) and do not use semantic zoom functionality. *Lifelines2* allows interactive alignment and summarization of qualitative data but has no interface for quantitative data [Wang et al., 2008, 2009]. Furthermore, many visualization methods provide high data density for time-series without considering qualitative abstractions. For example, *sparklines* [Tufte, 2006] are small line plots with minimal axis and label information. Often they are no larger than a single line of text. For the *horizon graph* [Reijner, 2008] the quantitative value range is split into equally sized bands, which are wrapped and layered. The data are displayed as a line plot and the area under the line is colored to indicate the band. In Lam et al. [2007], a multiple visual information resolution interface (*VIR*) is presented that encodes a time-series either with color and spatial position or with color alone. Finally, interactive zooming is often facilitated by distortion-based techniques [Leung and Apperley, 1994].

Comparative Evaluations KNAVE-II was benchmarked against paper charts and electronic spreadsheets in a comparative evaluation study with physicians [Martins et al., 2008]. It demonstrated less errors and shorter answer time, especially for complex clinical tasks. However, KNAVE-II was the only system which calculated and displayed qualitative abstractions. A non-interactive prototype of MIVA was also experimentally compared to paper charts and yielded generally better performance [Faiola and Newlon, 2011]. Horizon graphs were evaluated against line plots and showed better user performance for smaller chart size [Heer et al., 2009]. In another user study [Javed et al., 2010], horizon graphs yielded faster completion times than line plots for discrimination tasks but slower times for maximum and slope tasks. Lam et al. [2007] experimentally compared different arrangements of their VIR. LiveRAC and TimeRider were evaluated in qualitative user studies, which are not directly related to this study. Likewise, Lifelines2 was evaluated in case studies and comparatively evaluated with a less feature-rich version [Wang et al., 2008]. An insight-based comparison of bioinformatics visualizations is reported in [Saraiya et al., 2005].

Selecting Comparable Techniques The only visualization technique also using interval-based qualitative abstractions for the visualization of data is the representation used in KNAVE-II. It displays quantitative data and qualitative abstractions separately and uses spatial position as visual encoding for both attributes. To provide a fair comparison, this visualization method was selected as comparison benchmark for STZ. We refer to it as “KNAVE” throughout this paper. A further advantage of using KNAVE for comparison is that there already is empirical evidence on its performance and our study complements this body of research.

8.3 HYPOTHESES AND TASKS

We assume that the STZ technique is effective and efficient for lookup and comparison tasks on qualitative abstractions as well as for lookup and comparison tasks on quantitative values linked to qualitative abstractions when investigating a single and multiple time-oriented variables. Thus, we formulate two hypotheses—the first hypothesis dealing with qualitative abstractions alone and the second hypothesis involving quantitative data that are linked to specified qualitative abstractions—and compare the STZ technique experimentally against the KNAVE technique:

H1: There is *no difference* between the STZ technique and KNAVE in correctness and time spent for tasks involving lookup and comparison of qualitatively abstracted data when investigating time-oriented variables.

H2: The STZ technique performs *better* than KNAVE in correctness and time spent for tasks involving lookup and comparison of quantitative data within specified qualitative abstractions when investigating time-oriented variables.

The first hypothesis implies that spatial position coding of qualitative abstractions in KNAVE does not outperform color-coding in STZ. It is based on perceptual theory that both, spatial position and color are preattentively processed [Ware, 2004]. In addition, Mackinlay [1986] ranked spatial position and color hue as the most effective graphical devices for communicating nominal data and color saturation or density is also ranked second behind spatial position for ordinal data. If the vertical display size is sufficient, STZ will combine color-coded abstractions with spatial position coded representations of quantitative data, which will further increase perception of ordinal ranking.

The second hypothesis is based on the *proximity compatibility principle* [Wickens and Carswell, 1995] which specifies that displays relevant to a common task or mental operation should be rendered close together in perceptual space. This implies that reduced vertical span between the representations of the qualitative and quantitative aspects of a variable in STZ should result in better user performance.

User Tasks Representative user tasks are an important precondition for a comparative evaluation. We developed 12 conceptual tasks (Table 8.1), which were abstracted from real-life tasks a medical expert would perform to make it possible for the test persons to perform the tasks repeatedly in the experiment. Tasks 1–6 (task block 1) are solely concerned with qualitative abstractions of the data (H1) and tasks 7–12 (task block 2) involve raw quantitative data associated to qualitative abstractions (H2). The tasks were structured using the task taxonomy of Andrienko and Andrienko [2006]. This taxonomy distinguishes between elementary tasks dealing with individual elements and synoptic tasks dealing with the dataset as a whole or its subsets. Furthermore, direct and indirect lookup tasks are differentiated, depending

