Human-Computer Interaction Series

Wolfgang Aigner
Silvia Miksch
Heidrun Schumann
Christian Tominski

Visualization of Time-Oriented Data

Springer

Human-Computer Interaction Series

Human-computer interaction is a multidisciplinary field focused on human aspects of the development of computer technology. As computer-based technology becomes increasingly pervasive – not just in developed countries, but worldwide – the need to take a human-centered approach in the design and development of this technology becomes ever more important. For roughly 30 years now, researchers and practitioners in computational and behavioral sciences have worked to identify theory and practice that influences the direction of these technologies, and this diverse work makes up the field of human-computer interaction. Broadly speaking it includes the study of what technology might be able to do for people and how people might interact with the technology.

In this series we present work which advances the science and technology of developing systems which are both effective and satisfying for people in a wide variety of contexts. The Human-Computer Interaction series will focus on theoretical perspectives (such as formal approaches drawn from a variety of behavioral sciences), practical approaches (such as the techniques for effectively integrating user needs in system development), and social issues (such as the determinants of utility, usability and acceptability).

For further volumes:
www.springer.com/series/6033

Wolfgang Aigner • Silvia Miksch •
Heidrun Schumann • Christian Tominski

Visualization of Time-Oriented Data

Springer

Wolfgang Aigner
Vienna University of Technology
Vienna
Austria

Heidrun Schumann
University of Rostock
Rostock
Germany

Silvia Miksch
Vienna University of Technology
Vienna
Austria

Christian Tominski
University of Rostock
Rostock
Germany

ISSN 1571-5035
ISBN 978-0-85729-078-6 e-ISBN 978-0-85729-079-3
DOI 10.1007/978-0-85729-079-3
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2011929628

Cover design: VTeX UAB, Lithuania

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To our families.

Foreword

Time is central to life. We are aware of time slipping away, being used well or poorly, or of having a great time. Thinking about time causes us to reflect on the biological evolution over millennia, our cultural heritage, and the biographies of great personalities. It also causes us to think personally about our early life or the business of the past week. But thinking about time is also a call to action, since inevitably we must think about the future – the small decisions about daily meetings, our plans for the next year, or our aspirations for the next decades.

Reflections on time for an individual can be facilitated by visual representations such as medical histories, vacation plans for a summer trip, or plans for five years of university study to obtain an advanced degree. These personal reflections are enough justification for research on temporal visualizations, but the history and plans of organizations, communities, and nations are also dramatically facilitated by powerful temporal visual tools that enable exploration and presentation. Even more complex problems emerge when researchers attempt to understand biological evolution, geological change, and cosmic scale events.

For the past 500 years circular clock faces have been the prime representation for time data. These emphasize the twelve or 24-hour cycles of days, but some clocks include week-day, month or year indicators as well. For longer time periods, time lines are the most widely used, by historians as well as geologists and cosmologists.

The rise of computer display screens opened up new opportunities for time displays, challenging but not displacing the elegant circular clock face. Digital time displays are neatly discrete, clear and compact, but make time intervals harder to understand and compare. Increased use of linear time displays on computers has come with new opportunities for showing multiple time points, intervals, and future events. However, a big benefit of using computer displays is that multiple temporal variables can be shown above or below, or on the same time line. These kinds of overviews pack far more information in a compact space than was previously possible, while affording interactive exploration by zooming and filtering. Users can then see if the variables move in the same or opposite directions, or if one movement consistently precedes the other, suggesting causality.

These rich possibilities have payoffs in many domains including medical histories, financial or economic trends, and scientific analyses of many kinds. However, the design of interfaces to present and manipulate these increasingly complex and large temporal datasets has a dramatic impact on the users' efficacy in making discoveries, confirming hypotheses, and presenting results to others.

This book on Visualization of Time-Oriented Data by Aigner, Miksch, Schumann and Tominski represents an important contribution for researchers, practitioners, designers, and developers of temporal interfaces as it focuses attention on this topic, drawing together results from many sources, describing inspirational prototypes, and providing thoughtful insights about existing designs. While I was charmed by the historical review, especially the inclusion of Duchamp and Picasso's work, the numerous examples throughout the book showed the range of possibilities that have been tried – successes as well as failures. The analysis of the user tasks and interaction widgets made for valuable reading, provoking many thoughts about the work that remains to be done.

In summary, this book is not only about work that has been done, but it is also a call to action, to build better systems, to help decision makers, and to make a better world.

University of Maryland, *Ben Shneiderman*
February 2011

Preface

Time is an exceptional dimension. We recognize this every day: when we are waiting for a train, time seems to run at a snail's pace, but the hours we spend in a bar with a good friend pass by so quickly. There are times when one can wait endlessly for something to happen, and there are times when one is overwhelmed by events occurring in quick succession. Or it can happen that the weather forecast has predicted a nice and sunny summer day, but our barbecue has to be canceled due to a sudden heavy thunderstorm. Our perception of the world around us and our understanding of relations and models that drive our everyday life are profoundly dependent on the notion of time.

As visualization researchers, we are intrigued by the question of how this important dimension can be represented visually in order to help people understand the temporal trends, correlations, and patterns that lie hidden in data. Most data are related to a temporal context; time is often inherent in the space in which the data have been collected or in the model with which the data have been generated. Seen from the data perspective, the importance of time is reflected in established self-contained research fields around temporal databases or temporal data mining. However, there is no such sub-field in visualization, although generating expressive visual representations of time-oriented data is hardly possible without appropriately accounting for the dimension of time.

When we first met, we had all already collected experience in visualizing time and time-oriented data, be it from participating in corresponding research projects or from developing visualization techniques and software tools. And the literature had already included a number of research papers on this topic at that time. Yet despite our experience and the many papers written, we recognized quite early in our collaboration that neither we nor the literature spoke a common (scientific) language. So there was a need for a systematic and structured view of this important aspect of visualization.

We present such a view in this book – for scientists conducting related research as well as for practitioners seeking information on how their time-oriented data can be visualized in order to achieve the bigger goal of understanding the data and gaining valuable insights. We arrived at the systematic view upon which this book is based in the course of many discussions, and we admit that agreeing on it was not such an easy process. Naturally, there is still room for arguments to be made and for extensions of the view to be proposed. Nonetheless, we think that we have managed to lay the structural foundation of this area.

The practitioner will hopefully find the many examples that we give throughout the book useful. On top of this, the book offers a substantial survey of visualization techniques for time and time-oriented data. Our goal was to provide a review of existing work structured along the lines of our systematic view for easy visual reference. Each technique in the survey is accompanied by a short description, a visual impression of the technique, and corresponding categorization tags. But visual representations of time and time-oriented data are not an invention of the computer age. In fact, they have ancient roots, which will also be showcased in this book. A discussion of the closely related aspects of user interaction with visual representations and analytical methods for time-oriented data rounds off the book.

We now invite you to join us on a journey through time – or more specifically on a journey into the visual world of time and time-oriented data.

Vienna University of Technology &
University of Rostock,
February 2011

Wolfgang Aigner
Silvia Miksch
Heidrun Schumann
Christian Tominski

About the Authors

Wolfgang Aigner is assistant professor at the Institute of Software Technology & Interactive Systems at Vienna University of Technology, Austria. He received his PhD in computer science in 2006 for his work on "Visualization of Time and Time-Oriented Information: Challenges and Conceptual Design". From 2006 to 2010 he was research associate and deputy head of the Department of Information and Knowledge Engineering at Danube University Krems, Austria. Wolfgang has authored and co-authored several dozens of peer-reviewed articles and served as reviewer and program committee member for various scientific conferences, symposia, and workshops. From 2003 he was involved in a number of basic and applied research projects at national and international levels. Moreover, he participated in consulting projects and worked as a freelancer in the IT domain. His main research interests include visual analytics and information visualization, human-computer interaction (HCI), usability, and user-centered design.

Silvia Miksch has been head of the Information and Knowledge Engineering research group, Institute of Software Technology & Interactive Systems, Vienna University of Technology since 1998. From 2006 to 2010 she was professor and head of the Department of Information and Knowledge Engineering at Danube University Krems, Austria. In April 2010 she established the awarded Laura Bassi Centre of Expertise "CVAST – Center for Visual Analytics Science and Technology (Design, Interact & Explore)" funded by the Federal Ministry of Economy, Family and Youth of the Republic of Austria. Silvia has acquired, led, and has been involved in several national and international research projects. She has served on various program committees of international scientific conferences and was conference paper co-chair of the IEEE Conferences on Visual Analytics Science and Technology (IEEE VAST 2010, 2011) at VisWeek. She has more than 180 scientific publications and her main research interests are information visualization, visual analytics, plan management, and time.

Heidrun Schumann is a professor at the Institute for Computer Science at the University of Rostock, Germany, where she heads the Computer Graphics Research Group. Her research and teaching activities cover a number of topics related to computer graphics, particularly including information visualization, visual analytics, and rendering. More specifically, she is interested in the visualization of structures and multivariate data in space and time, in the design of scalable visual interfaces, and in terrain rendering techniques. Her current research projects are funded by public agencies and industry and span from fundamental research (e.g., scalable visualization methods and visual interfaces for smart environments) to applied research (e.g., computer graphics in the cockpit and visualization of bio-medical data). Heidrun is co-author of the first German textbook on visualization.

Christian Tominski is a lecturer and researcher at the Institute for Computer Science at the University of Rostock, Germany. Together with his colleagues from the Computer Graphics Research Group, Christian has authored and co-authored several articles on new visualization and interaction concepts as well as on aspects related to the software engineering of information visualization techniques. His current research interests are the visualization of multivariate data in time and space, the visualization of graph structures, and the promising opportunities of utilizing novel display and interaction devices for visualization. He is particularly interested in the role of interaction for the visual exploration and analysis of data. Christian developed a number of visualization systems and tools, including the LandVis system for spatio-temporal data, the VisAxes tool for time-oriented data, and the graph visualization system CGV.

Acknowledgements

Much of the information and insights presented in this book were obtained through the assistance of many of our students and colleagues. We would like to thank all of them for their feedback and discussions. Furthermore, we were kindly supported by our respective universities while developing and writing this book.

We particularly wish to thank all the authors of referenced material who gave feedback and provided images as well as the following publishers for their cooperation and unbureaucratic support in giving permission to reproduce material free of charge: Elsevier, Graphics Press, IEEE Press, International Cartographic Association, Springer, Third Millennium Press, and University of Chicago Press.

Valuable support was provided and insights gained within the VisMaster (Visual Analytics – Mastering the Information Age) project (a coordination action funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 225924) and the various research projects conducted within the research groups of the authors.

Contents

Chapter 1
Introduction

> Computers should also help us warp time, but the challenge here is even greater. Normal experience doesn't allow us to roam freely in the fourth dimension as we do in the first three. So we've always relied on technology to aid our perception of time.
>
> Udell (2004, p. 32)

Space and time are two outstanding dimensions because in conjunction they represent four-dimensional space or simply the world we are living in. Basically, every piece of data we measure is related and often only meaningful within the context of space and time. Consider for example the price of a barrel of oil. The data value of $129 alone is not very useful. Only if assessed in the context of where (space) and when (time) is the oil price valid and only then is it possible to meaningfully interpret the cost of $129.

Space and time differ fundamentally in terms of how we can navigate and perceive them. Space can in principle be navigated arbitrarily in all three spatial dimensions, and we can go back to where we came from. Humans have senses for perceiving space, in particular the senses of sight, touch, and hearing. Time is different; it does not allow for active navigation. We are constrained to the unidirectional character of constantly proceeding time. We cannot go back to the past and we have to wait patiently for the future to become present. And above all else, humans do not have senses for perceiving time directly. This fact makes it particularly challenging to visualize time – making the invisible visible.

Time is an important data dimension with distinct characteristics. Time is common across many application domains as for example medical records, business, science, biographies, history, planning, or project management. In contrast to other quantitative data dimensions, which are usually "flat", time has an inherent semantic structure, which increases time's complexity substantially. The hierarchical structure of granularities in time, as for example minutes, hours, days, weeks, and months, is unlike that of most other quantitative dimensions. Specifically, time comprises different forms of divisions (e.g., 60 minutes correspond to one hour, while 24 hours make up one day), and granularities are combined to form calendar systems

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_1,

(e.g., Gregorian, Julian, business, or academic calendars). Moreover, time contains natural cycles and re-occurrences, as for example seasons, but also social (often irregular) cycles, like holidays or school breaks. Therefore, time-oriented data, i.e., data that are inherently linked to time, need to be treated differently than other kinds of data and require appropriate visual and analytical methods to explore and analyze them.

The human perceptual system is highly sophisticated and specifically suited to spot visual patterns. Visualization strives to exploit these capabilities and to aid in seeing and understanding otherwise abstract and arcane data. Early visual depictions of time-series even date back to the 11th century. Today, a variety of visualization methods exist and visualization is applied widely to present, explore, and analyze data. However, many visualization techniques treat time just as a numeric parameter among other quantitative dimensions and neglect time's special character. In order to create visual representations that succeed in assisting people in reasoning about time and time-oriented data, visualization methods have to account for the special characteristics of time. This is also demanded by Shneiderman (1996) in his well-known task by data type taxonomy, where he identifies temporal data as one of seven basic data types most relevant for information visualization.

Creating good visualization usually requires good data structures. However, commonly only simple sequences of time-value-pairs $\langle(t_0, v_0), (t_1, v_1), \ldots, (t_n, v_n)\rangle$ are the basis for analysis and visualization. Accounting for the special characteristics of time can be beneficial from a data modeling point of view. One can use different calendars that define meaningful systems of granularities for different application domains (e.g., fiscal quarters or academic semesters). Data can be modeled and integrated at different levels of granularity (e.g., months, days, hours, and seconds), enabling for example value aggregation along granularities. Besides this, data might be given for time intervals rather than for time points, as for example in project plans, medical treatments, or working shift schedules. Related to this diversity of aspects is the problem that most of the available methods and tools are strongly focused on special domains or application contexts. Silva and Catarci (2000) conclude:

> It is now recognized that the initial approaches, just considering the time as an ordinal dimension in a 2D or 3D visualizations [sic], 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.
>
> Silva and Catarci (2000, p. 9)

The aim of this book is to present and discuss the multitude of aspects which are relevant from the perspective of visualization. We will characterize the dimension of time as well as time-oriented data, and describe tasks that users seek to accomplish using visualization methods. While time and associated data form a part of *what* is being visualized, user tasks are related to the question *why* something is visualized. *How* these characteristics and tasks influence the visualization design will be explained by several examples. These investigations will lead to a systematic categorization of visualization approaches. Because interaction techniques and analytical methods also play an important role in the exploration of and reasoning with time-oriented data, these will also be discussed. A large part of this book is devoted to a survey of existing techniques for visualizing time and time-oriented data. This survey presents self-contained descriptions of techniques accompanied by an illustration and corresponding references on a per-page basis.

Before going into detail on visualizing time-oriented data, let us first take a look at the basics and examine general concepts of information visualization.

1.1 Introduction to Visualization

Visualization is a widely used term. Spence (2007) refers to a dictionary definition of the term: *visualize* – to form a mental model or mental image of something. Visual representations have a long and venerable history in communicating facts and information. But only about twenty years have passed since visualization became an independent self-contained research field. In 1987 the notion of visualization in scientific computing was introduced by McCormick et al. (1987). They defined the term *visualization* as follows:

> Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights.
>
> McCormick et al. (1987, p. 3)

The goal of this new field of research has been to integrate the outstanding capabilities of human visual perception and the enormous processing power of computers to support users in analyzing, understanding, and communicating their data, models, and concepts. In order to achieve this goal, three major criteria have to be satisfied (see Schumann and Müller, 2000):

- expressiveness,
- effectiveness, and
- appropriateness.

Expressiveness refers to the requirement of showing exactly the information contained in the data; nothing more and nothing less must be visualized. *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. Finally, *appropriateness* involves a cost-value ratio in order to assess the benefit of the visualization process with respect to achieving a given task. While the value of a visual representation is not so easy to determine (see Van Wijk, 2006), cost is often related to time efficiency (i.e., the computation time spent) and space efficiency (i.e., the exploited screen space).

Expressiveness, effectiveness, and appropriateness are criteria that any visualization should aim to fulfill. To this end, the visualization process, above all else, has to account for two aspects: the data and the task at hand. In other words, we have to answer the two questions: "What has to be presented?" and "Why does it have to be presented?". We will next discuss both questions in more detail.

What? – Specification of the data

In recent years, different approaches have been developed to characterize data – the central element of visualization. In their overview article, Wong and Bergeron (1997) established the notion of *multidimensional multivariate data* as multivariate data that are given in a multidimensional domain. This definition leads to a distinction between *independent* and *dependent* variables. Independent variables define an n-dimensional domain. In this domain, the values of k dependent variables are measured, simulated, or computed; they define a k-variate dataset. If at least one dimension of the domain is associated with the dimension of time, we call the data *time-oriented data*.

Another useful concept for modeling data along cognitive principles is the *pyramid framework* by Mennis et al. (2000). At the level of data, this framework is based on three perspectives (also see Figure 3.29 on p. 63): *where* (location), *when* (time) and *what* (theme). The perspectives *where* and *when* characterize the data domain, i.e., the independent variables as described above. The perspective *what* describes what has been measured, observed, or computed in the data domain, i.e., the dependent variables as described above. At the level of knowledge, the *what* includes not only simple data values, but also objects and their relationships, where objects and relations may have arbitrary data attributes associated with them.

From the visualization point of view, all aspects need to be taken into account: The aspect *where* to represent the spatial frame of reference and to associate data values to locations, the aspect *when* to show the characteristics of the temporal frame of reference and to associate data values to the time domain, and the aspect *what* to represent individual values or abstractions of a multivariate dataset. As our interest is in time and time-oriented data, this book places special emphasis on the aspect *when*. We will specify the key properties of time and associated data in Chapter 3 and discuss the specific implications for visualization in Chapter 4.

Why? – Specification of the task

Similar to specifying the data, one also needs to know why the data are visualized and what tasks the user seeks to accomplish with the help of the visualization. On a very abstract level, the following three basic goals can be distinguished (see Ward et al., 2010):

- explorative analysis,
- confirmative analysis, and
- presentation of analysis results.

Explorative analysis can be seen as undirected search. In this case, no a priori hypotheses about the data are given. The goal is to get insight into the data, to begin extracting relevant information, and to come up with hypotheses. In a phase of *confirmative analysis*, visualization is used to prove or disprove hypotheses, which can originate from data exploration or from models associated with the data. In this sense, confirmative analysis is a form of directed search. When facts about the data have eventually been ascertained, it is the goal of the *presentation* step to communicate and disseminate analysis results.

These three basic visualization goals call for quite different visual representations. This becomes clear when taking a look at two established visualization concepts: filtering and accentuation. The aim of filtering is to visualize only relevant data and to omit less relevant information, and the goal of accentuation is to highlight important information. During explorative analysis, both concepts help users to focus on selected parts or aspects of the data. But filtering and accentuation must be applied carefully, because it is not usually known which data are relevant or important. Omitting or highlighting information indiscriminately can lead to misinterpretation of the visual representation and to incorrect findings. During confirmative analysis, filtering can be applied more easily as the data which is relevant, that is, the data that contribute to the hypotheses to be evaluated are usually known. Accentuation and de-accentuation are common means to enhance expressiveness and effectiveness, and to fine-tune visual presentations in order to communicate results and insight yielded by an exploratory or confirmative analysis process.

Although the presentation of results is very important, this book is more about visual analysis and interactive exploration of time-oriented data. Therefore, we will take a closer look at common analysis and exploration tasks. As Bertin (1983) describes, human visual perception has the ability to focus (1) on a particular element of an image, (2) on groups of elements, or (3) on an image as a whole. Based on these capabilities, three fundamental categories of interpretation aims have been introduced by Robertson (1991): *point*, *local*, and *global*. They indicate which values are of interest: (1) values at a given point of the domain, (2) values in a local region, or (3) all values of the whole domain. These basic tasks can be subdivided into more specific, concrete tasks, which are usually given as a list of verbal descriptions. Wehrend and Lewis (1990) define several such low-level tasks: identify or locate data values, distinguish regions with different values or cluster similar data, relate, compare, rank, or associate data, and find correlations and distributions. The task by data type taxonomy by Shneiderman (1996) lists seven high-level tasks that also include the notion of interaction with the data in addition to purely visual tasks:

- Overview: gain an overview of the entire dataset
- Zoom: zoom in on data of interest
- Filter: filter out uninteresting information
- Details-on-demand: select data of interest and get details when needed
- Relate: view relationships among data items
- History: keep a history of actions to support undo and redo
- Extract: allow extraction of data and of query parameters

Yi et al. (2007) further refine the aspect of interaction in information visualization and derive a number of categories of interaction tasks. These categories are organized around the user's intentions to interactively adjust visual representations to the tasks and data at hand. Consequently, a *show me* prefaces six categories:

- show me something else (explore)
- show me a different arrangement (reconfigure)
- show me a different representation (encode)
- show me more or less detail (abstract/elaborate)
- show me something conditionally (filter)
- show me related items (connect)

The *show me* tasks allow for switching between different subsets of the analyzed data (explore), different arrangements of visual primitives (reconfigure), and different visual representations (encode). They also address the navigation of different levels of detail (abstract/elaborate), the definition of data of interest (filter), and the exploration of relationships (connect).

In addition to the *show me* categories, Yi et al. (2007) introduce three further interaction tasks:

- mark something as interesting (select)
- let me go to where I have already been (undo/redo)
- let me adjust the interface (change configuration)

Mark something as interesting (select) subsumes all kinds of selection tasks, including picking out individual data values as well as selecting entire subsets of the data. Supporting users in going back to interesting data or views (undo/redo) is essential during interactive data exploration. Adaptability (change configuration) is relevant when a system is applied by a wide range of users for a variety of tasks and data types.

As we have seen, the purpose of visualization, that is, the task to be accomplished with visualization, can be defined in different ways. The above mentioned visualization and interaction tasks serve as a basic guideline to assist visualization designers in developing representations that effectively support users in conducting visual data exploration and analysis. In Chapter 4 we will come back to this issue and refine tasks with regard to the analysis of time-oriented data. The aspect of interaction will be taken up in Chapter 5.

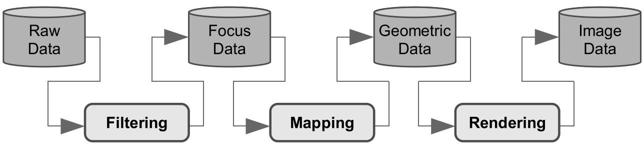
How? – The visualization pipeline

In order to generate effective visual representations, raw data have to be transformed into image data in a data-dependent and task-specific manner. Conceptually, raw data have to be mapped to geometry and corresponding visual attributes like color, position, size, or shape, also called *visual variables* (see Bertin, 1983; Mackinlay, 1986). Thanks to the capabilities of our visual system, the perception of visual stimuli is mostly spontaneous. As indicated earlier, Bertin (1983) distinguishes three levels of cognition that can be addressed when encoding information to visual variables. On the first level, elementary information is directly mapped to visual variables. This means that every piece of elementary information is associated with exactly one specific value of a visual variable. The second level involves abstractions of elementary information, rather than individual data values. By mapping the abstractions to visual variables, general characteristics of the data can be communicated. The third level combines the two previous levels and adds representations of further analysis steps and metadata to convey the information contained in a dataset in its entirety.

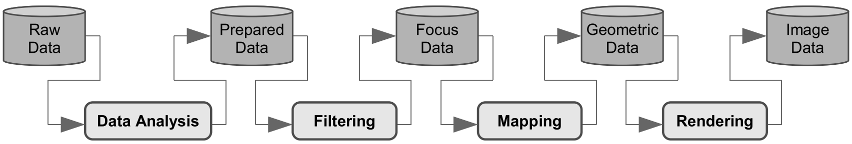
To facilitate generation of visual output at all three levels, a flexible mapping strategy is required. Such a strategy has been manifested as the so-called *visualization pipeline*, first introduced by Haber and McNabb (1990). The visualization pipeline consists of the three steps (see Figure 1.1(a)):

1. filtering,
2. mapping, and
3. rendering.

The filtering step prepares the raw input data for processing through the remaining steps of the pipeline. This is done with respect to the given analysis task and includes not only selection of relevant data but also operations for data enrichment or data reduction, interpolation, data cleansing, grouping, dimension reduction, and others. Literally, the mapping step maps the prepared data to appropriate visual variables. This is the most crucial step as it largely influences the expressiveness and effectiveness of the resulting visual representation. Finally, the rendering step generates actual images from the previously computed geometry and visual attributes. This general pipeline model is the basis for many visualization systems.



(a) Original variant (adapted from Haber and McNabb, 1990).



(b) Extended variant (adapted from dos Santos and Brodlie, 2004).

Fig. 1.1: The visualization pipeline.

The basic pipeline model has been refined by dos Santos and Brodlie (2004) in order to better address the requirements of higher dimensional visualization problems. The original filtering has been split up into two separate steps: data analysis and filtering (see Figure 1.1(b)). The data analysis carries out automatic computations like interpolations, clustering, or pattern recognition. The filtering step then extracts only those pieces of data that are of interest and need to be presented. In the case of large high-dimensional datasets, the filtering step is highly relevant because displaying all information will most likely lead to complex and overloaded visual representations that are hard to interpret. Because interests may vary among users, tasks, and data, the filtering step has to support the interactive refinement of filter conditions. Further input like the specific analysis task or hypothesis as well as application specific details can be used to steer the data extraction process.

In an effort to formally model the visualization process, Chi (2000) built upon the classic pipeline model and derived the *data state reference model*. This model reflects the stepwise transformation of abstract data into image data through several stages by using operators. While transformation operators transform data from one level of abstraction to another, within stage operators process the data only within the same level of abstraction (see Figure 1.2). This model broadens the capabilities of the visualization process and allows the generation of visual output at all of Bertin's levels. Different operator configurations lead to different views on the data, and thus, to comprehensive insight into the analyzed data. It is obvious that the selection and configuration of appropriate operators to steer the visualization process is a complex problem that depends mainly on the given visualization goal, which in turn is determined by the characteristics of the data and the task at hand.

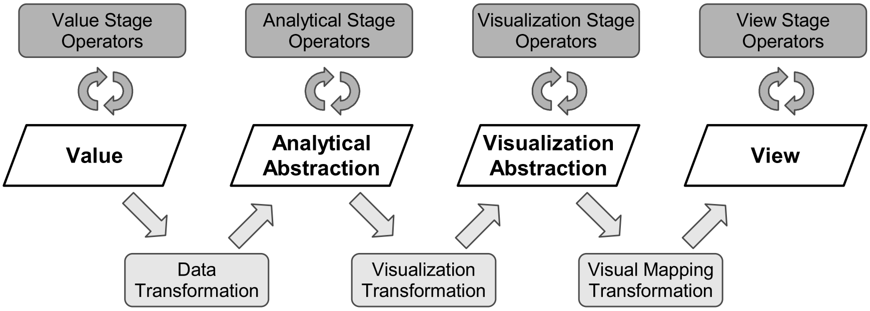


Fig. 1.2: The data state reference model (adapted from Chi, 2000).

The previous paragraphs may suggest that the image or view eventually generated by a visualization pipeline is an end product. But that is not true. In fact, the user controls the visualization pipeline and interacts with the visualization process in various ways. Views and images are created and adjusted until the user deems them suitable. Therefore, Card et al. (1999) integrate the user in their *information visualization reference model* (see Figure 1.3).

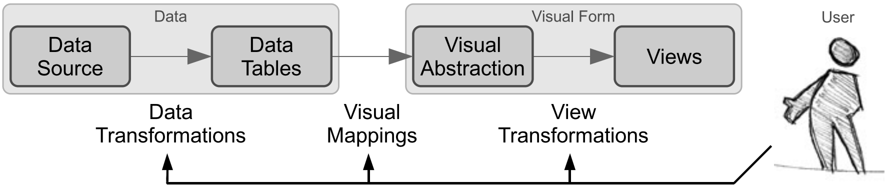


Fig. 1.3: The information visualization reference model (adapted from Card et al., 1999).

Having introduced the very basics of interactive visualization, we now move on to an application example. The goal is to illustrate a concrete visual representation and to demonstrate possible benefits for data exploration and analysis.

1.2 Application Example

Our particular example is in the domain of medicine. A considerable share of physicians' daily work time is devoted to searching and gathering patient-related information to form a basis for adequate medical treatment and decision-making. The amount of information is enormous and disorganized, and physicians might be overwhelmed by the information provided to them. Often, datasets comprise multiple variables of different data types that are sampled irregularly and independently from each other, as for example quantitative parameters (e.g., blood pressure or body temperature) and qualitative parameters (e.g., events like a heart attack) as well as instantaneous data (e.g., blood sugar measurement at a certain point in time) and interval data (e.g., insulin therapy from January to May 2010). Moreover, the data commonly originate from heterogeneous sources like electronic lab systems, hospital information systems, or patient data sheets that are not well integrated. Exploring such heterogeneous time-oriented datasets to get an overview of the history or the current health status of an individual patient or a group of patients is a challenging task.

Interactive visualization is an approach to representing a coherent view of such medical data and to catering for easy data exploration. In our particular example, an active discourse of the physician via interaction with the visual representation is of major importance since most static representations cannot satisfy task-dependent information needs seamlessly. In addition to presenting information intuitively, aiding clinicians in gaining new medical insights about patients' current health status, state changes, trends, or patterns over time is an important aspect.

VisuExplore is an interactive visualization tool for exploring a heterogeneous set of medical parameters over time (see Rind et al., 2010 and ↪ p. 231). VisuExplore uses multiple views along a common horizontal time axis to convey the different medical parameters involved. It is based on several well-known visualization meth-

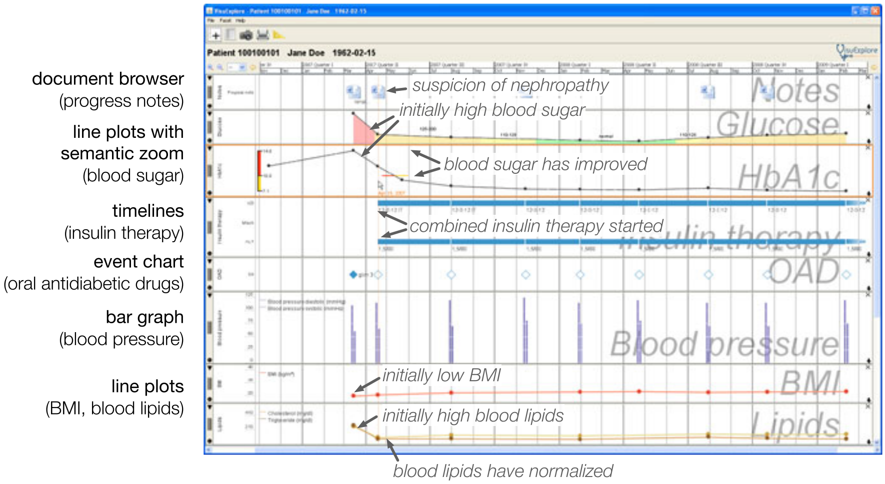


Fig. 1.4: Visualization of heterogeneous medical parameters of a diabetes patient.

ods, including line plots (↪ p. 153), bar graphs (↪ p. 154), event charts, and timelines (↪ p. 166), that are combined and integrated.

Figure 1.4 shows data of a diabetes patient over a period of two years and three months between November 2006 and March 2009. Beneath a panel that shows patient master data, eight visualization views are visible.

A document browser is placed on top that shows icons for medical documents, like for example diagnostic findings or x-ray images. In our example case, the document browser contains progress notes, as at the very beginning of treatment the physicians suspected renal failure. Next, a line plot with semantic zoom (see p. 112) is present which shows blood glucose values. Colored areas below the line provide qualitative information about normal (green), elevated (yellow), and high (red) value ranges which makes this semantic information easy to read. Below that, another line plot with semantic zoom functionality shows HbA1c (an indicator of a patient's blood glucose condition over the previous several weeks). In this case, more vertical space is devoted to the chart, thus allowing more exact readings of the values. Still, semantic information is added as color annotation of the y-axis, using small ticks to indicate when the variable's value crosses qualitative range boundaries (e.g., from critically high to elevated, as shown in the screenshot via a horizontal line that is colored red and yellow). Below the blood sugar values, there are two timeline charts showing the insulin therapy and oral anti-diabetic drugs. Insulin is categorized into rapid-acting insulin (ALT), intermediate-acting insulin (VZI), and a mixture of these (Misch). Details about brand name or dosage in free text are shown as labels that are located below the respective timeline. Oral anti-diabetic drugs are shown via an event chart below. There are also free text details about oral diabetes medication. The sixth view is a bar graph with adjacent bars for systolic and diastolic blood pressure. The bottom two views are line plots related to the body mass index (BMI) and blood lipids with two lines showing triglyceride and cholesterol values.

This arrangement has been chosen because it places views of medical tests directly above views of the related medical interventions. The height of some views has been reduced to fit on a single screen. This is possible because all information that is relevant for the physician's current task can still be recognized in this state.

The shown diabetes case is a 44-year-old patient with initially very high blood sugar values. From the interactive visual representations, several facts about the patient can be inferred as illustrated by the following insights that were gained by a physician using the VisuExplore system. The initially high blood sugar values were examined in detail via tooltips and showed exact values of 428 mg/dl glucose and 14.8% HbA1c. In addition, it can be seen in the bottom panel that blood lipid values are also high (256 mg/dl cholesterol, 276 mg/dl triglyceride). At the same time, the body mass index shown above is rather low (20.1). From the progress notes in the document browser it can be seen that the physician had the suspicion of a nephropathy. But these elevated values are also signs of latent autoimmune diabetes of adults, a special form of type 1 diabetes. After one month, blood sugar has improved (168 mg/dl glucose) and blood lipids have normalized. The patient switched to insulin therapy in a combination of rapid-acting insulin (ALT) and intermediate-acting insulin (VZI). Since April 2007, the insulin dosage has remained stable and concomitant medication is no longer needed. The patient's overall condition has improved through blood sugar management. Furthermore, the physician involved in the case study wondered about the very high HbA1c value of 11.9% in November 2006 and why diabetes treatment had only started four months later.

VisuExplore's interactive features allow physicians to get an overview of multiple medical parameters and focus on parts of the data. Physicians can add visualizations with one or more additional variables. They may resize and rearrange visualizations. Further, it is possible to navigate and zoom across the time dimension by dragging the mouse, by using dedicated buttons, or by selecting predefined views (e.g., last year). Moreover, the software allows the selecting and highlighting of data elements. Other time-based visualization and interaction techniques can extend the system to support special purposes. For example, a document browser shows medical documents (e.g., discharge letters or treatment reports) as document icons (e.g., PDF, Word) that physicians can click on if they want to open the document. VisuExplore integrates with the hospital information systems and accesses the medical data stored there.

This example demonstrated that visual representations are capable of providing a coherent view of otherwise heterogeneous and possibly distributed data. The integrative character also supports interactive exploration and task-specific focusing on relevant information.

1.3 Book Outline

With the basics of visualization and an application example, we have set the stage for the next chapters. Before going into detail about the contemporary visualization of time and time-oriented data, some inspiring and thought-provoking historical depictions and images from the arts are given attention in Chapter 2. The characteristics of time and data for modern interactive visualization on computers are the focus of Chapter 3. The actual visualization process, that is the transformation of abstract data to visual representations, will be discussed in Chapter 4, taking into account the key question words *what*, *why*, and *how* to visualize. In Chapters 5 and 6, we go beyond pure visualization methods and discuss cornerstones of interaction and analytical methods to support exploration and visual analysis. A major part of this book is devoted to a survey of existing information visualization techniques for time and time-oriented data in Chapter 7. Throughout the book we use the $\hookrightarrow$ symbol followed by a page number to refer the reader to a particular technique in the survey. A final summary along with a discussion of open challenges can be found in Chapter 8. Figure 1.5 provides a visual overview of the contents of the book.

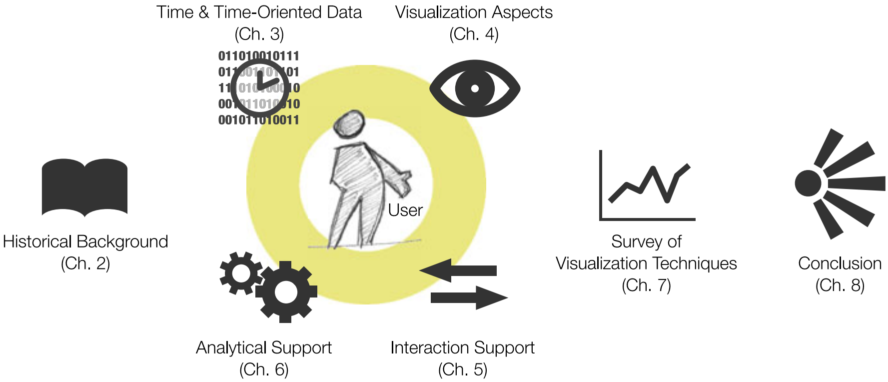


Fig. 1.5: Visual overview of the contents of the book.

Please refer to the companion website of the book for updates and additional resources including links to related material, visualization prototypes, and technique descriptions: http://www.timeviz.net.

References

Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, WI, USA. translated by William J. Berg.

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

Chi, E. H. (2000). A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 69–76, Washington, DC, USA. IEEE Computer Society.

dos Santos, S. and Brodlie, K. (2004). Gaining Understanding of Multivariate and Multidimensional Data through Visualization. *Computers & Graphics*, 28:311–325.

Haber, R. B. and McNabb, D. A. (1990). Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society, Los Alamitos, CA, USA.

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

McCormick, B. H., DeFanti, T. A., and Brown, M. D. (1987). Visualization in Scientific Computing. *Computer Graphics*, 21(6).

Mennis, J. L., Peuquet, D., and Qian, L. (2000). A Conceptual Framework for Incorporating Cognitive Principles into Geographical Database Representation. *International Journal of Geographical Information Science*, 14(6):501–520.

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, New York, NY, USA. ACM Press.

Robertson, P. K. (1991). A Methodology for Choosing Data Representations. *IEEE Computer Graphics and Applications*, 11(3):56–67.

Schumann, H. and Müller, W. (2000). *Visualisierung – Grundlagen und allgemeine Methoden*. Springer, Berlin, Germany.

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.

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.

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

Udell, J. (2004). Space, Time, and Data. *InfoWorld*, 26(26):32.

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

Ward, M., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters Ltd, Natick, MA, USA.

Wehrend, S. and Lewis, C. (1990). A Problem-Oriented Classification of Visualization Techniques. In *Proceedings of IEEE Visualization (Vis)*, pages 139–143, Los Alamitos, CA, USA. IEEE Computer Society.

Wong, P. C. and Bergeron, R. (1997). 30 Years of Multidimensional Multivariate Visualization. In Nielson, G. M., Hagen, H., and Müller, H., editors, *Scientific Visualization*, pages 40–62. IEEE Computer Society, Los Alamitos, CA, USA.

Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. (2007). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231.

Chapter 2
Historical Background

> There is a magic in graphs. The profile of a curve reveals in a flash a whole situation – the life history of an epidemic, a panic, or an era of prosperity. The curve informs the mind, awakens the imagination, convinces.
>
> Henry D. Hubbard in Brinton (1939, Preface)

Long before computers even appeared, visualization was used to represent time-oriented data. Probably the oldest time-series representation to be found in literature is the illustration of planetary orbits created in the 10th or possibly 11th century (see Figure 2.1). The illustration is part of a text from a monastery school and shows inclinations of the planetary orbits as a function of time.

To broaden the view beyond computer-aided visualization and provide background information on the history of visualization methods, we present historical and application-specific representations. They mostly consist of historical techniques of the pre-computer age, such as the works of William Playfair, Étienne-Jules Marey, or Charles Joseph Minard.

Furthermore, we will take the reader on a journey through the arts. Throughout history, artists have been concerned with the question of how to incorporate the dynamics of time and motion in their artworks. We present a few outstanding art movements and art forms that are characterized by a strong focus on representing temporal concepts. We believe that art can be a valuable source of inspiration; concepts or methods developed by artists might even be applicable to information visualization, possibly improving existing techniques or creating entirely new ones.

2.1 Classic Ways of Graphing Time

Representing business data graphically is a broad application field with a long tradition. William Playfair (1759–1823) can be seen as the protagonist and founding father of modern statistical graphs. He published the first known time-series depicting

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_2,

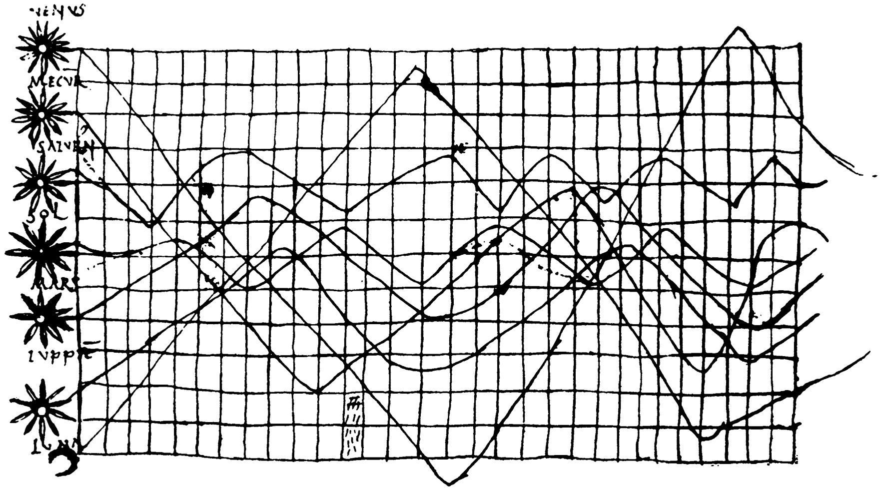


Fig. 2.1: Time-series plot depicting planetary orbits (10th/11th century). The illustration is part of a text from a monastery school and shows the inclinations of the planetary orbits as a function of time.
Source: Funkhouser (1936, p. 261). Used with permission of University of Chicago Press.

economic data in his *Commercial and Political Atlas* of 1786 (Playfair and Corry, 1786). His works contain basically all of the widely-known standard representation techniques (see Figures 2.2, 2.3, 2.5, and 2.4) such as the pie chart, the silhouette graph (↪ p. 175) , the bar graph (↪ p. 154), and the line plot (↪ p. 153).

In Figure 2.5 multiple heterogeneous time-oriented variables are integrated within a single view: the weekly wages of a good mechanic as a line plot, the price of a quarter of wheat as a bar graph, as well as historical context utilizing timelines (↪ p. 166). Playfair himself credits the usage of timelines to Joseph Priestley (1733–1804) who created a graphical representation of the life spans of famous historical persons divided into two groups of Statesmen and Men of Learning (see Figure 2.6). The usage of a horizontal line to represent an interval of time might seem obvious to us nowadays, but in Priestley's day this was certainly not the case. This is reflected in the fact that he devoted four pages of text to describe and justify his technique to his readers. A remarkable detail of Priestley's graphical method is that he acknowledged the importance of representing temporal uncertainties and provided a solution to deal with them using dots. Even different levels of uncertainty were taken into account, ranging from dots below lines to lines and dotted lines.