	No.	Subtasks	Task description	Var.
H1: Lookup tasks	1	EIL	How many intervals of <qualitative level a> occur in <variable x>?	s
	2	EIL	Mark the first interval where both variables <x> and <y> are within <qualitative level a>.	m
	3	EIL	Mark the first appearance of an interval of <qualitative level a> in <variable x>.	s
H1: Comparison tasks	4	EDL + ECO	<Variable x>: Is the <first> qualitative level in <week> higher/lower/equal than the <third> qualitative level?	s
	5	EIL + SCA + SCO	Which variable has the longest lasting interval of <qualitative level a>?	m
	6	EIL + SCA + SCO	Which variable has the most occurrences of <qualitative level a>?	m
H2: Lookup tasks	7	EIL + SPS	Which variable is <rising> when <variable x> enters <qualitative level a> the <first> time.	m
	8	EIL + EDL	What value has the next measured data point of <variable x> when <variable y> enters in <qualitative level a> the first time in <week>?	m
	9	EIL + EDL	How many measured values contains <variable x> <first> interval of <qualitative level a>.	s
H2: Comparison tasks	10	EIL + EDL + ECO	<Variable x>: Which interval of <qualitative level> contains the largest number of measured values?	s
	11	EIL + EDL + ECO + ECO	Which variable has the <highest/lowest> measured value in its <first> interval of <qualitative level y>?	m
	12	EIL + EDL + ECO	Find the <highest > measured value in <variable x>'s <first> interval of <qualitative level a>.	s

Table 8.1 – **Conceptual tasks.** The second column states the subtask types referring to the task taxonomy by Andrienko and Andrienko [2006] using these abbreviations: EDL = Elementary direct lookup, EIL = Elementary inverse lookup, SCA = Synoptic behavior characterization, SPS = Synoptic pattern search, ECO = Elementary comparison, SCO = Synoptic behavior comparison. The last column states whether the task involved a single variable (s) or multiple variables (m). .

on whether time is given or needs to be obtained. These task types are listed in the second column of Table 8.1. Note that every task involves at least one elementary lookup subtask concerning qualitative abstractions to ensure the inclusion of the qualitative abstractions. The first three tasks in each block are representative for the lookup tasks and the last three tasks in each block represent comparison tasks, as they include at least one comparison subtask. Synoptic pattern search tasks are classified as lookup tasks in the second block, since synoptic pattern search tasks correspond to inverse lookup tasks on the synoptic level (cf. [Andrienko and Andrienko, 2006]).

8.4 EXPERIMENT DESIGN

To mitigate the impact of individual differences of the test persons and to increase the output of the test results, a within-subjects crossover design was selected. The following independent variables are included in this study:

- *Visualization technique (V)*: STZ and KNAVE
- *Type of data (TD)*: Qualitative and combined (quantitative values and qualitative abstractions)
- *Task number (T)*: 6 different tasks for each data type

For these, we measured the dependent variables task completion time and task correctness. The number of conditions in a factorial design is determined by the number and levels of the independent variables which results in $V \times TD \times T = 2 \times 2 \times 6 = 24$ different conditions. To increase robustness, every task is repeated, resulting in 48 different conditions for each participant who had to perform every task with both visualization techniques. To mitigate learning and fatigue effects, the order of the visualization type and dataset was counterbalanced. The order of the tasks was randomized, resulting in an alternation of tasks involving qualitative and combined data. Also influences of certain sequences of tasks, which could be answered faster due to similar data in question, should be avoided by the random task order.

8.4.1 *Apparatus*

Hardware All test persons conducted the experiment on the same laptop (MacBook Pro 4.1 with 2 GB RAM running OS X 10.6) with the same symmetrically shaped optical mouse. The application used for the experiment was maximized on a 15.4" LCD screen set to a resolution of 1440x900 pixels.

Visualized Data Every task is defined for two datasets. The data were extracted from the “Diabetes” dataset of Frank and Asuncion [2010] and consist of blood glucose measurements for diabetes patients. This dataset was selected because it contains multivariate time-series data. Moreover, meaningful qualitative abstractions for blood glucose measurements exist. Also, the qualitative abstraction of these data should be easy to understand for non-experts. The number of variables was limited to the maximum number of variables that can be reasonably displayed with the KNAVE prototype on a single screen without the need to scroll. Based on this, four different variables were shown in the experiment. This design decision was necessary to ensure a fair comparison of both techniques, although it limits STZ’s benefit of being capable to show a high data density and reduces the necessity

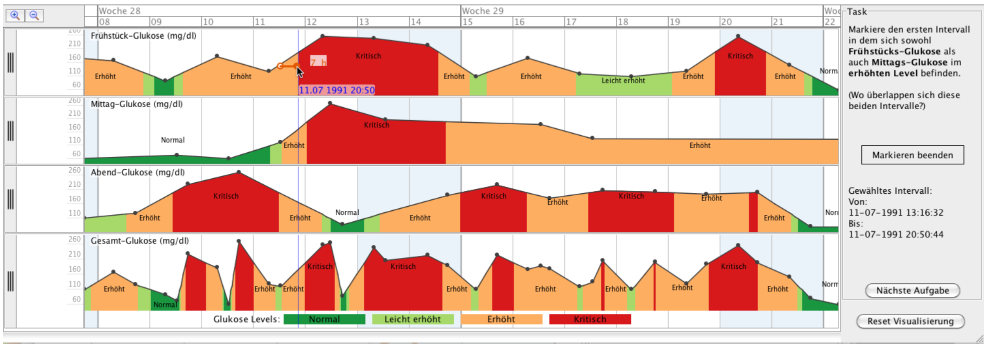


FIGURE 8.2 – **Screenshot of the STZ prototype during an evaluation session.** A legend at the bottom explains the color assignments of the qualitative levels. The task shown here was to find the first time-interval where both, pre-breakfast and pre-lunch blood glucose are in the elevated level (cf. Table 8.1, Task 2). The test persons had to select the time interval by dragging the mouse over the time axis to complete the task.

of semantic zooming. The datasets used in this study are subsets of these measurements from one patient over four weeks, and consist of the following variables: pre-breakfast, pre-lunch, pre-supper, and overall blood glucose. The associated qualitative abstractions can be grouped into four categories relating to hyperglycemia (normal; slightly elevated; elevated; critical).

Interactive Prototypes Fig. 8.2 and 8.3 show screenshots of the prototypes used during the evaluation sessions. In Fig. 8.2 qualitative and quantitative data for each variable are shown in a single diagram using color to visualize qualitative abstractions (STZ). The test persons could resize the panels containing the diagrams vertically using the mouse, which resulted in a change of the semantic zoom level. In Fig. 8.3 qualitative abstractions are shown in separate diagrams as bars in different vertical positions (KNAVE). Both prototypes offered the same interactions: tooltips for data points and qualitative intervals, resizable diagram panels and a mouse tracker showing the date and time of the current mouse position on the time axis.

To ensure repeatability of this study, all materials such as prototypes, datasets, and tasks as well as data collected on completion times and error rates can be found at http://ieg.ifs.tuwien.ac.at/research/semtimezoom/.

8.4.2 *Subjects*

20 test persons (12 male, 8 female) took part in the experiment. All test persons were volunteers, not color blind and had normal or corrected-to-normal vision. The average age of the test persons was 27 years and ranged between 22 and 30 years. Most of them were university students, with more

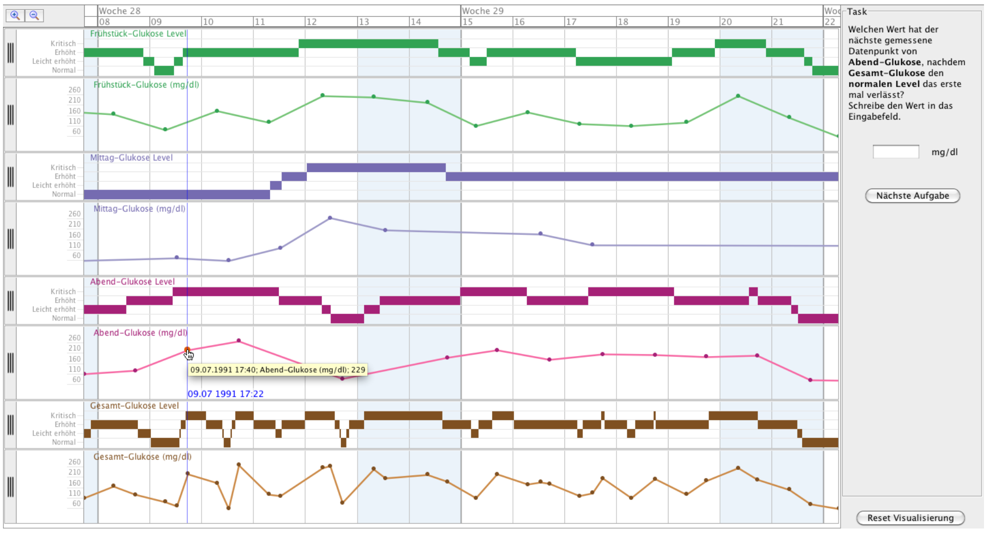


FIGURE 8.3 – **Screenshot of the prototype of the benchmark visualization technique based on the visual representations used in the KNAVE-II framework.** The task shown here was to find the value of the next measured data point of pre-supper blood glucose when overall blood glucose leaves the normal state the first time (cf. Table 8.1, Task 8). The test persons had to enter the read value of the data point (tooltip) in the text box on the right side of the window.

than half from the Faculty of Informatics. All test persons were at least in their second year of university or had reported to deal with graphical data representations frequently in their daily working routine.

8.4.3 *Procedure*

The test persons were given a short introduction to the purpose of the study before they were asked to fill out a questionnaire containing questions about personal information and self-assessment to computer experience and graph reading skills. Then they received a training session before each experiment round. A training session started with an introduction of the visualization technique and the corresponding interactions demonstrated by the test supervisor. After the introduction, the participants were instructed to solve three training tasks and encouraged to ask any questions before advancing to the actual experiment session.

The visualization prototypes were presented in full screen to avoid distraction and to offer enough space for the visualization itself along with the task description and answering possibilities. Before a task began, a pop-up message appeared with the task description, hiding the current visualization state. The participants were instructed to read the task instructions carefully and then press an "Ok" button. This initiated a task, causing the visualization to reappear and the timer to start for the given task. The task description was still visible on the right side of the visualization window (cf. Fig. 8.2 and 8.3). The tasks could be completed by either selecting an answer from a list, marking a time interval or entering a number in a text field, depending on the given task. A task was finalized by pressing the "Next Task" button, at which point the timer stopped and the task ended. Completion time and the provided input were recorded for each task. In addition, user interactions were recorded: tooltip activation, marking of time intervals, resizing of visualization panels, and in connection to that, semantic zoom level change when using the STZ prototype.