Even earlier than both Priestley and Playfair, Jacques Barbeu-Dubourg (1709–1779) created the earliest known modern timeline. His *carte chronographique* (Barbeu-Dubourg, 1753) consisted of multiple sheets of paper that were glued together and add up to a total length of 16.5 meters. A rare version of the chart is available at Princeton University Library where the paper is mounted on two rollers

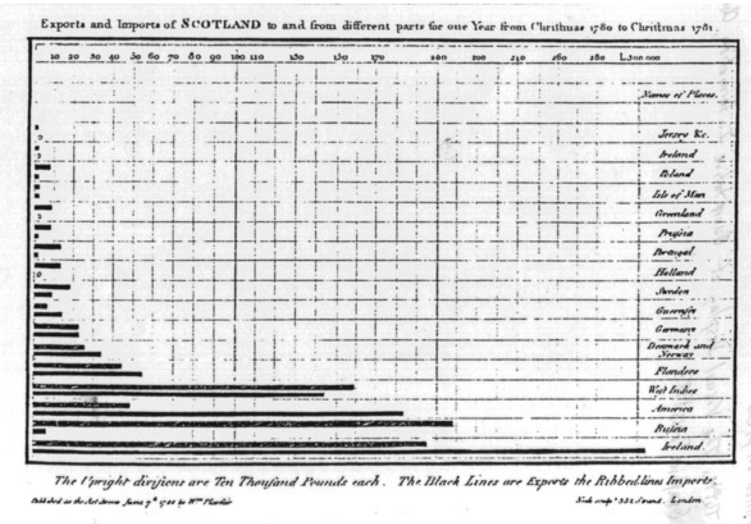


Fig. 2.2: Image from Playfair's *Commercial and Political Atlas* (1786) representing exports and imports of Scotland during one year via a bar graph.
Source: Playfair and Corry (1786).

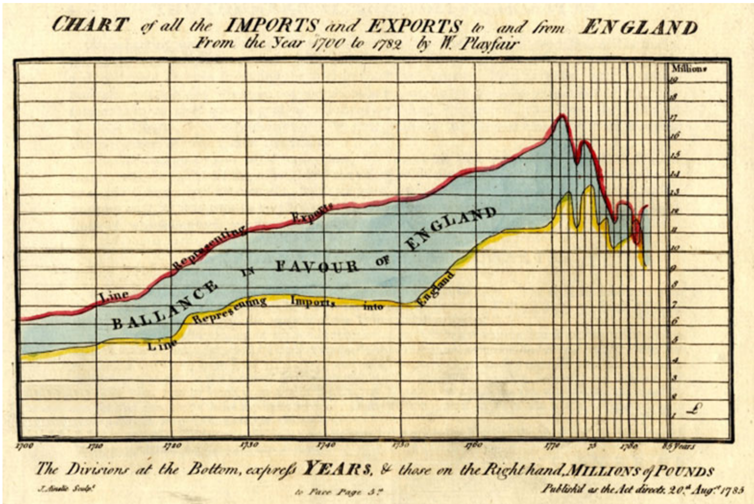


Fig. 2.3: Image from Playfair's *Commercial and Political Atlas* (1786) representing imports and exports of England from 1700 to 1782 via a line plot. The yellow line on the bottom shows imports into England and the red line at the top exports from England. Color shading is added between the lines to indicate positive (light blue) and negative (red; around 1781) overall balances.
Source: Playfair and Corry (1786).

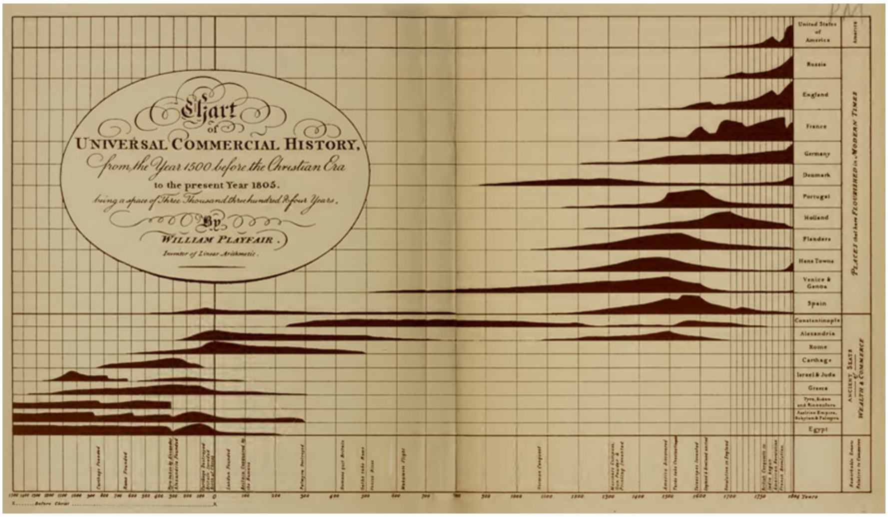


Fig. 2.4: Silhouette graph used by William Playfair to represent the rise and fall of nations over a period of more than 3000 years. A horizontal time scale is shown at the bottom that uses a compressed scale for the years before Christ on the left. Important events are indicated textually above the time scale. Countries are grouped vertically into Ancient Seats of Wealth & Commerce (bottom), Places that have Flourished in Modern Times (center), and America (top).
Source: Playfair (1805). Adapted from Brinton (1914).

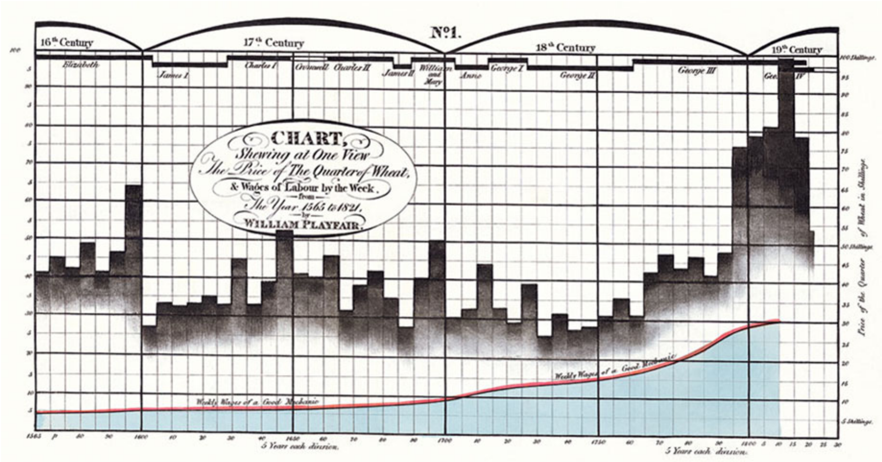


Fig. 2.5: Information rich chart of William Playfair that depicts the weekly wages of a good mechanic (line plot at the bottom), the price of a quarter of wheat (bar graph in the center), as well as historical context (timeline at the top) over a time period of more than 250 years.
Source: Playfair (1821).

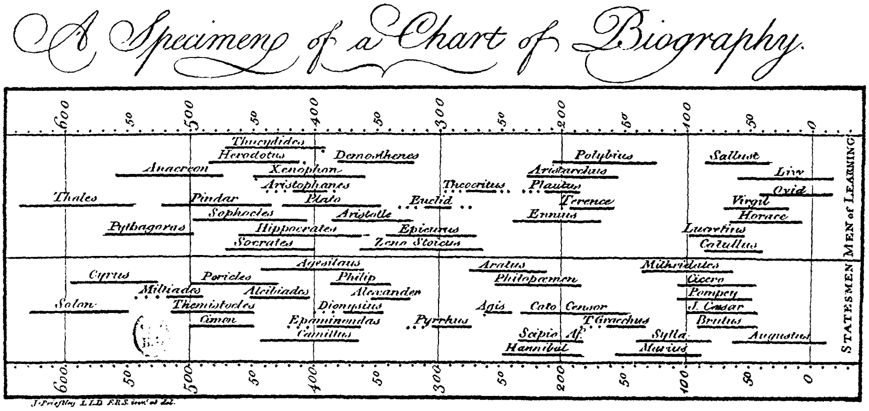


Fig. 2.6: Joseph Priestley's *chart of biography* that portrays the life spans of famous historical persons using timelines.
Source: Priestley (1765).

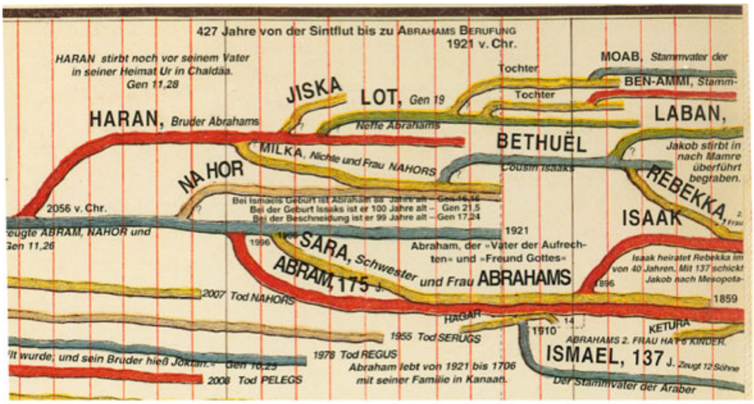
in a foldable case that can be scrolled via two handles (see Ferguson, 1991 for a detailed description).

Another prominent example of a graphical representation of historical information via annotated timelines is *Deacon's synchronological chart of universal history* which was originally published in 1890 and was drawn by Edmund Hull (see Figure 2.7). Various reprints and books extending the original historic facts to the present and adaptations for specialized areas like for example inventions and explorations can be found in the literature (e.g., Third Millennium Press, 2001).

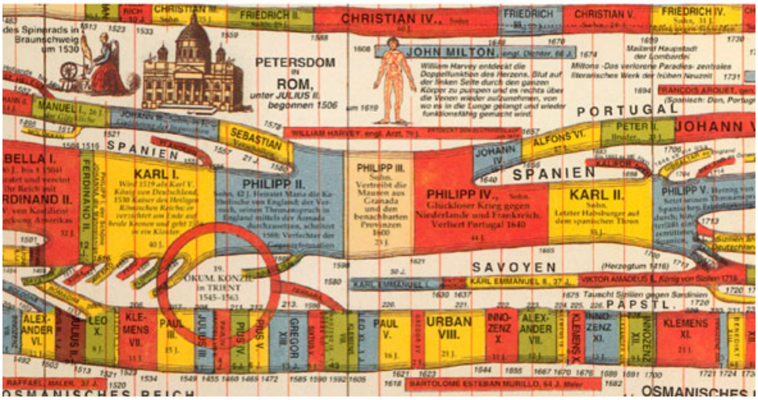
Charles Joseph Minard created a masterpiece of the visualization of historical information in 1861. His graphical representation of *Napoleon's Russian campaign* of 1812 is extraordinarily rich in information, conveying no less than six different variables in two dimensions (see Figure 2.8). Tufte (1983) comments on this representation as follows:

> It may well be the best statistical graphic ever drawn.
>
> Tufte (1983, p. 40)

The basis of the representation is a 2-dimensional map on which a band symbolizing Napoleon's army is drawn. The width of the band is proportional to the army's size; the direction of movement (advance or retreat) is encoded by color. Furthermore, various important dates are plotted and a parallel line graph shows the temperature over the course of time.



(a)



(b)

Fig. 2.7: Parts of Deacon's synchronological chart of universal history.
Source: Third Millennium Press (2001). © Third Millennium Press Ltd. Used with permission.

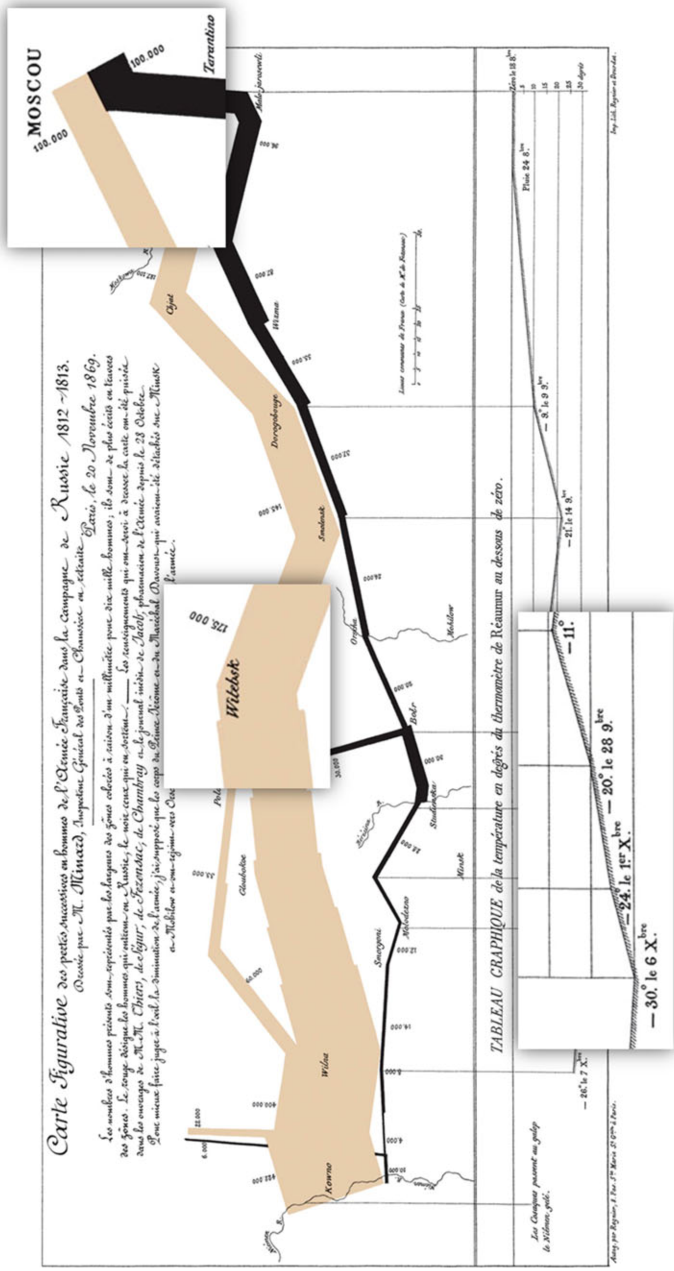


Fig. 2.8: Napoleon's Russian campaign of 1812 by Charles Joseph Minard (1861). A band visually traces the army's location during the campaign, whereby the width of the band indicates the size of the army and the color encodes advance or retreat of the army. Labels and a parallel temperature chart provide additional information.

Source: Adapted from http://commons.wikimedia.org/wiki/File:Minard.png; Retrieved Feb., 2011.

About 25 years after Minard portrayed Napoleon's march to Moscow, the prominent historic figure Florence Nightingale used a statistical graph to show numbers and causes of deaths over time during the Crimean War. When Nightingale was sent to run a hospital near the Crimean battlefields to care for British casualties of war, she made a devastating discovery: many more men were dying from infectious diseases they had caught in the filthy hospitals of the military than from wounds. By introducing new standards of hygiene and diet, and most importantly, by ensuring proper water treatment, deaths due to infectious diseases fell by 99% within a year. Florence Nightingale tediously recorded mortality data for two years and created a novel diagram to communicate her findings. Figure 2.9 shows two of these *rose charts*. This representation is also called *polar area graph* and consists of circularly arranged wedges that convey quantitative data. Unlike pie charts, all the segments of rose charts have the same angle. Bringing the data in this form clearly revealed the horrible fact that many more soldiers were dying because of preventable diseases they had caught in hospital than from wounds sustained in battle. Not only this fact was communicated, but also how this situation could be improved by the right measures; these can be seen from the left rose chart in Figure 2.9. Through this diagram, which was more a call to action than merely a presentation of data, she persuaded the government and the Queen to introduce wide-reaching reforms, thus bringing about a revolution in nursing, health care, and hygiene in hospitals worldwide.

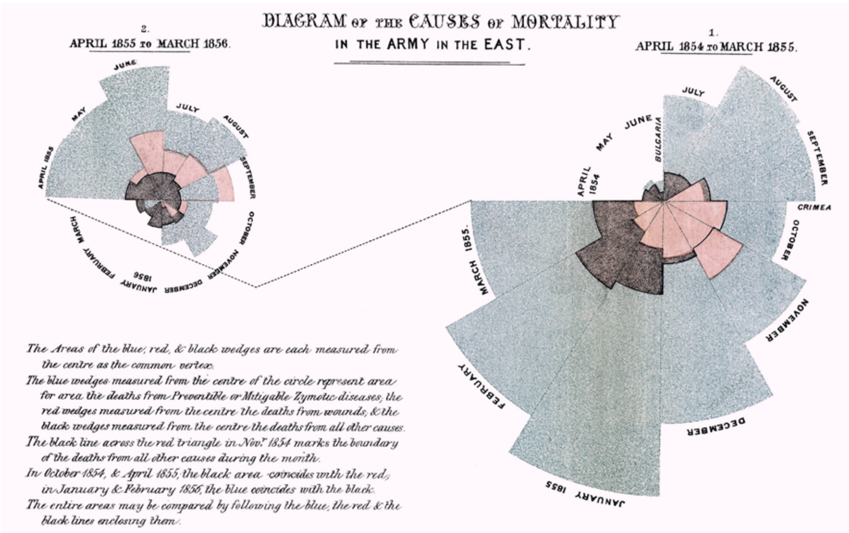


Fig. 2.9: Rose charts showing number of casualties and causes of death in the Crimean War by Florence Nightingale (1858). Red shows deaths from wounds, black represents deaths from accidents and other causes, and blue shows deaths from preventable infectious diseases soldiers caught in hospital. The chart on the right shows the first year of the war and the chart on the left shows the second year after measures of increased hygiene, diet, and water treatment had been introduced. *Source: http://en.wikipedia.org/wiki/File:Nightingale-mortality.jpg; Retrieved Feb., 2011.*

A quite different approach to representing historical information is the illustration of the *Cuban missile crisis* during the Cold War by Bertin (1983). The diagram shows decisions, possible decisions, and the outcomes thereof over time (see Figure 2.10). This representation is similar to the *decision chart* (↪ p. 159). Chapple and Garofalo (1977) provided an illustration of *Rock'n'Roll history* shown in Figure 2.11 that depicts protagonists and developments in the area as curved lines that are stacked according to the artists' percentage of annual record sales. The *ThemeRiver*™ technique (↪ p. 197) can be seen as further, more formal development of this idea.

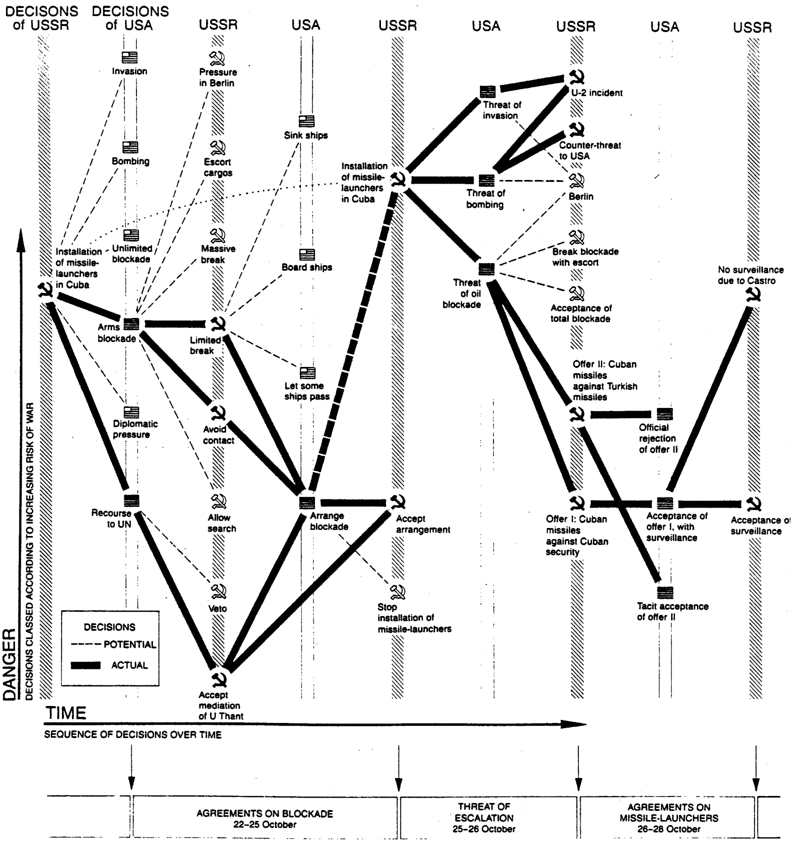


Fig. 2.10: Cuban missile crisis (threat level and decisions over time). The diagram shows decisions, possible decisions, and the outcomes thereof over time.
Source: Bertin (1983, p. 264). © 1983 by the Board of Regents of the University of Wisconsin System. Reprinted with permission of The University of Wisconsin Press.

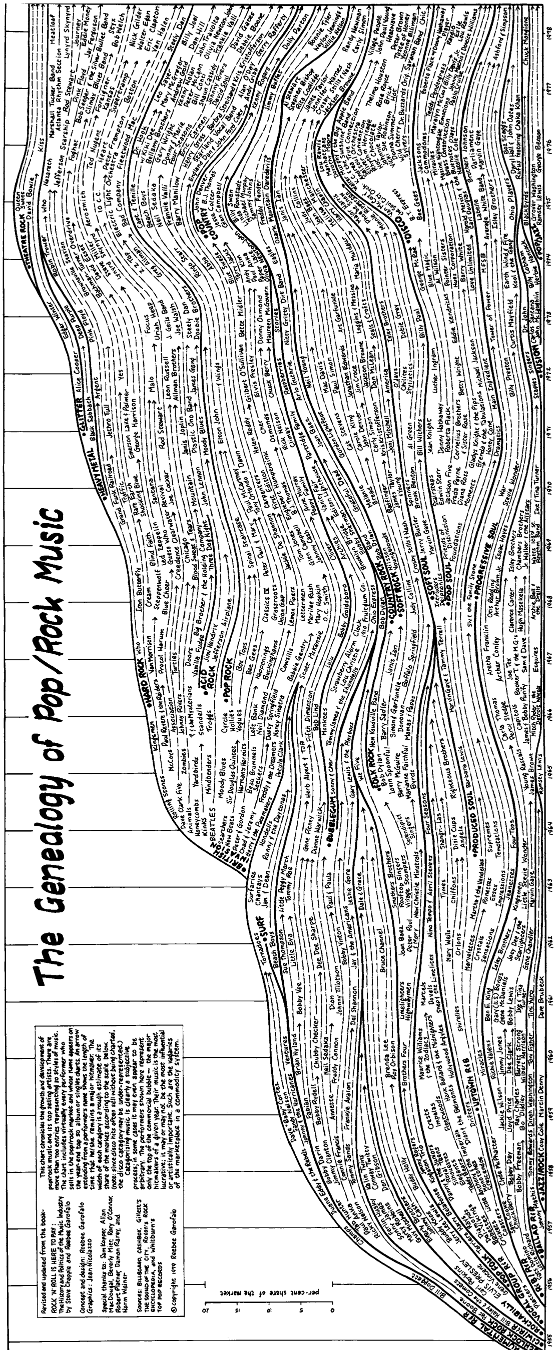


Fig. 2.11: Rock'n'Roll history that depicts protagonists and developments in the area as curved lines that are stacked according to the artists' percentage of annual record sales.
Source: Image courtesy of Reebee Garofalo.

With the advance of industrialization in the late 19th and early 20th century, optimizing resources and preparing time schedules became essential requirements for improving productivity. One of the main protagonists of the study and optimization of work processes was Frederick Winslow Taylor (1856–1915). His associate Henry Laurence Gantt (1861–1919) studied the order of steps in work processes and developed a family of timeline-based charts as intuitive visual representation to illustrate and record time-oriented processes (see Figures 2.12 and 2.13). Widely known as *Gantt charts* (↪ p. 167), these representations are such powerful analytical instruments that they are used nearly unchanged in modern project management.

Other interesting representations of work-related data can be seen in Figures 2.14 and 2.15. A record of hours worked per day by an employee is shown in Figure 2.14. It is interesting to note that both axes are used for representing different granularities of time, i.e., days on the horizontal axis and hours per day on the vertical axis. Figure 2.15 employs a radial layout of the time and allows a reading on multiple levels: the outer ring shows days without work and the inner rings show hours worked during the day, whereas the green areas indicate night hours.

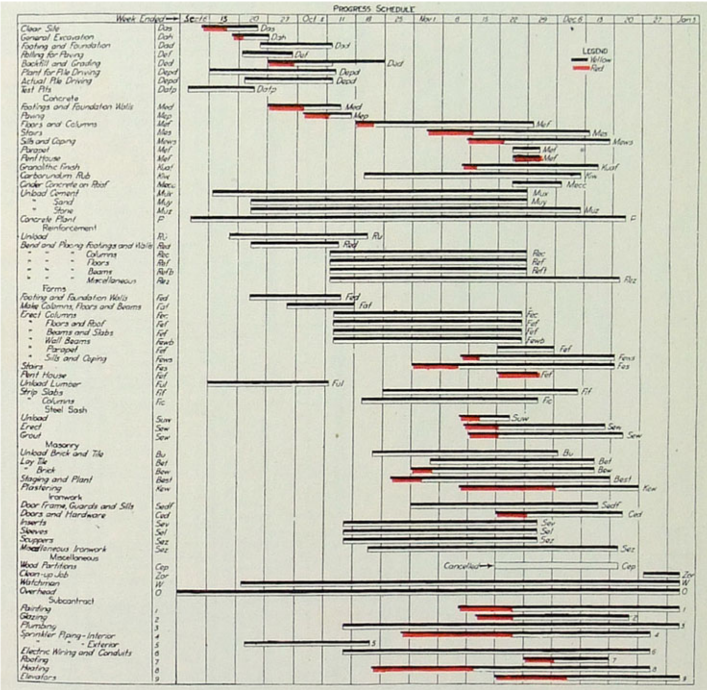


Fig. 2.12: Progress schedule based on the graphical method of Henry L. Gantt. Different work packages are shown as horizontal lines. Black lines indicate the planned timings; the actual quantity of work done is shown below in red.
Source: Brinton (1939, p. 259).

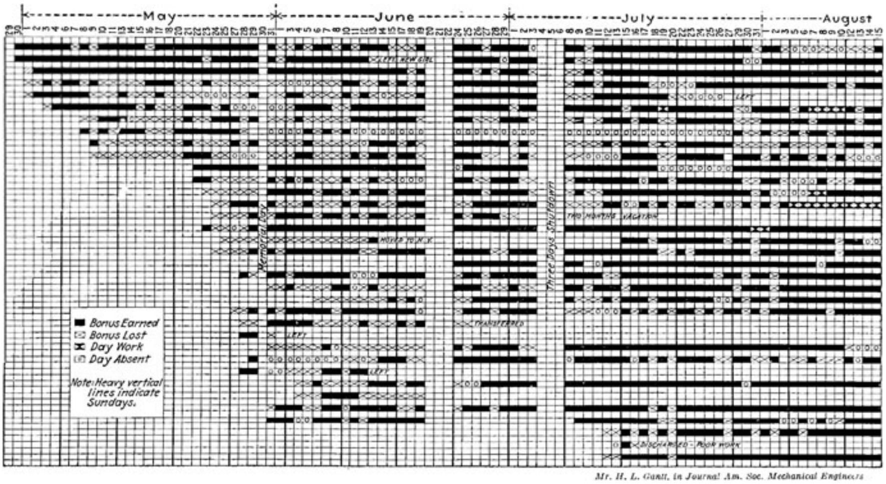


Fig. 2.13: Record of work carried out in one room of a Worsted Mill by Henry L. Gantt. Each row represents one worker and gives information about whether a bonus was earned and whether the worker was present.
Source: Brinton (1914, p. 52).

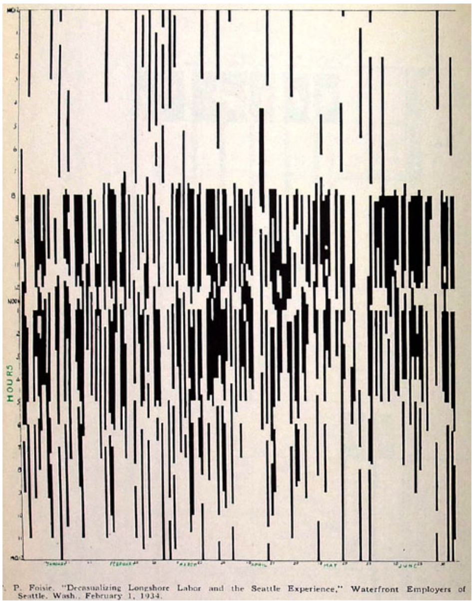


Fig. 2.14 Exact hours and days worked in 1929 by an employee at the Oregon ports. Days are mapped on the horizontal axis and hours per day worked are represented as bars on the vertical axis. The representation shows extreme irregularities in working hours.
Source: Brinton (1939, p. 250).

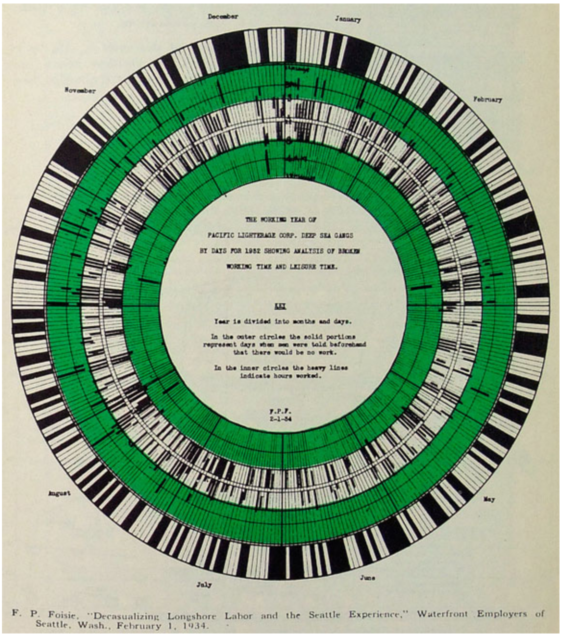


Fig. 2.15 An analysis of working time and leisure time in 1932. Uses a radial layout of time and allows a reading on multiple levels: the outer ring shows days without work and the inner rings show hours worked during the day, whereas the green areas indicate night hours. *Source: Brinton (1939, p. 251).*

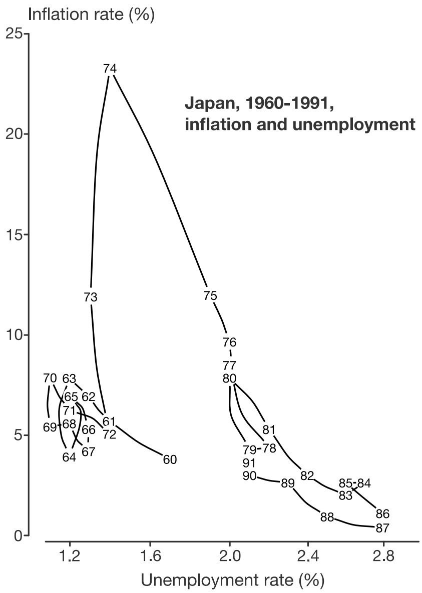


Fig. 2.16 Phillips curve. Unemployment rate (horizontal axis) is plotted against inflation rate (vertical axis). Each point in the plot corresponds to one year and is labeled accordingly. The markers of subsequent years are linked to create a visual trace of time. *Source: Adapted from Tufte (1997, p. 60). Used with permission of Graphics Press.*

A quite unique representation of economic data is the so-called *Phillips curve* – a 2D plot based on an economic theory that shows unemployment vs. inflation in a Cartesian coordinate system. In this representation, time is neither mapped to the horizontal nor the vertical axis, but is rather shown textually as labeled data points on the curve. This way, the dimension of time is slightly de-emphasized in favor of showing the relationship of two time-dependent variables (see Figure 2.16). Each year's combination of the two variables of unemployment rate and inflation rate leads to a data point in 2D space that is marked by the digits of the corresponding year. The markers of subsequent years are connected by a line resulting in a path over the course of time.

For representing positional changes within a set of elements, *rank charts* were already introduced in early statistical publications (see Figure 2.17). Elements are ordered according to their ranking and displayed next to each other in columns for different points in time. The positional change of individual elements is emphasized by connecting lines. This way, the degree of rank change is represented by the angles of the connecting lines, thus making big changes in rank stand out visually by the use of very steep lines.

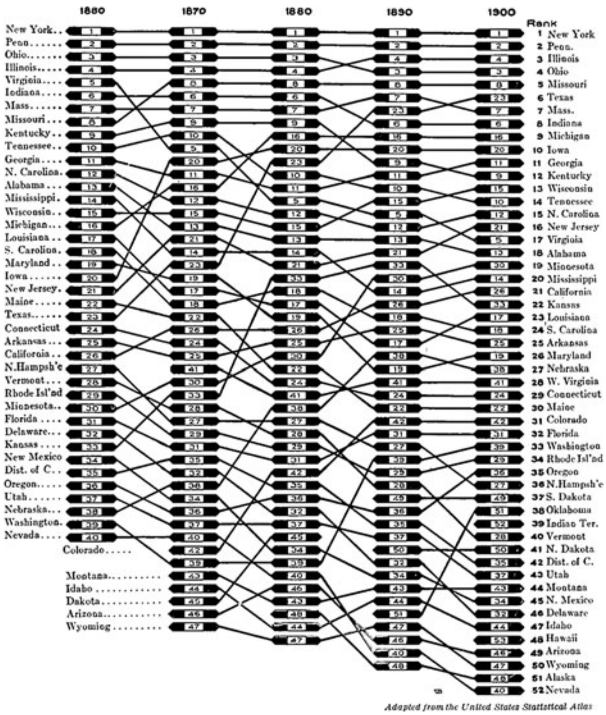


Fig. 2.17: Rank of states and territories in population at different census years from 1860 to 1900. *Source: Brinton (1914, p. 65).*

A remarkable representation of time-oriented information was created by Étienne-Jules Marey (1830–1904) in the 1880s (see Figure 2.18). It shows the train schedule for the track Paris to Lyon graphically. Basically, a 2D diagram is used which places the individual train stops according to their distance in a list on the vertical axis, while time is represented on the horizontal axis. Thus, horizontal lines are used to identify the individual stops and a vertical raster is used for timing information. The individual trains are represented by diagonal lines running from top-left to bottom-right (Paris–Lyon) and bottom-left to top-right (Lyon–Paris), respectively. The slope of the line gives information about the speed of the train – the steeper the line, the faster the respective train is travelling. Moreover, horizontal sections of the trains' lines indicate if the train stops at the respective station at all and how long the train stops. On top of that, the density of the lines provides information about the frequency of trains over time. This leads to a clear and powerful representation showing complex information at a glance while allowing for in-depth analysis of the data. Similar representations have also been used for the Japanese Shinkansen train line and the Javanese Soerabaja-Djokjakarta train line where the track's terrain profile is additionally shown.

Étienne-Jules Marey not only created the fabulous train schedule, but was also very interested in exploring all kinds of movement. Born in 1830 in France, he was a trained physician and physiologist. His interest in internal and external movements in humans and animals, such as blood circulation, human walking, horse gaits, or dragonfly flight, led to the decomposition of these movements via novel photography and representation methods (see Figures 2.19, 2.20, and 2.21). This photography method, which is called *chronophotography*, paved the way for the birth of modern film-making at the end of the nineteenth century.

Today, Marey is still a valuable source of inspiration. Reason enough to speak highly of him and his work:

> Tirelessly, this brilliant visionary stopped the passage of time, accelerated it, slowed it down to "see the invisible," and recreated life through images and machines.
>
> La maison du cinema and Cinematheque Francaise (2000)

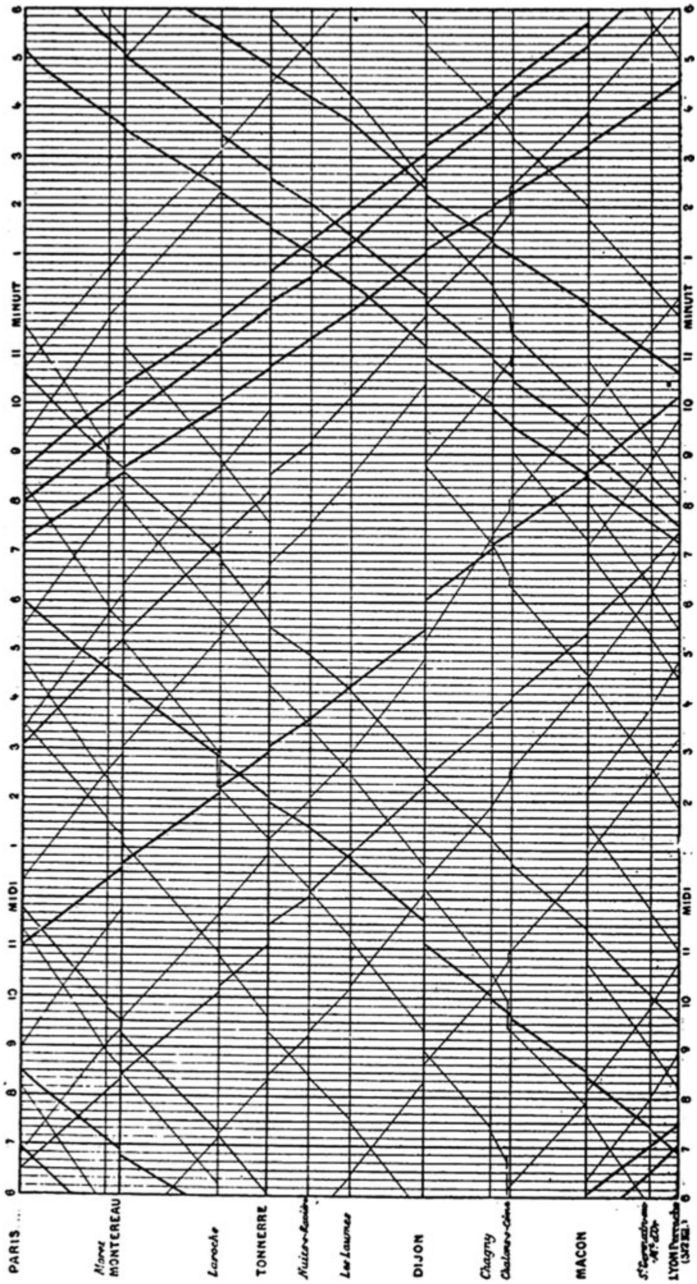


Fig. 2.18: Train schedule by Étienne-Jules Marey (19th century). Individual train stops are placed according to their distance in a list on the vertical axis, while time is represented on the horizontal axis (figure above is rotated by 90°). The individual trains are represented by diagonal lines running from top-left to bottom-right (Paris–Lyon) and bottom-left to top-right (Lyon–Paris) respectively. *Source: Marey (1875, p. 260).*

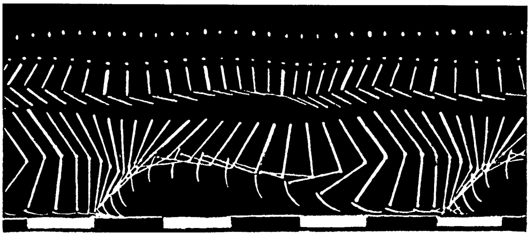


Fig. 2.19 A person walking. Studies of movement by Étienne-Jules Marey (19th century).
Source: Marey (1894, p. 61).

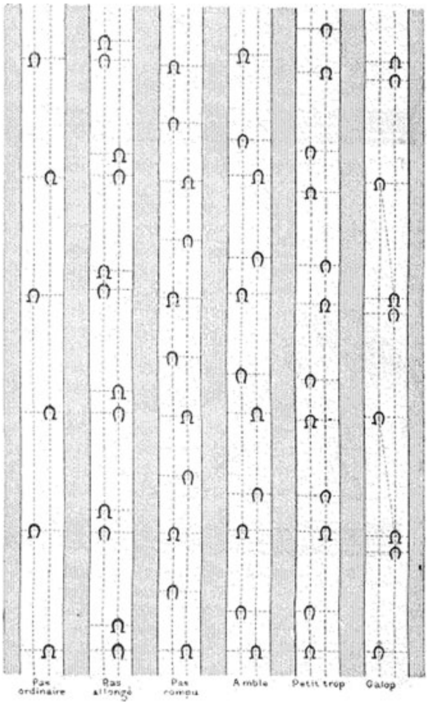
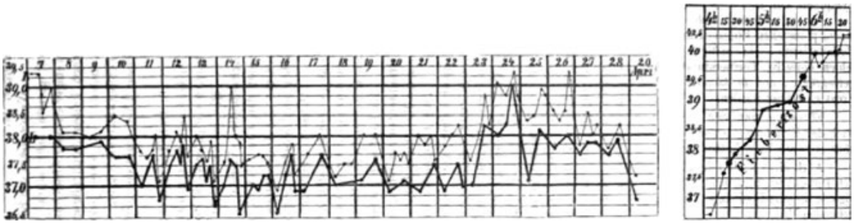


Fig. 2.20 Horse gaits. Studies of movement by Étienne-Jules Marey (19th century).
Source: Source: Marey (1894, p. 188).

Fig. 2.21 Chronophotography. A photo of flying pelican taken by Étienne-Jules Marey around 1882.
Source: http://commons.wikimedia.org/wiki/File:Marey_-_birds.jpg; Retrieved Feb., 2011.

In medicine, large amounts of information are generated which mostly have to be processed by humans. Graphical representations which help to make this myriad of information comprehensible play a crucial role in the workflow of healthcare personnel. These representations range from the *fever curves* of the nineteenth century (see Figure 2.22) and EEG time-series plots (see Figure 2.23) to information-rich patient status overviews (see Figure 2.24). Especially the graphical summary of patient status by Powsner and Tufte (1994) makes use of concepts such as *small multiples* (↪ p. 236), *focus+context* (see p. 111), or the integration of textual and graphical information. It manages to display information on a single page that would otherwise fill up entire file folders and would require serious effort to summarize.

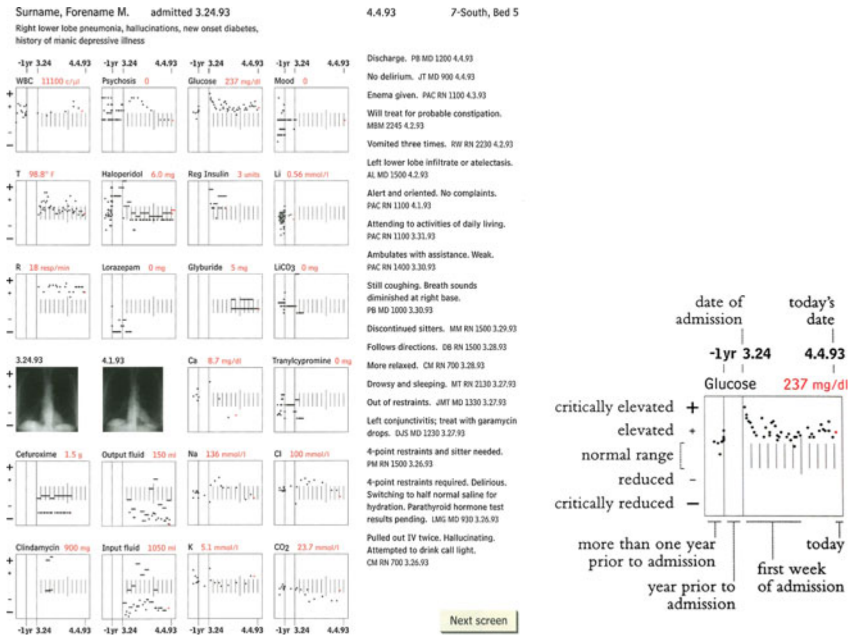


(a) Fever chart.

(b) Detail on rigor (Fieberfrost).

Fig. 2.22: Fever charts created by Carl August Wunderlich (1870).
Source: Wunderlich (1870, p. 161, 167).

Fig. 2.23 EEG time-series plot.
Source: *http://commons.wikimedia.org/wiki/File:Spike-waves.png; Retrieved Feb., 2011.*



(a) Overview. (b) Details.

Fig. 2.24: Graphical summary of patient status by Powsner and Tufte (1994). Concise summary of patient information. Uses *small multiples*, *focus+context*, and integrates textual as well as graphical information.
Source: Tufte (1997, p. 110–111). © Graphics Press. Used with permission.

Weather and climate are further well-known application areas dealing with time-oriented data. Here, developments over time are of greater interest than single snapshots. Figure 2.25 shows the adaptation of an extremely information-rich illustration provided by the New York Times for more than 30 years to show New York City's weather developments for a whole year. Monthly and yearly aggregates are displayed along with more detailed information on temperature, humidity, and precipitation. All in all, more than 2500 numbers are shown in this representation in a very compact and readable form. An even earlier example of a visual representation of the weather data of New York City is shown in Figure 2.26. Here, temperatures, wind velocity, relative humidity, wind direction, and the weather conditions of a single month (December, 1912) are displayed.

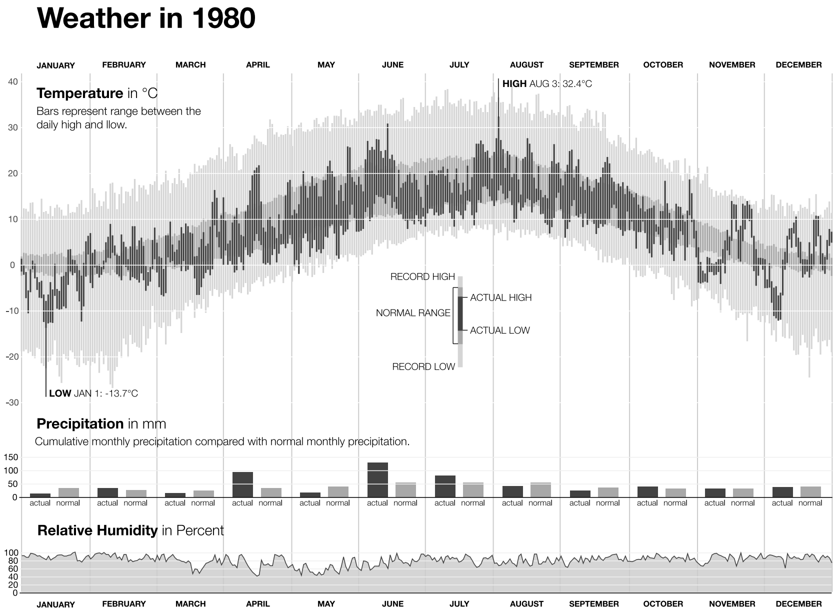


Fig. 2.25: Weather statistics for 1980. Aggregated values are displayed along with more detailed information on temperature, humidity, and precipitation. Similar illustrations have been printed annually by the New York Times for more than 30 years.
Source: Generated with the Protovis toolkit.

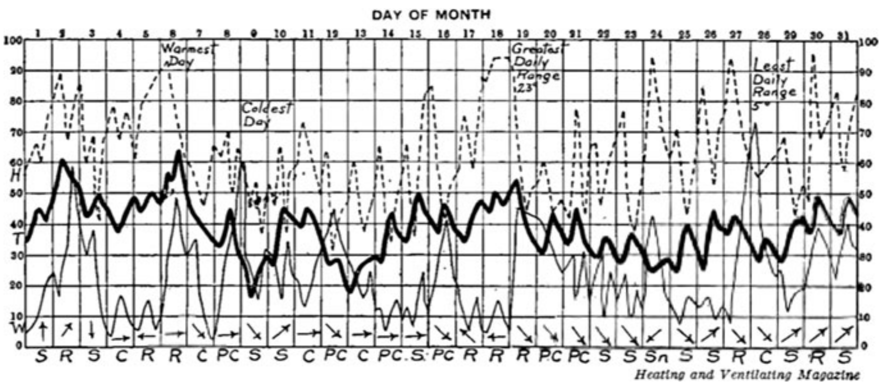


Fig. 2.26: Record of the Weather in New York City for December, 1912. The bold line indicates temperature in degrees Fahrenheit. The light solid line shows wind velocity in miles per hour. The dotted line depicts relative humidity in percentage from readings taken at 8 a.m. and 8 p.m. Arrows portray the prevailing direction of the wind. Initials at the base of the chart show the weather conditions as follows: S, clear; PC, partly cloudy; C, cloudy; R, rain; Sn, snow.
Source: Brinton (1914, p. 93).

2.2 Time in Visual Storytelling & Arts

Two disciplines that are seldom connected to time-oriented information are *visual explanations* and *visual storytelling*. Although ubiquitously used in various forms in daily life, they are rarely considered for visualizing abstract information. Visual explanations are often used in manuals for home electronics, furniture assembly, car repair, and many more (see Figures 2.28 and 2.29). Often, they are used to illustrate stepwise processes visually to an international audience to support the often poorly translated textual instructions. The stepwise nature conveys a temporal aspect and might also be applied to represent abstract information. Even older than everything we presented previously is the craft of *storytelling*, especially visual storytelling, starting from caveman paintings and Egyptian hieroglyphs to picture books and comic strips (see Figure 2.30). Time is the central thread that ties everything together in visual storytelling. Many interesting techniques and paradigms exist that might be applicable to visualization in general (see for example Gershon and Page, 2001) as well as to the representation of time-oriented information in particular.

Comics The art of *comics* is often dubbed as *visual storytelling over time* or *sequential art* (a term used by Will Eisner) because temporal flows are represented in juxtaposed canvases on a page. These descriptions already suggest that comics incorporate many concepts of time, while still retaining a static, 2-dimensional form. Scott McCloud (1994) analyzed many of the methods and paradigms of comics, concluding that powerful means of representing time, dynamics, and movement are applied which differ from those applied in painting or photography. Comics allow for the seamless representation of many temporal concepts that may be also applicable to visualization. Basically, the course of time is represented in comics via

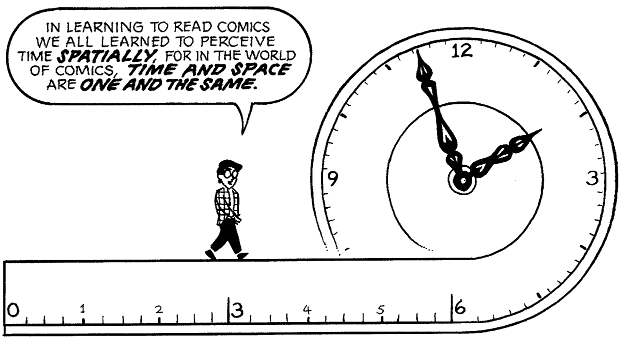


Fig. 2.27: In learning to read comics we all learned to perceive time spatially, for in the world of comics, time and space are one and the same.
Source: McCloud (1994, p. 100). © 1993, 1994 Scott McCloud. Reprinted with permission of HarperCollins Publishers.

Fig. 2.28: Visual explanation is used to illustrate stepwise processes.
Source: Adapted from Tomitsch et al. (2007).

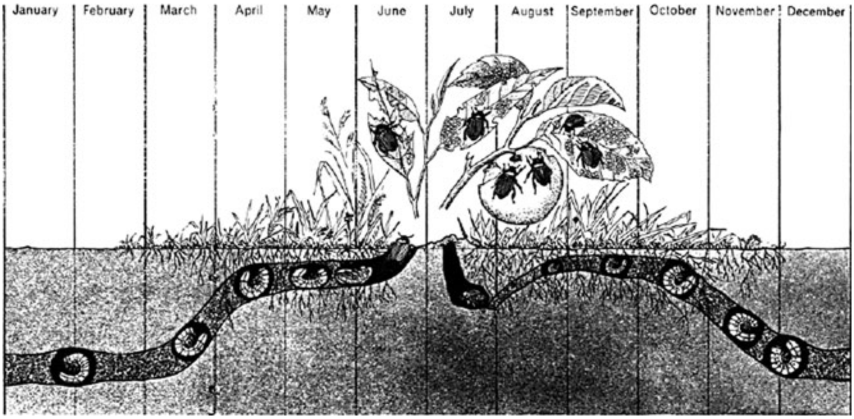
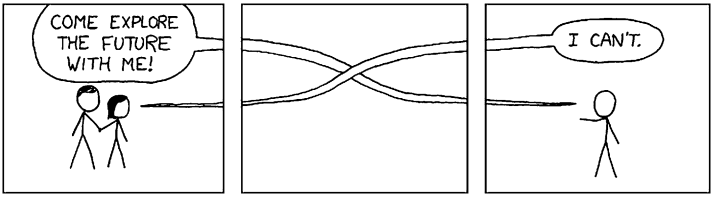


Fig. 2.29: Life Cycle of the Japanese Beetle (Popillia japonica Newman).
Source: Newman (1965, p. 104–105). Reproduced from Tufte (1990, p. 43). Used with permission of Graphics Press.

juxtaposition of panels. But the individual panels portray more than single frozen moments in time and are more than photos placed side by side. Rather, single panels contain whole scenes whose temporal extent may span from milliseconds to arbitrary lengths (see Figure 2.31). Not only the content of a panel sheds light on the length of its duration but also the shape of the panel itself can affect our perception of time. Even more freedom in a temporal sense is given by the transition from one panel to the next or by the space between panels, respectively (see Figure 2.32). Here, time might be compressed, expanded, rewound; deja vu's might be incorporated and much more. This also implies that comics are not just simply linearly told stories. Comics are very versatile and much more powerful in incorporating time in comparison to paintings, photographs, and even film. Besides the purely temporal aspect, motion is another important topic in comics. Several visual



(a) Randall Munroe, xkcd – A Webcomic of Romance, Sarcasm, Math, and Language.
Source: http://xkcd.com/338/; Retrieved Feb., 2011. Image courtesy of Randall Munroe.

(b) Greg Dean, RealLife – A daily online comic.
Source: http://www.reallifecomics.com/archive/991206.html; Created Dec. 6, 1999; Retrieved Feb., 2011. Image courtesy of Greg Dean.

Fig. 2.30: Comics where temporal flows are represented in juxtaposed canvases on a page.

techniques, such as motion lines or action lines with additional effects like multiple images, streaking effects, or blurring are applied (see Figure 2.33). In part, these techniques are borrowed from photography. Recently, research work on generating these comic-like effects from motion pictures has been conducted as for example in Markovic and Gelautz (2006).

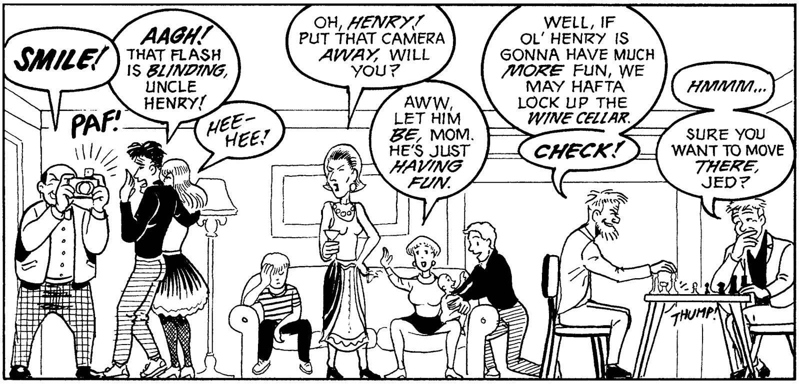


Fig. 2.31: A single comic panel contains more than a frozen moment in time.
Source: McCloud (1994, p. 95). © 1993, 1994 Scott McCloud. Reprinted with permission of HarperCollins Publishers.

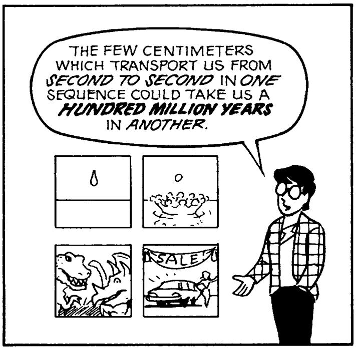


Fig. 2.32: Transitions between panels might span intervals of arbitrary length.
Source: McCloud (1994, p. 100). © 1993, 1994 Scott McCloud. Reprinted with permission of Harper-Collins Publishers.

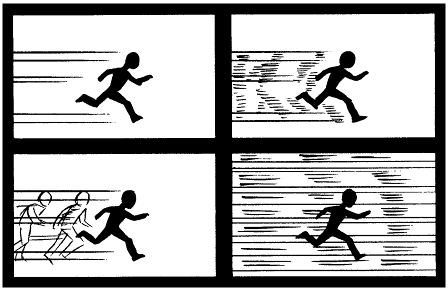


Fig. 2.33: Techniques to represent movement in comics (motion lines, streaking, multiple images, background streaking).
Source: McCloud (1994, p. 114). © 1993, 1994 Scott McCloud. Reprinted with permission of HarperCollins Publishers.

Music & dance Music notes are a notation almost everybody is aware of, but it is one which is rarely seen in conjunction with time-oriented information (see Figure 2.34). Nevertheless, music notes are clearly a visual representation of temporal information – even more than that. A rich set of different symbols, lines, and text constitute a very powerful visual language. Beat, rhythm, pitch, note length, pausing, instrument tuning, and parallelism are the most important visualized parameters. In fact, it is hard to imagine any other way of representing musical compositions than via music notes. Related to that, special notations are used for recording dance performances statically on paper (see Figure 2.35).

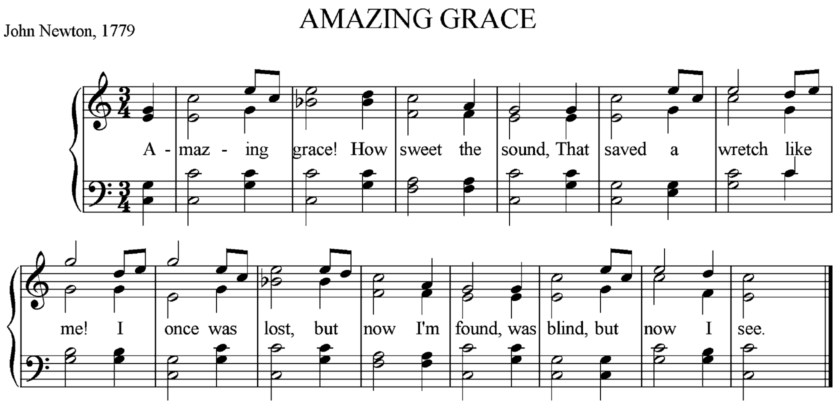


Fig. 2.34: Music notation. A rich set of symbols, lines, and text visualizes beat, rhythm, pitch, note length, pausing, instrument tuning, and parallelism.
Source: http://commons.wikimedia.org/wiki/File:AmazingGraceFamiliarStyle.png; Retrieved Feb., 2011.

Fig. 2.35: Dance notation. Used for recording dance performances statically on paper.
Source: Tufte (1990, p. 117). © 1990 Graphics Press. Used with permission.

Movies One art form that is only touched upon briefly here, but which might also offer interesting ideas for visualization, is *film*. We will present movies that exemplify how movie makers are able to transport highly non-linear stories in the temporally linear medium of film. These examples pertain to the plot of a film, and not to filming or cutting techniques.

Run Lola Run[1] is a movie that presents several possible successions of events sequentially throughout the film (compare *branching time* in Section 3.1.1). The individual episodes begin at the same point in time and show different possible strands of events.

The movie *Pulp Fiction*[2] comprises an even more complicated and challenging plot. It is a collection of different episodes that are semantically as well as temporally linked. Moreover, the movie ends by continuing the very first scene in the movie, thus closing the loop.

A further example of the use of interesting temporal constellations in film is the movie *Memento*[3]. The main character of the movie is a man who suffers from short-term memory loss, who uses notes and tattoos to hunt for his wife's killer. What makes the storytelling so challenging is the fact that time flows backwards from scene to scene (i.e., the end is shown at the beginning and the story progresses to the beginning from there).

Music videos are also often used as an innovation playground where directors can experiment with unconventional temporal flows such as the *reverse narrative* as used in Coldplay's *The Scientist*[4].

Paintings A very interesting approach to overcoming the limitations of time can be found in *Renaissance* paintings. Here, sequences of different temporal episodes are shown in a single composition. Figure 2.36 for example shows a painting by Masolino da Panicale that presents two scenes in the life of St. Peter within a single scenery. While this method of showing different stages or episodes within a unifying scenery was well understood by the people at that time (the Middle Ages), it might not be as easily understood by a modern viewer. This technique provides evidence for the following statement:

> [...] paintings have always been able to capture more than a fleeting moment in time.
> Crabbe (2003)

The beginning of the 20th century was characterized by new findings and breakthroughs in the natural sciences, especially in mathematics and physics, such as Einstein's theory of relativity. But not only the world of science was shaken by these developments; artists also addressed these topics in their own way. Foremost among these were the protagonists of the art movement of *Cubism*, who focused on incorporating time in their artworks. They coined the term *Four-dimensional Art*.

---

[1] Run Lola Run (Lola rennt), written and directed by Tom Twyker, 1998.

[2] Pulp Fiction, written by Quentin Tarantino et al., directed by Quentin Tarantino, 1994.

[3] Memento, written by J. and C. Nolan, directed by Christopher Nolan, 2000.

[4] The Scientist, recorded by Coldplay, music video directed by Jamie Thraves, 2001.

Fig. 2.36: Masolino da Panicale, Curing the Crippled and the Resurrection of Tabitha (Brancacci Chapel, S. Maria del Carmine, Florence, Italy), 1420s. Different stages or episodes of a single person are shown within a unifying scenery.
Source: http://commons.wikimedia.org/wiki/File:Cappella_brancacci,_Guarigione_dello_storpio_e_resurrezione_di_Tabita_(restaurato),_Masolino.jpg; Retrieved Feb., 2011.

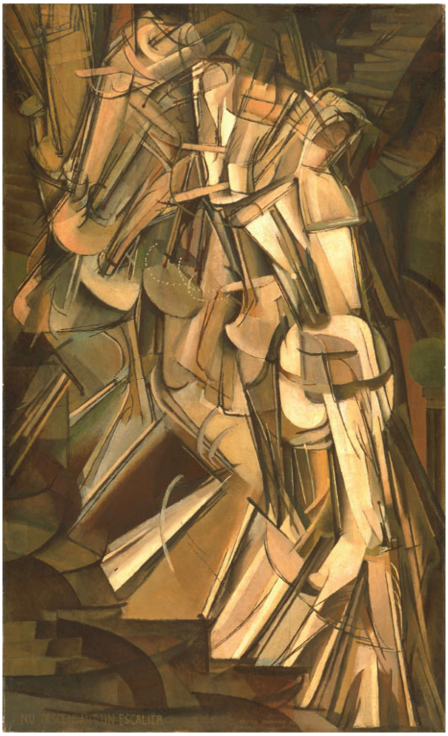


Fig. 2.37 Marcel Duchamp, Nude Descending a Staircase (No. 2), 1912. The dimension time is incorporated by overlaying different stages of a person's movement. (Philadelphia, Philadelphia Museum of Art. Oil on canvas, 57 7/8 x 35 1/8' (147 x 89.2 cm). The Louise and Walter Arensberg Collection, 1950. © VBK, Vienna 2010.) *Source: © 2010. Photo The Philadelphia Museum of Art/Art Resource/Scala, Florence.*

Fig. 2.38 Pablo Picasso, Portrait of Ambroise Vollard, 1910. Many different observations are composed and partly overlaid to form a single picture. (Moscow, Pushkin Museum. © Succession Picasso/VBK, Vienna 2010.) *Source: © 2010. Photo Scala, Florence.*

According to Volaric (2003), the first documented occurrence of time as the fourth dimension appeared in Paris in 1910. As already mentioned, the concept of the n-dimensional space in mathematics and physics inspired artists to think about 4D space. Figure 2.37 shows Marcel Duchamp's painting *Nude Descending a Staircase* which incorporates the dimension of time in a very interesting way by overlaying different stages of a person's movement. Another example is Pablo Picasso's *Portrait of Ambroise Vollard* (see Figure 2.38), where many different observations are composed and partly overlaid to form a single picture. The artists wanted to put emphasis on the *process* of looking and recording over time (in contrast to taking a photo). These new ways of bringing the fourth dimension into the static domain of pictures are still a challenge to viewers today.

2.3 Summary

We have provided a brief review of relevant historical and application-specific visualization techniques and representations of time in the visual arts. Our aim was to provide an historical context for developments in this area and to present some ideas from related areas that might act as a further source of inspiration for designing visualizations. Furthermore, this chapter has demonstrated the enormous breadth of the topic which we are only able to cover in part.

Readers interested in more information about historical representations of time-oriented data and historical representations in general are referred to the wonderful books of Tufte (1983, 1990, 1997, 2006), Wainer (2005), and Rosenberg and Grafton (2010). Michael Friendly's great work on the history of data visualization can be studied in numerous articles such as (Friendly, 2008) as well as online in his Data Visualization Gallery[5] and the Milestones Project[6]. Additionally, interesting historic facts related to time representations are discussed on the Chronographics Weblog[7] of Stephen Boyd Davis.

Now, after setting the stage and considering various concepts and ideas from related disciplines, we will narrow our focus and present a systematic view of the visualization of time-oriented data. In this sense, we will first discuss important aspects that make the handling of time and time-oriented data possible. Following that, the visualization problem itself will be systematically explained and discussed.

References

Barbeu-Dubourg, J. (1753). Carte chronographique. Paper roll in scroll case, 40 cm x 16.5 m. Princeton University Library.

Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, WI, USA. translated by William J. Berg.

Brinton, W. C. (1914). *Graphic Methods for Presenting Facts*. The Engineering Magazine Company, New York, NY, USA.

Brinton, W. C. (1939). *Graphic Presentation*. Brinton Associates, New York, NY, USA.

Chapple, S. and Garofalo, R. (1977). *Rock 'N' Roll is Here to Pay: The History and Politics of the Music Industry*. Burnham Inc Pub, Chicago, IL, USA.

Crabbe, A. (2003). Different Pictures of Time. URL, http://davinci.ntu.ac.uk/rws/researchers/design/tonycrabbe/different.htm. Retrieved Nov. 12, 2003.

Ferguson, S. (1991). The 1753 Carte chronographique of Jacques Barbeu-Dubourg. *Princeton University Library Chronicle*, 52(2):190–230.

Friendly, M. (2008). A Brief History of Data Visualization. In Chen, C.-h., Härdle, W., and Unwin, A., editors, *Handbook of Data Visualization*, pages 15–56. Springer, Berlin, Heidelberg, Germany.

Funkhouser, H. G. (1936). A Note on a Tenth Century Graph. *Osiris*, 1(1):260–262.

Gershon, N. and Page, W. (2001). What Storytelling Can Do for Information Visualization. *Communications of the ACM*, 44(8):31–37.

La maison du cinema and Cinematheque Francaise (2000). Étienne-Jules Marey: Movement in Light. URL, http://www.expo-marey.com/ANGLAIS/home.html. Retrieved Nov. 18, 2004.

Marey, E.-J. (1875). *La Méthode Graphique dans les Sciences Expérimentales (Suite)*, volume 1, pages 255–278. Masson, G., Paris, France.

Marey, E.-J. (1894). *Le mouvement*. Masson, G., Paris, France.

Markovic, D. and Gelautz, M. (2006). Comics-Like Motion Depiction from Stereo. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 155–160, Plzen, Czech Republic. University of West Bohemia.

[5] http://www.datavis.ca/gallery/; Retrieved Feb., 2011.

[6] http://www.datavis.ca/milestones/; Retrieved Feb., 2011.

[7] http://chronographics.blogspot.com/; Retrieved Feb., 2011.

McCloud, S. (1994). *Understanding Comics*. HarperPerennial, New York, NY, USA.

Newman, L. H. (1965). *Man and Insects*. Aldus Books, London, UK.

Playfair, W. (1805). *An Inquiry into the Permanent Causes of the Decline and Fall of Powerful and Wealthy Nations*. Greenland and Norris, London, UK.

Playfair, W. (1821). *A Letter on our Agricultural Distresses, their Causes and Remedies*. William Sams, London, UK.

Playfair, W. and Corry, J. (1786). *The Commercial and Political Atlas: Representing, by Means of Stained Copper-Plate Charts, the Progress of the Commerce, Revenues, Expenditure and Debts of England during the Whole of the Eighteenth Century*. printed for J. Debrett; G. G. and J. Robinson; J. Sewell; the engraver, S. J. Neele; W. Creech and C. Elliot, Edinburgh; and L. White, London, UK.

Powsner, S. M. and Tufte, E. R. (1994). Graphical Summary of Patient Status. *The Lancet*, 344(8919):386–389.

Priestley, J. (1765). *A Chart of Biography*. Johnson, J., London, UK.

Rosenberg, D. and Grafton, A. (2010). *Cartographies of Time: A History of the Timeline*. Princeton Architectural Press, New York, NY, USA.

Third Millennium Press (2001). *Zeittafel der Weltgeschichte. Den letzen 6000 Jahren auf der Spur.* Könemann Verlagsgesellschaft mbH, Cologne, Germany.

Tomitsch, M., Aigner, W., and Grechenig, T. (2007). A Concept to Support Seamless Spectator Participation in Sports Events Based on Wearable Motion Sensors. In *Proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA07)*, pages 209–214, Los Alamitos, CA, USA. IEEE Computer Society.

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

Tufte, E. R. (1990). *Envisioning Information*. Graphics Press, Cheshire, CT.

Tufte, E. R. (1997). *Visual Explanations*. Graphics Press, Cheshire, CT.

Tufte, E. R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, CT.

Volaric, A. (2003). Four-Dimensional Art. URL, http://www.mathsci.appstate.edu/˜sjg/class/3510/final/geoartws.pdf. Retrieved Feb., 2011.

Wainer, H. (2005). *Graphic Discovery: A Trout in the Milk and Other Visual Adventures*. Princeton University Press, Princeton, NJ, USA.

Wunderlich, C. A. (1870). *Das Verhalten der Eigenwärme in Krankheiten*. Otto Wigand, Leipzig, Germany, 2nd edition.

Chapter 3
Time & Time-Oriented Data

> What, then, is time?
> If no one asks me, I know what it is.
> If I wish to explain it to him who asks, I do not know.
>
> Saint Augustine (AD 354-430, The Confessions)

The fundamental phenomenon of time has always been of interest for mankind. Many different theories for characterizing the physical dimension of time have been developed and discussed over literally thousands of years in philosophy, mathematics, physics, astronomy, biology, and many other disciplines. As reported by Whitrow et al. (2003), a 1981 literature survey by J.T. Fraser found that the total number of entries judged to be potentially relevant to the systematic study of time reached about 65,000. This illustrates the breadth of the topic and the restless endeavor of man to uncover its secrets. What can be extracted as the bottom line across many theories is that time is *unidirectional* (arrow of time) and that time gives *order* to events.

The most influential theories for the natural sciences are probably Newton's concepts of absolute vs. relative time, and Einstein's four-dimensional spacetime. Newton assumed an absolute, true, mathematical time that exists in itself and is not dependent on anything else. Together with space, it resembles a container for all processes in nature. This image of an absolute and independent dimension prevailed until the beginning of the 20th century. Then, Einstein's relativity theory made clear that time in physics depends on the observer. Thus, Einstein introduced the notion of *spacetime*, where space and time are inherently connected and cannot be separated. That is, each event in the universe takes place in four-dimensional space at a location that is defined by three spatial coordinates at a certain time as the fourth coordinate (see Lenz, 2005). Both Newton's notion of absolute time and Einstein's spacetime are concepts that describe time as a fundamental characteristic of the universe. In contrast to that, the way humans deal with time in terms of deriving it essentially from astronomical movements of celestial bodies or phenomena in nature is what Newton called relative time.

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_3,

The first signs of the systematic use of tools for dealing with time have been found in the form of bone engravings that resembled simple calendars based on the cycle of the moon. In this regard, the most fundamental natural rhythm perceived by humans is the day. Consequently, it is the basis of most calendars and was used to structure the simple life of our ancestors who lived in close contact with nature (see Lenz, 2005). More complex calendars evolved when man settled into agricultural communities, moving away from the life of a hunter-gatherer, and began to live from agriculture. Until very late in human history, time was kept only very roughly. Industrialization and urban civilization brought about the need for more precise, regular, and synchronized overall timekeeping.

Today, the most commonly used calendric system is the Gregorian calendar. It was introduced by Pope Gregory XII in 1582, primarily to correct the drift of the previously used Julian calendar, which was slightly too long in relation to the astronomical year and the seasons[1]. Apart from this calendric system, many other systems are in use around the world, such as the Islamic, the Chinese, or the Jewish calendars, or calendars for special purposes, like academic (semester, trimester, etc.) or financial calendars (quarter, fiscal year, etc.).

In this book, we will not look at the physical dimension of time itself and its philosophical background, how time is related to natural phenomena, or how clocks have been developed and used. We focus on how the physical dimension of time and associated data can be modeled in a way that facilitates interactive visualization using computer systems. As a next step we are now going to examine the design aspects for modeling time.

3.1 Modeling Time

First of all, it is important to make a clear distinction between the physical dimension time and a model of time in information systems. When modeling time in information systems, the goal is not to perfectly imitate the physical dimension time, but to provide a model that is best suited to reflect the phenomena under consideration and support the analysis tasks at hand. Moreover, as Frank (1998) states, there is nothing like a single correct model or taxonomy of time – there are many ways to model time in information systems and time is modeled differently for different applications depending on the particular problem. Extensive research has been conducted in order 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 which includes many references to fundamental publications is provided by Hajnicz (1996). However, as she points out, the terminology is not consistent across the different fields, and hence, does not integrate well with visualization. Moreover, as Goralwalla et al. (1998) note, most research focuses on the development of specialized models with different features for particular domains.

[1] Interestingly, much more precise calendars were known hundreds of years earlier in other cultures, such as those developed by the Mayas and the Chinese.

But apart from the many time models created for specific purposes and applications, attempts have been made to capture the major design aspects underlying all specific instances, as for example by Frank (1998), Goralwalla et al. (1998), Peuquet (1994, 2002), and Furia et al. (2010).

Here, we want to present the overall design aspects of modeling time, and not a particular model. To do this, we will describe a number of major design aspects and their features which are particularly important when modeling time. Application-specific models can be derived from these as particular configurations.

3.1.1 Design Aspects

To define the design aspects relevant for time, we adapted the works of Frank (1998) and Goralwalla et al. (1998), where principal orthogonal aspects are presented to characterize different types of time. These aspects will now be described in detail.

Scale: ordinal vs. discrete vs. continuous As a first perspective, we look at time from the scale along which elements of the model are given. In an *ordinal* time domain, only relative order relations are present (e.g., before, after). For example, statements like "Valentina went to sleep before Arvid arrived" and "Valentina woke up after a few minutes of sleep" can be modeled using an ordinal scale. Note that only relative statements are given and one cannot discern from the given example whether she woke up before or after he arrived (see Figure 3.1). This might be sufficient if only qualitative temporal relationships are of interest or no quantitative information is available.

In *discrete* domains temporal distances can also be considered. Time values can be mapped to a set of integers which enables quantitative modeling of time values (e.g., quantifiable temporal distances). Discrete time domains are based on a smallest possible unit (e.g., seconds or milliseconds as in UNIX time) and they are the most commonly used time models in information systems (see Figure 3.2). *Continuous* time models are characterized by a possible mapping to real numbers, i.e., between any two points in time, another point in time exists (also known as dense time, see Figure 3.3).

Examples of visualization techniques capable of representing the three types of scale are the *point and figure chart* (see Figure 3.4) for an ordinal scale, *tile maps* (see Figure 3.5 and ↪ p. 178) for a discrete scale, and the *circular silhouette graph* (see Figure 3.6 and ↪ p. 175) for a continuous time scale.

Scope: point-based vs. interval-based Secondly, we consider the scope of the basic elements that constitute the structure of the time domain. *Point-based* time domains can be seen in analogy to discrete Euclidean points in space, i.e., having a temporal extent equal to zero. Thus, no information is given about the region between two points in time. In contrast to that, *interval-based* time domains relate to subsections of time having a temporal extent greater than zero. This aspect is also closely related to the notion of granularity, which will be discussed in Section 3.1.2.

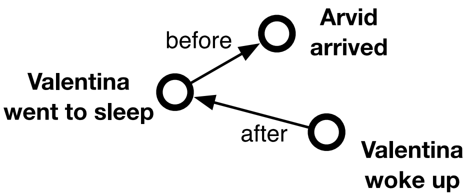


Fig. 3.1 Ordinal scale. Only relative order relations are present. At this level it is not possible to discern whether Valentina woke up before or after Arvid arrived.

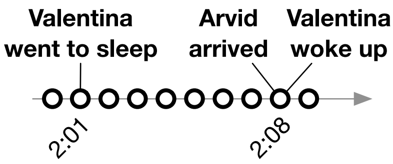


Fig. 3.2 Discrete scale. Smallest possible unit is minutes. Although Arvid arrived and Valentina woke up within the same minute, it is not possible to model the exact order of events.

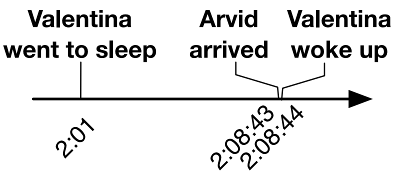


Fig. 3.3 Continuous scale. Between any two points in time, another point in time exists. Here, it is possible to model that Arvid arrived shortly before Valentina woke up.

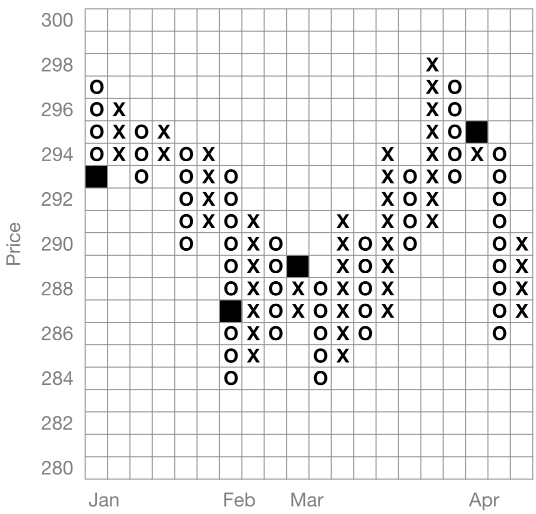


Fig. 3.4 Point and figure chart. Visualization technique tracking price and price direction changes. Uses an *ordinal time scale*. ○...positive price change of a certain amount, ×...negative price change of a certain amount, ■...begin/end of a trading period. *Source: Adapted from Harris (1999).*

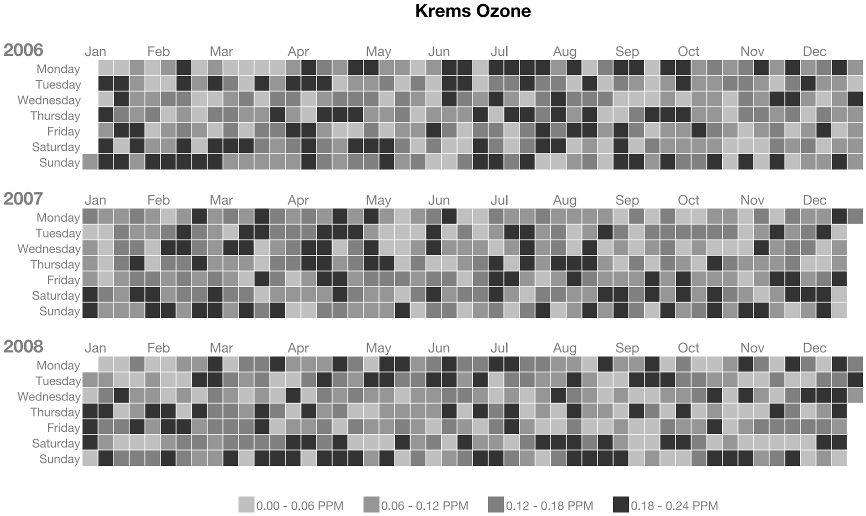


Fig. 3.5: Tile maps. Example shows average daily ozone measurements (scale: *discrete*, scope: *interval-based*) over the course of three years.
Source: Adapted from Mintz et al. (1997) with permission of David Mintz.

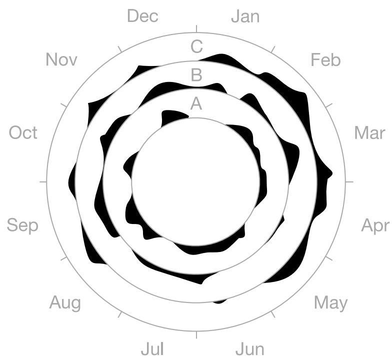


Fig. 3.6 Circular silhouette graph. Enables the representation of time along a *continuous scale* with a *cyclic arrangement*. The representation emphasizes the visual impression by filling the area below the plotted line in order to create a distinct silhouette. This eases comparison when placed side by side.
Source: Adapted from Harris (1999).

For example the time value August 1, 2008 might relate to the single instant August 1, 2008 00:00:00 in a point-based domain, whereas the same value might refer to the interval [August 1, 2008 00:00:00, August 1, 2008 23:59:59] in an interval-based domain (see Figures 3.7 and 3.8).

Examples of visualization techniques capable of representing the two types of scope are the *TimeWheel* (see Figure 3.9 and ↪ p. 200) for a point-based domain and *tile maps* (see Figure 3.5 and ↪ p. 178) for an interval-based time domain.

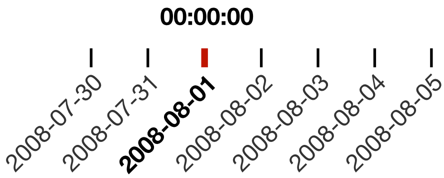


Fig. 3.7 Time value "August 1, 2008" in a point-based domain. No information is given in between two time points.

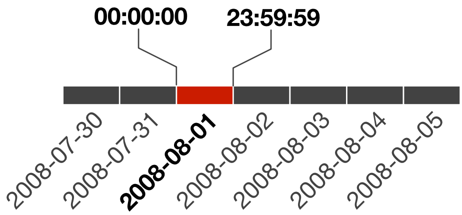


Fig. 3.8 Time value "August 1, 2008" in an interval-based domain. Each element covers a subsection of the time domain greater than zero.

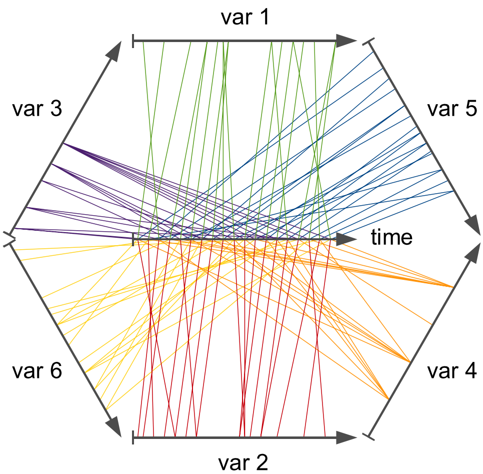


Fig. 3.9 TimeWheel. Parameter axes are arranged around the horizontal point-based time axis in a regular polygonal manner. For each time-step, lines descend from the time axis to the corresponding points on the parameter axes.

Arrangement: linear vs. cyclic As the third design aspect, we look at the arrangement of the time domain. Corresponding to our natural perception of time, we mostly consider time as proceeding *linearly* from the past to the future, i.e., each time value has a unique predecessor and successor (see Figure 3.10). However, periodicity is very common in all kinds of data, for example seasonal variations, monthly averages, and many more. In a *cyclic* organization of time, the domain is composed of a set of recurring time values (e.g., the seasons of the year, see Figure 3.11). Hence, any time value A is preceded and succeeded at the same time by any other time value B (e.g., winter comes before summer, but winter also succeeds summer). In order to enable meaningful temporal relationships in cyclic time, Frank (1998) suggests the use of the relations *immediately before* and *immediately*

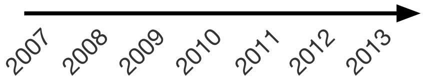


Fig. 3.10 Linear time. Time proceeds linearly from past to future.

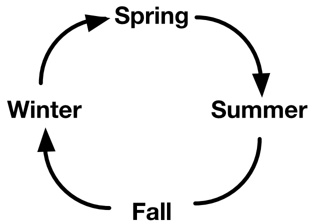


Fig. 3.11 Cyclic time. Set of recurring time values such as the seasons of the year.

after. Strictly cyclic data, where the linear progression of time from past to future is neglected, is very rare (e.g., records for the day of week not considering month or year). The combination of periodic and linear progression denoted by the term *serial periodic data* (e.g., monthly temperature averages over a couple of years) is much more common. Periodic time-oriented data in this sense includes both strictly cyclic data and serial periodic data.

Examples of visualization techniques capable of representing the two types of arrangement are the *TimeWheel* (see Figure 3.9 and ↪ p. 200) for linear time and the *circular silhouette graph* (see Figure 3.6 and ↪ p. 175) for cyclic time.

Viewpoint: ordered vs. branching vs. multiple perspectives The fourth subdivision is concerned with the views of time that are modeled. *Ordered* time domains consider things that happen one after the other. On a more detailed level, we might also distinguish between totally ordered and partially ordered domains. In a totally ordered domain only one thing can happen at a time. In contrast to this, simultaneous or overlapping events are allowed in partially ordered domains, i.e., multiple time primitives at a single point or overlapping in time. A more complex form of time domain organization is the so-called *branching* time (see Figure 3.12). Here, multiple strands of time branch out and allow the description and comparison of alternative scenarios (e.g., in project planning). This type of time supports decision-making processes where only one of the alternatives will actually happen. Note that branching is not only useful for future scenarios but can also be applied for investigating the past, e.g., for modeling possible causes of a given decision. In contrast to branching time where only one path through time will actually happen, *multiple perspectives* facilitate simultaneous (even contrary) views of time, which are necessary, for instance, to structure eyewitness reports. A further example of multiple perspectives are stochastic multi-run simulations. For a single experiment, there might be completely different output data progressions depending on the respective initialization.

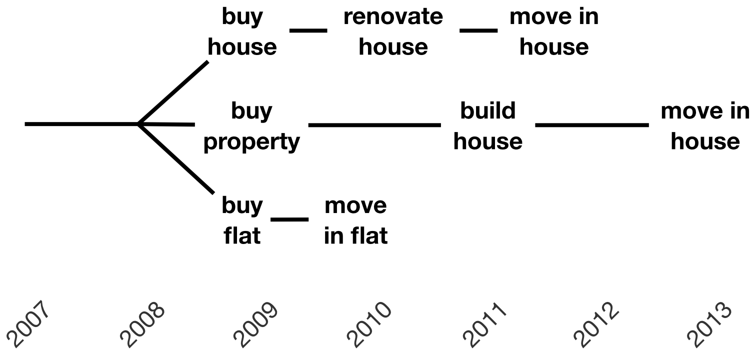


Fig. 3.12: Branching time. Alternative scenarios for moving into a new living space.

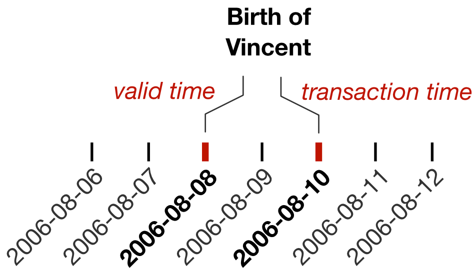


Fig. 3.13 Multiple perspectives. Vincent was born on August 8, 2006 (valid time) and this fact was stored in the register of residents two days later on August 10, 2006 (transaction time).

In temporal databases, the two perspectives *valid time* and *transaction time* are often modeled (see Figure 3.13). The valid time perspective of a fact is the time when the fact is true in the modeled reality (e.g., “Vincent was born on August 8, 2006”). In contrast to that, the transaction time perspective of a fact denotes when it was stored in the database (e.g., the birth of Vincent is stored in the register of residents after filling out a form two days after his birth). Multiple perspectives often need to be condensed into a single consistent view of time (see for example Wolter et al., 2009).