Every test person used one of the visualization techniques to master a set of 24 tasks with one dataset. Afterwards, they were offered the chance to take a break to stay alert and then continued to master another set of 24 tasks with the second visualization technique with another dataset. After the experiment, they were asked to decide which of the visualization techniques they personally preferred over the other one. The procedure and the estimated duration are outlined in Table 8.2. To verify these assumptions and to find flaws in the design, a pilot test has been carried out with one test person before recruiting the test persons for the actual experiment. The pilot test verified the experiment duration of about one hour and did not reveal any serious problems in the design.

8.4.4 *Analysis Approach*

The collected data were checked for possible errors and preprocessed for further statistical analysis. The goal was to find significant differences in task completion time and task correctness for a visualization technique with statistical hypothesis tests.

The influence of the used dataset on timing was tested using a paired t-test. It was found that the time samples violated the normality assumptions of the t-test, so the logarithm of the time was used. This also makes sense in order to

Activity	**Time [min]**
Pre-experiment Questionnaire	5.0
Training Round One	5.0
Experiment Round One	22.5
Training Round Two	5.0
Experiment Round Two	22.5
Post-experiment Questionnaire	5.0
Total	**65.0**

TABLE 8.2 – **Overview of experiment procedure**.

dampen the influence of overly long answering timings that would distort the results otherwise. The result of the t-test yielded no significant influence of the used dataset ($t(479) = 1.557$, $p = 0.12$, Cohen's $d = 0.071$). The correctness rate did not follow a normal distribution or log normal distribution, but a Mann-Whitney's U test also did not show a significant influence of the used dataset (the mean ranks of STZ and KNAVE were 23.8 and 25.2, respectively; $U = 271$, $Z = -0.37$, $p = 0.72$, $r = 0.053$). Therefore, the following analysis will not take into account which dataset was used for the experiment trials.

Even though the order of the visualization types was counterbalanced to reduce possible learning effects or fatigue, the carryover effect seems unbalanced for visualization types. On the one hand, the median of the completion time for STZ in the first round of the experiment was 17.0 seconds and in the second round 15.3 seconds resulting in an average improvement of 1.7 seconds. On the other hand, the median of the completion time for KNAVE in the first round was 24.9 seconds and in the second round 18.1 seconds with an average improvement of 6.8 seconds. Also, individual task completion times were considerably faster in the second round and therefore the completion times for each round needed to be compared separately, though the personal differences of the test persons will not be taken into account by this analysis. A Mann-Whitney's U test did show a significant influence of the experiment round on correctness (the mean ranks of STZ and KNAVE were 20.3 and 28.7, respectively; $U = 186.5$, $Z = -2.2$, $p < 0.05$, $r = 0.32$). Therefore, success rate data were also analyzed separately for the first and second round.

Task completion times and error rates (1−success rate) were aggregated for each task set according to Table 8.1 to test the hypotheses stated in Section 8.3. Completion times were summed up for each task set and error rates were calculated as ratio of errors to the overall number of tasks in a task set.

Completion times for the task sets were tested for normal or log-normal distributions using the Shapiro-Wilk test for every task set and visualization type. Task completion times tend to be right skewed [Sauro and Lewis, 2010]; presumably this is the reason that the completion times for all task sets follow a log-normal distribution. The logarithmized task set pairs of completion time also show equal variance for both visualization types in round 1 and 2, which was detected using an F-Test. As a result, a t-test could be used to test significant differences of the logarithmized completion times for the task sets and thereby testing the hypotheses.

Error rates for the task sets have been quite low with both visualizations and do not follow a normal or a log-normal distribution. Therefore, a non-parametric Mann-Whitney's U test was used to test the significance of error rates, since the error rate pairs for each task set did show equal variance for both visualizations. Due to the fact that error rates were very low for both techniques, we will mainly report on differences in task completion times in the results section.

Additionally, every individual task was tested for significant differences be-

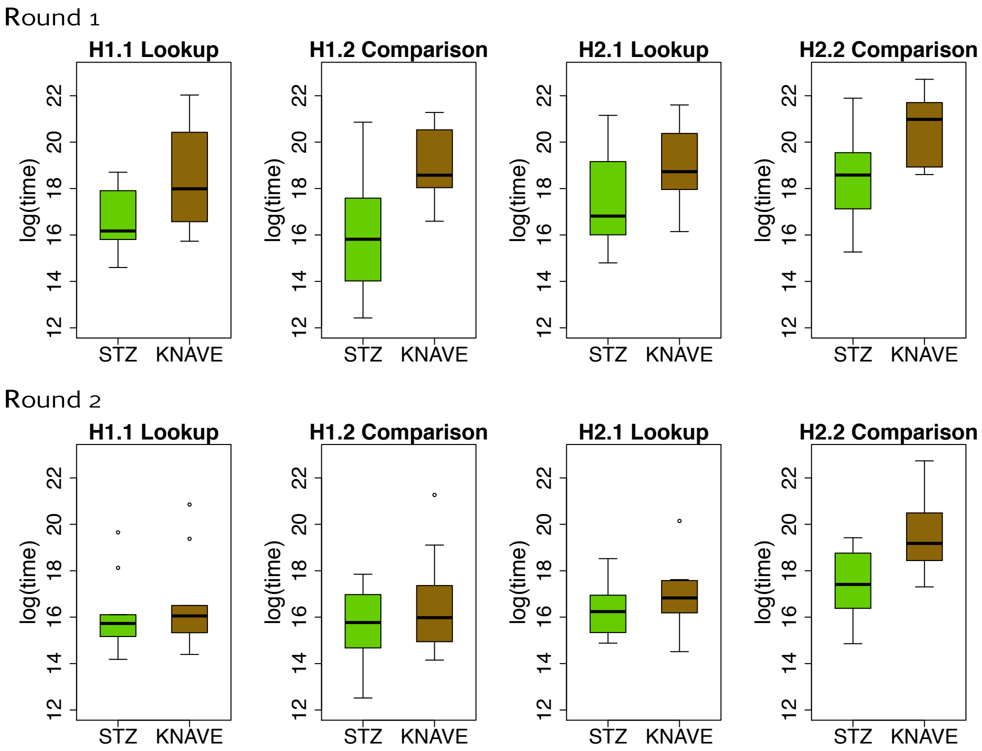


FIGURE 8.4 – **Box plots for completion times** per task set/round.

tween the visualization techniques. The task completion times had log-normal distributed completion times and equal variance between visualization types in each round, so t-tests could be used again for the analysis. Mann-Whitney's U tests were run to test the error rates for the individual tasks.

8.5 RESULTS

Fig. 8.4 and Table 8.3 show the completion time for each task set according to Table 8.1 and visualization type in the first and second round.

8.5.1 *Hypothesis 1 – Qualitative Data*

The first part of this analysis is focused on tasks involving only the qualitative abstractions of the data. In the case of this experiment, these tasks included questions regarding the temporal behavior, number of occurrences, and ordinal characteristics of episodes of normal, slightly elevated, elevated, and critical blood glucose measurements. Lookup tasks were analyzed separately from comparison tasks.

Completion times (s)	*Round 1* mean	std.dev.	*Round 2* mean	std.dev
H1 Lookup Tasks		**p = 0.020**		p = 0.286
STZ	116.3	32.8	99.3	30.0
KNAVE	157.6	60.6	108.8	39.8
H1 Comparison Tasks		**p = 0.003**		p = 0.170
STZ	107.5	46.0	94.4	24.0
KNAVE	169.2	53.0	110.9	44.4
H2 Lookup Tasks		p = 0.069		p = 0.190
STZ	138.3	61.4	100.1	18.8
KNAVE	160.3	46.1	111.6	36.8
H2 Comparison Tasks		**p = 0.009**		p = 0.005
STZ	154.2	47.9	125.9	34.0
KNAVE	222.1	51.4	176.7	47.2

TABLE 8.3 – **Completion times per task set and round.** Statistically significant results are marked in bold.

Lookup Tasks A one sided t-test showed a significant difference in completion time between the visualization types in round one ($t(15) = 2.2$, $p < 0.05$, Cohen's $d = 1.00$) with STZ outperforming KNAVE. In the second round no significant difference between both visualization types ($t(17) = 0.6$, $p = 0.29$, Cohen's $d = 0.26$) was found regarding completion time.

The error rates have an equal median for both visualization types in round one (8.3%) and two (0%). Consequently, no significant difference was found by a Mann-Whitney's U test between visualization types. Nevertheless, a learning effect is also evident in the error rates as the median is reduced from 8.3% to 0% in the second round.

Comparison Tasks The mean of completion times of the STZ users was about 1 minute lower than of the KNAVE users in the first round and 16.5 seconds lower in the second round. In the first round, a one sided t-test revealed a significantly faster completion time for test persons using the STZ technique ($t(16) = 3.16$, $p < 0.01$, Cohen's $d = 1.63$). Again, no significant difference was found on error rates depending on the visualization technique in both rounds.

Recap Hypothesis 1 expects that there is no difference in completion time and error rate for lookup and comparison tasks involving only qualitative data between STZ and KNAVE. This was confirmed for error rates, as there is no significant difference in both rounds and both task sets. However, it was observed that STZ performed significantly better than KNAVE in terms of completion time for both task sets in the first round, but no significant difference was found for the second round.

8.5.2 *Hypothesis 2 – Qualitative & Quantitative Data*

This part of the analysis investigates the completion time and error rates for tasks involving quantitative data mapped to specified qualitative abstractions. Again, lookup tasks will be discussed separately from comparison tasks.

Lookup Tasks In the first round, the mean completion time of the KNAVE users was 15% higher to master a lookup task than STZ users and 10% higher in the second round. The completion time was not found to be significantly faster for any visualization technique in the first and second round. Error rates did not show any significant differences. Interestingly, the mean of the errors rose in the second round compared to the first round with KNAVE. The medians of error rates are zero for both visualization types and rounds.

Comparison Tasks Comparison tasks involving both, qualitative and quantitative data seem to be the most complex tasks, which is also reflected in the longest task completion times. The test persons were 40% to 45% faster with the STZ visualization than with KNAVE. The completion time was significantly faster with STZ in both rounds: $t(18) = 1.8$, $p < 0.05$, Cohen's $d = 0.82$ (round 1) and $t(18) = 2.9$, $p < 0.01$ Cohen's $d = 1.29$ (round 2). Once more, the error rates were lower in the second round but the median is constantly zero for both rounds and visualizations.