Both branching time and multiple perspectives introduce the need to deal with probability (or uncertainty), to convey, for instance, which path through time will most likely be taken, or which evidence is believable. The *decision chart* (see Figure 3.14 and ↪ p. 159) is an example of a visualization technique capable of representing branching time.

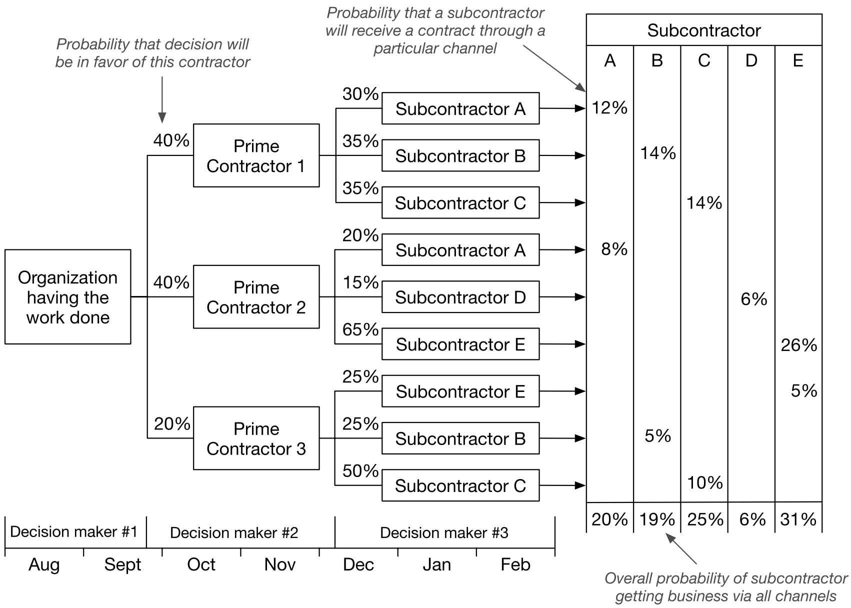


Fig. 3.14: Decision chart. Example of a visualization technique capable of representing branching time. Future decisions and potential alternative outcomes along with their probabilities can be depicted over time.
Source: Adapted from Harris (1999).

3.1.2 Granularities & Time Primitives

The previous section introduced design aspects to adequately model the time domains' scale, scope, and arrangement as well as possible viewpoints onto the time domain. Besides these general aspects, the hierarchical organization of time as well as the definition of concrete time elements used to relate data to time need to be specified. In the following, we will discuss this facet in more detail.

Granularity and calendars: none vs. single vs. multiple To tame the complexity of time and to provide different levels of granularity, useful abstractions can be employed. Basically, granularities can be thought of as (human-made) abstractions of time in order to make it easier to deal with time in every-day life (like minutes, hours, days, weeks, months). More generally, granularities describe mappings from time values to larger or smaller conceptual units[2] (see Figure 3.15 for an example of time granularities and their relationships).

If a granularity and calendar system is supported by the time model, we categorize it as *multiple* granularities. Besides this complex variant, there might be a *single*

[2] An overview and formalization of time granularity concepts is given by Bettini et al. (2000).

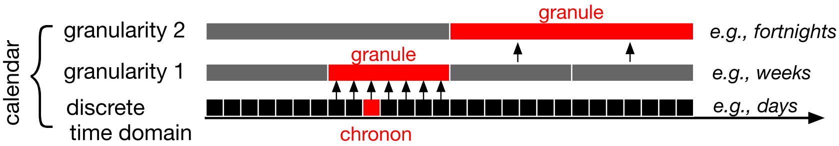


Fig. 3.15: Example of a discrete time domain with multiple granularities. The smallest possible unit (chronon) is one *day*. Based on this, the granularity *weeks* contains granules that are defined as being a continuous set of seven days. Moreover, the granularity *fortnights* consists of granules that are a set of two consecutive weeks.

granularity only (e.g., every time value is given in terms of milliseconds) or *none* of these abstractions are supported (e.g., abstract ticks).

Most information systems that deal with time-oriented data are based on a discrete time model that uses a fixed smallest granularity also known as *bottom granularity* (e.g., Java's Date class uses milliseconds as the smallest granularity). Hence, the underlying time domain can be described as a sequence of non-decomposable, consecutive time intervals of identical duration called *chronons* (see Jensen et al., 1998). This allows for a simple representation of a point in time as number of chronons relative to a reference point (e.g., milliseconds (=chronons) since January 1, 1970 00:00:00 GMT). Chronons may be grouped into larger segments, termed *granules*. Based on this, a granularity is a non-overlapping mapping of so-called *granules* to subsets of the time domain (see Dyreson et al., 2000). Granularities are related in the sense that the granules in one granularity may be further aggregated to form larger granules belonging to a coarser granularity. For example, 60 consecutive seconds are mapped to one minute.

A system of multiple granularities in lattice structures is referred to as a *calendar* (see Figure 3.16 for the granularity lattice of the Gregorian calendar). More precisely, it is a mapping between human-meaningful time values and an underlying time domain. Thus, a calendar consists of a set of granularities including mappings between pairs of granularities that can be represented as a graph (see Dyreson et al., 2000). Calendars most often include cyclic elements, allowing human-meaningful time values to be expressed succinctly. For example, dates in the common Gregorian calendar may be expressed in the form <day, month, year> where each of the fields day, month, and year circle as time passes (see Jensen et al., 1998).

Moreover, mappings between granularities might be regular or irregular. A regular mapping exists for example between the granularities *seconds* and *minutes* where one minute is always mapped to 60 seconds[3]. In contrast to that, the mapping of *days* to *months* is irregular because one month might be composed of 28, 29, 30, or 31 days depending on the context (particular year and month).

[3] We are not considering the exception of leap seconds here.

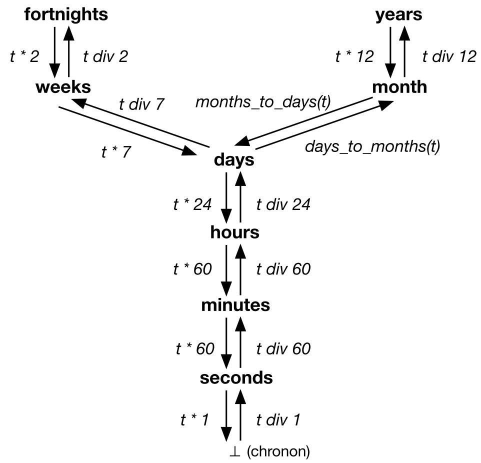


Fig. 3.16 Annotated granularity lattice of the Gregorian calendar that contains regular and irregular mappings (leap seconds are not considered in the granularity lattice).

To manage time granularities and calendars accordingly, appropriate models and, even more importantly, conversion operators have to be supported. These include for example the definition of granularities, their relationships, and calendars, and operations like conversions from one granularity to another or for combining calendars. Particularly, operations that convert from one granularity to another, as for example from days to months, can be quite complex due to the irregularities in granularities. Basic implementations of the described functionalities are present in most programming languages and database systems in terms of the widely used Gregorian calendar (e.g., java.util.Calendar and java.util.GregorianCalendar)[4].

Moreover, granularities influence equality relationships. Take for example the time interval between Tuesday, December 30, 2008 and Thursday, January 1, 2009 (see Figure 3.17). While this interval is entirely within a single week on the granularity of weeks, it overlaps two years on the granularity of years. Note that this is contradictory to the naive assumption that when an equality relationship holds true on a fine granularity it also holds true on a coarser one.

An example of a visualization technique that uses time granularities is the *cycle plot* (see Figure 3.18 and ↪ p. 176).

The concepts chronon, granule, granularity, and calendar have been introduced to hierarchically organize the time domain which reflects our common perception and usage of time.

Time primitives: instant vs. interval vs. span Next, we present a set of basic elements used to relate data to time, so-called time primitives: instant, interval, and span. These time primitives can be seen as an intermediary layer between data elements and the time domain. Basically, time primitives can be divided into anchored

[4] More sophisticated systems and models that support multiple (user-defined) granularities and calendars are described in Dyreson et al. (2000) and Lee et al. (1998).

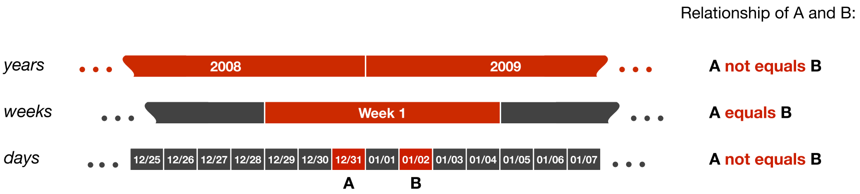


Fig. 3.17: Granularities influence equality relationships. The times of A and B are not equal on the granularity of days, but are equal on the granularity of weeks, and then again are not equal on the coarser granularity of years.

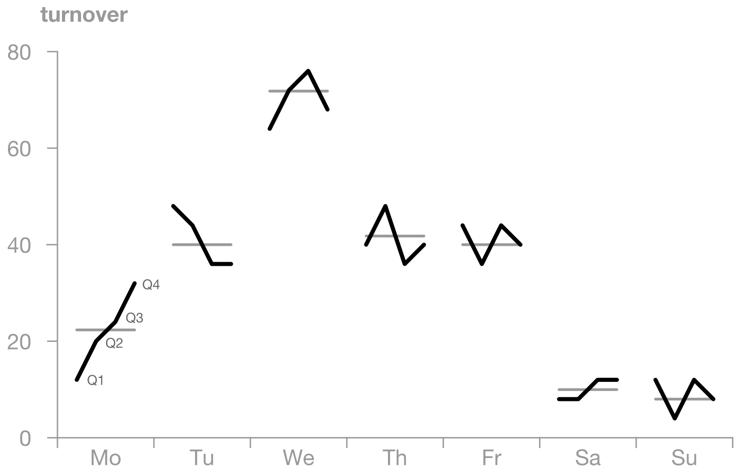


Fig. 3.18: Cycle plot. Visualization technique that utilizes two time granularities to represent cycles and trends. The example shows trends of measurements of weekdays over quarters. For example, on Mondays, the values show an increasing trend over the year while on Tuesdays the trend is decreasing. Furthermore, the general shape of a week's cycle is visible.
Source: Adapted from Cleveland (1993) with permission of William Cleveland.

(absolute) and unanchored (relative) primitives. Instant and interval are primitives that belong to the first group, i.e., they are located on a fixed position along the time domain. In contrast to that, a span is a relative primitive, i.e., it has no absolute position in time.

An *instant*[5] is a single point in time, e.g., May 23, 1977. Depending on the scope, i.e., whether a point-based or interval-based time model is used (see previous section), an instant might also have a duration (see Figure 3.19 and Figure 3.20). Time primitives can be defined at all levels of granularity representing chronons, granules, or sets of both. Examples of instants are the date of birth "May 23, 1977" and

[5] Sometimes also referred to as *time point*.

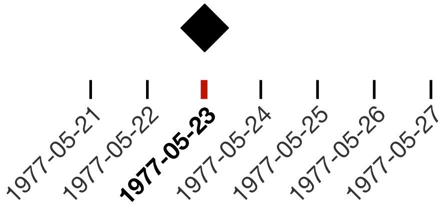


Fig. 3.19 Instant in a point-based time model. A point in time that has no duration.

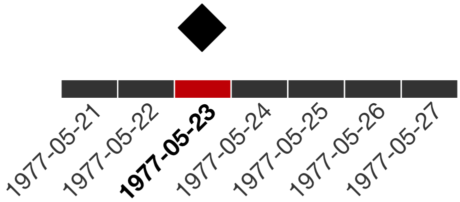


Fig. 3.20 Instant in an interval-based time model. A point in time that has a duration which depends on its granularity.

the beginning of a presentation on "January 10, 2009 at 2 p.m." whereas the first instant (date of birth) is given at a granularity of *days* and the second (beginning of presentation) at a granularity of *hours*.

An *interval* is a portion of time of the underlying time domain that can be represented by two instants that denote the beginning and end of the interval, e.g., [June 13, 2009; June 19, 2009] (see Figure 3.21). Alternatively, intervals can be modeled as beginning instant + duration (positive span), or as duration (positive span) + end instant. An interval that is defined in terms of beginning and end is modeled as a closed interval including the beginning as well as the end instant.

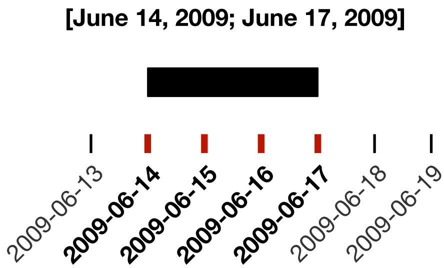


Fig. 3.21 Interval [June 14, 2009; June 17, 2009] in a point-based time model.

The *span* is the only unanchored primitive. It represents a directed duration of time, e.g., 4 days (see Figure 3.22). A time span is defined as a directed, unanchored primitive that represents a directed amount of time in terms of a number of granules in a given granularity. Examples of spans are the length of a vacation of "10

Fig. 3.22 Span. Example of the span "four days" which is formed by four granules of the granularity *days*.

days" and the duration of a lecture of "150 minutes". Figure 3.22 illustrates this graphically by showing an example span of "four days" which is a count of four granules of the granularity *days*. A span is either positive, denoting forward motion of time, or negative, denoting backwards motion of time (see Jensen et al., 1998). In case of irregular granularities (e.g., "months"), the exact length of a span is not known precisely. Consider for example the granularity *months*, where a span of "two months" might be 59, 60, 61, or 62 days depending on the particular time context. This implies that the exact length of spans within irregular granularities can only be determined exactly when related absolutely to the time domain (anchored). Otherwise, mean values might be used for calculations (e.g., mean month and mean year).

Most of the previously given visualization examples are suited for representing instants. *Gantt charts* (↪ p. 167) are an example of a visualization technique that shows time intervals (see Figure 3.23).

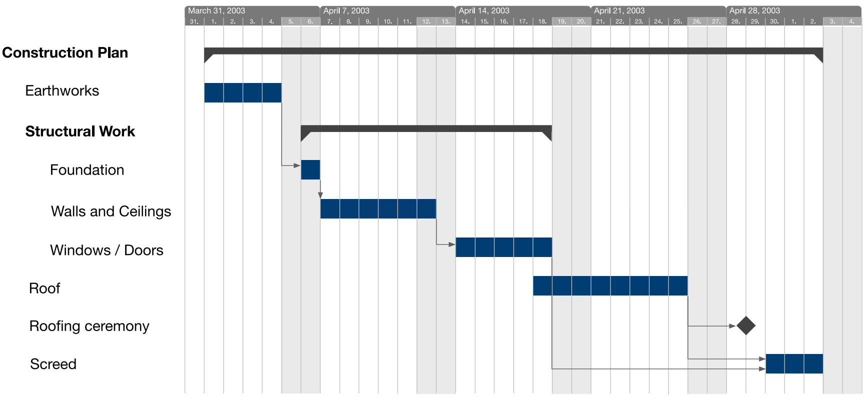


Fig. 3.23: Gantt chart. Example of a visualization technique capable of representing intervals. The tasks of a project plan are displayed as a list in the left part of the diagram. For each task, a horizontal bar (timeline) displays the extent of the task in time.

Relations between time primitives Between individual time primitives relations might exist, such as *before* and *after*. As presented by Peuquet (1994), these relations can be specified in different ways. We will present these relations in terms of topology, i.e., relative locations of time elements. Depending on the time primitives used, different relations make sense.

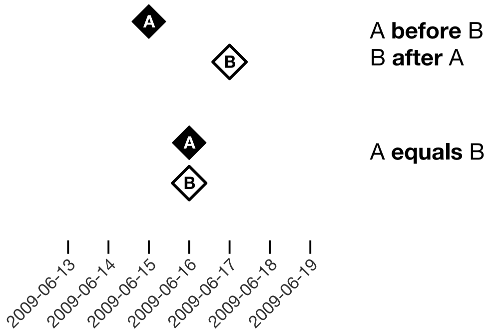


Fig. 3.24 Instant relations. Instants can be related in three different ways.

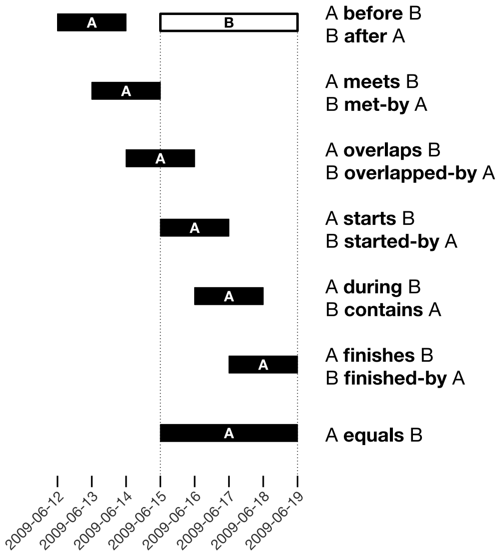


Fig. 3.25 Interval relations. Instants can be related in thirteen different ways.

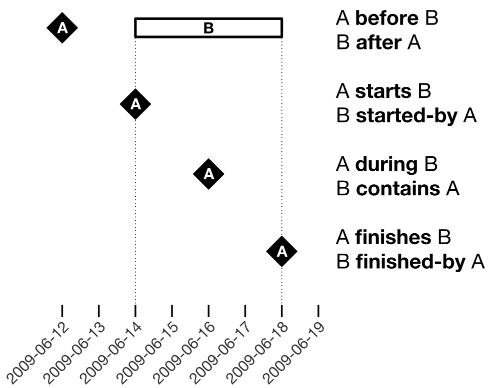


Fig. 3.26 Instant+interval relations. Instants and intervals can be related in eight different ways.

Between two instants A and B, three relationships are possible (see Figure 3.24). Either A is *before* B, A is *after* B, or A *equals* B (i.e., A and B are at the same time). For relations between time intervals, things get more complex. Allen (1983) defined a set of thirteen basic relations that are very common in time modeling (see Figure 3.25). These are A *before* B or B *after* A (i.e., interval A ends before interval B starts), A *meets* B or B *met-by* A (i.e., interval A ends right when interval B starts), A *overlaps* B or B *overlapped-by* A (i.e., intervals A and B overlap whereas interval A ends during interval B), A *starts* B or B *started-by* A (i.e., intervals A and B start at the same time but interval A ends earlier), A *during* B (i.e., interval A starts later and ends earlier as interval B), A *finishes* B or B *finished-by* A (i.e., interval A and B end at the same time but interval A starts later), and A *equals* B (i.e., both intervals start and end at the same time). When looking at relations between an instant A and an interval B, eight options exist (see Figure 3.26). Either, A *before* B or B *after A* (i.e., instant A is before the start of interval B), A *starts* B or B *started-by* A (i.e., instant A and the start of interval B are the same), A *during* B or B *contains* A (i.e., instant A is after the start and before the end of interval B), or A *finishes* B or B *finished-by* A (i.e., instant A and the end of interval B are the same).

These relationships are important concepts, especially when reasoning about time. Furthermore, the set of possible relations is determined by further design aspects.

Determinacy: determinate vs. indeterminate Uncertainty is another important aspect when considering time-oriented data. If there is no complete or exact information about time specifications or if time primitives are converted from one granularity to another, uncertainties are introduced and have to be dealt with. Therefore, the *determinacy* of the given time specification needs to be considered. A determinate specification is present when there is complete knowledge of all temporal aspects. Prerequisites for determinate specification are either a continuous time domain or only a single granularity within a discrete time domain. Information that is temporally indeterminate can be characterized as *don't know when* information, or more precisely, *don't know exactly when* information (see Jensen et al., 1998). Examples of this are inexact knowledge (e.g., "time when the earth was formed"), future planning data (e.g., "it will take 2-3 weeks"), or imprecise event times (e.g., "one or two days ago"). Notice that temporal indeterminacy as well as the relativity of references to time are mainly qualifications of statements rather than of the events they denote. Indeterminacy might be introduced by explicit specification (e.g., earliest beginning and latest beginning of an interval) or is implicitly present in the case of multiple granularities. Consider for example the statement "Activity A started on June 14, 2009 and ended on June 17, 2009" – this statement can be modeled by the beginning instant "June 14, 2009" and the end instant "June 17, 2009" both at the granularity of *days*. If we look at this interval from a granularity of *hours*, the interval might begin and end at any point in time between 0 a.m. and 12 p.m. of the specified day (see Figure 3.27).

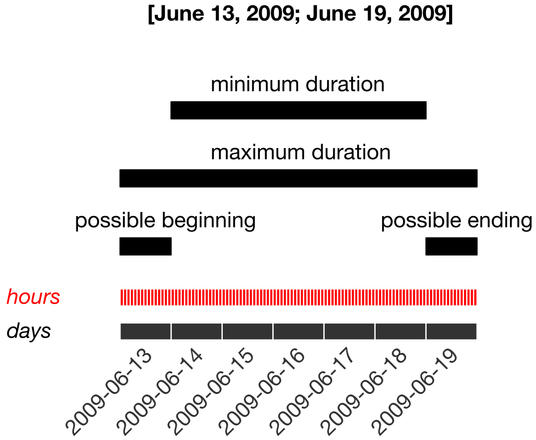


Fig. 3.27: Indeterminacy. Implicit indeterminacy when representing the interval [June 14, 2009; June 17, 2009] that is given at a granularity of *days* on a finer granularity of *hours*.

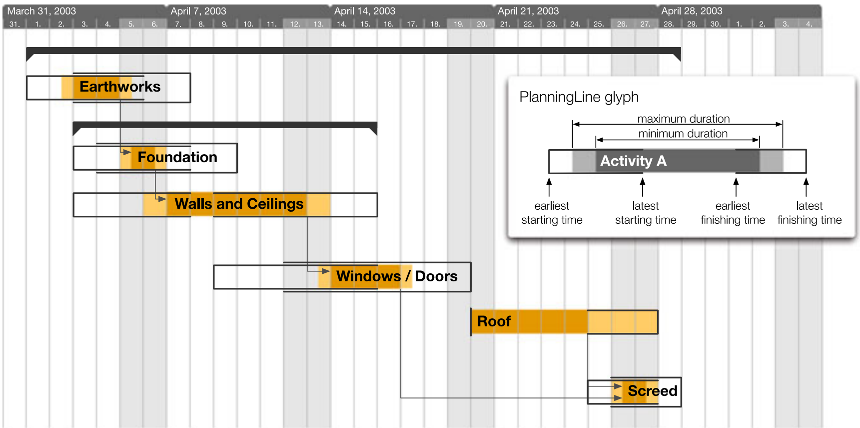


Fig. 3.28: PlanningLines allow the depiction of temporal indeterminacies via a glyph consisting of two encapsulated bars representing minimum and maximum duration, that are bounded by two caps that represent the start and end intervals.
Source: Adapted from Aigner et al. (2005).

Examples of time models that consider temporal indeterminacy are HMAP[6] by Combi and Pozzi (2001) and the time model underlying the time annotations used in the medical treatment plan specification language Asbru by Shahar et al. (1998). A visualization technique capable of depicting temporal indeterminacy is for example *PlanningLines* (see Figure 3.28 and ↪ p. 172).

3.2 Characterizing Data

After discussing the question of modeling the time domain itself, we now move on to the question of characterizing time-oriented data. When we speak of time-oriented data, we basically mean data that are somehow connected to time. More precisely, we consider data values that are associated with time primitives.

The available modeling approaches are manifold and range from considering continuous to discrete data models (see Tory and Möller, 2004). In the former case, time is seen as an observational space and data values are given relative to it (e.g., a time-series in form of time-value pairs (t,v)). For the latter, data are modeled as objects or entities which have attributes that are related to time (e.g., calendar events with attributes *beginning* and *end*). Moreover, certain analytic situations even demand domain transformations, such as a transformation from the time domain into the frequency domain (Fourier transformation).

A useful concept for modeling time-oriented data along cognitive principles is the *pyramid framework* by Mennis et al. (2000) (see Figure 3.29), which has already been mentioned briefly in Section 1.1. The model is based on the three perspectives location (*where* is it?), time (*when* is it?), and theme (*what* is it made of?) at the level of data. Derived interpretations of these data aspects form objects (*what* is it?) on the cognitively higher level of knowledge, along with their taxonomy (classification; super-/subordinate relationships) and partonomy (interrelationships; part-whole relationships).

Depending on the phenomena under consideration and the purpose of the analysis, different points of view can be taken. An example of this would be considering distinct conceptual entities that are related to time (objects) vs. the observation of a continuous phenomenon, like temperature over time (values). There cannot be a single model that is ideal for all kinds of applications. However, certain fundamental design alternatives can be identified to characterize time-oriented data. In the context of this book, we focus on the data component, i.e., the lower part of the pyramid framework as depicted in Figure 3.29.

Scale: quantitative vs. qualitative Due to the given data domain we distinguish between quantitative and qualitative variables. Quantitative variables are based on a metric (discrete or continuous) range that allows numeric comparisons. In contrast, the scale of qualitative variables includes an unordered (nominal) or ordered (ordi-

[6] The word HMAP is not an abbreviation, but it is the transliteration of the ancient Greek poetical word *day*.

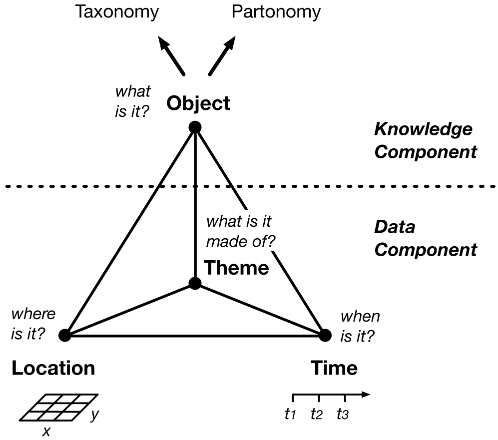


Fig. 3.29 Pyramid framework. Data are conceptualized along the three perspectives of location, time, and theme. Derived interpretations form objects on the cognitively higher level of knowledge. *Source: Adapted from Mennis et al. (2000).*

nal) set of data values. It is of fundamental importance to consider the characteristics of the data scale to design appropriate visual representations.

Frame of reference: abstract vs. spatial Furthermore, it makes sense to distinguish abstract and spatial data. By abstract data we mean a data model that does not include the *where* aspect with regard to the pyramid framework, i.e., abstract data are not connected per se to some spatial location. In contrast to this, spatial data contain an inherent spatial layout, i.e., the underlying data model includes the *where* aspect. The distinction between abstract and spatial data reflects the way the time-oriented data should be visualized. For spatial data, the inherent spatial information can be exploited to find a suitable mapping of data to screen. The *when* aspect has to be incorporated into that mapping, where it is not always easy to achieve an emphasis on the time domain. For abstract data, no a priori spatial mapping is given. Thus, first and foremost an expressive spatial layout has to be found. This spatial layout should be defined such that the time domain is exposed.

Kind of data: events vs. states This criterion refers to the question of whether events or states are dealt with. Events, on the one hand, can be seen as markers of state changes, like for example the departure of a plane. States, on the other hand, can be characterized as phases of continuity between events (e.g., plane is in the air). As one can see, states and events are two sides of the same coin. However, it should be clearly communicated whether states or events, or even a combination of both, are visualized.

Number of variables: univariate vs. multivariate This criterion concerns the number of time-dependent variables. In principle, it makes a difference if we have to represent data where each time primitive is associated with only one single data value (i.e., univariate data) or if multiple data values (i.e., multivariate data) must be represented. Compared to univariate data, for which many methods have been developed, the range of methods applicable for multivariate data is significantly smaller.

3.3 Relating Data & Time

Aspects regarding time dependency of data have been extensively examined in the field of temporal databases. Here, we adapt the notions and definitions developed in that area (see Steiner, 1998; Liu and Özsu, 2009). According to them, any dataset is related to two temporal domains:

- internal time $\mathfrak{T}_i$ and
- external time $\mathfrak{T}_e$.

Internal time is considered to be the temporal dimension inherent in the data model. Internal time describes when the information contained in the data is valid. Conversely, *external time* is considered to be extrinsic to the data model. The external time is necessary to describe how a dataset evolves in (external) time. Depending on the number of time primitives in internal and external time, time-related datasets can be classified as shown in Figure 3.30.

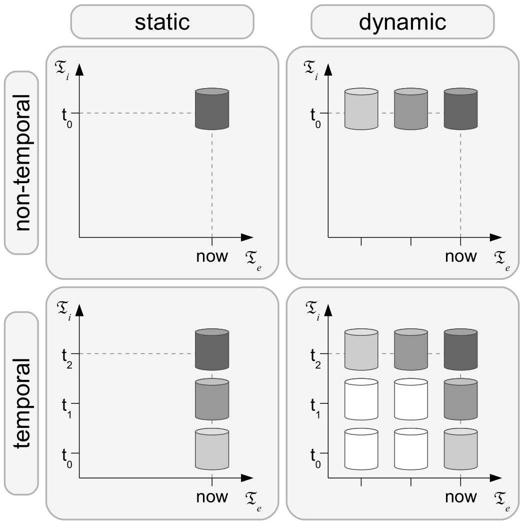


Fig. 3.30 Temporal characteristics of time related data. A dataset is related to the two temporal domains internal time $\mathfrak{T}_i$ and external time $\mathfrak{T}_e$. *Source: Adapted from Steiner (1998).*

Static non-temporal data If both internal and external time are each comprised of only one temporal element, the data are completely independent of time. A fact sheet containing data about the products offered by a company is an example of static non-temporal data. This kind of data is not addressed in this book.

Static temporal data If the internal time contains more than one time primitive, while the external time contains only one, then the data can be considered dependent on time. Since the values stored in the data depend on the internal time, static temporal data can be understood as an historical view of how the real world or some model looked at the various elements of internal time. Common time-series are a prominent example of static temporal data. Most of today's visualization approaches that explicitly consider time as a special data dimension address static temporal data, for instance the TimeSearcher (see Hochheiser and Shneiderman, 2004 and ↪ p. 188).

Dynamic non-temporal data If the internal time contains only one, but the external time is composed of multiple time primitives, then the data depend on the external time. To put it simply, the data change over time, i.e., they are dynamic. Dynamic data that change at high rate are often referred to as *streaming data*. Since the internal time is not considered, only the current state of the data is preserved; an historical view is not maintained. There are fewer visualization techniques available that explicitly focus on dynamic non-temporal data. These techniques are mostly applied in monitoring scenarios, for instance to visualize process data (see Matković et al., 2002 and ↪ p. 222). However, since internal time and external time can usually be mapped from one to the other, some of the known visualization techniques for static temporal data can be applied for dynamic non-temporal data as well.

Dynamic temporal data If both internal and external time are comprised of multiple time primitives, then the data are considered to be bi-temporally dependent. In other words, the data contain variables depending on (internal) time, and the actual state of the data changes over (external) time. Usually, in this case, internal and external time are strongly coupled and can be mapped from one to the other. Examples of such data could be health data or climate data that contain measures depending on time (e.g., daily number of cases of influenza or daily average temperature), and that are updated every 24 hours with new data records of the passed day. An explicit distinction between internal and external time is usually not made by current visualization approaches, because considering both temporal dimensions for visualization is challenging. Therefore, dynamic temporal data are beyond the scope of this book.

3.4 Summary

In this chapter, we structured and specified the characteristics of time and time-oriented data. We approached this from three perspectives: First, we characterized time and time models by discussing the related design aspects and abstractions. Second, we presented relevant data aspects and third, we analyzed different types of time-orientation. Figure 3.31 summarizes these perspectives and their corresponding aspects.

The first perspective mainly addresses time and the complexity of modeling time. Therefore, we needed to clarify the concepts of scale, scope, arrangement, and viewpoints in order to specify the design space, and to define granularity and calendars, time primitives, as well as temporal relations and determinacy of temporal elements in order to specify the abstractions.

The second perspective focuses on relevant aspects of the data variables using the understanding of time models explained above. This resulted in the definitions of

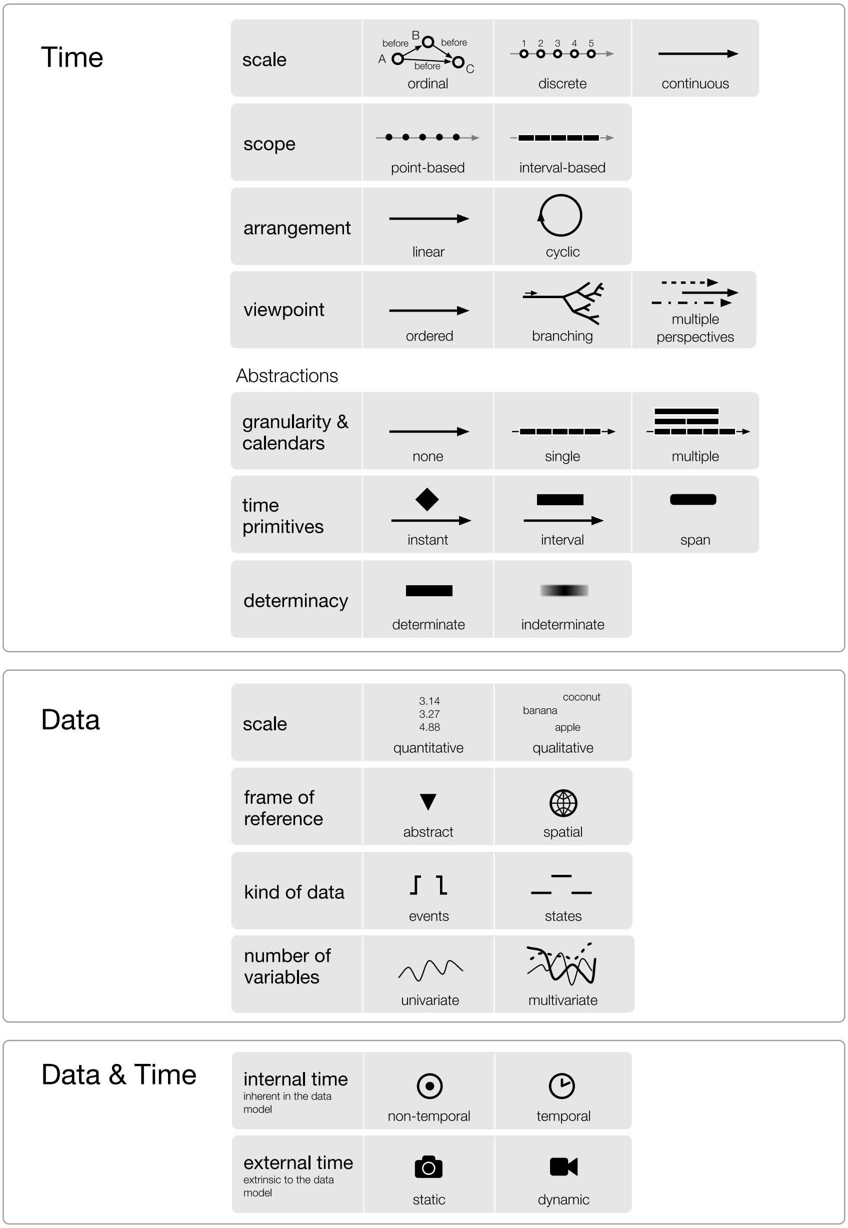


Fig. 3.31: Design aspects of time-oriented data.

scale, frame of reference, kind of data, and number of variables. The third perspective helps to identify how time and data are related in a particular setting. Therefore, we took a look at how data variables are associated with time elements using the distinction of internal and external time. All of these aspects need to be considered when visualizing and analyzing data variables over time.

In terms of characterizing data, we mainly focused on the temporal relations of data variables, i.e., relations between time primitives as well as between data and internal and external time. Due to our focus on time aspects in this book, we did not discuss other issues regarding data structures and the relationships between different data variables that are not strictly related to time. We are aware that the relationships between data variables are of importance, too. However, these aspects have been widely discussed in database and data modeling theories. Also, many useful modeling alternatives and reference models have been developed and can be adopted, such as continuous models using scalars, vectors, or tensors, etc. (see Wright, 2007) or discrete models using structures like trees, graphs, etc. (see Shneiderman, 1996).

We took this hard road – which required us to consider a number of characterizations and modeling concerns – because we are convinced that we can only develop visualization methods that successfully support people in carrying out their tasks if we have a clear understanding of what the data look like. In the next chapter we explore the aspect of visualization design in more depth.

References

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.

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

Bettini, C., Jajodia, S., and Wang, X. S. (2000). *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer, Secaucus, NJ, USA, 1st edition.

Cleveland, W. (1993). *Visualizing Data*. Hobart Press, Summit, NJ, USA.

Combi, C. and Pozzi, G. (2001). HMAP - A Temporal Data Model Managing Intervals with Different Granularities and Indeterminacy from Natural Language Sentences. *The VLDB Journal*, 9(4):294–311.

Dyreson, C. E., Evans, W. S., Lin, H., and Snodgrass, R. T. (2000). Efficiently Supporting Temporal Granularities. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):568–587.

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*, pages 40–62. Oxford University Press, New York, NY, USA.

Furia, C. A., Mandrioli, D., Morzenti, A., and Rossi, M. (2010). Modeling Time in Computing: A Taxonomy and a Comparative Survey. *ACM Computing Surveys*, 42:6:1–6:59.

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

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

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

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.

Jensen, C. S., Dyreson, C. E., Böhlen, M. H., Clifford, J., Elmasri, R., Gadia, S. K., Grandi, F., Hayes, P. J., Jajodia, S., Käfer, W., Kline, N., Lorentzos, N. A., Mitsopoulos, Y. G., Montanari, A., Nonen, D. A., Peressi, E., Pernici, B., Roddick, J. F., Sarda, N. L., Scalas, M. R., Segev, A., Snodgrass, R. T., Soo, M. D., Tansel, A. U., Tiberio, P., and Wiederhold, G. (1998). The Consensus Glossary of Temporal Database Concepts – February 1998 Version. In Etzion, O., Jajodia, S., and Sripada, S., editors, *Temporal Databases: Research and Practice*, volume 1399 of *Lecture Notes in Computer Science*, pages 367–405. Springer, Berlin, Germany.

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.

Lenz, H. (2005). *Universalgeschichte der Zeit*. Marixverlag, Wiesbaden, Germany.

Liu, L. and Özsu, M. (2009). *Encyclopedia of Database Systems*. Springer, Berlin, Heidelberg, Germany.

Matković, K., Hauser, H., Sainitzer, R., and Gröller, E. (2002). Process Visualization with Levels of Detail. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 67–70, Los Alamitos, CA, USA. IEEE Computer Society.

Mennis, J. L., Peuquet, D., and Qian, L. (2000). A Conceptual Framework for Incorporating Cognitive Principles into Geographical Database Representation. *International Journal of Geographical Information Science*, 14(6):501–520.

Mintz, D., Fitz-Simons, T., and Wayland, M. (1997). Tracking Air Quality Trends with SAS/GRAPH. In *Proceedings of the 22nd Annual SAS User Group International Conference (SUGI97)*, pages 807–812, Cary, NC, USA. SAS.

Peuquet, D. J. (1994). It's about Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. *Annals of the Association of American Geographers*, 84(3):441–461.

Peuquet, D. J. (2002). *Representations of Space and Time*. The Guilford Press, New York, NY, USA.

Shahar, Y., Miksch, S., and Johnson, P. (1998). The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. *Artificial Intelligence in Medicine*, 14(1-2):29–51.

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.

Steiner, A. (1998). *A Generalisation Approach to Temporal Data Models and their Implementations*, PhD thesis, Swiss Federal Institute of Technology.

Tory, M. and Möller, T. (2004). Rethinking Visualization: A High-Level Taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 151–158, Los Alamitos, CA, USA. IEEE Computer Society.

Whitrow, G. J., Fraser, J. T., and Soulsby, M. P. (2003). *What is Time?: The Classic Account of the Nature of Time*. Oxford University Press, New York, NY, USA.

Wolter, M., Assenmacher, I., Hentschel, B., Schirski, M., and Kuhlen, T. (2009). A Time Model for Time-Varying Visualization. *Computer Graphics Forum*, 28(6):1561–1571.

Wright, H. (2007). *Introduction to Scientific Visualization*. Springer, Berlin, Germany.

Chapter 4
Visualization Aspects

> The graphical method has considerable superiority for the exposition of statistical facts over the tabular. A heavy bank of figures is grievously wearisome to the eye, and the popular mind is as capable of drawing any useful lessons from it as of extracting sunbeams from cucumbers.
>
> Farquhar and Farquhar (1891, p. 55)

Many different types of data are related to time. Meteorological data, financial data, census data, medical data, simulation data, news articles, photo collections, or project plans, to name only a few examples, all contain temporal information. In theory, because all these data are time-oriented, they should be representable with one and the same visualization approach. In practice, however, the data exhibit specific characteristics and hence each of the above examples requires a dedicated visualization. For instance, stock exchange data can be visualized with flocking boids (see Vande Moere, 2004 and ↪ p. 223), census data can be represented with Trendalyzer (see Gapminder Foundation, 2010 and ↪ p. 220), and simulation data can be visualized efficiently using MOSAN (see Unger and Schumann, 2009 and ↪ p. 209). News articles (or keywords therein) can be analyzed with ThemeRiver (see Havre et al., 2002 and ↪ p. 197) and project plans can be made comprehensible with PlanningLines (see Aigner et al., 2005 and ↪ p. 172). Finally, meteorological data are visualized for us in the daily weather show. Apparently, this list of visualizations is not exhaustive. The aforementioned approaches are just examples from a substantial body of techniques that recognize the special role of the dimension of time. We shall complete this list in a rich survey of information visualization techniques in Chapter 7.

Besides these dedicated techniques, 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 (natural or real numbers), general visualization frameworks like the Xmdv-Tool, Visage, or Polaris (see Ward, 1994; Kolojejchick et al., 1997; Stolte et al., 2002) as well as standard visualization techniques like parallel coordinates by Inselberg and Dimsdale (1990) or more or

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_4,

less sophisticated diagrams and charts, as surveyed by Harris (1999), are applicable 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 line plot). However, in many cases, time is treated just as one quantitative variable among many others, not more, not less. Therefore, generic approaches usually do not support establishing a direct visual connection between multiple variables and the time axis, they do not communicate the specific aspects of time (e.g., the different levels of temporal granularity), and they are limited in terms of direct interactive exploration and browsing of time-oriented data, which are essential for a successful visual analysis.

The bottom line is that time must be specifically considered to support the visual analysis. Different types of time-oriented data need to be visualized with dedicated methods. As the previous examples suggest, a variety of concepts for analyzing time-oriented data are known in the literature (see for example the work by Silva and Catarci, 2000; Müller and Schumann, 2003; Aigner et al., 2008). This variety makes it difficult for researchers to assess the current state of the art, and for practitioners to choose visualization approaches most appropriate to their data and tasks.

What is required is a systematic view of the visualization of time-oriented data (see Aigner et al., 2007). In this chapter we will develop such a systematic view. The different design options derived from the systematic view will be discussed and illustrated by a number of visualization examples.

4.1 Characterization of the Visualization Problem

In the first place, we need a structure to organize our systematic view. But instead of using formal or theoretical constructs, we decided to present a structure that is geared to three practical questions that are sufficiently specific for researchers and at the same time easy to understand for practitioners:

1. *What* is presented? – *Time & data*
2. *Why* is it presented? – *User tasks*
3. *How* is it presented? – *Visual representation*

Because any visualization originates from some data, the first question addresses the structure of time and the data that have been collected over time. The motivation for generating a visualization is reflected by the second question. It relates to the aim of the visualization and examines the tasks carried out by the users. How the data are represented is covered by the third question. The following sections will provide more detailed explanations and refinements for each of these questions.

4.1.1 What? – Time & Data

It goes without saying that the temporal dimension itself is a crucial aspect that any visualization approach for representing time and time-oriented data has to consider. It is virtually impossible to design effective visual representations without knowledge about the characteristics of the given data and time domain. The characteristics of time and data as well as corresponding design aspects have already been explained in detail in Sections 3.1 and 3.2. Here, we will only briefly summarize these aspects.

Characteristics of time The following list briefly reiterates the key criteria of the dimension of time that are relevant for visualization:

- *Scale – ordinal vs. discrete vs. continuous*: In an ordinal time model, only relative order relations are present (e.g., before, during, after). In discrete and continuous domains, temporal distances can also be considered. In discrete models, time values can be mapped to a set of integers based on a smallest possible unit (e.g., seconds). In continuous models, time values can be mapped to the set of real numbers, and hence, between any two points in time, another point can be inserted.
- *Scope – point-based vs. interval-based*: Point-based time domains have basic elements with a temporal extent equal to zero. Thus, no information is given about the region between two points in time. Interval-based time domains relate to subsections of time having a temporal extent greater than zero.
- *Arrangement – linear vs. cyclic*: Linear time corresponds to an ordered model of time, i.e., time proceeds from the past to the future. Cyclic time domains are composed of a finite set of recurring time elements (e.g., the seasons of the year).
- *Viewpoint – ordered vs. branching vs. multiple perspectives*: Ordered time domains consider things that happen one after the other. In branching time domains, multiple strands of time branch out and allow for description and comparison of alternative scenarios, but only one path through time will actually happen (e.g., in planning applications). Multiple perspectives facilitate simultaneous (even contrary) views of time (as for instance required to structure eyewitness reports).

In addition to these criteria, which describe the dimension of time, aspects regarding the presence or absence of different levels of granularity, the time primitives used to relate data to time, and the determinacy of time elements are relevant (see Section 3.1 in the previous chapter).

Characteristics of time-oriented data Like the time domain, the data have a major impact on the design of visualization approaches. Let us briefly reiterate the key criteria for data that are related to time:

- *Scale – quantitative vs. qualitative*: Quantitative data are based on a metric scale (discrete or continuous). Qualitative data describe either unordered (nominal) or ordered (ordinal) sets of data elements.

- *Frame of reference – abstract vs. spatial*: Abstract data (e.g., a bank account) have been collected in a non-spatial context and are not per se connected to some spatial layout. Spatial data (e.g., census data) contain an inherent spatial layout, e.g., geographical positions.
- *Kind of data – events vs. states*: Events, on the one hand, can be seen as markers of state changes, whereas states, on the other hand, characterize the phases of continuity between events.
- *Number of variables – univariate vs. multivariate*: Univariate data contain only one data value per temporal primitive, whereas in the case of multivariate data each temporal primitive holds multiple data values.

These primary categories form a basis for finding answers to the *what* question of our systematic view. But having characterized what has to be visualized is just the first step. The subsequent step is to focus on the *why* question.

4.1.2 Why? – User Tasks

It is commonly accepted that software development has to start with an analysis of the problem domain users work in (see Hackos and Redish, 1998; Courage and Baxter, 2005). This applies accordingly to the development of visualization solutions for time-oriented data.

To specify the problem domain, so-called task models are widely used in the related field of human-computer interaction (see Constantine, 2003). A prominent example of such task models is the ConcurTaskTree (CTT) by Paternò et al. (1997). It describes a hierarchical decomposition of a goal into tasks and subtasks. Four specific types of tasks are supported in the CTT notation: abstract tasks, interaction tasks, user tasks, and application tasks. Abstract tasks can be further decomposed into subtasks (including abstract subtasks). Leaf nodes are always interaction tasks, user tasks, or application tasks. They have to be carried out either by the user, by the application system, or by interaction between user and system. The CTT notation is enriched with a set of temporal operators that define temporal relationships among tasks and subtasks (e.g., independent concurrency, concurrency with information exchange, disabling, enabling). Usually, CTT models are constructed manually by a domain expert, and mostly for the purpose of driving automatic user interface generation (see for example Paternò and Santoro, 2002). First approaches have begun to use this notation for visualization purposes. For instance, Winckler et al. (2004) apply the CTT notation for structured tests and for the evaluation of interactive visualization techniques.

In the visualization domain, however, tasks are usually given at a lower level in the form of informal verbal lists only. An accepted low-level task description specifically addressing the temporal domain has been introduced by McEachren (1995). The tasks are defined by a set of important questions that users might seek to answer with the help of visual representations:

- *Existence of data element*
  Question: Does a data element exist at a specific time?
  Starting point: time point or time interval
  Search for: data element at that time
  Example: "Was a measurement made in June, 1960?"
- *Temporal location*
  Question: When does a data element exist in time?
  Starting point: data element
  Search for: time point or time interval
  Example: "When did the Olympic Games in Vancouver start?"
- *Time interval*
  Question: How long is the time span from beginning to end of the data element?
  Starting Point: data element
  Search for: duration, i.e., length of time of a data element from its beginning to its end
  Example: "How long was the processing time for dataset A?"
- *Temporal pattern*
  How often does a data element occur?
  Starting point: interval in time
  Search for: frequency of data elements within a certain portion of time and based on this the detection of a pattern
  Example: "How often was Jane sick last year?"
- *Rate of change*
  Question: How fast is a data element changing or how much difference is there from data element to data element over time?
  Starting point: data element
  Search for: magnitude of change over time
  Example: "How did the price of gasoline vary in the last year?"
- *Sequence*
  Question: In what order do data elements occur?
  Starting point: data elements
  Search for: temporal order of different data elements
  Example: "Did the explosion happen before or after the car accident?"
- *Synchronization*
  Question: Do data elements exist together?
  Starting point: data elements
  Search for: occurrence at the same point or interval in time
  Example: "Is Jill's birthday on Easter Monday this year?"

This list of tasks covers two basic cases. First, having at hand one or more data values, the user is searching for time primitives that exhibit these values, and second, having at hand one or more time primitives, the user seeks to discern the data values associated with them. This reflects the well-established distinction between *identification* (i.e., looking for data values) and *localization* (i.e., looking for when and where in time and space).

This distinction is also the basis for the formal task model by Andrienko and Andrienko (2006). They describe tasks using two basic notions: *references*, which constitute the domain (spatial or temporal) in which the data values have been collected, and *characteristics*, which are the data values that were collected. On the first level, the Andrienko model is subdivided into two classes of tasks: elementary and synoptic tasks. *Elementary* tasks address individual data elements. This may include individual values, but also individual groups of data. The main point here is that data are taken into account separately; they are not considered as a whole. *Synoptic* tasks, on the other hand, involve a general view and consider sets of values or groups of data in their entirety.

Elementary tasks are further divided into lookup, comparison, and relation seeking. The *lookup* task includes direct and inverse lookup, which stand for searching for data values and searching for points in space and time, respectively. *Relation seeking* tasks address the search for occurrences of relations specified between data characteristics or references, for example, looking for courses that are scheduled on Mondays. In a broader sense, *comparison* can also be seen as relation seeking, but the relations to be determined are not specified beforehand. Direct comparison tasks interrelate characteristics, whereas inverse comparison tasks interrelate references.

Synoptic tasks are further divided into descriptive and connectional tasks. *Descriptive* tasks specify the properties of either a set of references or a set of characteristics. The first case belongs to the group of identification tasks. Here, a set of references is given, and the task is to find a pattern that describes the behavior of the given reference points. The second case belongs to the group of localization tasks. Here, a concrete pattern is given, and the task is to search for those reference points in time and space that exhibit the pattern. Besides specifying the properties of a set of characteristics or references, the comparison of those sets is highly relevant. As in the case of elementary tasks, we have to distinguish between direct and inverse comparison tasks, depending on whether sets of references or sets of characteristics are compared. Moreover, the synoptic relation seeking task considers two sets of characteristics or references to come up with relationships between these sets.

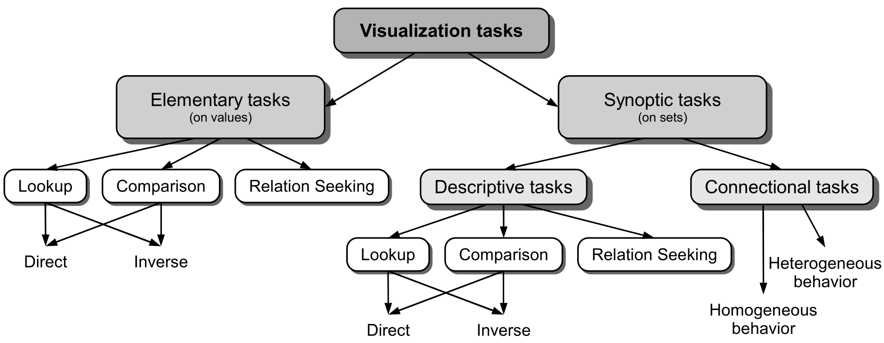


Fig. 4.1: Taxonomy of visualization tasks.

In contrast to descriptive tasks, *connectional* tasks establish connections between at least two sets, taking into account the relational behavior of two or more variables. Depending on the set of underlying references – either variables are considered over the same set or over different sets of references – homogeneous and heterogeneous behavior tasks are distinguished.

To illustrate how the different tasks of the Andrienko model are related, we arranged them into a task taxonomy. Figure 4.1 shows more clearly how the visualization tasks are organized. The quasi-hierarchical structure of this taxonomy allows the later refinement of classes of tasks depending on application-specific needs. The following list provides illustrative examples of the different types of tasks:

Elementary tasks:

- *Direct lookup:* What was the price of Google stocks on January 14?
- *Inverse lookup:* On which day(s) was the lowest stock price for Amazon in 2010?
- *Direct comparison:* Compare the stock prices of Sun and Oracle on January 14.
- *Inverse comparison:* Did the price of Apple stocks reach $200 before or after January 14?
- *Relation seeking:* On which days was the price of Adobe stocks higher than the price of AOL stocks?

Synoptic tasks:

- *Direct lookup (pattern definition):* What was the trend of Oracle stocks during January?
- *Inverse lookup (pattern search):* Find months in which the price of Novell stocks decreased.
- *Direct (pattern) comparison:* Compare the behavior of the stock price of Hewlett-Packard in January and June.
- *Inverse (pattern) comparison:* How is a decreasing trend of Dell stocks related to the period of summer vacation?
- *Relation seeking:* Find two contiguous months with opposite trends in the stock price of Lenovo.
- *Homogeneous behavior:* Is the behavior of Nokia stocks influencing the behavior of Motorola stocks?
- *Heterogeneous behavior:* Do the phases of the moon influence the behavior of Intel stocks?

From a practical perspective, the verbal task descriptions by McEachren (1995) are very helpful because they are easy to understand. They can serve as a guideline when designing visual representations of time and time-oriented data. However, when shifting to a more scientific or theoretical point of view, a more formal notation is desirable. In order to go beyond the guideline character and to automate the design process, formal task descriptions are indispensable. Andrienko and Andrienko (2006) made a significant contribution in this regard. Later in this chapter, we will examine the impact that user tasks (i.e., the *why* aspect) can have on the visualization design. But before, we shall complete the description of visualization aspects by focusing on the *how* perspective.

4.1.3 How? – Visual Representation

The answers to the questions what the data input is and why the data are analyzed very much determine the answer to the last remaining question: How can time-oriented data be represented visually? More precisely, the question is how time and associated data are to be represented. Chapter 7 will show that a large variety of visual approaches provide very different answers to this question. To abstract from the subtle details of this variety, we concentrate on two fundamental criteria: the mapping of time and the dimensionality of the presentation space.

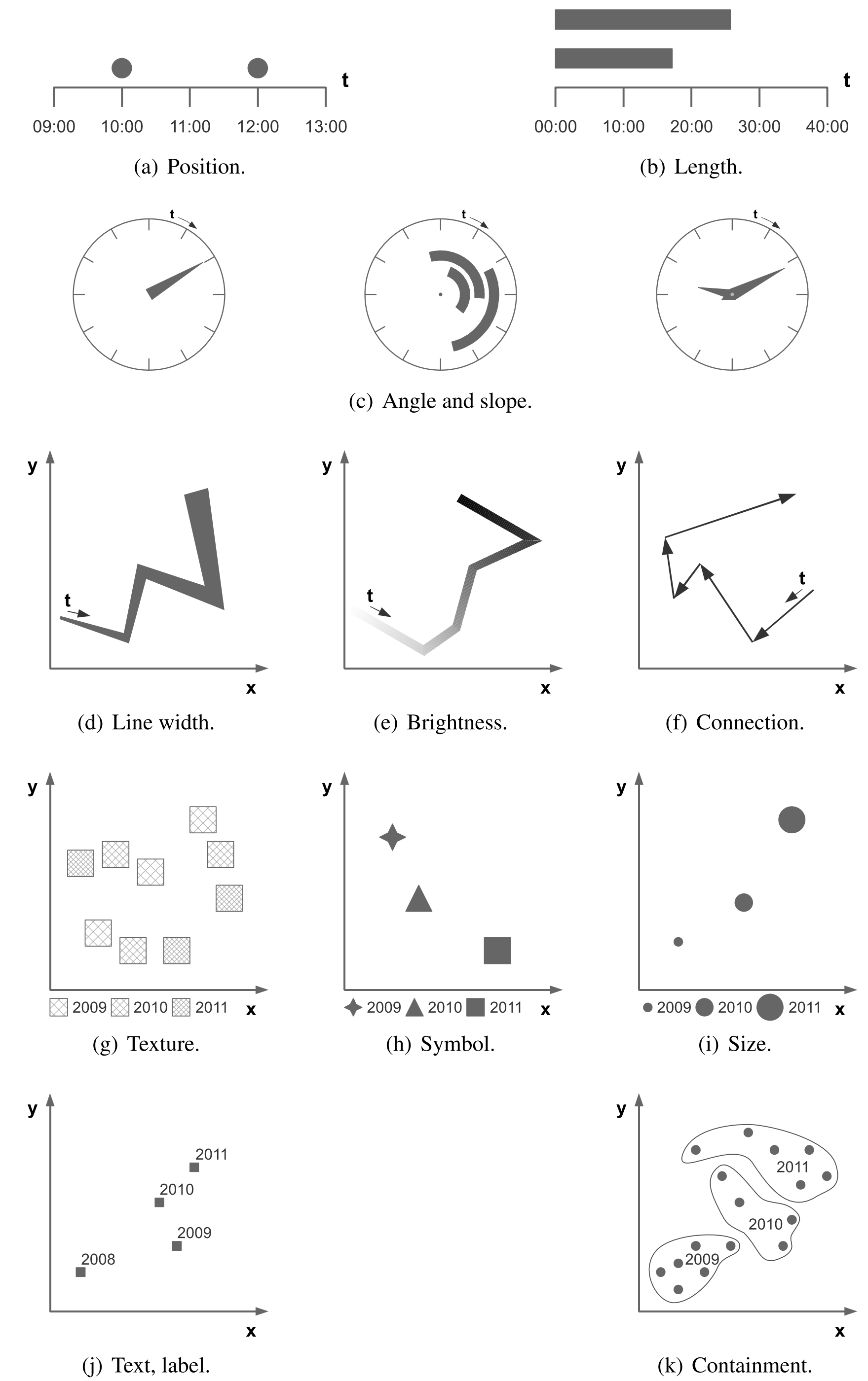
Mapping of time

Like any data variable that is to be visualized, the dimension of time has to pass the mapping step of the visualization pipeline. Usually, abstract data are made visually comprehensible by mapping them to some geometry (e.g., two-dimensional shapes) and corresponding visual attributes (e.g., color) in the presentation space. On top of this, human perception has an intrinsic understanding of time that emphasizes the progression of time, and visualization can make use of this fact by mapping the dimension of time to the dynamics of a visual representation.

So practically, there are two options for mapping time: the mapping of time to space and the mapping of time to time. When speaking of a mapping from time to space, we mean that time and data are represented in a single coherent visual representation. This representation does not automatically change over time, which is why we call such visualizations of time-oriented data *static*. In contrast to that, *dynamic* representations utilize the physical dimension of time to convey the time dependency of the data, that is, time is mapped to time. This results in visualizations that change over time automatically (e.g., slide shows or animations). Note that the presence or absence of interaction facilities to navigate in time has no influence on whether a visualization approach is categorized as static or dynamic.

Static representations There are various ways of mapping time to visual variables (see Bertin, 1983 and Figure 4.2). Most visualization approaches that implement a time-to-space mapping use one display dimension to represent the time axis. Classic examples are charts where time is often mapped to the horizontal x-axis and time-dependent variables are mapped to the vertical y-axis (see Figure 4.3). More complex mappings are possible when two or more display dimensions are used for representing time. Mappings that generate two-dimensional spirals or three-dimensional helices are examples that emphasize the cyclic character of time. The different granularities of time are often illustrated by a hierarchical subdivision of the time axis.

The actual data can then be visualized in manifold ways. It is practical to use a data mapping that is orthogonal to the mapping of time. For example point plots (↪ p. 152), line plots (↪ p. 153), or bar graphs (↪ p. 154) map data values to position or size relative to the time axis. Time-dependency is immediately perceived and can be recognized easily, which facilitates the interpretation of the temporal



(a) Position. (b) Length.

(c) Angle and slope.

(d) Line width. (e) Brightness. (f) Connection.

(g) Texture. (h) Symbol. (i) Size.

(j) Text, label. (k) Containment.

Fig. 4.2: Examples of static visual mappings of time.

Fig. 4.3: Mapping time to position. The horizontal axis of the chart encodes positions of points in time, whereas the vertical axis encodes data values.

character of the data. In fact, for quantitative variables (discrete or continuous time and data), using position or length is more efficient than using color or other visual variables such as texture, shape, or orientation (see Mackinlay, 1986). For ordinal variables, color coding is a good alternative. Each point or interval on the time axis can be visualized using a unique color from a color scale. But, as Silva et al. (2007) demonstrate, care must be taken when using color for visualization. It is absolutely mandatory that the applied color scale be capable of communicating order[1]. Only then are users able to interpret the visualization and to relate data items to their temporal context easily.

Because time is often considered to be absolute, position or length encodings are predominant and only rarely is time mapped to other visual variables. When time is interpreted relatively rather than absolutely, for instance, when considering the age of a data item or the duration between two occurrences of a data item, then visual variables such as transparency, color, and others gain importance. An example of encoding duration to color is given in Figure 4.4.

Instead of encoding data to basic graphical primitives such as points, lines, or bars that are aligned with the time axis, one can also create fully fledged visual representations and align multiple thumbnails of them along the time axis – a concept that Tufte (1983) refers to as *small multiples* (↪ p. 236). The advantage is that a single thumbnail may contain much more visual information than basic graphical primitives. But this comes at the price that the number of time primitives (i.e., the number of thumbnails) that can be shown on screen simultaneously is limited. This

[1] Borland and Taylor (2007) warn that this is not the case for the most commonly used rainbow color scale. The ColorBrewer tool by Harrower and Brewer (2003) is a good source of useful color scales.

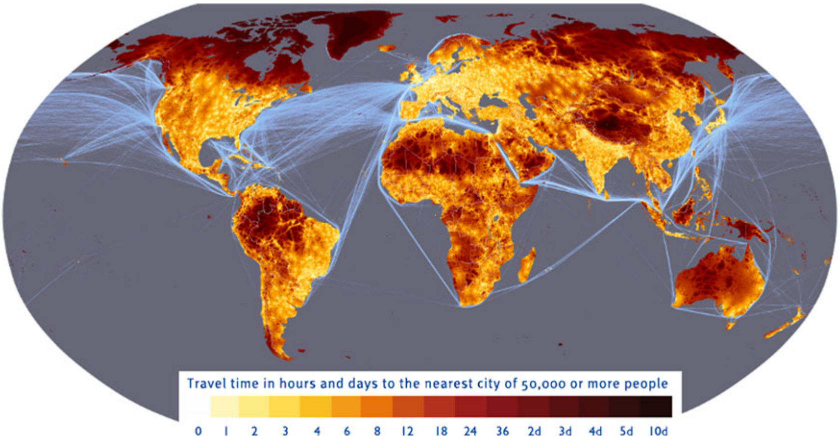


Fig. 4.4: Mapping time to color. Color encodes the time it takes to travel from a location on our planet to the nearest major city.
Source: Nelson (2008). Used with permission.

reflects the general need to find a good trade-off between the complexity of the visual encoding of time and that of the data. In Chapter 7, we will see that a variety of suitable solutions exist, each with an individually determined trade-off depending on the addressed data and tasks.

Dynamic representations In cases where much screen space is required to convey characteristics and relationships of data items (e.g., geographical maps, multivariate data visualization, visualization of graph structures), it is difficult to embed the time axis into the display space as well. As an alternative, physical time can be utilized to encode time. To this end, several visualizations (also called *frames*) are rendered successively for the time steps in the data. In theory, a one-to-one mapping between time steps and frames can be implemented, which means that the dynamic visualization represents time authentically. In practice, however, this is only rarely possible. More often, dynamic visualizations perform interpolation to compute intermediate results in cases where only few time steps are present, or perform aggregation or sampling to compress the length of an animation in cases where large numbers of time steps have to be visualized (see Wolter et al., 2009).

Self-evidently, dynamic approaches have to take human perception into account when representing a series of successively generated visualization frames. Depending on the number of images shown per second, dynamic visualizations are either perceived as animations or as slide shows. Animations usually show between 15 and 25 frames per second, while slide shows usually show a new frame every 2 to 4 seconds. On the one hand, data that contain only a few snapshots of the underlying phenomenon should preferably be represented as slide shows to avoid creating false impressions of dynamics. On the other hand, large numbers of observations of highly dynamic processes are best represented using animations, because they communi-

Fig. 4.5 A typical animation widget to control the mapping of time to time.

cate quite well the underlying dynamics in the data. Figure 4.5 gives an example of a typical VCR-like widget for controlling the mapping of time to time in an animation.

The distinction between the mapping of time to space and that of time to time is crucial, because different visualization tasks and goals are supported by these mappings. Dynamic representations are well suited to convey the general development and the major trends in the analyzed data. However, there are also critical assessments of animations used for the purpose of visualization (see Tversky et al., 2002; Simons and Rensink, 2005). Especially when larger multivariate time-series have to be visualized, animation-based approaches reach their limits. In such cases, users are often unable to follow all of the changes in the visual representation, or the animations simply take too long and users face an indigestible flood of information. This problem becomes aggravated when using animations in multiple views. On the other hand, if animations are designed well and if they can be steered interactively by the user (e.g., slow motion or fast forward), mapping the dimension of time to the physical time can be beneficial (see Robertson et al., 2008). This is not only the case from the point of view of perception, but it is also because using physical time for visual mapping implies that the spatial dimensions of the presentation space can be used exclusively to visualize the time-dependent data.

This is not the case, however, for static representations. In contrast to animations, static representations require screen real estate to represent the time axis itself and the data in an integrated fashion. On the one hand, the fact that static representations show all of the information on one screen is advantageous because one can fully concentrate on the dependency of time and data. Especially visual comparison of different parts of the time axis can be accomplished easily using static representations. On the other hand, integration of time and data in one single view tends to lead to overcrowded representations that are hard to interpret. In the face of larger time-oriented datasets, analytical methods and interaction are mandatory to avoid visual clutter.

Finally, it is worth mentioning that any (non-temporal) data visualization can be extended to a visualization for time-oriented data simply by repetition. Repetition in time leads to dynamic representations, where each frame shows a snapshot of the data and repetition in space leads to static multiple view representations (or Tufte's *small multiples*, ↪ p. 236), where each view shows an individual part of the time axis. While static representations always have to deal with the issue of finding a good layout for the views, dynamic representations encode time linearly in a straight-forward manner. Perhaps this is the reason why many visualization solutions resort to simple animations, even though these might not be the best option for the data and tasks at hand.

Dimensionality of the presentation space

The presentation space of a visualization can be either two-dimensional or three-dimensional, or 2D or 3D for short. Two-dimensional visualizations address the spatial dimensions of computer displays, that is, the x-axis and the y-axis. All graphical elements are described with respect to x and y coordinates. Dots, lines, circles, or arcs are examples of 2D geometry. The semantics of the data usually determine the layout of the geometry on screen. 3D visualizations use a third dimension, the z-axis, for describing geometry. This allows the visualization of more complex and volumetric structures. As human perception is naturally tuned to the three-dimensional world around us, 3D representations potentially communicate such structures better than 2D approaches. Since the z-axis does not physically exist on a computer display, projection is required before rendering 3D visualizations. The projection is usually transparent to the user and is commonly realized through standard computer graphics methods which require no additional effort.

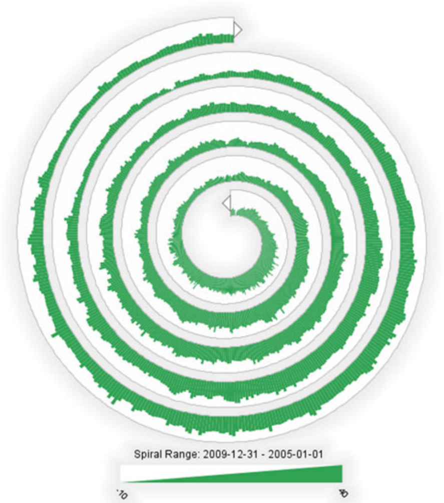


Fig. 4.6: Mapping to 2D. Spiral geometry is used to represent the time axis and data are encoded to the width of spiral segments.

Visualization approaches using a 2D presentation space usually map the time axis to a visual axis on the display (provided that the approach is not dynamic). In many cases, the time axis is aligned with either coordinate axis of the display. However, this is not necessarily always the case. Circular time axes (e.g., the spiral in Figure 4.6) use polar coordinates, which actually can be mapped to Cartesian coordinates and vice versa. It is also possible to apply affine transformations to the time axis.

Because one dimension of the display space is usually occupied for the representation of the dimension of time, the possibilities of encoding the data depending on time are restricted. One data variable can be encoded to the remaining spatial dimension of the presentation space, as for instance in a bar graph, where the x-axis encodes time and the y-axis, more precisely the height of bars, encodes a time-dependent variable. In order to visualize multiple variables further graphical attributes like shape, texture, or color can be used.

Multidimensional data, that is, data with more independent variables than just the dimension of time, are hard to visualize in 2D without introducing overlap and visual clutter. Particularly, data with a spatial frame of reference can benefit from the additional dimension available in a 3D presentation space. It is common practice to apply the so-called *space-time cube* concept (see Kraak, 2003 and ↪ p. 245), according to which the z-axis encodes time and the x- and y-axes represent two independent variables (e.g., longitude and latitude). Further variables, dependent or independent, are then encoded to color, size, shape or other visual attributes (see Figure 4.7 and ↪ p. 252).

Fig. 4.7: Mapping to 3D. Three-dimensional helices represent time axes for individual regions of a map and associated data are encoded by color.

The question of whether or not it makes sense to exploit three dimensions for visualization has been discussed at length by the research community (see Card et al., 1999). One camp of researchers argues that two dimensions are sufficient for effective data analysis. In their thinking the third dimension introduces unnecessary difficulties (e.g., information hidden on back faces, information lost due to occlusion, or information distorted through perspective projection) which 2D representations are not or are only marginally affected by. But having just two dimensions for the visual mapping might not be enough for large and complex datasets.

This is where the other camp of researchers make their arguments. They see the third dimension as an additional possibility to naturally encode further information. Undoubtedly, certain types of data (e.g., geospatial data) might even require the third dimension for expressive data visualization, because there exists a one-to-one mapping between the data dimensions and the dimensions of the presentation space.

We do not argue for either position in general. The question whether to use 2D or 3D is rather a question of which data has to be visualized and what are the analytic goals to be achieved. The application background and user preferences also influence the decision for 2D or 3D. But definitely, when developing 3D visualizations, the previously mentioned disadvantages of a three-dimensional presentation space have to be addressed (e.g., by providing ways to cope with occlusion as suggested by Elmqvist and Tsigas, 2007). Moreover, intuitive interaction techniques are mandatory and additional visual cues are usually highly beneficial.

In the previous discussion of the questions *what*, *why*, and *how* we have outlined the basic aspects that need to be considered when visualizing time and time-oriented data. In the next section, we will return to each of these aspects and show in more detail and by means of examples how the visualization design is influenced by them.

4.2 Visualization Design Examples

In the previous section, we introduced three basic questions that have to be taken into account when designing visual representations for time and time-oriented data:

1. Data level: *What* is presented?
2. Task level: *Why* is it presented?
3. Presentation level: *How* is it presented?

We will now demonstrate the close interrelation of the three levels. By means of examples we will illustrate the necessity and importance of finding answers to each of these questions in order to arrive at visual representations that allow viewers to gain insight into the analyzed data.

4.2.1 Data Level

In the first place, the characteristics of time-oriented data strongly influence the design of appropriate visual representations. Two examples will be used to demonstrate this: one is related to the time axis itself, the other will deal with the data. First, we point out how significantly different the expressiveness of a visual representation can be depending on whether the time domain is linear or cyclic. Secondly, we will illustrate that spatial time-oriented data[2] require a visualization design that is quite different from that of abstract time-oriented data, and that is usually more complex and involves making well-balanced design decisions.

[2] Commonly referred to as *spatio-temporal data*.

Time characteristics: linear vs. cyclic representation of time

Figure 4.8 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. The data exhibit a strong cyclic pattern. The leftmost image of Figure 4.8 uses a simple line plot (↪ p. 153) to visualize the data. Although peaks in time can be recognized easily when examining this representation, the cyclic behavior of the data, however, can only be guessed and it is hard to discern which cyclic temporal patterns in fact do exist. In contrast, the middle and the right image of Figure 4.8 show a circular representation that emphasizes cyclic characteristics of time-oriented data by using a spiral-shaped time axis (see Weber et al., 2001 and ↪ p. 185). For the left spiral, the cyclic pattern is not visible. This is due to the fact that the cycle length has been set to 24 days, which does not match the pattern in the data. The right spiral representation in Figure 4.8 is adequately parameterized with a cycle length of 28 days, which immediately reveals the periodic pattern present in the data. The significant difference in the number of cases of influenza reported on Sundays and Mondays, respectively, is quite obvious. We would also see this weekly pattern if we set the cycle length to 7 or 14 days, or any (low) multiple of 7.

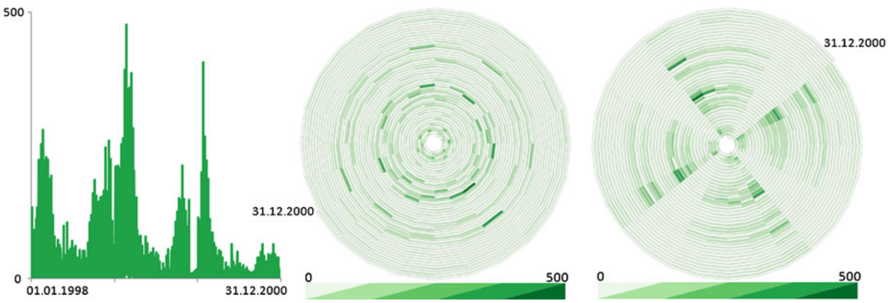


Fig. 4.8: Different insights can be gained from visual representations depending on whether the linear or cyclic character of the data is emphasized.

The example illustrates that in addition to using the right kind of representation of time (linear vs. cyclic), it is also necessary to find an appropriate parametrization of the visual representation. Interaction (see Chapter 5) usually enables users to re-parameterize the visualization, but the difficulty is to find parameter settings suitable to discover patterns in unknown datasets. Automatically animating through possible parameter values – for the spiral's cycle length in our example – is one option to assist users in finding such patterns. During the course of the animation, visual patterns emerge as the spiral's cycle length is approaching cycles in the data that match in length. Upon emergence of such patterns, the user stops the animation and can fine-tune the display as necessary. Analytical methods (see Chapter 6) can help in narrowing down the search space, which in our example means finding promising candidates with adequate cycle length (see Yang et al., 2000). Combining interactive exploration and analytical methods is helpful for finding less sharp or uncommon patterns, which are hard to distill using either approach alone.

Data characteristics: abstract data vs. spatial data

We used linear vs. cyclic time to demonstrate the impact of the characteristics of time on the visualization design. Let us now do likewise with abstract vs. spatial data to illustrate the impact of data characteristics.

Abstract data are not associated per se with a spatial visual mapping. Therefore, when designing a visual representation of such data, one can fully concentrate on aspects related to the characteristics of the dimension of time. The *ThemeRiver* approach by Havre et al. (2000) is an example of an approach in which the time aspect is focused on (↪ p. 197). The dimension of time is mapped to the horizontal display axis and multiple time-dependent variables are mapped to the thickness of individually colored currents, which form an overall visual stream of data values along the time axis. Figure 4.9 illustrates the ThemeRiver approach. Because time is the only dimension of reference in abstract time-oriented data, the visual representation can make the best of the available screen space to convey the variables' dependency on time. The full-screen design, where the ThemeRiver occupies the entire screen,

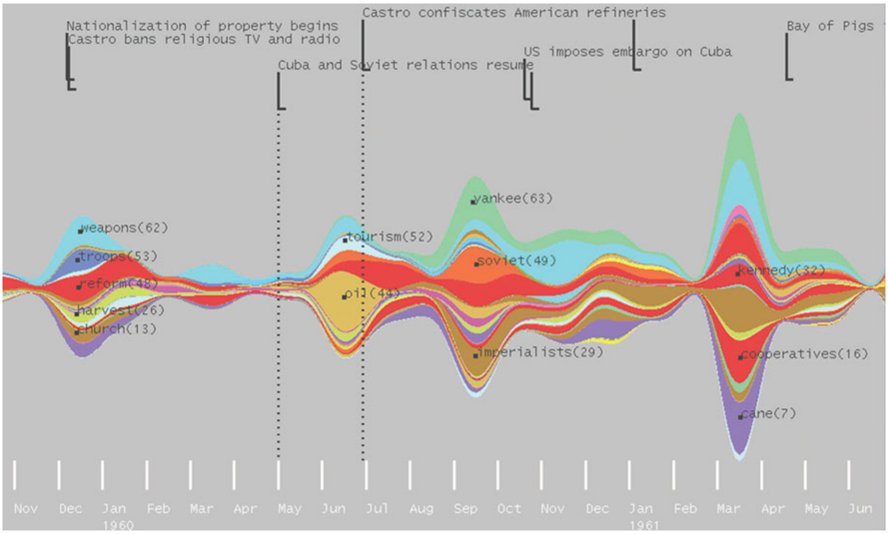


Fig. 4.9: The ThemeRiver technique is fully focused on communicating the temporal evolution of abstract time-oriented data.
Source: Havre et al. (2002). © 2002 IEEE. Used with permission.

even makes it possible to display additional information, such as a time scale below the ThemeRiver, labels in the individual currents, or extra annotations for important events in the data.

When considering time-oriented data with spatial references, the visualization design has to address an additional requirement; not only the temporal character of the data needs to be communicated, but also the spatial dependencies in the data must be revealed. Of course, this implies a conflict in which the communication of temporal aspects competes with the visualization of the spatial frame of reference for visual resources, such as screen space, visual encodings, and so forth. Providing too many resources to the visualization of aspects of time will most likely lead to a poorly represented spatial context – and vice versa. The goal is to find a well-balanced compromise. An example of such a compromise is given in Figure 4.10, where the data are visualized using ThemeRiver thumbnails superimposed on a two-dimensional map display (↪ p. 240). The position of a ThemeRiver thumbnail on the map is the visual anchor for the spatial context of the data. The ThemeRiver thumbnail itself encodes the temporal context of the data. The compromise that has been made implies that the map display is rather basic and avoids showing any geographic detail; just the borders of regions are visible. On the other hand, the ThemeRiver representation has to get along with much less screen space (compared

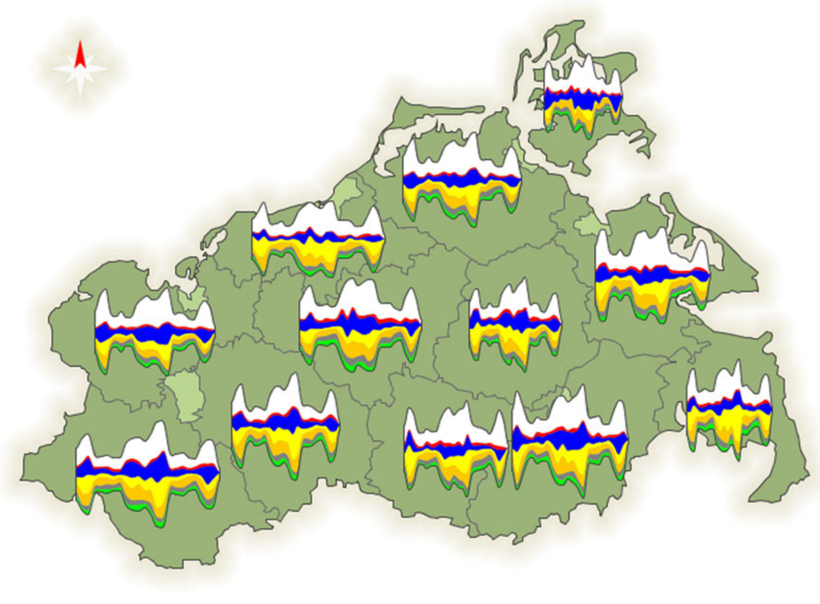


Fig. 4.10: Embedding ThemeRiver thumbnails on a map allows for communicating both temporal and spatial dependencies of spatial time-oriented data.

to the full-screen counterpart). This is the reason why labels or annotations are no longer visible constantly, but instead are displayed only on demand.

On top of the compromises made, all visualization approaches that embed (time-representing) thumbnails (or glyphs or icons) into a map share a common problem: finding a good layout. What makes a good layout is heavily application dependent, but there is consensus that having an overlap-free layout is generally a good starting point. However, finding a layout that minimizes occlusion among thumbnails and overlap of thumbnails with geographic features is a difficult problem. In fact, the problem is related to the general map labeling problem, which is NP-hard. Pursuing a globally optimized solution is computationally complex (see Petzold, 2003; Been et al., 2006), whereas locally optimizing approaches usually perform less expensive iterative adjustments that lead to suitable, but not necessarily optimal layouts (see Fuchs and Schumann, 2004; Luboschik et al., 2008). We will not go into any details of possible solutions, but instead refer the interested reader to the original publications.

The bottom line is that the characteristics of time and time-oriented data shape the design of visual representations to a great extent. As with the example of abstract vs. spatial data we see that the more aspects need to be communicated, the more intricate the visualization can become. One has to make acceptable trade-offs and may face NP-hard computational challenges. Moreover, the example of linear vs. cyclic time illustrates that the right visual mapping is essential for crystallizing answers and insight from visualizations of time-oriented data.

4.2.2 Task Level

We introduced the user task as a second important visualization aspect. Incorporating the users' tasks into the visualization design process on a general level is a challenging endeavor. Therefore, the illustrative example we present here is a pragmatic solution for the specific case of *color coding*. Earlier in this chapter we indicated that in addition to positional encoding of data values along a time axis, color coding plays an important role when visualizing multiple time-dependent data variables. The design of the color scale employed for the visual encoding substantially influences the overall expressiveness of the visual representation. To obtain expressive visual results, flexible color coding schemes are needed that can be adapted to the data as well as to the task at hand. In the following, we will explain how color scales can be generated in a task-dependent manner, and how they can be applied to visualize time-oriented data. But first let us briefly review general aspects of color coding.

Color coding

The general goal of color coding is to find an expressive mapping of data to color. This can be modeled as a color mapping function $f : D \rightarrow C$ that maps values of a dataset D to colors from a color scale C. A fundamental requirement for effective color coding is that the color mapping function f be injective, that is, every data value (or every well-defined group of data values) is associated with a unique color. This, in turn, allows users to mentally associate that unique color with a distinct data value (or group of values). On the one hand, mapping two quite different data values should result in two colors that are easy to discern visually. On the other hand, users spotting visually similar colors infer that these colors represent data values that are similar. Figure 4.11 demonstrates a basic mapping strategy.

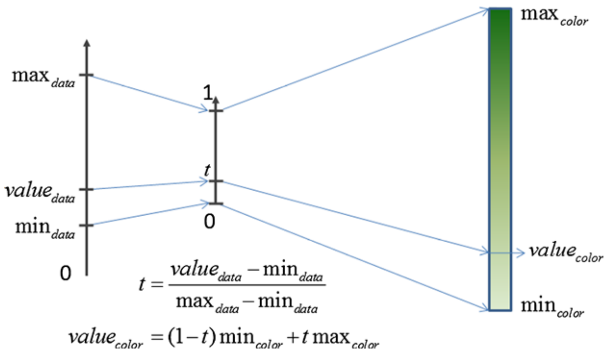


Fig. 4.11: Simple strategy for mapping data to color.

Telea (2007) describes these and further factors as relevant for color coding. We adapted his statements for the case of visualization of time-oriented data under consideration of the characteristics of time and data (as identified in Chapter 3):

- *Characteristics of the data*: First and foremost, the statistical features of the data and the time scale should be taken into account: extreme values, overall distribution of data values as well as data variation speeds and domain sampling frequencies. For example, using a linear color mapping function on a skewed dataset will result in the majority of data values being compressed to a narrow range of colors.
- *Characteristics of the tasks at hand*: Different tasks require different color coding schemes. A main distinction here is whether the task requires the comparison of exact quantitative values or the assessment of qualitative differences. Furthermore, certain goals may lead to specific regions of interest in the data domain. These regions should be accentuated, for instance by using bright, warm, and fully saturated colors.

- *Characteristics of the user*: The capabilities and the cultural as well as professional background of users have to be considered when designing appropriate color scales. Individual color perception has to be taken into account, and for users suffering from color vision defects, data values have to be mapped redundantly to additional visual attributes. The conventions of the application background need to be considered as well. Medical experts, for instance, are very much used to interpreting red-black-green color scales, despite the problems such colors may cause for people with color vision deficits.
- *Characteristics of the output device*: Different output devices use different systems to define and display colors. Thus, a color coding scheme which is appropriate for displaying data on a computer display might be inappropriate when showing the same data on other media. A common example is that colors that have been carefully tuned for the slides of a talk appear completely different when projected onto a wall.

The problem is that most of today's visualizations that use color do not consider these aspects to an adequate level. It is often the case that just basic color coding schemes are used, most prominently the classic rainbow color scale. However, this can lead to a loss of expressiveness of the generated images (see Borland and Taylor, 2007; Silva et al., 2007). In the following, we focus on the task aspect in more detail.

Task-dependent color coding

In order to define color scales in a task-specific manner, an adequate specification of tasks is required. For this purpose, one can use the task model of Andrienko and Andrienko (2006), which we described in Section 4.1.2. The model basically investigates tasks at three different levels. The first level draws a distinction between individual data values and sets of data values (elementary tasks vs. synoptic tasks). At the second level, the Andrienkos distinguish lookup, comparison, and relation seeking tasks. In a broader sense, relation seeking can be seen as a specific case of comparison[3]. This fact allows us to focus on the distinction between the two tasks: lookup and comparison. The third level addresses two types of tasks: identification and localization. Accomplishing identification tasks (direct lookup or direct comparison) requires recognizing data values and characteristic patterns as precisely as possible, whereas performing localization tasks (inverse lookup or inverse comparison) requires locating those references in time (and/or space) that exhibit certain characteristics of interest. In summary, task-dependent color scales can be generated based on the following distinctions:

- Individual values vs. sets of values,
- Identification vs. localization, and
- Lookup vs. comparison.

[3] For relation seeking the user has to compare different data items and/or time points to find interesting relations, where relations of interest are defined beforehand, which is not the case for plain comparison.