Recap Hypothesis 2 proposes that the STZ visualization is more appropriate for tasks involving quantitative data within specified qualitative levels than the KNAVE visualization and should outperform the KNAVE visualization in terms of task completion time and error rate. This was confirmed regarding significantly shorter duration in both rounds for comparison tasks. Lookup tasks involving quantitative values did not have significant findings. The hypothesis was not confirmed regarding error rates, as no significant effect was found in both rounds for both task sets. Also, the error rates did not have a tendency to either visualization technique.

8.5.3 *Results on Individual Task Level*

In the first round, one-sided t-tests for every individual task revealed significantly faster completion times with STZ for tasks 1, 3, 5, 6, 9, 10, and 12 (cf. Table 8.1). Analysis of the completion times in the second round showed significant faster completion times for tasks 6, 7, 10, 11, and 12 with STZ. The only three tasks that were significantly faster in both rounds are task 6, 10, and 12, noteworthy all three tasks include comparison subtasks. In the first round of the experiment, every task had a faster mean completion time with STZ than with KNAVE, except for task 4. Also in the second round, task 4 had a longer mean duration with STZ. Mann-Whitney's U tests were run to evaluate the differences of error rates between the visualization techniques on individual tasks separately for each round. The tests did not reveal significant findings for any task in either round.

8.5.4 *User Preference*

After the test persons had finished both rounds of the experiment, they were asked to decide which of the visualization techniques they personally preferred over the other one. 19 out of 20 test persons preferred the STZ visualization technique. A Chi-square test revealed a significant difference for personal preference ($\chi^2 = 16.2$, $p < 0.001$).

8.5.5 *User Interactions*

The interaction log included activation of tooltips and resizing of visualization panels that trigger a representation mode change using the STZ prototype. The latter was intended to provide insight into which tasks needed a representation mode change. Although the test persons were encouraged to use this feature in the training session and got a demonstration on how to use it, it was barely used in the experiment session. A Mann-Whitney's U test on the number of tooltips needed for each task was used between visualization types. The test showed that KNAVE users needed significantly less tooltips for task 4, 8, and 7; STZ users needed significantly less tooltips for task 6.

8.6 DISCUSSION

While no significant difference of error rate could be found, the results of the analysis of task completion time showed that the STZ visualization technique, despite using 40% less display space in the initial experiment setting, outperforms the KNAVE technique for comparison tasks involving quantitative values mapped to qualitative abstractions. Additional analysis on individual task level has revealed that comparison tasks involving multiple variables were also performed significantly faster with STZ. The KNAVE technique did not show significantly faster completion times on any individual task number nor on any task group relating to the hypothesis. The only task that was on average mastered faster with KNAVE than with STZ was task number 4. This task is the only one concerning the ordinal characteristics of the qualitative abstractions, which are not immediately visible in STZ. It is also suspected that the task description was misleading for some test persons, explaining the rather high error rate in the first round with both visualization techniques.

The analysis of the interaction logs showed that the STZ visualization technique was more interaction-intensive than the KNAVE visualization technique, relating to the number of activated tooltips. This does not conflict with the idea of STZ as an interactive visualization technique, although the test persons did hardly ever use the semantic zoom feature. Despite the higher interaction activity for STZ there is no increase in completion times.

With respect to the test results, we believe that the combined visualization of the quantitative and qualitative aspects of a variable in one view excels especially for comparison tasks of quantitative values in defined qualitative abstractions. We attribute that mainly to the reduced distance between the different aspects for a variable. KNAVE requires the user's gaze to travel vertically between the diagrams that belong to the same variable to find the quantitative values that make up a distinct qualitative area. This difficulty would probably increase, if the diagrams were not grouped together by variable like in the KNAVE experiment setting in this study. This belief is also supported by the *proximity compatibility principle*, which specifies that displays relevant to a common task or mental operation (mental proximity) should be rendered close together in perceptual space (close display proximity) [Wickens and Carswell, 1995]. A second reason for the better performance of STZ over KNAVE is probably the use of distinct color hues for different qualitative abstractions. This can be backed by the fact that the features color hue and intensity are preattentively processed and "pop out" from their surroundings [Ware, 2004]. This advantage was also pointed out by several test persons after the experiment.

Limitations The error rate was rather low throughout both visualization types and all tasks. This indicates that the test persons were equally careful, regardless of the visualization technique. We also believe that the error rates were rather low because of the nature of the tasks, which did not require the test persons to estimate values, and the answers could be found straightforwardly. We are attributing the reason for the mistakes that have still been made mainly to glitches or misinterpretations of task descriptions.

Another limitation of the study was the relatively low number of subjects used in the experiment. Though the study was initially planned as a within-subject experiment, the analysis showed that the differences between the first and second round of the experiment were unbalanced according to the learning effect for task completion times and error rate. Possibly, the training sessions have been too short to understand the visualization techniques completely. As a result, the rounds had to be analyzed separately as a between-subject design for each round. Of course, this also reduced the size of the groups for each round to half of the initial group size of 20. A larger number of test persons would have improved the statistical power of the results and maybe resulted in clearer results. Furthermore, task number

4 showed an unusual behavior, both in completion time and error rate. The instructions for the test persons seem to have been confusing for some test persons and should have been explained more clearly.