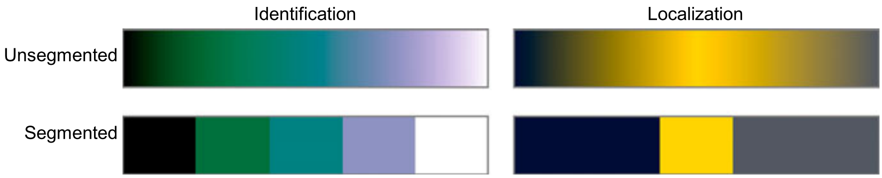


Fig. 4.12: Examples of unsegmented and segmented color scales for identification and localization of data values in a visual representation.

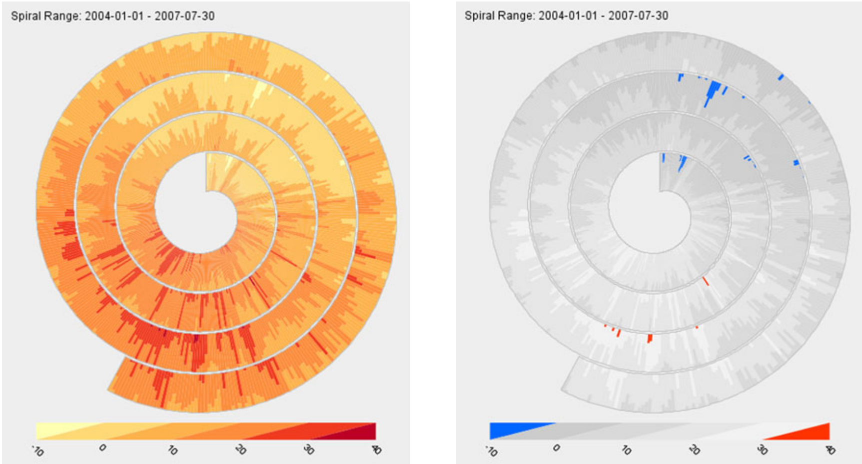
Color-coding individual data values requires unsegmented color scales. Unsegmented color scales associate unique colors with all individual data values, that is, every color of the color scale represents exactly one data value. In contrast to that, segmented color scales should be used to encode sets of data values. Each color of a segmented color scale stands for a set, usually a range of data values.

In order to facilitate identification tasks, it should be made easy for the user to mentally map the perceived color to a concrete data value (or set of values). Moreover, distances in the color scale should correspond to distances in the data. To support localization tasks, color scales should be designed in such a way that they exploit pre-attentive perception of temporal areas of interest, for instance by using accentuation and de-accentuation.

The specification of color scales for identification and localization of individual data values and sets of values is a well investigated problem (see Bergman et al., 1995; Harrower and Brewer, 2003; Silva et al., 2007). Figure 4.12 shows examples of such color scales. The segmented color scale for identification represents five sets of values, the unsegmented version can be used to identify individual values. The segmented color scale for localization supports users in making a binary decision: yellow encodes a match of some selection criteria, otherwise there is no match. The unsegmented color scale represents a smooth interpretation of the selection criteria.

Figure 4.13 illustrates the difference between color scales for identification and localization for the case of time-oriented data. The figure shows daily temperature values for about three and a half years mapped to a color-coded spiral display (↪ p. 184). While the color scale in Figure 4.13(a) supports identification, that is, one can easily associate a color with a particular range of values, the color scale in Figure 4.13(b) is most suited to locate specific data values in time. In our example, the highest and lowest values are accentuated using saturated red and blue, respectively. All other values are encoded to shades of gray, effectively attenuating these parts of the data. Thus it is easy to locate where in time high and low values occur.

The distinction between lookup and comparison tasks deserves a more detailed investigation. Supporting the lookup task basically requires color scales that allow for precise association of particular colors with concrete data values. In order to facilitate comparison tasks, all variables involved in the comparison must be represented by a common unified color scale, which can be problematic when variables exhibit quite different value ranges. The next paragraphs will provide more detail on how efficient color scales for lookup and comparison tasks can be designed.

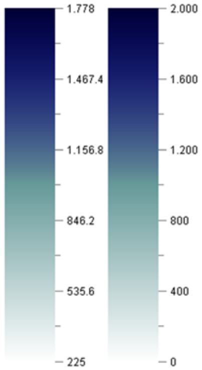


(a) Color scale for identification of values. (b) Color scale for localization of extrema.

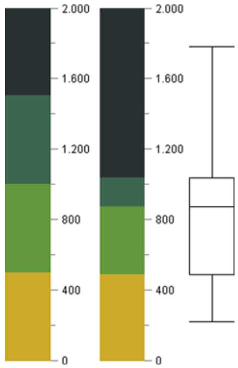
Fig. 4.13: Daily temperature values visualized along a spiral time axis using different color scales for different tasks.

Color-coding for the lookup task As mentioned earlier, there are two kinds of lookup tasks: inverse lookup and direct lookup. Inverse lookup tasks are basically a search for certain references in time that exhibit specific data characteristics (localization). For the inverse lookup task, relevant data values (or subsets) are known beforehand and hence can be easily accentuated using a highlighting color. On the other hand, the design of color scales for direct lookup (identification) is intricate because the whole range of data values is potentially relevant and must be easily identifiable. One way to facilitate lookup tasks is to extract statistical metadata from the underlying dataset and utilize them to adjust predefined color scales (see Schulze-Wollgast et al., 2005; Tominski et al., 2008). Let us take a look at three possible ways of adaptation.

Expansion of the value range The labels displayed in a color scale legend are the key to an easy and correct interpretation of a color-coded visualization. Commonly a legend shows labels at uniformly sampled points between the data's minimum and maximum. As the left color scale in Figure 4.14(a) illustrates, this usually results in odd and difficult-to-interpret labels. Even if the user has a clear picture of the color, it takes considerable effort to mentally compute the corresponding value, or even the range of plausible values. The trick of value range expansion is to extend the data range that is mapped to the color scale. This is done in such a way so as to arrive at a color mapping that is easier to interpret. The right color scale in Figure 4.14(a) demonstrates this positive effect.



(a) Value range expansion.



(b) Control point adjustment.

Fig. 4.14: Value range expansion and control point adjustment help to make color legends more readable and to better adapt the color coding to the underlying data distribution, which is depicted as a box-whisker plot.

Adjustment of control points A color map is defined by several control points, each of which is associated with a specific color. Appropriate interpolation schemes are used to derive intermediate colors in between two control points. The left color scale in Figure 4.14(b) shows an example where control points are uniformly distributed (interpolation is not applied for this segmented color scale). While this is generally a good starting point, more information can be communicated when using an adapted control point distribution. This is demonstrated in the right color scale of Figure 4.14(b), where control points have been shifted in accordance with the data distribution. The advantage is that users can easily associate colors with certain ranges of the data distribution[4].

Skewing of the color mapping function Uneven value distributions can be problematic because they lead to situations where the majority of data values is represented by only a narrow range of colors. This is unfavorable for lookup tasks. Logarithmic or exponential color mapping functions are useful when visualizing data with skewed value distributions. In cases where the underlying data distribution cannot be described by an analytical function, equalization can be applied to generate adapted color scales. The net effect of equalization is that the scale of colors is in accord with the data's value distribution. Histogram equalization and box-whisker equalization are examples of this kind of adaptation:

- Histogram equalization works as follows. First, one subdivides the value range into n uniform bins and counts the number of data values falling into the bins. Secondly, the color scale is sampled at $n+1$ points, where the points' locations

[4] The box-whisker plot or boxplot used in the figures depict minimum, 1st quartile, median, 3rd quartile, and maximum value (horizontal ticks from bottom to top).

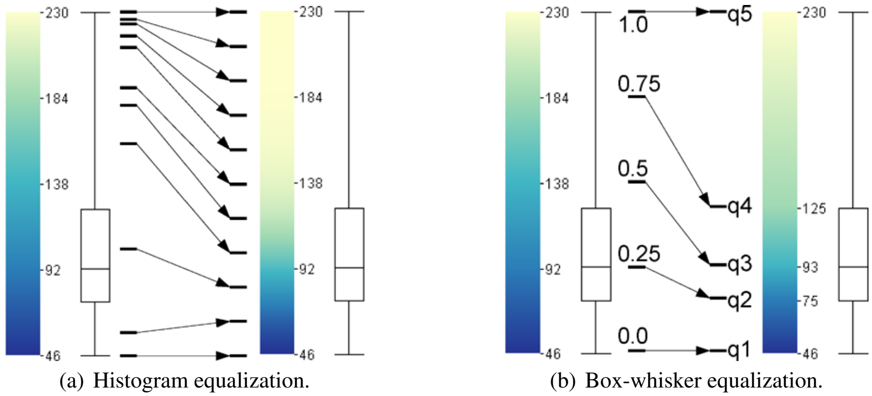


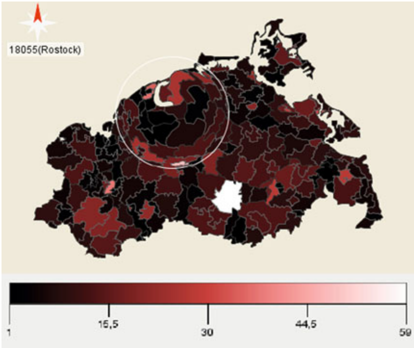
Fig. 4.15: Equalization schemas for adapting a color scale to the data distribution, which is depicted as box-whisker plots.

are determined by the cumulative frequencies of the bins. Finally, the colors at these sample points are used to construct an adapted color scale as illustrated in Figure 4.15(a). As a result, more colors are provided there where larger numbers of data values are located, making values in high density regions easier to distinguish.

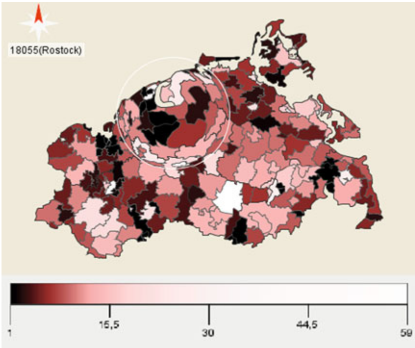
- Box-whisker equalization works similarly. Here, colors are sampled at points determined by quartiles. Quartiles partition the original data into four parts, each of which contains one-fourth of the data. The second quartile is defined as the median of the entire set of data (one half of the data lies below the second quartile, the other half lies above it). The first and the third quartile are the medians of the lower and upper half of the data, respectively. The adapted color scale is constructed from the colors sampled at the quartiles (see Figure 4.15(b)).

How equalization affects the visualization of spatio-temporal data compared to using unadapted color scales is shown in Figure 4.16. It can be seen that colors are hard to distinguish in dense parts of the data unless histogram or box-whisker equalization is applied, which improves discriminability.

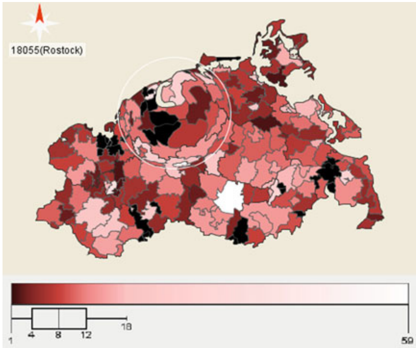
Color-coding for the comparison task The comparison of two or more time-dependent variables requires a global color scale that comprises the value ranges of all variables participating in the comparison. Particularly problematic are comparisons where the individual value ranges are quite different. For example, a variable with a small value range would be represented by only a small fraction of the global color scale, which makes it hard for viewers to differentiate colors in that range. An approach to alleviating this problem is to derive distinct intervals from the union of all value ranges and to create a separate encoding for each interval. To this end, a unique constant hue is assigned to each interval, while varying only brightness and saturation to encode data values. Finally, the separately specified color scales for the



(a) Color coding without equalization.



(b) Histogram equalization.

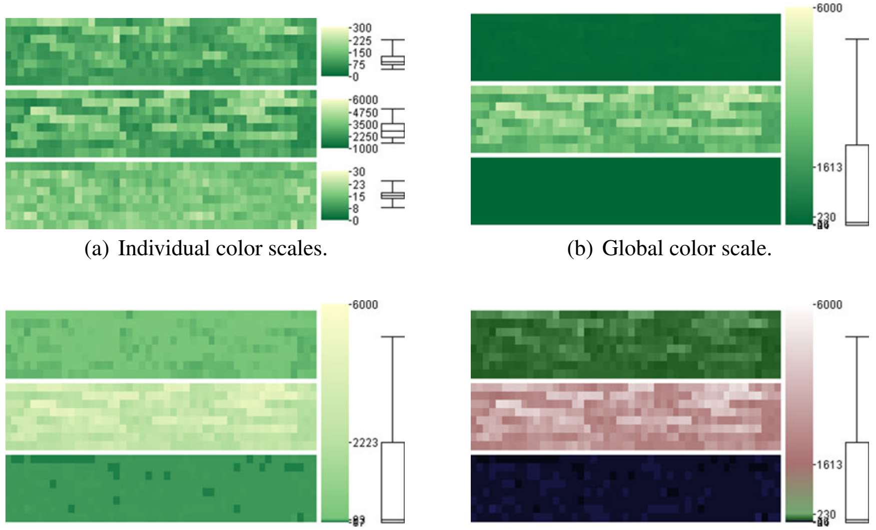


(c) Box-whisker equalization.

Fig. 4.16: Color scale equalization applied to the visualization of time-oriented health data on a map.

intervals are integrated into one global comparison color scale. To avoid discontinuities at the tieing points of two intervals, brightness and saturation values of one interval have to correspond with the respective values of the adjacent interval. In other words, within one interval the hue is constant while brightness and saturation vary, whereas at the boundary from one interval to the next the hue is modified while brightness and saturation are kept constant. This way, even small value ranges will be represented by their own brightness-varying subscale of the global color scale and the differentiation of data values is improved.

Figure 4.17 shows how different color coding schemes influence the task of comparing three time-dependent variables. Figure 4.17(a) uses individual color scales for each variable. Visual comparison is hardly possible because one and the same color stands for three different data values (one in each value range). A global color scale as shown in Figure 4.17(b) allows visual comparison, but data values of the first and third variable are no longer distinguishable because their value ranges



(a) Individual color scales.

(b) Global color scale.

(c) Box-whisker equalized color scale.

(d) Global comparison color scale.

Fig. 4.17: Different color scales for visual comparison of three time-dependent variables.

are rather small compared to the one of the second variable. Figure 4.17(c) illustrates that adapting the color scale to the global value distribution is beneficial. Figure 4.17(d) shows the visualization outcome when applying the color scale construction as described above: the recognition of values has been improved significantly. However, these results cannot be guaranteed for all cases, in particular, then when the merging process generates too many or too few distinct value ranges.

In the previous paragraphs we discussed the influence of the task at hand on the visualization of time-oriented data. The example of color-coding served to demonstrate how the task can be taken into account in the visualization process. As we have seen, visualization results can be improved when task-based concepts are applied. But still more research is required to investigate new methods of task-orientation, in particular in the light of collaborative visualization environments.

4.2.3 Presentation Level

Finally, there are design issues at the level of the visual representation. Communicating the time-dependence of data primarily requires a well-considered placement of the time axis. This will make it easier for users to associate data with a particular time, and vice versa. In Section 4.1.3, we have differentiated between 2D and 3D presentations of time-oriented data. Let us take up this distinction as an example of a design decision to be made at the level of the visual representation. Visualization approaches that use a 2D presentation space have to ensure that the time axis is emphasized, because time and data dimensions often have to share the two available display dimensions. In the case of 3D representations, a third display dimension is allocatable. In fact, many techniques utilize it as a dedicated dimension for the time axis, clearly separating time from other (data) dimensions. In the following, we will illustrate the 2D and the 3D approach with two examples.

2D presentation of time-oriented data

We discuss the presentation of time-oriented data in 2D by the example of axes-based visualizations. Axes-based visualization techniques are a widely used approach to represent multi-dimensional datasets in 2D. The basic idea is to construct a visual axis for each dimension of the n-dimensional data space, and to scale the axes with respect to the corresponding value range. In a second step, a suitable layout of the visual axes on the display has to be found. Finally, the data representation is realized by placing additional visual objects along the visual axes and in accord with the data. In this way, a lossless projection of the n-dimensional data space onto the 2-dimensional screen space can be accomplished. *Parallel coordinates* by Inselberg and Dimsdale (1990) are a well known example of this approach. Parallel coordinates use equidistant and parallel axes to represent multiple variables, and each data tuple is represented by a polygonal line linking the corresponding variable values. In the case of time-oriented data, however, this means that the axis encoding time is considered as one of many, not taking into account the outstanding importance of this axis (see Figure 4.18).

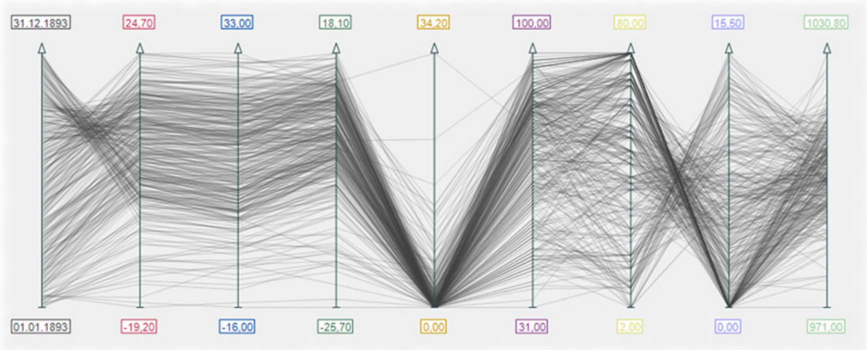


Fig. 4.18: In parallel coordinates, the time axis (here the leftmost axis) is just one of many axes and it is not treated in any particular way to emphasize the importance of time.

In contrast, Tominski et al. (2004) describe an axes-based visualization called *TimeWheel*, which focuses on one specific axis of interest, in our case the time axis (↪ p. 200). The basic idea of the TimeWheel technique is to distinguish between one independent variable, in our case time, and multiple dependent variables representing the time-oriented data. The dimension of time is presented by the reference time axis in the center of the display and time-dependent variables are shown as data axes that are circularly arranged around the time axis, where each dependent variable has a specific color hue associated with it. For each time value on the time axis, colored lines are drawn that connect the time value with the corresponding data value at each of the data axes, effectively establishing a visual link between time and multivariate data. By doing so, the time dependency of all variables can be visualized. Note that the interrelation of time values and data values of a variable can be explored most efficiently when a data axis is parallel to the time axis. Interactive rotation of the TimeWheel can be used to move data axes of interest into such a parallel position.

Two additional visual cues support data interpretation and guide the viewer's attention: color fading and length adjustment. Color fading is applied to attenuate lines drawn from the time axis to axes that are almost perpendicular to the time axis. During rotation, lines gradually fade out and eventually become invisible when the associated data axis approaches an upright orientation. To provide more display space for the data variables of interest, the length of data axes is adjusted according

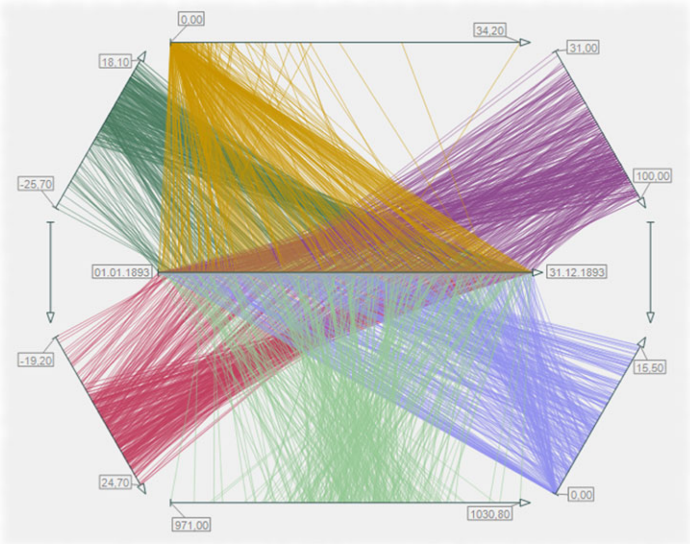


Fig. 4.19: The TimeWheel shows the reference time axis in a prominent central position and arranges data axes representing time-dependent variables around the time axis. Data are visualized by drawing lines between points at the time axis and values at data axes.

to their angle to the time axis. When the TimeWheel is rotated, data axes that are going to become parallel to the time axis are stretched to make them longer and data axes that head for an upright orientation are shrunk to make them shorter. Figure 4.19 shows a TimeWheel that visualizes eight time-dependent variables, where color fading and length adjustment have been applied to focus on the orange and the light green data axes.

The TimeWheel is an example of a 2D visualization technique that acknowledges the important role of the time axis. The time axis' central position emphasizes the temporal character of the data and additional visual cues support interactive analysis and exploration of multiple time-dependent data variables.

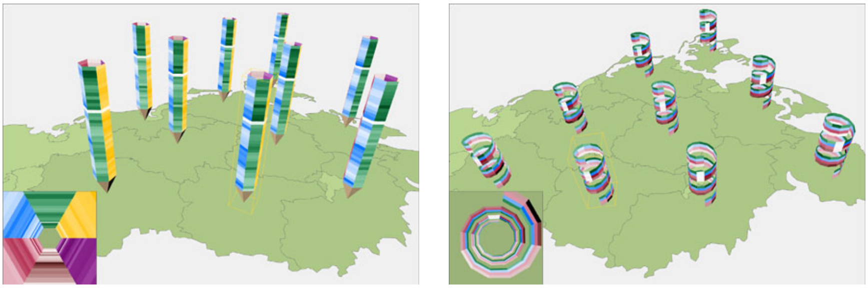
3D presentation of time-oriented data

3D presentation spaces provide a third display dimension. This opens the door to additional possibilities of encoding time and time-oriented data. Particularly, the visualization of data that have further independent variables in addition to the dimension of time can benefit from the additional dimension of the display space.

Spatio-temporal data are an example where data variables do not only depend on time, but also on space (e.g., on points given by longitude and latitude or on geographic regions). When visualizing such data, the temporal frame of reference as well as the spatial frame of reference have to be represented. We already mentioned that applying the *space-time cube* design (see Kraak, 2003 and ↪ p. 245) is common practice: the z-axis of the display space exclusively encodes time, while the x- and y-axes represent spatial dimensions. Spatio-temporal data are then encoded by embedding visual objects into the space-time cube (e.g., visual markers or icons) and by mapping data to visual attributes (e.g., color or texture). Kristensson et al. (2009) provide evidence that space-time cube representations can facilitate intuitive recognition and interpretation of data in their spatio-temporal context.

Figure 4.20 shows two examples of this approach as described by Tominski et al. (2005). Figure 4.20(a) represents multiple time-dependent variables by so-called pencil icons (↪ p. 249). The linear time axis is encoded along the pencil's faces starting from the tip. Each face of the pencil is associated with a specific data variable and a specific color hue, and represents the corresponding data values by varying color saturation. Figure 4.20(b) uses so-called helix icons (↪ p. 252). Here, we assume a cyclic character of time and thus, a ribbon is constructed along a spiral helix. For each time step the ribbon extends in angle and height, depending on the number of time elements per helix cycle and the number of cyclic passes. Again color coding is used to encode the data values. To represent more than one data variable, the ribbon can be subdivided into narrower sub-ribbons.

The 3D display space used in the previous examples is advantageous in terms of the prominent encoding of time, but it also exhibits two problems that one needs to address: perspective distortions and occlusion (see Section 4.1.3). Perspective distortions are problematic because they could impair the interpretation of the visualized data. Therefore, the visual mapping should avoid or reduce the use of geo-



(a) Pencil icons for linear time. (b) Helix icons for cyclic time.

Fig. 4.20: 3D visualization of spatio-temporal data using color-coded icons embedded into a map display.

metric visual attributes that are subject to perspective projections (e.g., shape, size, or orientation). This is the reason why the pencil icons and helix icons apply color coding instead of geometric encoding. The occlusion aspect has to be addressed by additional mechanisms. For example, users should be allowed to rotate the icons or the whole map in order to make back faces visible. Another option is to incorporate additional 2D views that do not suffer from occlusion. Such views are shown for a user-selected region of interest in the bottom-left corner of Figures 4.20(a) and 4.20(b). Again this approach is a compromise. On the one hand, the 2D view is occlusion-free, but on the other hand, one can show only a limited number of additional views, and moreover, one unlinks the data from their spatial reference.

Irrespective of whether one uses a 2D or 3D representation, the visualization design for time-oriented data requires a special handling of the time axis to effectively communicate the time-dependence of the data. Both approaches have to take care to emphasize the dimension of time among other data dimensions.

4.3 Summary

Solving the visualization problem primarily requires answering the three questions: (1) What is visualized? (2) Why is it visualized? (3) How is it visualized? The answers to the first two questions determine the answer to the third question.

In the case of visualizing time-oriented data, answering the what-question requires both specifying the characteristics of the time domain as well as specifying the characteristics of the data associated with time. In Chapter 3, we have shown that many different aspects characterize time and time-oriented data. It is virtually impossible to simultaneously cover all of them within a single visualization process. On top of this, there exists no visualization technique that is capable of handling all of the different aspects simultaneously and presenting all of them in an appropriate way. Here, the answer to "why are we visualizing the data" comes into play. Those aspects of the data that are of specific interest with regard to the tasks at hand have to be communicated by the visual representation, while others can be diminished or even omitted. However, this is an intricate problem, and most of today's visualization systems do not support the process of generating suitable task-specific visual representations. Thus, our primary aim can only be to communicate the problem, and also to demonstrate the necessity and potential of considering the interrelation between the what, why, and how aspects by example, as we have done in Section 4.2.

Fig. 4.21: Three key questions of the visualization problem.

Figure 4.21 again summarizes the key characteristics of the three aspects. The what-aspect addresses characteristics of time and data as detailed in Chapter 3. For describing the why-aspect, we rely on an abstracted view of the tasks by Andrienko and Andrienko (2006) (see Section 4.1.2). The how-aspect is mainly categorized by the differentiation of static and dynamic as well as 2D and 3D representations (see Section 4.1.3).

We will see that there are a variety of techniques for handling and accounting for these key characteristics. Accordingly, many different visual representations of time-oriented data can be generated. Chapter 7 will attest to this statement.

References

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.

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.

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.

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

Been, K., Daiches, E., and Yap, C.-K. (2006). Dynamic Map Labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780.

Bergman, L., Rogowitz, B. E., and Treinish, L. A. (1995). A Rule-based Tool for Assisting Colormap Selection. In *Proceedings of IEEE Visualization (Vis)*, pages 118–125, Washington, DC, USA. IEEE Computer Society.

Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, WI, USA. translated by William J. Berg.

Borland, D. and Taylor, R. (2007). Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17.

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

Constantine, L. L. (2003). Canonical Abstract Prototypes for Abstract Visual and Interaction Design. In Jorge, J., Nunes, N. J., and e Cunha, J. F., editors, *Interactive Systems: Design, Specification, and Verification*, volume 2844 of *Lecture Notes in Computer Science*. Springer, Berlin, Germany.

Courage, C. and Baxter, K. (2005). *Understanding Your Users*. Morgan Kaufmann, San Francisco, CA, USA.

Elmqvist, N. and Tsigas, P. (2007). A Taxonomy of 3D Occlusion Management Techniques. In *Proceedings of the IEEE Conference on Virtual Reality (VR)*, pages 51–58, Los Alamitos, CA, USA. IEEE Computer Society.

Farquhar, A. B. and Farquhar, H. (1891). *Economic and Industrial Solutions*. G. B. Putnam's Sons, New York, NY.

Fuchs, G. and Schumann, H. (2004). Intelligent Icon Positioning for Interactive Map-Based Information Systems. In *Proceedings of the International Conference of the Information Resources Management Association (IRMA)*, pages 261–264, Hershey, PA, USA. Idea Group Inc..

Gapminder Foundation (2010). Gapminder Trendalyzer. URL, http://www.gapminder.org/world/. Retrieved Feb., 2011.

Hackos, J. T. and Redish, J. C. (1998). *User and Task Analysis for Interface Design*. John Wiley & Sons, Inc., New York, NY, USA.

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

Harrower, M. A. and Brewer, C. A. (2003). ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps. *The Cartographic Journal*, 40(1):27–37.

Havre, S., Hetzler, E., and Nowell, L. (2000). ThemeRiver: Visualizing Theme Changes Over Time. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 115–124, Los Alamitos, CA, USA. IEEE Computer Society.

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.

Inselberg, A. and Dimsdale, B. (1990). Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. In *Proceedings of IEEE Visualization (Vis)*, pages 361–378, Los Alamitos, CA, USA. IEEE Computer Society.

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

Kraak, M.-J. (2003). The Space-Time Cube Revisited from a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages 1988–1995, Newcastle, UK. The International Cartographic Association (ICA).

Kristensson, P., Dahlback, N., Anundi, D., Bjornstad, M., Gillberg, H., Haraldsson, J., Martensson, I., Nordvall, M., and Stahl, J. (2009). An Evaluation of Space Time Cube Representation of Spatiotemporal Patterns. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):696–702.

Luboschik, M., Schumann, H., and Cords, H. (2008). Particle-Based Labeling: Fast Point-feature Labeling Without Obscuring Other Visual Features. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1237–1244.

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

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

Müller, W. and Schumann, H. (2003). Visualization Methods for Time-Dependent Data - An Overview. In *Proceedings of Winter Simulation Conference (WSC)*, pages 737–745, Los Alamitos, CA, USA. IEEE Computer Society.

Nelson, A. (2008). *Travel Time to Major Cities: A Global Map of Accessibility*. Office for Official Publications of the European Communities, Luxembourg.

Paternò, F., Mancini, C., and Meniconi, S. (1997). ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In *Proceedings of IFIP TC13 International Conference on Human-Computer Interaction (INTERACT)*, pages 362–369, Boston, MA, USA. Kluwer Academic Publishers.

Paternò, F. and Santoro, C. (2002). One Model, Many Interfaces. In *Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces (CADUI)*, pages 143–154, Boston, MA, USA. Kluwer Academic Publishers.

Petzold, I. (2003). *Beschriftung von Bildschirmkarten in Echtzeit*, PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn.

Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J. (2008). Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14:1325–1332.

Schulze-Wollgast, P., Tominski, C., and Schumann, H. (2005). Enhancing Visual Exploration by Appropriate Color Coding. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 203–210, Plzen, Czech Republic. University of West Bohemia.

Silva, S., Madeira, J., and Santos, B. S. (2007). There is More to Color Scales than Meets the Eye: A Review on the Use of Color in Visualization. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 943–950, Los Alamitos, CA, USA. IEEE Computer Society.

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.

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

Stolte, C., Tang, D., and Hanrahan, P. (2002). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65.

Telea, A. C. (2007). *Data Visualization: Principles and Practice*. A K Peters, Ltd., Natick, MA, USA.

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247, New York, NY, USA. ACM Press.

Tominski, C., Fuchs, G., and Schumann, H. (2008). Task-Driven Color Coding. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 373–380, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

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

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

Unger, A. and Schumann, H. (2009). Visual Support for the Understanding of Simulation Processes. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64, Los Alamitos, CA, USA. IEEE Computer Society.

Vande Moere, A. (2004). Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 97–104, Los Alamitos, CA, USA. IEEE Computer Society.

Ward, M. O. (1994). XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of IEEE Visualization (Vis)*, pages 326–333, Los Alamitos, CA, USA. IEEE Computer Society.

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing Time-Series on Spirals. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 7–14, Los Alamitos, CA, USA. IEEE Computer Society.

Winckler, M. A., Palanque, P., and Freitas, C. M. D. S. (2004). Tasks and Scenario-Based Evaluation of Information Visualization Techniques. In *Proceedings of the Annual Conference on Task Models and Diagrams (TAMODIA)*, pages 165–172, New York, NY, USA. ACM Press.

Wolter, M., Assenmacher, I., Hentschel, B., Schirski, M., and Kuhlen, T. (2009). A Time Model for Time-Varying Visualization. *Computer Graphics Forum*, 28(6):1561–1571.

Yang, J., Wang, W., and Yu, P. S. (2000). Mining Asynchronous Periodic Patterns in Time Series Data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 275–279, New York, NY, USA. ACM Press.

Chapter 5
Interaction Support

> 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. The best graphic operations are those carried out by the decision-maker himself.
>
> Bertin (1981, p. 16)

The previous chapter discussed various options for designing visual representations that help people in understanding time and time-oriented data. 'Seeing' trends, correlations, and patterns in a visual representation is indeed a powerful way for people to gain knowledge from data. However, we have also seen earlier in this book that manifold aspects are involved in the generation of a visual representation: characteristics of time and data, user tasks, choice and parametrization of visualization techniques, and so forth. Particularly when feeding unknown data into a visualization method, the visual outcome might not turn out as expected. But we usually do not know exactly what it is that is expected or whether the visual representation is effective with regard to the task that the user is trying to accomplish. And because there are things that we do not know, we have to seek assistance from the user. So, visual exploration and analysis is not a one-way street where data are transformed into images, but in fact is a human-in-the-loop process controlled and manipulated by one or more users.

Having said that, it becomes clear that in addition to visual methods, a high degree of interactivity and advanced interaction techniques for working with time-oriented data are important. Interaction helps users in understanding the visual mapping, in realizing the effect of visualization parameters, in carving out hidden patterns, and in becoming confident about the data. But interaction also provokes curiosity – users want to get their hands on their data – which is particularly useful when exploring unknown data. The importance of interaction is nicely reflected in the following statement:

> Visual representations alone cannot satisfy analytical needs. Interaction techniques are required to support the dialogue between the analyst and the data.
>
> Thomas and Cook (2005, p. 30)

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_5,

5.1 Motivation & User Intents

The constantly increasing size and complexity of today's datasets are major challenges for interactive visualization. Large datasets cannot simply be loaded to limited computer memory and then be mapped to an even smaller display. Users are only able to digest a fraction of the available information at a time. Complex data contain many different aspects and may stem from heterogeneous sources. As complexity increases, so does the number of questions that one might ask about the data and that should be answered with the help of visual representations.

In our particular case, we need to account for the specific aspects of time and time-oriented data in the context of what, why, and how they are visualized (see Chapters 3 and 4). Any attempt to indistinctively encode all facets of a complex time-oriented dataset into a single visual representation is condemned to fail, because it would lead to a confusing and overloaded display that users could hardly interpret.

Instead, the big problem has to be split into smaller pieces by focusing on relevant data aspects and particular tasks per visual representation. Several benefits can be gained: computational costs are reduced in a kind of divide-and-conquer way, the visual representations become more effective because they are tailored to emphasize a particular point, and users find it easier to explore or analyze the data since they can concentrate on important and task-relevant questions.

Dividing the visualization problem and separating different aspects into individual views raises the question of how users can visually access and mentally combine these. The answer is *interaction*. In an iterative process, the user will focus on different parts of the data, look at them from alternative perspectives, and will seek answers to diverse questions. Starting with an overview of the data, which requires appropriate data abstraction and dedicated visual representations, the user will most certainly identify parts to focus on for more detailed examination (see Shneiderman, 1996). From there, it might make sense to move on to data that are related or similar, or it might be better to return to the overview and investigate the data from a different point of view, or with regard to a different question. In other words, the user forms a mental model of the data by interactively navigating from one focus to the next, where the term focus includes data subsets, data aspects, analysis tasks, and so forth. While exploring data this way, users develop understanding and insight.

The general motivation for interaction is clear now, but what specifically motivates the user to interact? Yi et al.'s (2007) study on a deeper understanding of interaction gives an answer to this question. They identified several user intents for interaction and introduced a list of categories that describe on a high level why users would like to interact. In the following, we make use of the categories from Yi et al. (2007) and adapt them to the case of interacting with time and time-oriented data:

Select – Mark something as interesting When users spot something interesting in the visual representation, they want to mark and visually highlight it as such, be it to temporarily tag an intriguing finding or to permanently memorize important analysis results. The subjects to be marked can be manifold: interesting points in time, an entire time-dependent variable, a particular temporal pattern, or certain identified events.

Explore – Show me something else In order for the visualization of larger or more complex multivariate time-oriented data to be practically usable, it has to focus on only a subrange of time and on only a subset of the data variables. As a consequence, users have to interactively visit different parts of the time domain and consider alternative variables for the inclusion in the visual encoding to arrive at an overall view of the data.

Reconfigure – Show me a different arrangement Different arrangements of time and associated data can communicate completely different aspects, a fact which becomes obvious when recalling the distinction between linear and cyclic representations of time. As users want to look at time from different angles, they need to be provided with facilities that allow them to interactively generate different spatial arrangements of time-oriented data.

Encode – Show me a different representation Similarly to what was said about the spatial arrangement, the visual encoding of data values has a major impact on what can be derived from a visual representation. Because data and tasks are versatile, users need to be able to adapt the visual encoding to suit their needs, be it to carry out localization or comparison tasks, or to confirm a hypothesis generated from one visual encoding by checking it against an alternative one.

Abstract/Elaborate – Show me more or less detail During visual analysis, users need to look at certain things in detail, while for other things schematic representations are sufficient. The hierarchically structured levels of granularity of time, where higher-level abstractions provide aggregated overviews, and lower levels hold the corresponding details, are a natural match to drive such an interactive information drill-down into time-oriented data.

Filter – Show me something conditionally When users search for particular information in the data or evaluate a certain hypothesis about the data, it makes sense to restrict the visualization to show only those data items that adhere to the conditions imposed by the search criteria or the hypothesis' constraints. Interactively filtering out or attenuating irrelevant data items clears the view for users to focus on their current task.

Connect – Show me related items When users make a potentially interesting finding in the data, they usually ask themselves whether similar or related discoveries can be made in other parts of the data. So users intend to interactively find, compare, and evaluate such similarities or relations, for example, to see whether a trend they discovered in one season of a year is present for other data variables or is repeated at the same time in subsequent years.

Undo/Redo – Let me go to where I have already been Users have to navigate in time and look at it at different levels of granularity, they have to try different arrangements and visual encodings, and they have to experiment with filtering conditions and similarity thresholds. To account for the explorative nature of interactive analytic reasoning, a history mechanism with undo and redo operations is needed. Undo/redo enables users to try out new views on the data and to return effortlessly to the previous visual representation if the new one did not work out as expected.

Change configuration – Let me adjust the interface In addition to adapting the visual representation to the data and tasks at hand, users also want to adapt the overall system that provides the visualization. This includes adapting the user interface (e.g., the arrangement of windows or the items in toolbars), but also the general management of system resources (e.g., the amount of memory to be used).

Taken together, these intents represent what a visualization system for time-oriented data should support in terms of interaction in order to take full advantage of the synergy of the human's and the machine's capabilities. Many of the approaches we describe in the survey in Chapter 7 offer support for one or the other user intent. While marking (or selecting) interesting data items and navigation in time are quasi-mandatory, facilities for other intents are often rudimentary or not considered at all. This is most likely due to the extra effort one has to expend for implementing effective interaction methods. But in fact, all of these user intents are equally important and corresponding techniques should be provided.

5.2 Fundamental Principles

Now that we know about the general motivation and more specific user intents behind interaction, we can move on and take a look at how interaction is actually performed. To this end, we will describe fundamental interaction concepts next.

Conceptual considerations

When users interact they express their intent to change their point of view and they expect that the visual representation reflects the intended change. Norman (2002) models interaction as a loop of two gulfs: the gulf of execution and the gulf of evaluation. The first part subsumes steps that are related to the execution of interaction, including the intention, the mental construction of an interaction plan, and the actual physical interaction (e.g., pressing a button). The second part is related to the feedback, which in our case is of a visual nature. It involves perceiving and understanding the feedback as well as evaluating the success of the interaction. Figure 5.1 illustrates Norman's conceptual model.

Different modes of interaction can be identified depending on how the interaction loop is performed. Spence (2007) distinguishes two modes of active user interaction:

- stepped interaction and
- continuous interaction.

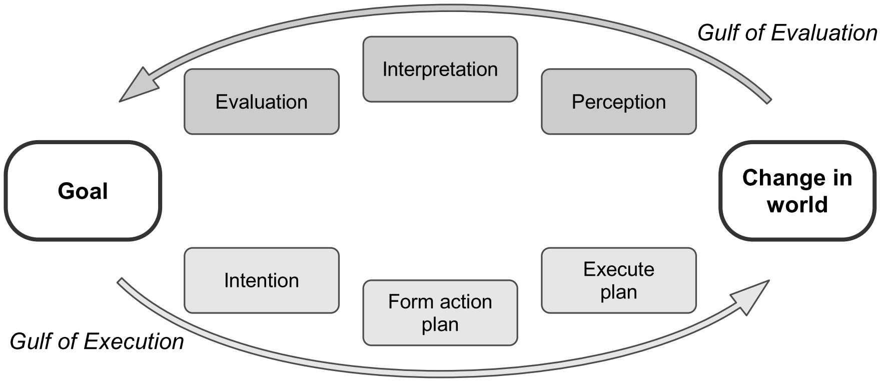


Fig. 5.1: Norman's model of interaction (adapted from Norman, 2002).

For *stepped interaction* the interaction loop is executed in a discrete fashion. That is, the user performs one interaction step and evaluates the visual feedback. Much later the user might perform another step of interaction. As an example one can imagine a user looking at a visualization of the data at the granularity of years. If more details are required, the user might take an interaction step to switch to a finer granularity of months.

The term *continuous interaction* is used to describe interaction for which the loop is iterated at high frequency. The user continuously performs interaction and evaluates feedback for a significant period of time. This is particularly useful, because examining multiple 'what if' scenarios is a key aspect of exploratory analysis of time-oriented data.

An example could refer to setting the cycle length for a spiral visualization in order to find cyclic patterns. For stepped interaction, the user has to explicitly specify the cycle length of interest in a successive manner. The stepped approach requires much time to explore parameter ranges and one does not see patterns emerge as a suitable cycle length is approached. Continuous interaction (e.g., by dragging a slider) allows the user to explore any range at any speed and reduces the risk of losing interesting patterns.

Technical considerations

Technically, Jankun-Kelly et al. (2007) model the loop of user interaction as adjustments of visualization parameters, where concrete parameters can be manifold, e.g., the rotation angle of a 3D helix glyph, the focus point of a fisheye-transformed time axis, thresholds of a filter operation, or parameters that control a clustering algorithm.

For smooth and efficient interaction, the ensemble of visual and interaction methods has to generate feedback in a timely manner (within 50 - 100 ms according to

Shneiderman (1994) and Spence (2007)). However, even data of moderate size can pose computational challenges. On the one hand, mapping and rendering the visual representation might take some time, particularly if complex visual abstractions have to be displayed. On the other hand, analytical methods (see Chapter 6) involved in the visualization process consume processing time before generating results. The adverse implication for interaction is that visual feedback might lag, disrupting the interaction loop.

Another aspect adds to the time costs for presenting visual feedback. As interaction involves change, we want users to understand what is happening. However, abrupt changes in the visual display will hurt the mental model that users are developing while exploring unknown data. Pulo (2007) and Heer and Robertson (2007) provide evidence that smoothly interpolating the parameter change and applying animation to present the visual feedback is a better solution. However, animation consumes time as well, not to mention the possibly costly calculations for interpolating parameter changes.

Thus there are two conflicting requirements. On the one hand, interaction needs synchrony. An interactive system has to be responsive at all times and should provide visual feedback immediately. From the interaction perspective, a system that is blocked and unresponsive while computing is the worst scenario. On the other hand, interaction needs asynchrony – for both generating the feedback (i.e., computation) and presenting the feedback (i.e., animation). The difficulty is to integrate synchrony and asynchrony.

In order to utilize the power of the human-in-the-loop, Norman's model must be outfitted with intuitive interaction methods to allow users to change the visualization, and with intuitive ways of presenting the visual feedback to reflect the change. From a technical perspective, both the gulf of execution and the gulf of evaluation must work together smoothly and seamlessly under the umbrella of an effective user interface.