The interaction logs revealed that the test persons hardly ever changed between the representation modes of the STZ technique (i.e. resizing of visualization panels). Thus, the experiment was in fact a comparison study between the hybrid-representation with filled qualitative regions used in STZ with the KNAVE visualization. Consequently, interactive compression, which is a major strength of the STZ technique, was not covered by the results. From the visualization design point of view, labels have been used in the colored regions of the STZ technique (cf. Fig. 8.2) but not for the LifeLines in KNAVE, because such labels are not used in the original technique of the KNAVE-II framework either. Nevertheless these labels may have introduced some advantage for the STZ technique. Likewise, the KNAVE technique used color to differentiate between variables (cf. Fig. 8.3), which again may have been an advantage for KNAVE.

8.7 CONCLUSION AND FUTURE WORK

We investigated a novel visualization technique (STZ) that is capable of displaying quantitative data and qualitative abstractions of time-oriented, multivariate data. It uses a combined representation of different visual encodings, whereas spatial position is used to encode the quantitative data and color-coding is used to display the related qualitative abstractions. In order to assess the effectiveness and efficiency of this technique, a comparative evaluation was performed with a related visualization technique (KNAVE-II) also using interval-based qualitative abstractions for the visualization of data. It displays the quantitative and qualitative data separately and uses spatial position as visual encoding for both attributes. An earlier experiment revealed significant differences in favor of KNAVE-II for the dependent variables task completion time, errors, and user preference when compared against paper charts and electronic spreadsheets. Our experiment showed that a combined visualization of quantitative and qualitative data using different visual encodings (STZ) performs at least equally than comparable techniques (KNAVE) and excels especially for more complex tasks. The combined visualization was also clearly preferred by the users. Although the evaluation was carried out in a context of patient data analysis, the results appear to be generalizable for other data with similar characteristics.

Implications Two major learnings of our research concern the usage of visual variables for heterogeneous, multivariate data and the spatial separation of views. First, the ranking of visual variables in [Mackinlay, 1986] implies that information encoded by spatial position is more accurately perceived than any other encoding such as color, size, or orientation. However, our results show that different visual encodings might be beneficial if different data types are to be combined (e.g. quantitative and qualitative). Moreover, color hue is very well suited for displaying nominal characteristics of the data. If it is necessary to additionally display the ordinal ranking of qualitative data, color intensity and brightness can be used to encode this ordinal ranking [Harrower and Brewer, 2003]. But in that case, the number of different variables that can be displayed reasonably is limited. Second, using spatially separated representations for different data types (e.g. qualitative and quantitative) requires more movement by the head and eyes, because the user has to look for potential targets in different places. Thus, combined displays following the *proximity compatibility principle* [Wickens and Carswell, 1995] and displaying all relevant information in one representation should be used for multilevel data, if possible. The evaluation presented in this work showed that a combined representation particularly excels for more complex tasks involving both lookup and comparison subtasks of qualitative and quantitative data.

Future Work We plan to run follow-up studies with larger number of variables that take advantage of the semantic zoom ability in STZ. Another aspect that has not been covered in this study is that the hybrid representation with filled qualitative regions used in STZ emphasizes higher quantitative values because of the larger colored areas below the curve. It should be investigated if this influences the identification of distinct qualitative levels. In parallel, it would be necessary to conduct experiments to find the optimal heights for the representation transitions in STZ. Further, insight-based evaluations should be carried out with domain experts in order to better assess the utility of the STZ technique in medical contexts.

ACKNOWLEDGEMENTS

This work was supported by the Centre for Visual Analytics Science and Technology (CVAST; #822746) funded by the Austrian Federal Ministry of Economy, Family and Youth in the exceptional Laura Bassi Centres of Excellence initiative. Many thanks to Natalia and Gennady Andrienko for their help in task categorization, Silvia Miksch for her support, and Theresia Gschwandtner for her feedback to our manuscript.

REFERENCES

Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of Time-Oriented Data*. Springer, London.

Andrienko, N. and Andrienko, G. (2006). *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, Berlin.

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting time-oriented data and information to a coherent interactive visualization. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 105–112. ACM.

Clancey, W. J. (1985). Heuristic Classification. *Artificial Intelligence*, 27(3):289–350.

Combi, C., Keravnou-Papailiou, E., and Shahar, Y. (2010). *Temporal Information Systems in Medicine*. Springer, Berlin.

Faiola, A. and Newlon, C. (2011). Advancing critical care in the ICU: a human-centered biomedical data visualization systems. In *Ergonomics and Health Aspects, Proc. HCII 2011*, LNCS 6779, pages 119–128. Springer.

Farrington, J. (2011). Seven plus or minus two. *Performance Improvement Quarterly*, 23(4):113–116.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Harrower, M. and Brewer, C. (2003). Colorbrewer.org: An online tool for selecting colour schemes for maps. *Cartographic Journal*, 40(1):27–37.

Heer, J., Kong, N., and Agrawala, M. (2009). Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 1303–1312. ACM.

Javed, W., McDonnel, B., and Elmqvist, N. (2010). Graphical perception of multiple time series. *IEEE Trans. Visualization and Computer Graphics*, 16(6):927–934.

Lam, H., Munzner, T., and Kincaid, R. (2007). Overview use in multiple visual information resolution interfaces. *IEEE Trans. Visualization and Computer Graphics*, 13:1278–1285.