User interface

The user interface is the channel through which a human and a machine exchange information (i.e., interaction input and visual feedback). This interface is the linchpin of interactive visual exploration and analysis of time-oriented data. Any visual representation is useless if the user interface fails to present it to the user in an appropriate way, and the diversity of available visualization techniques lies idle if the user interface fails to provide interactive access to them. In order to succeed, the user interface has to bridge the gap between the technical aspects of a visualization approach and the users' mental models of the problems to be solved. In this regard, Cooper et al. state:

> [...] user interfaces that are consistent with users' mental models are vastly superior to those that are merely reflections of the implementation model.
>
> Cooper et al. (2007, p. 30)

The user interface is responsible for numerous tasks. It has to provide visual access to time-oriented data and to information about the visualization process itself at different levels of graphical and semantic detail. Appropriate controls need to be integrated to allow users to steer exploration and analysis with regard to the interaction intents mentioned before, including marking interesting points in time, navigating in time at different levels of granularity, rearranging data items and elements of the visual representation, or filtering for relevant conditions. Moreover, the user interface has to support bookkeeping in terms of the annotation of findings, storage of results, and management of the working history (undo/redo).

In general, the user interface has to offer facilities to present information to the user and to accept interaction input from the user. This separation is reflected in the *model-view-controller* (MVC) architecture by Krasner and Pope (1988), where views provide visual representations of some model (in our case time, time-oriented data, and visualization parameters) and controllers serve for interactive (or automatic) manipulation of the model.

Visualization views Especially the different temporal granularities make it necessary to present the data at different levels of graphical and semantic detail. *Overview+detail* as well as *focus+context* are the key strategies to address this demand. Overview+detail methods present overview and detail separately. The separation can be either spatial or temporal. Spatial separation means that separate views show detail and overview. For example, on the bottom of Figure 5.2, an overview shows the entire time domain at a high level of abstraction. On top of the overview there is a separate detail view, which shows the data in full detail (i.e., detailed planning information), but only for a narrow time interval. Temporal separation means a view is capable of showing any level of detail, but only one at a time. This is

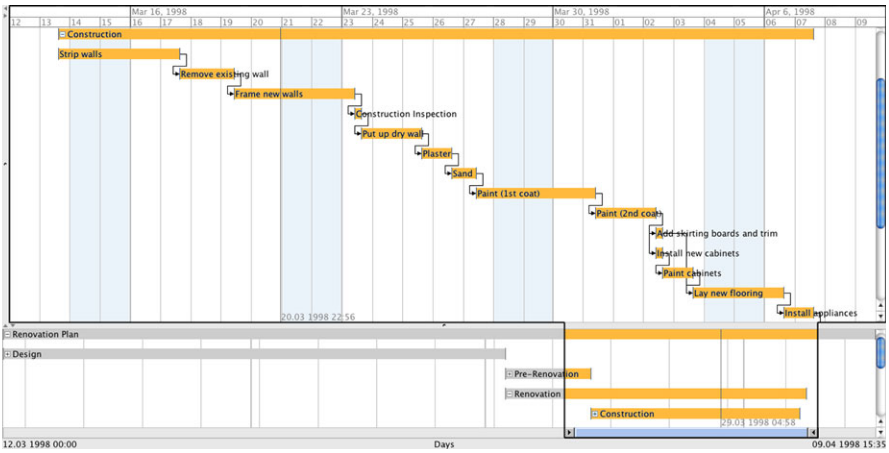


Fig. 5.2: Overview+detail. The detail view at the top shows individual steps of the construction phase of a renovation plan. In the overview at the bottom, the entire project is shown, including the design, pre-renovation, renovation, and construction phases.

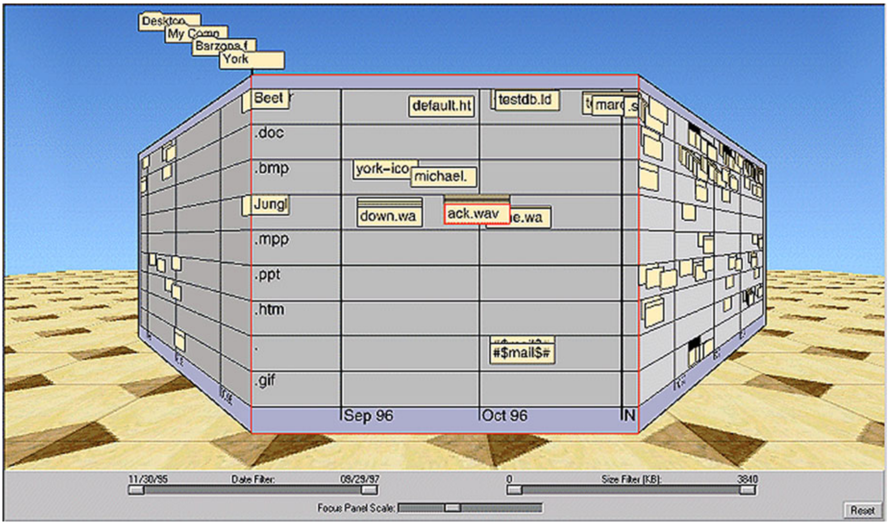


Fig. 5.3: Focus+context. The center of the perspective wall shows the focus in full detail. The focus is flanked on both sides by context regions. Due to perspective distortion, the context regions intentionally decrease in size and show less detail.
Source: © Inxight Federal Systems.

usually referred to as *zooming*, where the user can interactively zoom into details or return to an overview. *Semantic zooming* denotes zooming that is performed in the data space, whereas *graphical zooming* operates in the presentation space. Contrary to overview+detail, focus+context methods smoothly integrate detail and overview. For the user-chosen focus, full detail is presented, and the focus is embedded into a less-detailed display of the context. Figure 5.3 shows the perspective wall technique (↪ p. 168) as a prominent example of the focus+context approach. Cockburn et al. (2009) provide a comprehensive survey of overview+detail, zooming, and focus+context and discuss the advantages and disadvantages of these concepts.

When visualizing time-oriented data, a sensible approach is to provide *multiple coordinated views*[1], each of which is dedicated to particular aspects of time, certain data subsets, or specific visualization tasks. Views are coordinated to help develop and maintain a consistent overall image of the visualized data. This means that an interaction which is initiated in one view is automatically propagated to all coordinated views, which in turn update themselves to reflect the change visually. A practical example is browsing in time. When the user navigates to a particular range of the time axis in one view, all other views (that are coordinated) follow the navigation automatically, which otherwise would be a cumbersome task to be manually accomplished by the user on a per-view basis. Figure 5.4 shows an example where multiple coordinated views are applied to visualize spatio-temporal data in the VIS-STAMP system (↪ p. 244).

[1] Baldonado et al. (2000) provide general guidelines for when to use multiple coordinated views.

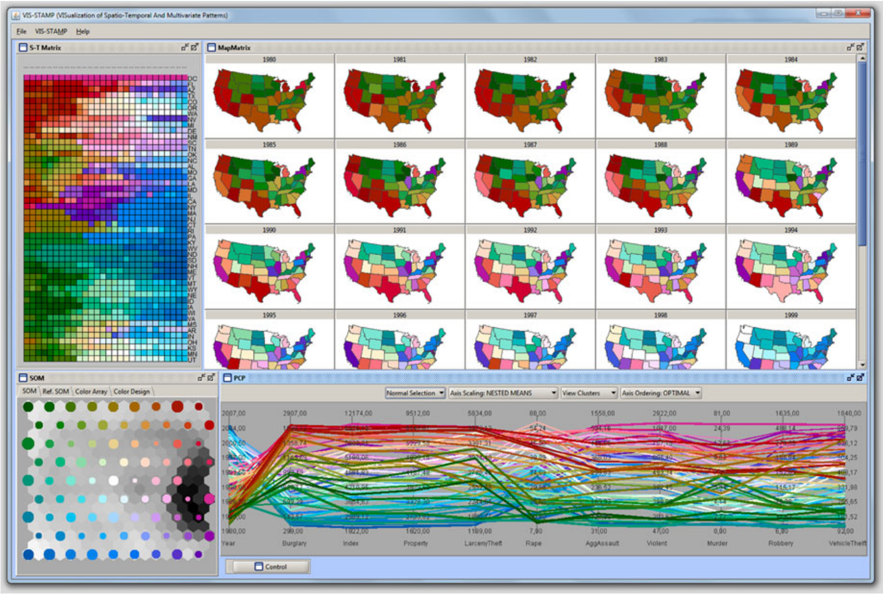


Fig. 5.4: Multiple coordinated views. Analysts can look at the data from different perspectives. The views are coordinated, which means selecting objects in one view will automatically highlight them in all other views as well.
Source: Generated with the VIS-STAMP system.

In addition to coordination, the arrangement of multiple views on the display plays a significant role. The two extremal positions one could take are to use a fixed arrangement that has been designed by an expert and has proved to be efficient, or to provide users with the full flexibility of windowing systems, allowing them to move and resize views arbitrarily. Both extremes have their advantages and disadvantages and both are actually applied. An interesting alternative is to make use of view docking. The main reason for applying docking is that while it maintains flexibility, it also imposes a certain order in terms of what arrangements are possible. For instance, it is preferable that views do not overlap partially; a view should either be visible or not. To this end, docking works with the available screen space being partitioned into regions, each of which contains one or more views. The regions can be resized and moved with the constraint that the arrangement remains a partition, that is, remains overlap-free. A region that contains more than one view provides an interface to switch between them (usually tab-based).

Interaction controls The user interface also consists of various interaction controls to enable users to tune the visualization process to the data and task at hand. Figure 5.5 shows a simple example with only a single spiral view (see Tominski and Schumann, 2008 and ↪ p. 184), which however already depends on a number of parameters, which in turn demands a corresponding number of controls in the control panel. The figure shows sliders for continuous adjustments of parameters such

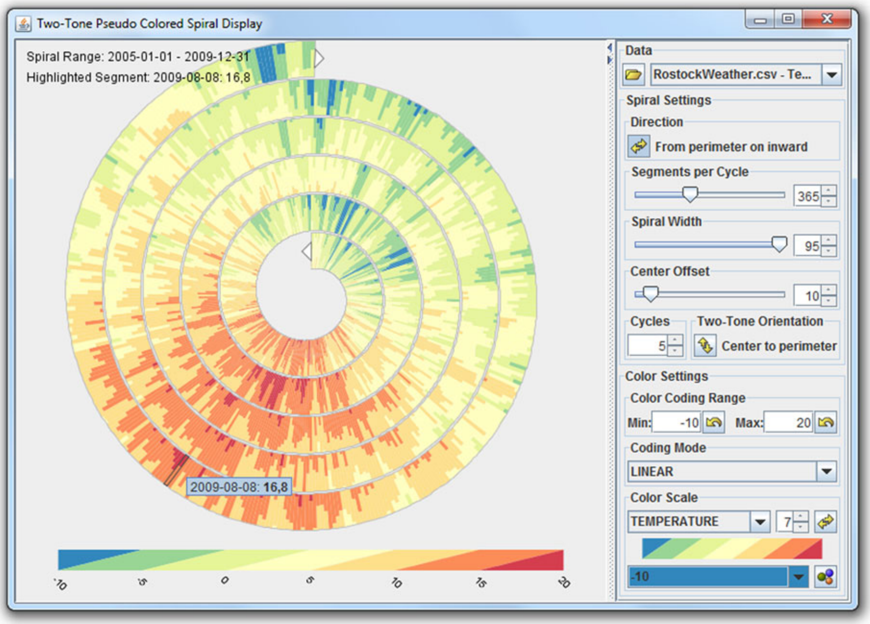


Fig. 5.5: User interface for a spiral visualization. The interface consists of one spiral view and one control panel, which in turn consists of various controls to adjust visualization parameters.

as *segments per cycle*, *spiral width*, and *center offset*. Buttons, drop boxes, and custom controls are provided for selecting different modes of encoding (e.g., adjusting individual colors or choosing different color scales).

In this example, user input (e.g., pressing a button or dragging a slider) is immediately committed to the system in order to realize continuous interaction. However, this puts high demands on the system in terms of maintaining responsiveness and generating visual feedback at interactive rates (see Piringer et al., 2009). Therefore, a commonly applied alternative is to allow users to perform a number of adjustments and to commit the adjustments as a single transaction in a stepped manner only when the user presses the “Apply” button.

Certainly, there are visualization parameters that are adjusted more often than others during interactive visual exploration. Resources should preferably be spent on facilitating continuous interaction for important parameters. Moreover, Gajos et al. (2006) provide evidence that duplicating important functionality from an all-encompassing control panel to an exposed position is a useful way to drive adaptable user interfaces. For example, toolbars allow for interaction that is most frequently used, whereas rarely applied tools have to be selected from an otherwise collapsed menu structure.

5.3 Basic Methods

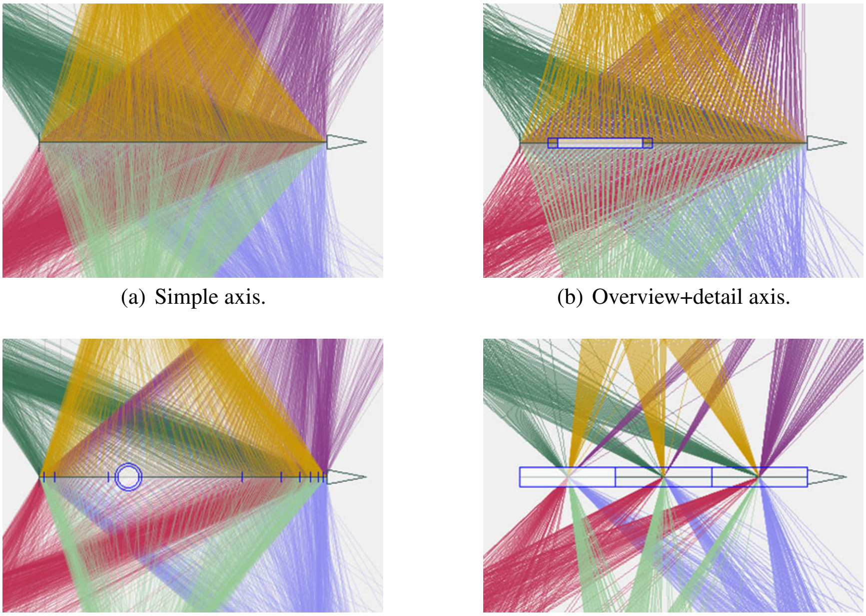
It is clear now that we need visualization views on the one hand, and interaction controls on the other hand. Views are usually equipped with visual representations, and we described many examples for time and time-oriented data in the previous chapters. Let us now take a closer look at interactive means of controlling the visualization beyond standard graphical user interface controls. To this end, we briefly describe *direct manipulation*, *brushing & linking*, and *dynamic queries* as key methods for interactive visualization and show how these methods can be applied for exploration and analysis of time-oriented data.

Direct manipulation

Working with a graphical user interface has the advantage that standardized components can be used to control the visualization process. However, a disadvantage is that visual feedback usually does not appear where the interaction was performed, but in a different part of the display, i.e., the view where the visualization is shown (see Figure 5.5). Direct manipulation as introduced by Shneiderman (1983) is the classic means to address this disadvantage. The goal is to enable users to directly manipulate the visual representation without a detour. To this end, a visualization view or graphical elements are implemented so as to be responsive to user input. A visualization may for instance allow zooming into details under the mouse cursor simply by rotating the mouse wheel, or visiting different parts of the visual representation simply by dragging the view. Such functionality is often present in zoomable user interfaces (see Cockburn et al., 2009). Virtual trackballs (see Henriksen et al., 2004) are more object-centric in that they allow users to grab and rotate virtual objects to view them from different angles.

In terms of interacting with visual representations of time-oriented data, navigating time is of particular importance. Many tools provide standard slider or calendar controls in the user interface to support navigation. However, these controls do not realize direct manipulation. In order to do so, interaction has to be tightly coupled with the display of the time axis. We explain what this means by two examples. Take a look back at Figure 5.5. You may notice that a slider for navigating in time is missing. Instead, the spiral display allows direct manipulation. For this purpose, small arrow-shaped handles are displayed at the beginning and at the end of the spiral. Clicking these handles navigates back and forth in time. Clicking and dragging the mouse off a handle adjusts the navigation speed, which is determined by the distance of the mouse cursor to the handle. A textual label provides precise feedback about the time span currently mapped to the spiral.

For a second example of direct manipulation, we refer to the interactive axes of the TimeWheel (↪ p. 200). Besides a simple non-interactive axis representation, the TimeWheel provides three different types of interactive axes: (1) overview+detail axis, (2) focus+context axis, and (3) hierarchical axis (see Figure 5.6). Each of these interactive axes displays time and additionally offers different options for direct



(a) Simple axis.
(b) Overview+detail axis.
(c) Focus+context axis.
(d) Hierarchical axis.

Fig. 5.6: The TimeWheel's mapping of time along the time axis can be manipulated directly in different ways. The simple axis uses a fixed linear mapping of time. The overview+detail axis allows users to select any particular range of the time domain to be mapped linearly to the time axis. The focus+context axis can be used to untangle dense parts of the time domain by applying a non-linear mapping. The hierarchical axis represents time at different levels of granularity, where individual axis segments can be expanded and collapsed.

manipulation. The overview+detail axis basically extends the simple axis with three interactive handles to control the position and extent of the time interval to be displayed in the TimeWheel, effectively allowing users to zoom and scroll into any particular part of the data. The focus+context axis applies a non-linear distortion to the time axis in order to provide more drawing space for the user's focus and less space for the context. This allows users to untangle dense parts of the data. Finally, for the hierarchical axis, the display is hierarchically subdivided into segments according to the different granularities of time (e.g., years, quarters, months, days). Users can expand or collapse these segments interactively to view the data at different levels of abstraction. Figure 5.6 illustrates the three types of interactive axes and the corresponding visual mapping compared to the simple non-interactive axis.

The advantage of directly manipulating the visual representation is, as indicated, that interaction and visual feedback take place at the very same location. However, direct manipulation always involves some learning and training of the interaction facilities provided. This is necessary because most of the time the interaction is not standardized but custom-made to fit the visual mapping.

Brushing & linking

Brushing & linking is a classic interaction concept, which takes up the idea of direct manipulation. Becker and Cleveland (1987) describe brushing as a technique that enables users to select interesting data items directly from a data display. There are various options one can follow when selecting data items. We will often find brushing being implemented as point and click interaction to select individual data items. Rubber-band or lasso interaction serve the purpose of brushing subranges in the data or multiple data items at once. Hauser et al. (2002) introduce brushing based on angles between data items, and Doleisch and Hauser (2002) go beyond binary selection to allow for smooth brushing (i.e., data items can be partly selected).

After brushing, selected data items are highlighted in order to make them stand out against the rest of the data. The key benefit of brushing & linking is that data selected in one view are automatically highlighted in all other views, effectively linking the views. In this sense, brushing & linking is a form of coordination among multiple views. This is especially useful when visualizing the variables of a multivariate time-oriented dataset individually in separate views. Then, brushing a temporal interval of interest in one view will highlight the same interval and corresponding data values in all views, and we can easily compare how the individual variables develop during the brushed time period.

For complex data, using a single brush is often unsatisfactory. Instead, users need to perform multiple brushes on different time-dependent variables or in different views. Compound brushing as explained by Chen (2004) allows the combination of individual brushes into composite brushes by using various operators, including logical, analytical, data-centric, and visual operations. With such facilities, brushing is much like a visual query mechanism.

Dynamic queries

Shneiderman (1994) describes dynamic queries as a concept for visual information seeking. It is strongly related to brushing & linking in that the goal is to focus on data of interest, which in the case of dynamic queries is often realized by filtering out irrelevant data. Because time-oriented data are often large, dynamically omitting data with respect to task-specific conditions can be very helpful.

Depending on the view characteristics and visualization tasks, two alternatives can be applied to display filtering results: filtered objects can be displayed in less detail or they can be made invisible. Reducing detail is useful in views that maintain an overview, where all information needs to be displayed at all times, but filtered objects need only to be indicated. Making objects invisible is useful in views that notoriously suffer from cluttering.

Filter conditions are usually specified using dedicated mechanisms. Threshold or range sliders are effective for filtering time or any particular numerical variable; textual filters are useful for extracting objects with specific labels (e.g., data tagged by season). Similar to what has been said for brushing & linking, the next logical

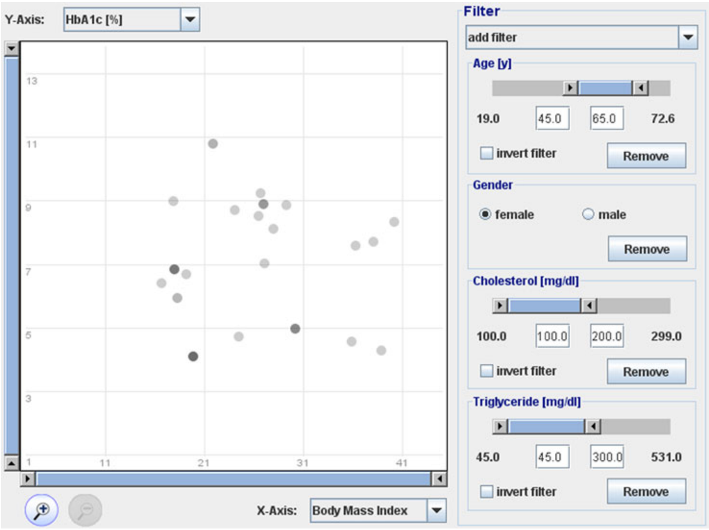


Fig. 5.7: Several filters can be adjusted in order to dynamically restrict the scatter plot visualization to data items that conform with the formulated conditions.

step is to combine filters to provide some form of multidimensional data reduction. For instance, a logical AND combination generates a filter that can be passed only if an object obeys all filter conditions; an object can pass a logical OR filter if it satisfies any of the involved filter conditions. Figure 5.7 shows an example of a dynamic query interface.

While some systems offer only fixed filter combinations or require users to enter syntactic constructs of some filter language, others implement a visual interface where the user can interactively specify filter conditions. Examples for querying time-oriented data that are visualized as line plot (↪ p. 153) are the time boxes by Hochheiser and Shneiderman (2004) and the relaxed selection techniques by Holz and Feiner (2009).

Time boxes are used to filter out variables of a multivariate line plot. To this end, the user marks regions in the visual display by creating one or more elastic rectangles that specify particular value ranges and time intervals. The system then filters out all variables whose plots do not overlap with the rectangles, effectively performing multiple AND-combined range queries on the data. Figure 5.8 depicts a query that combines three time boxes to restrict the display to stocks that performed well in the first and the last weeks of the year, but had a bad performance in the middle of year.

The relaxed selection techniques are useful for finding specific patterns in the data. For that purpose, the user specifies a query pattern by sketching it directly on the display. When the user is performing the sketching, either the distance of the sketch to the line plot or the user's sketching speed are taken into consideration in order to locally relax the query pattern. This relaxation is necessary to allow for a

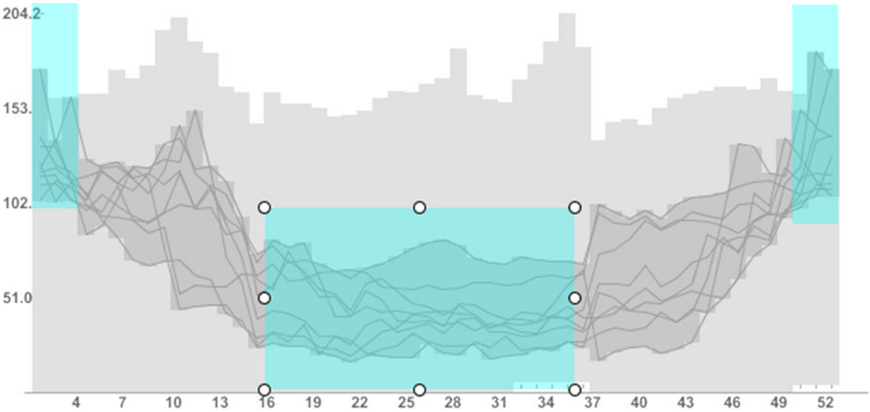


Fig. 5.8: Three time boxes are used to dynamically query stock data. Only those stocks are displayed that are high at the beginning, but low in the middle, and again high at the end of the year.
Source: Generated with the TimeSearcher software.

certain tolerance when matching the pattern in the data. An interactive display of the query sketch can be used to fine-tune the query pattern. Once the query pattern is specified, the system computes corresponding pattern matches and displays them in the line plot as depicted in Figure 5.9.

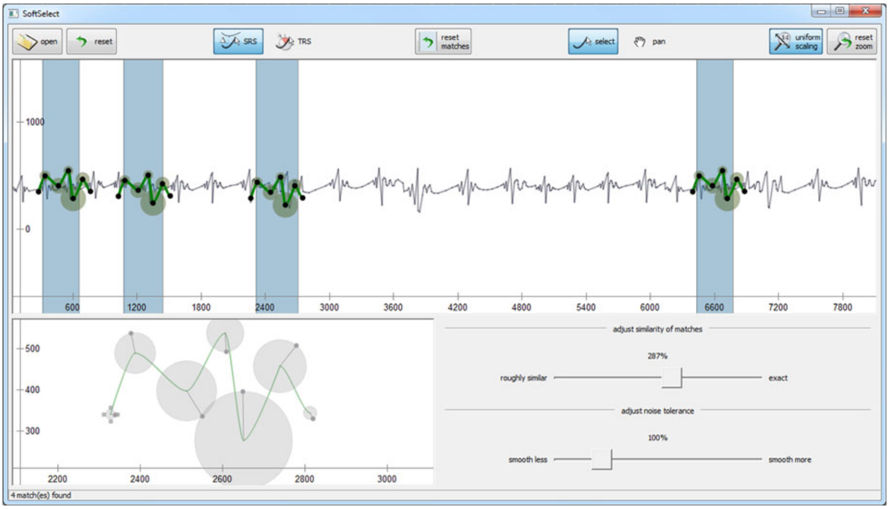


Fig. 5.9: The user can sketch a query pattern directly in the line plot and optionally refine it locally in a dedicated query view. The line plot then shows where in time the query matches with a certain tolerance.
Source: Image courtesy of Christian Holz.

Specifying dynamic queries visually as illustrated by the previous two examples is definitely very useful. However, the user can sketch or mark only those things which are displayed. Formulating queries with regard to potential but not yet existing patterns in the data beyond some tolerance requires additional formal query languages, whose expressiveness, in turn, depends on the formalism used.

Direct manipulation, brushing & linking, and dynamic queries are vital for effective exploration and analysis of time-oriented data. Although these concepts have existed for quite some time now, many visualization tools offer only a fraction of what is possible. Again one can find a reason for that in the higher costs accruing from designing and implementing efficient interaction methods. Moreover, because visual and interactive means must be coupled tightly, it is difficult to develop interaction techniques that can be interchanged among the different visualization techniques for time-oriented data. Finding a solution to this problem is an open research question.

5.4 Integrating Interactive and Automatic Methods

So far in this chapter, we have shown that interaction is mandatory to allow users to parameterize the visualization according to their needs and tasks. However, with increasing complexity of data and visualization methods alike, it is not always easy for users to find parameter values that suit the analysis task at hand. Particularly if parameters are not self-explanatory, they are not easy-to-set manually. So, some form of support is needed to assist users in the parametrization process.

A possible solution is to employ the concept of *event-based visualization*, which combines visualization with event methodology (see Reinders et al., 2001; Tominski, 2011). In diverse application fields, including active databases, software engineering, and modeling and simulation, events are considered happenings of interest that trigger some automatic actions. In the context of visualization, such an event-action-scheme is useful for complementing manual interaction with automatic parametrization of visual representations.

The basic idea of event-based visualization is (1) to let users specify their interests, (2) to detect if and where these interests match in the data, and (3) to consider detected matches when generating the visual representation. This general procedure requires the three main components: (1) *event specification*, (2) *event detection*, and (3) *event representation*. Figure 5.10 illustrates how they are attached to the visualization pipeline. Next we will look at each of these components in more detail.

Describing user interests

The event specification is an interactive step where users describe their interests as *event types*. To be able to find actual matches of user interests in the data, the event specification must be based on formal descriptions. Tominski (2011) uses elements

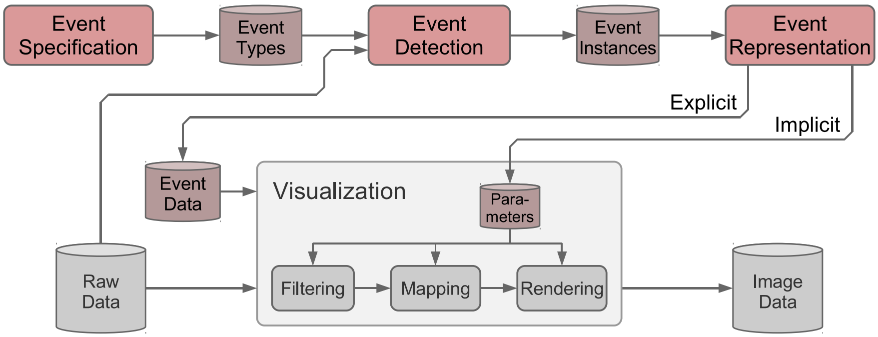


Fig. 5.10: The main ingredients of event-based visualization – event specification, event detection, and event representation – attached to the visualization pipeline.

of predicate logic to create well-defined event formulas that express interests with respect to relational datasets (e.g., data records whose values exceed a threshold or attribute with the highest average value). For an analysis of time-oriented data, sequence-related notations (for instance as introduced by Sadri et al., 2004) have to be added to enable users to specify conditions of interest regarding temporally ordered sequences (e.g., sequence of days with rising stock prices). A combination of event types to composite event types is possible via set operators.

To give a simple example of a sequence event type, we formulate the interest: *"Find three successive days where the number of people suffering from influenza increases by more than 15% each day."* This interest is expressed as:

$$\{(x,y,z)_{date} \mid z.flu \geq y.flu \cdot 1.15 \wedge y.flu \geq x.flu \cdot 1.15\}$$

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, casual users will have difficulties in describing their interests by using event formulas directly. Sufficient specification support starts with providing individual means for experts, regular users, and visualization novices. In this regard, one can think of three different levels of specification: *direct specification*, *specification by parametrization*, and *specification by selection*. Although all levels are based on the same formalism, the complete functionality of the formalism is available only to expert users at the level of direct specification. The idea for the second level is to hide the complexity of event formulas from the user. To this end, parameterizable templates are provided. The user can adjust the templates via easy-to-set parameters, but otherwise has no access to the possibly complicated internal event formula. The amount of increase in the previous event type example is a good candidate for a template parameter. The increase can then be changed from 15% to any particular value without rephrasing the entire event type formula. The third level of event specification is based on simple selection. The idea is to provide a predefined collection of event types that are particularly tailored to the application context, and that are equipped with expressive labels and descriptions, so that users can easily select what 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). Devising such a semantic structure and describing event types expressively is a task for domain experts.

Finding relevant data portions

The event detection is an automatic step that determines whether the interests defined interactively are present in the data. The outcome of the event detection is a set of *event instances*. They 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 data entities. In a second step, predicates, functions, and logical connections are evaluated, so that the event formula as a whole can be evaluated as either true or false. Because this procedure can be very costly in terms of computation time, efficient methods must be utilized for the event detection. A combination of the capabilities of relational database management systems and efficient algorithms (e.g., the OPS algorithm by Sadri et al., 2004) is useful for static data. When dynamic data (i.e., data that change over time, see Section 3.3) 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, because they do not have access to the entire dataset.

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. The three requirements that have to be considered are:

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.

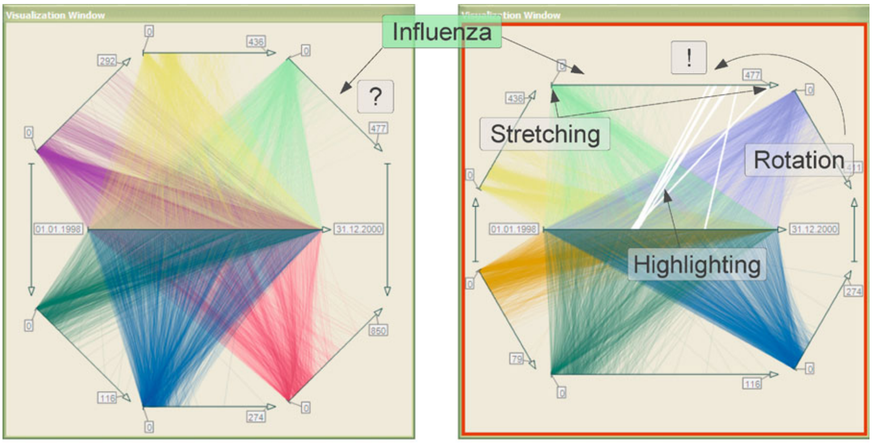
Most importantly, the visual representation must clearly express that something interesting is contained in the data. To meet this requirement, easy-to-perceive visual cues (e.g., a red frame around the visual representation, exclamation marks, or annotations) must be incorporated. Alpha blending can 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 the particular event type is).

However, when facing arbitrarily definable event formulas, this last requirement is difficult to fulfill.

We can distinguish two basic options for representing events: *explicit* and *implicit* event representation. For the explicit case, the focus is set exclusively on event instances, neglecting the raw data. Since the number of events is usually smaller than the number of data items, even large datasets can be analyzed. In the implicit case, the data context is retained and visualization parameters are set automatically so as to highlight points of interest 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 solely by adapting visualization parameters. Apparently, availability of adequate visualization parameters is a prerequisite for implicit event representation.

We will illustrate the potential of event-based visualization with an example. Let us assume a user who has to analyze multivariate time-dependent human health data for uncommonly high numbers of cases of influenza. The task at hand is to find out if and where in time these situations have occurred. A possible way to accomplish this task is to use the TimeWheel technique (↪ p. 200).

Figure 5.11(a) shows a TimeWheel that uses the standard parametrization, where time is encoded along the central axis and multiple diagnoses are mapped to the



(a) Default parametrization. (b) Targeted parametrization.

Fig. 5.11: Default vs. targeted parameterization of a TimeWheel. (a) TimeWheel representing a time-dependent health dataset using the default configuration, which aims at showing main trends, but does not consider the interests of the user. (b) TimeWheel representing the same data, but matches with the user's interests have been detected and corresponding data are emphasized via highlighted lines and automatic rotation and stretching; the presentation is better targeted to the user's task at hand.

axes surrounding the time axis. In particular, influenza happens to be the diagnosis that is mapped to the upper right axis (light green). Alpha-blending is applied by default to reduce visual clutter. Looking at this TimeWheel, the user can only guess from the labels of the axis showing influenza that there are higher numbers of cases, because 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 this, in an event-based visualization environment, the user can specify the interest *"Find days with a high number of cases of influenza."* as the event type ($\{x \mid x.flu \geq 300\}$). If the event detection step confirms the existence of such events in the data, visualization parameters are altered automatically so as to provide an individually adjusted TimeWheel that reflects the special situation. In our particular example, we switch 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. Additionally, rotation and stretching is applied such that the axis representing influenza is moved gradually to an exposed position and is provided with more display space. 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 automatically in response to event instances is depicted in Figure 5.11(b). In this TimeWheel, the identification of days with higher numbers of influenza infections is easy.

5.5 Summary

The focus of this chapter was on interaction. We started with a brief overview of intents that motivate users to interact with the visualization. The most notable intent in the context of time-oriented data is the intent to navigate in time in order to visit different parts of the data. Users also need to view time-oriented data at different levels of detail, because the data are often given at multiple granularities. Further intents are related to interactively adjusting the visual mapping according to data and tasks at hand, and to managing the exploration process.

We explained that interactive visualization is an iterative loop where the computer generates feedback in oder to visually reflect the change that resulted from user interaction. This human-in-the-loop process brings together the computational power of the machine and the intellectual power of human beings. In order to take full advantage of this synergy, an efficient user interface is needed that bridges the gap between the algorithmic structures used for visualizing time and time-oriented data, and the mental models and analytic workflows of users. This also includes tackling technical challenges to guarantee smooth execution of the interaction loop.

This chapter also presented brief descriptions of basic interaction concepts, including direct manipulation, brushing & linking, and dynamic queries. These concepts are vital for data exploration tasks where the user performs an undirected search for potentially interesting data features. But still, the potential of these concepts has not been fully exploited by current visualization techniques for time and time-oriented data. There is room for future work to better adapt existing interaction methods or to develop new ones according to the specific needs of time-oriented data.

The example of event-based visualization indicates that a combination of interactive, automatic, and visual methods is quite useful for generating visual representations that are better targeted for the user's task at hand than standard representations. This holds true as long as users know what they are looking for (i.e., users perform a directed search). Event-based visualization as described here is not suited to automatically mine potential events in time-oriented data. In order to do so, one needs to integrate analytical methods as described in the next chapter.

References

Baldonado, M. Q. W., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for Using Multiple Views in Information Visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 110–119, New York, NY, USA. ACM Press.

Becker, R. A. and Cleveland, W. S. (1987). Brushing Scatterplots. *Technometrics*, 29:127–142.

Bertin, J. (1981). *Graphics and Graphic Information-Processing*. de Gruyter, Berlin, Germany. translated by William J. Berg and Paul Scott.

Chen, H. (2004). Compound Brushing Explained. *Information Visualization*, 3(2):96–108.

Cockburn, A., Karlson, A., and Bederson, B. B. (2009). A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, 41(1):2:1–2:31.

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

Doleisch, H. and Hauser, H. (2002). Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 147–154, Plzen, Czech Republic. University of West Bohemia.

Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. (2006). Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 201–208, New York, NY, USA. ACM Press.

Hauser, H., Ledermann, F., and Doleisch, H. (2002). Angular Brushing of Extended Parallel Coordinates. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 127–130, Los Alamitos, CA, USA. IEEE Computer Society.

Heer, J. and Robertson, G. (2007). Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247.

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

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.

Holz, C. and Feiner, S. (2009). Relaxed Selection Techniques for Querying Time-Series Graphs. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 213–222, New York, NY, USA. ACM Press.

Jankun-Kelly, T., Ma, K.-L., and Gertz, M. (2007). A Model and Framework for Visualization Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):357–369.

Krasner, G. E. and Pope, S. T. (1988). A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49.

Norman, D. A. (2002). *The Design of Everyday Things*. Basic Books (Perseus), New York, NY, USA.

Piringer, H., Tominski, C., Muigg, P., and Berger, W. (2009). A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120.

Pulo, K. (2007). Navani: Navigating Large-Scale Visualisations with Animated Transitions. In *Proceedings of the International Conference Information Visualisation (IV)*, Los Alamitos, CA, USA. IEEE Computer Society.

Reinders, F., Post, F. H., and Spoelder, H. J. (2001). Visualization of Time-Dependent Data with Feature Tracking and Event Detection. *The Visual Computer*, 17(1):55–71.

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.

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

Shneiderman, B. (1994). Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77.

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.

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.

Tominski, C. (2011). Event-Based Concepts for User-Driven Visualization. *Information Visualization*, 10(1):65–81.

Tominski, C. and Schumann, H. (2008). Enhanced Interactive Spiral Display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity*, pages 53–56, Linköping, Sweden. Linköping University Electronic Press.

Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. (2007). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231.

Chapter 6
Analytical Support

> It is useful to think of the human and the computer together as a single cognitive entity, with the computer functioning as a kind of *cognitive coprocessor* to the human brain. [...] Each part of the system is doing what it does best. The computer can pre-process vast amounts of information. The human can do rapid pattern analysis and flexible decision making.
>
> Ware (2008, p. 175)

Visualization and interaction as described in the previous chapters help users to visually analyze time-oriented data. Analysts can look at the data, explore them, and in this way understand them. This is possible thanks to human visual perception and the fact that humans are quite good at recognizing patterns, finding interesting and unexpected solutions, combining knowledge from different sources, and being creative in general[1]. This holds true unless the problem to be solved exceeds a certain size. Very large time-series or data that consist of many thousands of time-dependent variables can usually not be grasped by human observers. In such cases, we need the proficiency of computing systems to assist the knowledge crystallization from time-oriented data. Apparently, if the problem size is sufficiently large, computers are better (i.e., faster and more accurate) than humans at numeric and symbolic calculations, logical reasoning, and searching.

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 (see Fayyad et al., 1996, 2001; Han and Kamber, 2005). A variety of concepts and methods are involved in achieving this goal, including databases, statistics, artificial intelligence, neural networks, machine learning, information retrieval, pattern recognition, data visualization, and high-performance computing.

This chapter will illustrate how automatic analytical calculations can be utilized to facilitate the exploration and analysis of larger and more complex time-oriented

[1] Wegner (1997) makes some interesting statements about why interaction is better than algorithms.

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_6,

data. To this end, we will give a brief overview of typical temporal analysis tasks. For selected tasks, we will present examples that demonstrate how visualization can benefit from considering analytical support. Our descriptions will intentionally be kept at a basic level. For details on the sometimes quite complex matter of temporal data analysis, we refer interested readers to the relevant literature.

6.1 Temporal Analysis Tasks

Temporal analysis and temporal data mining are especially concerned with extracting useful information from time-oriented data. More specifically, analytical methods for time-oriented data address the following categories of tasks (see Antunes and Oliveira, 2001; Laxman and Sastry, 2006; Hsu et al., 2008; Brockwell and Davis, 2009; Mitsa, 2010):

Classification Given a predefined set of classes, the goal of classification is to determine which class a dataset, sequence, or subsequence belongs to. Applications such as speech recognition and gesture recognition apply classification to identify specific words spoken or interactions performed. The analysis of sensor data or spatio-temporal movement data often requires classification to make the enormous volumes of data to be handled manageable.

Clustering Clustering is concerned with grouping data into clusters based on similarity, where the similarity measure used is a key aspect of the clustering process. In the context of time-oriented data, it makes sense to cluster similar time-series or subsequences of them. For example, in the analysis of financial data, one may be interested in stocks that exhibit similar behavior over time. In contrast to classification, where the classes are known a priori, clusters are not defined upfront.

Search & retrieval This task encompasses searching for a priori specified queries in possibly large volumes of data. This is often referred to as *query-by-example*. Search & retrieval can be applied to locate exact matches for an example query or approximate matches. In the latter case, similarity measures are needed that define the degree of exactness or fuzziness of the search (e.g., to find customers whose spending patterns over time are similar but not necessarily equal to a given spending profile).

Pattern discovery While search & retrieval requires a predefined query, pattern discovery is concerned with *automatically* discovering interesting patterns in the data (without any a priori assumptions). The term *pattern* usually covers a variety of meanings, including sequential pattern, periodic pattern, but also temporal association rules. In a sense, a pattern can be understood as a local structure in the data or combinations thereof. Often, frequently occurring patterns are of interest, for example when analyzing whether a TV commercial actually leads to an increase in sales. But patterns that occur very rarely can also be interesting because they might indicate malicious behavior or failures.

Prediction An important task in analyzing time-oriented data is the prediction of likely future behavior. The goal is to infer from data collected in the past and present how the data will evolve in the future. To achieve this goal one first has to build a predictive model for the data. Examples of such models are autoregressive models, non-stationary and stationary models, or rule-based models.

In the context of visualization, these tasks share a common goal: *data abstraction* in order to reduce the workload when computing visual representations and to keep the perceptual efforts required to interpret them low. For classification and clustering, we abstract from the raw data and work with classes and clusters. For search & retrieval and pattern discovery we are foremost interested in relevant patterns and de-emphasize irrelevant data. For prediction, we focus on the future.

A variety of methods have to play in concert in order to accomplish temporal analysis tasks. Statistical aggregation operators (e.g., sum, average, minimum, maximum, etc.), methods from time-series analysis, as well as dedicated temporal data mining techniques are needed.