Leung, Y. K. and Apperley, M. D. (1994). A review and taxonomy of Distortion-Oriented presentation techniques. *ACM Trans. Computer-Human Interaction*, 1(2):126–160.

Lin, J., Keogh, E. J., Wei, L., and Lonardi, S. (2007). Experiencing SAX: A Novel Symbolic Representation of Time Series. *Data Mining and Knowledge Discovery*, 15(2):107–144.

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Trans. Graphics*, 5:110–141.

Martins, S. B., Shahar, Y., Goren-Bar, D., Galperin, M., Kaizer, H., Basso, L. V., McNaughton, D., and Goldstein, M. K. (2008). Evaluation of an architecture for intelligent query and exploration of time-oriented clinical data. *Artificial Intelligence In Medicine*, 43:17–34.

McLachlan, P., Munzner, T., Koutsofios, E., and North, S. (2008). LiveRAC: interactive visual exploration of system management time-series data. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 1483–1492. ACM.

Miksch, S., Horn, W., Popow, C., and Paky, F. (1996). Context-sensitive and expectation-guided temporal abstraction of high-frequency data. In *Proc. Int. Workshop for Qualitative Reasoning (QR-96)*, pages 154–163. AAAI.

Plaisant, C. (2004). The challenge of information visualization evaluation. In *Proc. Working Conf. Advanced Visual Interfaces (AVI)*, pages 109–116. ACM.

Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. (1996). Lifelines: visualizing personal histories. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 221–227. ACM.

Powsner, S. M. and Tufte, E. R. (1994). Graphical summary of patient status. *Lancet*, 344(8919):386–389.

Reijner, H. (2008). The development of the horizon graph. In *Proc. Vis08 Workshop From Theory to Practice: Design, Vision and Visualization*.

Rind, A., Aigner, W., Miksch, S., Wiltner, S., Pohl, M., Drexler, F., Neubauer, B., and Suchy, N. (2011). Visually exploring multivariate trends in patient cohorts using animated scatter plots. In *Ergonomics and Health Aspects, Proc. HCII 2011*, LNCS 6779, pages 139–148. Springer.

Saraiya, P., North, C., and Duca, K. (2005). An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Trans. Visualization and Computer Graphics*, 11(4):443–456.

Sauro, J. and Lewis, J. R. (2010). Average task times in usability tests: what to report? In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 2347–2350. ACM.

Shahar, Y., Goren-Bar, D., Boaz, D., and Tahan, G. (2006). Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. *Artificial Intelligence In Medicine*, 38:115–135.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages (VL)*, pages 336–343.

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, Los Alamitos, CA, USA.

Tufte, E. R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, CT, USA.

Wang, T. D., Plaisant, C., Quinn, A. J., Stanchak, R., Murphy, S., and Shneiderman, B. (2008). Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 457–466. ACM.

Wang, T. D., Plaisant, C., Shneiderman, B., Spring, N., Roseman, D., Marchand, G., Mukherjee, V., and Smith, M. (2009). Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. *IEEE Trans. Visualization and Computer Graphics*, 15(6):1049–1056.

Ware, C. (2004). *Information Visualization - Perception for Design*. Morgan Kaufmann, San Francisco, CA, USA.

Wickens, C. D. and Carswell, C. M. (1995). The proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(3):473–494.

Acknowledgements

This thesis, and the many years of work behind it, would not have been possible without the help of many people. First and foremost I would like to thank my family – my wife Beatrix and my two lovely daughters, Emilia and Valentina, for their love, support, and understanding, especially during the last months when family time was a precious good.

I would like to express my gratitude to my mentor Silvia Miksch, who supported me throughout my scientific career and is a great role model for being an enthusiastic and motivated researcher. I would also like to thank A Min Tjoa, the head of the Institute of Software Technology and Interactive Systems (ISIS) at Vienna University of Technology, for ensuring a fruitful work environment and academic freedom for my research.

A big share of gratitude goes to my co-authors, collaborators, and colleagues, especially at the Laura Bassi Centre of Expertise for Visual Analytics Science and Technology (CVAST) @ Vienna University of Technology and the Department of Information and Knowledge Engineering (IKE) @ Danube University Krems. In particular, I would like to thank Bilal Alsallakh, Markus Bögl, Paolo Federico, Theresia Gschwandtner, Stephan Hoffmann, Katharina Kaiser, Tim Lammarsch, and Alexander Rind. Furthermore, I owe thanks to the representatives of the company partners with whom I worked in collaborative research projects, as well as the students and interns that were involved in my work. A special mention goes to the administrative and technical staff at Vienna University of Technology and Danube University Krems who tried to keep as much burdensome work away from me as possible.

Many thanks also to all my research collaborators with whom I had the pleasure to work througout the years. Specifically, I would like to thank my long-time collaborators Heidrun Schumann and Christian Tominski for their commitment and endurance, especially when working on our joint book project.

I am also grateful to the funding agencies who supported parts of this research, specifically the Austrian Science Fund (FWF), the Austrian Research Promotion Agency (FFG), the Austrian Federal Ministry of Economy, Family and Youth in the Laura Bassi Centres of Excellence initiative, the "FIT-IT Visual Computing" program of the Austrian Federal Ministry of Transport, Innovation and Technology, and the European Commission.

Thank you all!