In what follows, we demonstrate the applicability of analytical methods for the analysis of time-oriented data using the three examples: clustering, temporal data abstraction, and principal component analysis. Clustering decreases the number of data items to be represented, and allows the discernment of similarities and unexpected behavior. Temporal data abstraction reduces data complexity by deriving qualitative statements, which are much easier to understand. Principal component analysis decreases the number of time-dependent variables by switching the focus to major trends in the data.

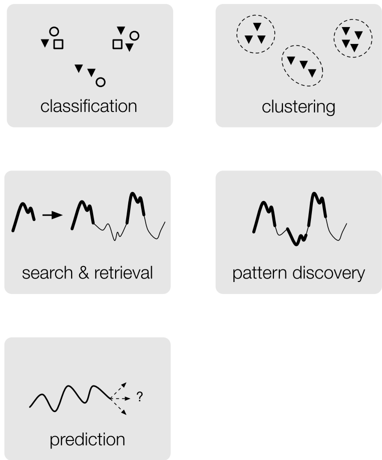


Fig. 6.1 Overview of temporal analysis tasks.

6.2 Clustering

In general, grouping data into clusters and concentrating on the clusters rather than on individual data values allows the analysis of much larger datasets. Appropriate *distance* or *similarity measures* lay the ground work for clustering. Distance and similarity measures are profoundly application dependent and range from average geometric distance, to measures based on longest common subsequences, to measures based on probabilistic models. Based on computed distances, clustering methods create groups of data, where the number of available techniques is large, including hierarchical clustering, partitional clustering, and sequential clustering. Due to the diversity of methods, selecting appropriate algorithms is typically difficult. Careful adjustment of parameters and regular validation of the results are therefore essential tasks in the process of clustering. More details on clustering methods and distance measures can be found in the work by Jain et al. (1999), Gan et al. (2007), and Xu and Wunsch II (2009).

A prominent example of how analytical methods can assist the visualization of time-oriented data is the work by Van Wijk and Van Selow (1999). The goal is to identify common and uncommon subsequences in large time-series data and to understand their distribution over time. The problem is that simply drawing line plots for all subsequences is not a satisfactory solution due to the overwhelmingly large number of time points and line plots. In order to tackle this problem, clustering methods and a calendar-based visualization are used.

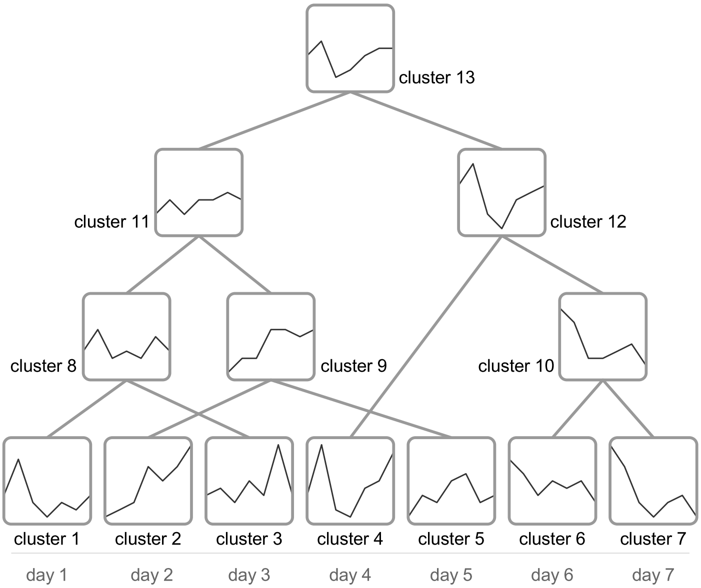


Fig. 6.2: By repeatedly merging the two most similar sequences into new clusters, a clustering hierarchy is generated. The root cluster is an aggregated representative of the entire dataset.

In particular, the approach works as follows. As Van Wijk and Van Selow (1999) are interested in patterns on the granularity of days, the first step is to split a large time-series into k day patterns, each of which stores the subsequence for one day. The clustering process starts with the k day patterns as initial clusters. Then the differences of all possible combinations of two clusters are computed and the two most similar clusters are merged into a new cluster (i.e., an aggregated representative of the two clusters). This process runs repeatedly and results in a *clustering hierarchy* with $2k-1$ clusters, where the root of the hierarchy represents the entire dataset in an aggregated fashion. Figure 6.2 illustrates the clustering process with data for seven days.

The visualization of the clustered day patterns uses two different views for the two analysis tasks: (1) assess similarity among day patterns and (2) locate common and uncommon patterns over time. The first task is facilitated by a basic line plot (↪ p. 153) that shows a selected number of clusters, where each plot uses a unique color. To accomplish the second task, a calendar display is used where individual days are color-coded according to cluster affiliation. This way, analysts can see the day pattern and at the same time understand when during a year this pattern occurs. Various interaction methods allow adjustments of the visual representation and data exploration. In terms of assessing similarities, the user can select a day from the calendar and with the help of the clustering hierarchy, similar days (and clusters) can be retrieved automatically.

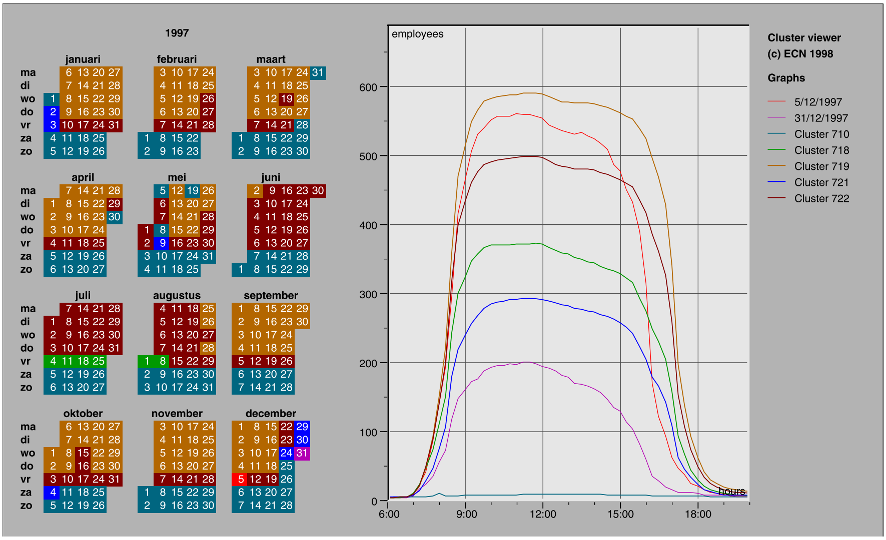


Fig. 6.3: Visual analysis of the number of employees at work. Day patterns for selected days and clusters are visualized as line plots (right). Individual days in a calendar display (left) are colored according to cluster affiliation.
Source: Van Wijk and Van Selow (1999), © 1999 IEEE. Used with permission.

Figure 6.3 shows an example of the visualization design. The data displayed in the figure contain the number of employees at work. The line plot currently shows the day patterns of two days (5/12/1997 and 31/12/1997) and five clusters (710, 718, 719, 721, and 722). Van Wijk and Van Selow (1999) demonstrate that several conclusions can be drawn from the visual representation. To give only a few examples:

- Employees follow office hours quite strictly and work between 8:30 am and 5:00 pm in most cases.
- Fewer people work on Fridays during summer (cluster 718).
- During weekends and holidays only very few people are at work (cluster 710).
- It is common practice to take a day off after a holiday (cluster 721).

These and similar statements were more difficult or even impossible to derive without the integration of clustering. Van Wijk and Van Selow (1999) most convincingly demonstrate the advantages of analytical support for the visual analysis of time-oriented data. While here the benefit lies in the abstraction from raw data to aggregated clusters, we will see in the next section that other kinds of abstraction are useful as well.

6.3 Temporal Data Abstraction

In practice, time-oriented datasets are often large and complex and originate from heterogeneous sources. The challenging question is how huge volumes of possibly continuously measured data can be analyzed to support 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 as well. What is needed is a way to abstract the data in order to make them eligible for subsequent visualization.

The term *data abstraction* was originally introduced by Clancey (1985) in his classic proposal on heuristic classification. In general, the objective of data abstraction is:

> ... to create an abstraction that conveys key ideas while suppressing irrelevant details.
>
> Thomas and Cook (2005, p. 86)

The basic idea is to use qualitative values, classes, or concepts, rather than raw data, for further analysis or visualization processes (see Lin et al., 2007; Combi et al., 2010). This helps in coping with data size and data complexity. To arrive at suitable data abstractions, several tasks must be conducted, including selecting relevant information, filtering out unneeded information, performing calculations, sorting, and clustering.

Principles

Let us now illustrate the concept of *temporal data abstraction* in medical contexts with a simple example. Figure 6.4 shows time-oriented data as generated when monitoring newborn infants that have to be ventilated artificially. The figure visualizes three variables plotted as points against a horizontal time axis: S_aO_2 (arterial oxygen saturation), $P_{tc}CO_2$ (transcutaneous partial pressure of carbon dioxide), and P_aCO_2 (arterial partial pressure of carbon dioxide). S_aO_2 and $P_{tc}CO_2$ are measured continuously at a regular rate, but with different frequency. New values for P_aCO_2 arrive irregularly and some values for $P_{tc}CO_2$ are missing.

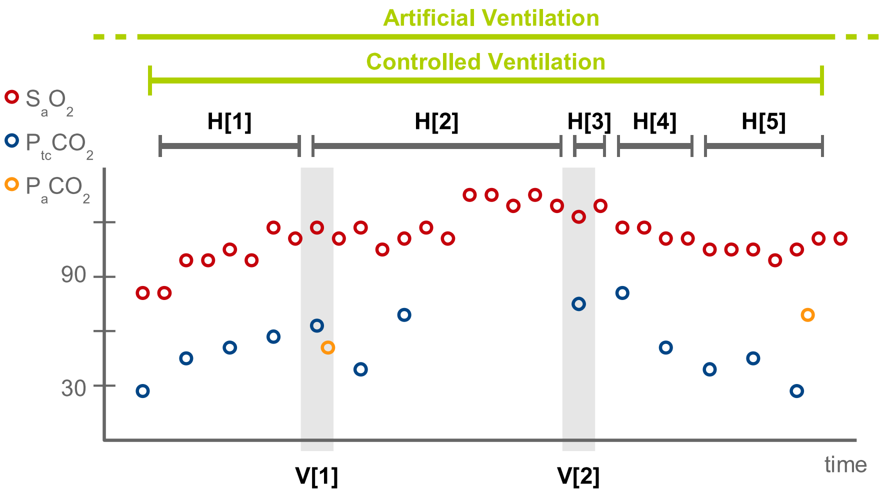


Fig. 6.4: Temporal data abstraction in the context of artificial ventilation. Vertical temporal abstractions are illustrated as V[1] and V[2] and horizontal temporal abstraction are illustrated as H[1] – H[5]. The context is given as "artificial ventilation" and its sub-context "controlled ventilation".

The aim of temporal data abstraction is to arrive at qualitative values or patterns over time intervals. *Vertical* temporal abstraction (illustrated in V[1] and V[2]) considers multiple variables over a particular time point and combines them into a qualitative value or pattern. *Horizontal* temporal abstraction (illustrated as H[1] – H[5]) infers a qualitative value or pattern from one or more variables and a corresponding time interval. Usually the abstraction process is context-dependent. In Figure 6.4, the abstraction is done in the context of artificial ventilation and in the sub-context of controlled ventilation.

In medical applications, there are different types of abstraction methods, ranging from rather simple to quite complicated ones. However, as pointed out by Combi et al. (2010), no exhaustive schema exists to categorize the available methods. Nevertheless, the common understanding is that even in very simple cases the process is knowledge-driven. The use of knowledge is the main characteristic that distinguishes data abstraction from statistical data analysis (e.g., trend detection using time-series analysis).

Simple methods involve single data values and usually do not need to consider time specifically. They generate vertical abstractions. The knowledge used are concept associations or concept taxonomies. Combi et al. (2010) distinguish three types of simple methods:

- *Qualitative abstraction* means converting numeric expressions to qualitative expressions. For example, the numeric value of 34.8°C of body temperature can be abstracted to the qualitative value "hypothermia".
- *Generalization abstraction* involves a mapping of instances into classes. For example, "hand-bagging is administered" is abstracted to "manual intervention is administered", where "hand-bagging" is an instance of the concept class "manual intervention".
- *Definitional abstraction* is a mapping across different concept categories. The movement here is not within the same concept taxonomy, as for the generalization abstraction, but across two different concept taxonomies.

More complex methods consider one or more variables jointly and specifically integrate the dimension of time in a kind of temporal reasoning. These methods generate horizontal temporal abstractions. According to Combi et al. (2010), four types of complex methods exist:

- *Merge (or state) abstraction* is the process of deriving maximal time intervals for which some constraints of interest hold. For example, several consecutive days with high fever and increased blood values can be mapped to "bed-ridden".
- *Persistence abstraction* means applying persistence rules to project maximal intervals for some property, both backwards and forwards in time. For example, "headache in the morning", can be abstracted to "headache in the evening before" or "headache in the afternoon afterwards".
- *Trend (or gradient or rate) abstraction* is concerned with deriving significant changes and rates of change in the progression of some variable. For example, $P_{tc}CO_2$ has decreased from 130 to 90 in the last 20 minutes would result in "$P_{tc}CO_2$ is decreasing too fast".
- *Periodic abstraction* aims to derive repetitive occurrence, with some regularity in the pattern of repetition. For example, "headache every morning, but not during the day", would result in "repetitive headache in the morning".

Application examples

The principles described in the previous paragraphs can be applied in various ways. In the following, we will give a few examples of systems that utilize temporal data abstraction. For more examples, we refer to the survey of temporal data abstraction in clinical data analysis by Stacey and McGregor (2007).

Monitoring artificially ventilated infants VIE-VENT is an open-loop knowledge-based monitoring and therapy planning system for artificially ventilated infants (see Miksch et al., 1996). In order to derive qualitative descriptions for different kinds of temporal trends (i.e., very-short, short, medium, and long-term trends) from continuously arriving quantitative data, the system utilizes context-sensitive and expectation-guided methods and incorporates background knowledge about data points, data intervals, and expected qualitative trend patterns. Smoothing and adjustment mechanisms help to keep qualitative descriptions stable in case of shifting contexts or data oscillating near thresholds. Context-aware schemata for data point transformation and curve fitting are used to express the dynamics of and the reaction to different data abnormalities. For example, during intermittent positive pressure ventilation (ippv), the transformation of the quantitative value $P_{tc}CO_2 = 56mmHg$ results in the qualitative abstraction "$P_{tc}CO_2$ substantially above target range". During intermittent mandatory ventilation (imv) however, $56mmHg$ represents the "target value". Qualitative abstractions and schemata of curve fitting are subsequently used to decide if the value progression happens too fast, at normal rate, or too slow.

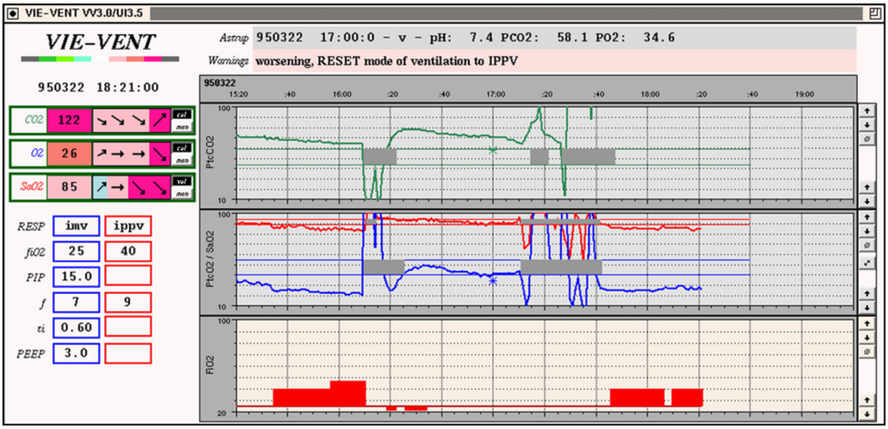


Fig. 6.5: VIE-VENT displays measured quantitative values as line plots. Qualitative abstractions and trends are represented by different colors and arrows in the top three boxes on the left. *Source: Miksch et al. (1996), © 1996 Elsevier. Used with permission.*

Figure 6.5 shows the user interface of VIE-VENT. In the top-left corner, the system displays exact values of the quantitative blood gas measurements CO2, O2, SaO2. Arrows depict trends and qualitative abstractions are indicated by different colors (e.g., deep pink represents "extremely above target range"). The left panel further shows current and recommended ventilator settings in blue and red boxes, respectively. The right-hand side shows line plots of the most important variables for the last four hours.

Dealing with oscillating data Strongly oscillating data pose a formidable challenge for methods that aim to extract qualitative abstractions and patterns from the data. The problem is that derived abstractions could change too quickly as to be interpretable by the observer. Therefore, Miksch et al. (1999) developed the Spread, a time-oriented data abstraction method that is capable of deriving steady qualitative abstractions from oscillating high-frequency data. The tool performs the following steps of processing and data abstraction:

1. *Eliminate data errors*: Sometimes up to 40% of the input data are obviously erroneous, i.e., exceed the limits of plausible values.
2. *Clarify the curve*: Transform the still noisy data into the *spread*, which is a steady curve with some additional information about the distribution of the data along that curve.
3. *Qualify the curve.* Abstract from quantitative values to qualitative values like "normal" or "high" and concatenate intervals with equal qualitative values.

Figure 6.6 illustrates how the analytical abstractions can enhance the visualization. The Spread smooths out the strongly oscillating raw data. Even the increased oscillation in the center of the display is dealt with gracefully: it leads to increased width of the spread, but not to a change of the qualitative value. With these abilities, the Spread can support physicians in making better qualitative assessments of otherwise difficult-to-interpret data.

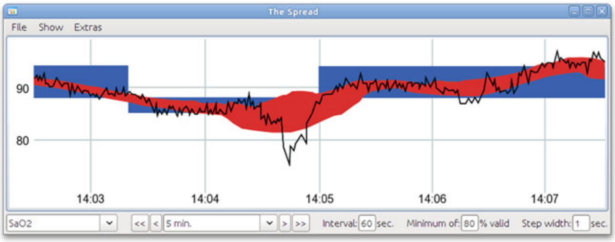


Fig. 6.6: The thin line shows the raw data. The red area depicts the *spread* and the blue rectangles represent the derived temporal intervals of steady qualitative values. The lower part of the figure shows the parameter settings.
Source: Adapted from Miksch et al. (1999), © 1999 Springer. Used with permission.

Linking temporal and visual abstraction In interactive environments, the visualization of time-oriented data and abstractions thereof can change dynamically due to user interaction, where resizing and zooming are among the most commonly applied operations. In such scenarios, the visualization must be able to capture as much temporal information as possible without losing overview and details, even if the available display space is very limited. Bade et al. (2004) demonstrate that this is possible by means of *semantic zooming* (see p. 112 and ↪ p. 230). The semantic zoom functionality relies on an appropriate set of temporal data abstractions and associated visual representations for different levels of detail as illustrated in Figure 6.7. Depending on the available display space (or the current zoom level), a suitable temporal abstraction is selected automatically and its corresponding visual abstraction is displayed. The advantage of this procedure is that it relieves the user of managing the levels of abstraction by hand. Moreover, the semantic zoom corresponds much better with the interactive nature of flexible and dynamic visual analysis scenarios.

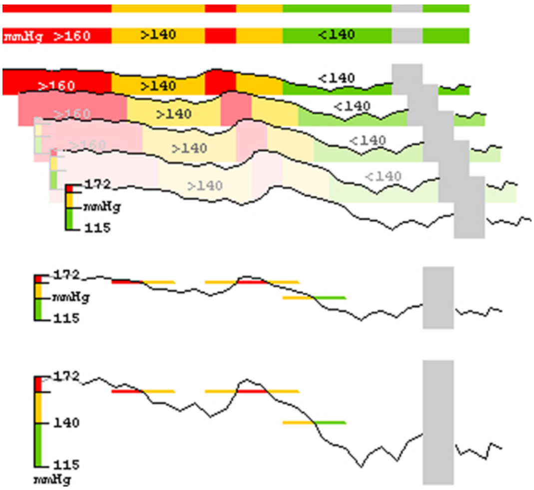


Fig. 6.7 Different steps of semantic zooming of a time-series visualization from a broad overview with qualitative values (top) to a detailed view with fine structures and quantitative details (bottom). Gray areas indicate missing data.

The examples described can only indicate the possible benefits that basic and complex temporal abstraction methods and their integration with the visualization can have for dealing with time-oriented data in medical applications. We know of quite positive feedback from medical experts who found it easy to capture the health conditions of their patients. Moreover, these qualitative abstractions can be used for further reasoning or in guideline-based care for a simplified representation of treatment plans.

What our previous examples have in common, however, is that they consider only a relatively small number of time-dependent variables. As we will see in the next section, if the number of variables gets larger, we need further analytical methods.

6.4 Principal Component Analysis

Time-oriented data are often of multivariate nature, but too large a number of variables poses considerable difficulties for the visualization. These difficulties can be overcome by applying principal component analysis (PCA), which offers an excellent basis for data abstraction (see Jolliffe, 2002; Jackson, 2003; Jeong et al., 2009).

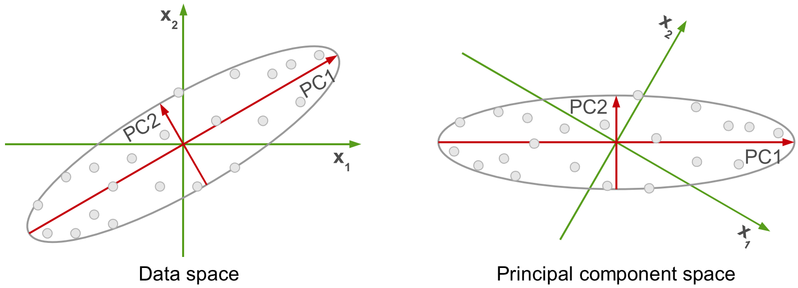


Fig. 6.8: Principal component analysis transforms multivariate data (with variables $\mathbf{x_1}$ and $\mathbf{x_2}$ in this case) into a new space, the so-called principal component space, which is spanned by the principal components (here PC1 and PC2).

The key principle of PCA is a transformation of the original data space into the principal component space (see Figure 6.8). In the principal component space, the first coordinate, that is, the first principal component represents most of the original dataset's variance, the second principal component, which is orthogonal to the first one, represents most of the remaining variance, and so on. Visualizing the data in the new principal component space shows us how closely individual data records are related to the major trends, and thus PCA helps us to reveal the internal structure of the data. Moreover, since principal components are ordered by their significance, we can focus on fewer principal components than we have variables in our data.

In the following we will take a brief look at the basics of principal component analysis and illustrate by means of examples the benefit that this analytical concept has for the visual analysis of time-oriented data.

Basic method

Assume that we have modeled our multivariate dataset as a matrix:

$$\mathbf{X} = \left(\mathbf{x_1}\mathbf{x_2}\cdots\mathbf{x_m}\right) = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ x_{2,1} & \cdots & x_{2,m} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix}$$

where the columns of $\mathbf{X}$ correspond to the m variables $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}$ of the dataset, and the rows represent n records of data (e.g., n repetitions of an experiment). For a time-oriented dataset, one of the $\mathbf{x_i}$ is usually the dimension of time.

Depending on the application it can make sense to prepare the data such that they are mean-centered and normalized (by subtracting off the mean of each variable and scaling each variable according to its variance). Now our goal is to transform the data into the principal component space that is spanned by $r \leq m$ principal components.

For the purpose of explanation, we resort to *singular value decomposition (SVD)* according to which any matrix $\mathbf{X}$ can be decomposed as:

$$\mathbf{X} = \mathbf{W} \cdot \Sigma \cdot \mathbf{C}^T$$

where $\mathbf{W}$ is an $n \times r$ matrix, Σ is an $r \times r$ diagonal matrix, and $\mathbf{C}^T$ is an $r \times m$ matrix:

$$\mathbf{X} = \begin{pmatrix} w_{1,1} & \cdots & w_{1,r} \\ w_{2,1} & \cdots & w_{2,r} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ w_{n,1} & \cdots & w_{n,r} \end{pmatrix} \cdot \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{pmatrix} \cdot \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \vdots & \vdots & & \vdots \\ c_{r,1} & c_{2,2} & \cdots & c_{r,m} \end{pmatrix}$$

The matrix $\mathbf{C}^T$ has in its rows the transposed eigenvectors $\mathbf{c_1}^T, \ldots, \mathbf{c_r}^T$ of the matrix $\mathbf{X}^T\mathbf{X}$, which corresponds to the *covariance matrix* of the original dataset. The $\mathbf{c_i}$ form the orthonormal basis of the principal component space; they are the principal components. Each $\mathbf{c_i}$ is the result of a linear combination of the original variables where the factors (or *loadings*) of the linear combination determine how much the original variables contribute to a principal component. The first principal component $\mathbf{c_1}$ is chosen so as to be the one that captures most of the original data's variance, the second principal component most of the remaining variance, and so forth. The significance values $\sigma_1, \ldots, \sigma_r$ in Σ are determined by the likewise ranked square roots of the eigenvalues $\sqrt{\lambda}_1, \ldots, \sqrt{\lambda}_r$ of the eigenvectors (i.e., the principal components) $\mathbf{c_1}, \ldots, \mathbf{c_r}$. Finally, the i-th row of the matrix $\mathbf{W}$ contains the coordinates of the i-th data record in the new principal component space. The individual coordinates are often referred to as the *scores*.

This brief formal explanation provides a number of key take-aways. Let us summarize the ones that are most relevant for visualization:

- the significance values determine the ranking of principal components,
- the ranking is the basis for data abstraction, where principal components that bear little information can be omitted,
- the loadings describe the relationship of the original data variables and the principal components, and
- the scores describe the location of the original data records in the principal component space.

Application examples

We will now demonstrate how PCA can be applied to enhance the visual analysis of time-oriented data. Our general goal is to uncover structure in the data and to reduce the analysis complexity by focusing on significant trends. In a first example, we will see that even a single principal component can bear sufficient information for discerning main trends in the data. Secondly, an example will illustrate how one can determine the principal components to be retained for the visualization as well as the ones that can be omitted due to their low significance.

Before we start with the examples, however, it is important to mention that PCA does not distinguish between independent and dependent variables. In particular, the dimension of time is processed indiscriminately, which sacrifices the temporal dependencies in the data. Therefore, it is often preferable to exclude time from the analysis, and to rejoin time and computed principal components afterwards to restore the temporal context. This is what we will do in the next example.

Revealing internal structures with PCA We consider the visual analysis of a meteorological dataset that contains daily observations of temperature (T_{min}, T_{avg}, and T_{max}) for a period of 105 years, which amounts to approximately 38,000 data records (see Nocke et al., 2004). As we are only interested in the summer seasons' weather conditions, the daily raw data are first aggregated into yearly data. To this end, five new variables are calculated for each year:

- *total heat (p1)* as the sum of the maximum temperatures for days with $T_{max} \geq 20°C$,
- *summer days (p2)* as the number of days with $T_{max} \geq 25°C$,
- *hot days (p3)* the number of days with $T_{max} \geq 30°C$,
- *mean of average (p4)* as the mean of the daily average temperatures T_{avg}, and
- *mean of extreme (p5)* as the mean of the daily maximum temperatures T_{max}.

These five quantitative variables are strongly correlated. The extracted dataset can be visualized as a centered layer area graph (↪ p. 195), as illustrated in Figure 6.9. This visual representation is quite useful to get an overview of the data. We can clearly distinguish valleys and peaks in the graph, which indicate particularly cold and hot summers, respectively. The general trend in the data is communicated quite well.

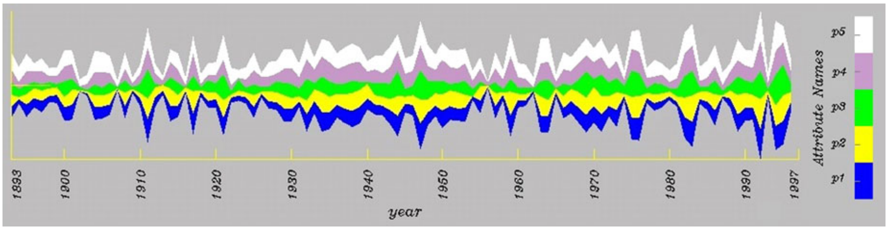


Fig. 6.9: Summer conditions (*p1–p5*) visualized as a centered layer area graph.
Source: Image courtesy of Thomas Nocke.

As we will see next, we can confirm our previous findings and gain further insight with the help of PCA and a simple bar graph (↪ p. 154). But instead of visualizing all five parameters, our visual analysis will be based on just a single principal component. So what we do is to apply PCA to the five variables extracted from the raw data. The dimension of time is excluded from the PCA. The computed PCA results are then fed to the visualization. In order to restore the temporal context, the bar graph in Figure 6.10 shows time along the horizontal axis, and the first principal component (PC1), to which all variables contribute because of their strong correlation, at the vertical axis. For each year, a bar is constructed that connects the baseline with the year's PC1 coordinate (i.e., the year's score in principal component space). This effectively means upward bars encode a positive deviation from the major trend, that is, they stand for warmer summers, where long bars indicate summers with extreme conditions. In contrast, downward bars represent colder-than-normal summers. As an additional visual cue, frequencies of score values are mapped onto color to further distinguish typical (saturated green) and outlier (bright yellowish green) years. This visual representation allows us to discern the following interesting facts:

- The first third of the time axis is dominated by moderately warm summers mixed with the coldest summers.
- The hot summers in the 1910s and 1920s are immediately followed by cold summers.
- There were relatively nice summer seasons between 1930 and 1950.
- In general, outlier summers, positive and negative ones, accumulate at the end of the time axis.

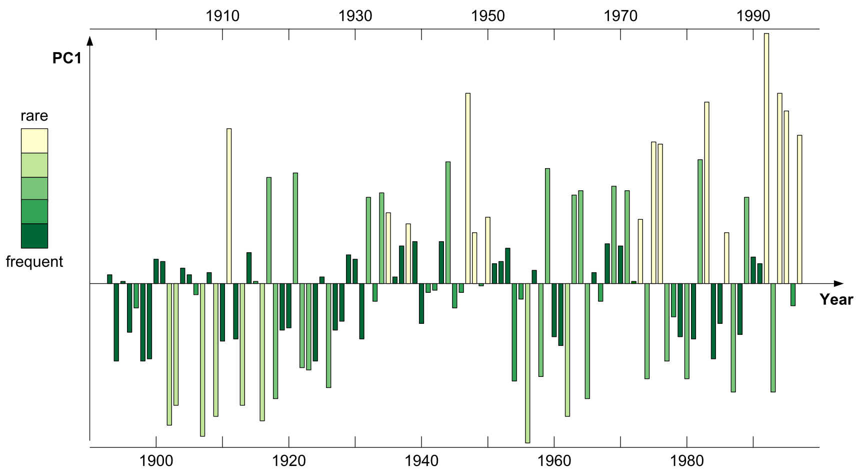


Fig. 6.10: The bar graph encodes years along the horizontal axis and the scores of the first principal component (PC1) along the vertical axis. Color indicates the frequency of score values.

Although the visualization in Figure 6.10 shows only the first principal component, rather than the five data variables, it depicts corresponding trends very well. Nonetheless, one should recall that our data represent a special case where all five variables are strongly correlated. This correlation is the reason why PC1 separates warm and cold summers so well. When analyzing arbitrary time-oriented datasets, further principal components might be necessary to capture major structural relationships. The following example will illustrate how users can be assisted in making informed decisions about which principal component's scores to display.

Determining significant principal components We now deal with a census dataset with multiple variables, including population, gross domestic product, literacy, and life expectancy. As before, the independent dimensions (i.e., time and space) are excluded to maintain the data's frame of reference, leaving ten variables to be processed analytically by the PCA. Accordingly, the analysis yields ten principal components, which correspond to the major trends in the data. The principal components' significance-weighted loadings indicate how individual variables participate in these trends.

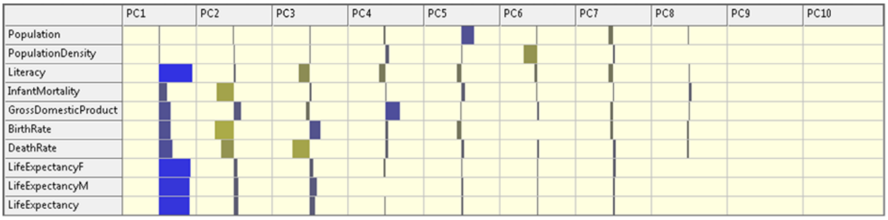


Fig. 6.11: The bars in the table cells visualize the loadings of principal components weighted by their significance. This clearly echoes the ranking of the principal components.

The significance-weighted loadings of our example are depicted in Figure 6.11, where longer bars stand for stronger participation, and blue and yellow color are used for positive and negative values, respectively. By definition, the principal components are ranked according to their significance from left to right. The figure indicates that the data's major trends (PC1-PC4) are largely influenced by the eight variables from literacy to life expectancy. But we can also see that if we consider only these first four principal components, we certainly lose the relation to the two variables population and population density, which do not contribute to the top four trends. Therefore, at least the principal components up to PC5, which is proportional to population, and PC6, which is indirectly proportional to population density, should be retained. In turn, if we are interested in the main trends only, we can safely omit the remaining principal components (PC7-PC10).

If we are interested in outlier trends as well, we should be less generous with dropping principal components. This can be illustrated by a visualization of the plain (i.e., unweighted) loadings of the principal components as shown in Figure 6.12. The figure clearly reveals contradictory contributions of the variables to the lower-ranked trends. In particular, we can see a contradiction between life expectancy of females and males in the ninth principal component (PC9).

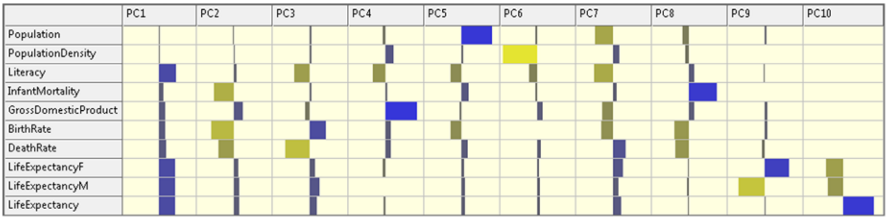


Fig. 6.12: The bars in the table cells visualize the unweighted loadings of principal components, that is, they indicate how much the individual variables contribute to any particular principal component.

The visualization of the loadings helped us in identifying the top-ranked principal components and those that might bear potentially interesting outlier information. The knowledge that we derived about the principal components can also be interpreted in terms of the variables of the original data space. A number of findings can be gained, including:

- All the positive loadings in the main trend (PC1) indicate a direct proportional relationship for the literacy, infant mortality, gross domestic product, birth rate, death rate, and life expectancy.
- The second trend (PC2) is constituted by the gross domestic product, life expectancy as well as infant mortality, death rate, and birth rate, where the latter three variables are indirectly proportional to this trend.
- The major trends in the data (PC1-PC3) are largely independent of population and population density.
- An outlier trend is present in PC9, where the contradictory loadings of life expectancy of females and males might hint at an interesting aspect.

In summary we have seen in this section that PCA is a useful tool for crystallizing major structural relationships in the data and for identifying possible candidates for data reduction.

6.5 Summary

The information seeking mantra proposed by Shneiderman (1996) should guide the users when exploring the data visually:

> Overview first,
> zoom and filter,
> then details-on-demand.
>
> Shneiderman (1996, p. 2)

However, with massive, heterogeneous, dynamic, and ambiguous datasets at hand, it is difficult to create overview visualizations without losing interesting patterns. Therefore, Keim et al. (2006) revised the information seeking mantra, in order to indicate that it is not sufficient to just retrieve and display the data using a visual metaphor:

> Analyze First -
> Show the Important -
> Zoom, Filter and Analyse Further -
> Details on Demand.
>
> Keim et al. (2006, p. 6)

In fact, it is necessary to analyze the data according to aspects of interest, to show the most relevant features of the data, and at the same time to provide interaction methods that allow the user to get details of the data on demand (see Keim et al., 2010).

In this chapter, we provided a brief overview of how analytical methods can support the visual analysis of time-oriented data. We gave a list of typical temporal analysis tasks and illustrated the utility of analysis methods with the three examples: clustering, temporal data abstraction, and principal component analysis. All of these examples perform a particular kind of data abstraction. Admittedly, our examples are simple, but still we believe that they demonstrate the benefits of analytical methods quite well.

In fact, when confronted with really huge datasets, a single analytical method alone will most certainly not suffice. Instead, a number of analytical methods must play in concert to cope with the size and complexity of time-oriented data. Moreover, analytical methods are not solely a preprocessing step to support the visualization of data. The full potential of analytical methods unfolds only if they are considered at all stages of interactive exploration and visual analysis processes in an integrated fashion depending on the data, users, and tasks.

However, as we will see in the next chapter, more in-depth research and development is necessary to arrive at an intertwined integration of visual, interactive and analytical methods for the bigger goal of gaining insight into large and complex time-oriented data.

References

Antunes, C. M. and Oliveira, A. L. (2001). Temporal Data Mining: An Overview. Workshop on Temporal Data Mining at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 105–112, New York, NY, USA. ACM Press.

Brockwell, P. J. and Davis, R. A. (2009). *Time Series: Theory and Methods*. Springer, New York, USA, 2nd edition.

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, Germany.

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.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*, ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, PA, USA.

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

Hsu, W., Lee, M. L., and Wang, J. (2008). *Temporal and Spatio-Temporal Data Mining*. IGI Global, Hershey, PA, USA.

Jackson, J. E. (2003). *A User's Guide to Principal Components*, Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York, NY, USA.

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

Jeong, D. H., Ziemkiewicz, C., Fisher, B., Ribarsky, W., and Chang, R. (2009). iPCA: An Interactive System for PCA-based Visual Analytics. *Computer Graphics Forum*, 28(3):767–774.

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

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

Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in Visual Data Analysis. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 9–16, Los Alamitos, CA, USA. IEEE Computer Society.

Laxman, S. and Sastry, P. (2006). A Survey of Temporal Data Mining. *Sādhanā*, 31:173–198.

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.

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. *Artificial Intelligence 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 *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM)*, pages 281–290, Berlin, Germany. Springer.

Mitsa, T. (2010). *Temporal Data Mining*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Boca Raton, FL, USA.

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

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.

Stacey, M. and McGregor, C. (2007). Temporal Abstraction in Intelligent Clinical Data Analysis: A Survey. *Artificial Intelligence in Medicine*, 39(1):1–24.

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.

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 4–9, Los Alamitos, CA, USA. IEEE Computer Society.

Ware, C. (2008). *Visual Thinking for Design*. Morgan Kaufmann, Burlington, MA, USA.

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

Xu, R. and Wunsch II, D. C. (2009). *Clustering*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Chapter 7
Survey of Visualization Techniques

> Today we live in an information-based technological world. The problem is that this is an invisible technology. Knowledge and information are invisible. They have no natural form. It is up to the conveyor of the information and knowledge to provide shape, substance, and organization [...]
>
> Norman (1993, p. 104)

A major part of this book is dedicated to a survey of existing visualization techniques for time and time-oriented data. The complexity of the visualization problem, which results from the multitude of aspects having an impact on the visual representation, already suggests that there must be a variety of techniques – and indeed there are numerous ones. The following survey lists many techniques, some of them very specific to a particular application domain, others more general with potential applicability in other fields than the one described. We are aware that our survey cannot be exhaustive. This is due to the fact that visualization of time-oriented data is a hot research area which is constantly yielding new techniques. Moreover, we have seen that visualization solutions might be highly application-dependent (what and why aspects, see Chapter 4), and hence, it is virtually impossible to dig out every tiny variation of existing visualization approaches that might be hidden in the vast body of scientific literature across application domains. Therefore, we took care to include a wide spectrum of key techniques, both classic ones with proven usefulness and contemporary ones with potential impact.

The survey lists the techniques on a per-page basis. This allows for easy access when a quick reference to a particular technique is sought by the reader. Each page briefly describes the background, explains the main idea and concepts, and indicates the application of a particular technique. The description is accompanied with a reference to the original publication or a list of references in the case that multiple publications propose or make use of the same approach. As this is a visualization book, a figure demonstrates the technique in use or the conceptual construction of the visual representation. Additionally, we provide a side-bar that categorizes the technique. To keep the categorization at a manageable level, we do not use the full-

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_7,

scale classification introduced in the previous chapters, but instead focus on three key criteria: data, time, and vis(ualization). For each key criterion, we introduce further sub-criteria and corresponding characteristics. The side-bar information follows this pattern:

- **data**
  - *frame of reference* – abstract vs. spatial
  - *variables* – univariate vs. multivariate
- **time**
  - *arrangement* – linear vs. cyclic
  - *time primitives* – instant vs. interval
- **vis**
  - *mapping* – static vs. dynamic
  - *dimensionality* – 2D vs. 3D

Where possible, a distinct classification will be given. However, this is not always possible, particularly for more general and flexible visualization approaches. In such cases, we will indicate that multiple characteristics hold per category.

7.1 Techniques

It was not easy to decide on a good order for the techniques in the survey. If we sorted by the name of a technique or by the year of first publication, we would lose the semantic relationships of techniques, and similar techniques would be scattered across the survey just because they have different names. Therefore, we use the categories provided in the side-bar to structure the survey. At the top-most levels, the order is determined by the data characteristics. The survey will start with techniques for abstract time-oriented data, and later we will move on to techniques for spatial data. Accordingly, techniques for univariate time-oriented data will precede those for multivariate data. At the subsequent levels, we order the techniques by their affiliation to the categories *arrangement*, *time primitives*, *mapping*, and *dimensionality*.

Table 7.1 provides an overview of all techniques that are included in our survey along with their categorization. This table might also be used to search for techniques that fulfill certain criteria. Suppose we have an abstract univariate dataset containing time instants as well as intervals and would like to find appropriate visualization techniques. In this case, we would look at the columns *abstract*, *univariate*, *instant*, and *interval* in Table 7.1 and search for lines that match these criteria. For the given example, we would find the techniques Gantt chart, perspective wall, and spiral display as primary matches. For quick access to a technique that is known by name, the index at the end of the book provides the corresponding page number.

	data				**time**				**vis**				
	frame of reference		variables		arrangement		time primitives		mapping		dimensionality		
	abstract	spatial	univariate	multivariate	linear	cyclic	instant	interval	static	dynamic	2D	3D	page
Point Plot	■		■		■		■		■		■		152
Line Plot	■		■		■		■		■		■		153
Bar Graph, Spike Graph	■		■		■		■		■		■		154
Sparklines	■		■		■		■		■		■		155
SparkClouds	■		■		■		■		■		■		156
Horizon Graph	■		■		■		■		■		■		157
TrendDisplay	■		■		■		■		■		■		158
Decision Chart	■		■		■		■		■		■		159
TimeTree	■		■		■		■		■		■		160
Arc Diagrams	■		■		■		■		■		■		161
Interactive Parallel Bar Charts	■		■		■		■		■			■	162
TimeHistogram 3D	■		■		■		■		■			■	163
Intrusion Monitoring	■		■		■		■			■	■		164
Anemone	■		■		■		■			■	■		165
Timeline	■		■		■			■	■		■		166
Gantt Chart	■		■		■		■	■	■		■		167
Perspective Wall	■		■		■		■	■	■			■	168
DateLens	■		■		■			■	■		■		169
TimeNets	■		■		■			■	■		■		170
Paint Strips	■		■		■			■	■		■		171
PlanningLines	■		■		■			■	■		■		172
Time Annotation Glyph	■		■		■			■	■		■		173
SOPO Diagram	■		■		■			■	■		■		174
Silhouette Graph, Circular Silhouette Graph	■		■		■	■	■		■		■		175
Cycle Plot	■		■		■	■	■		■		■		176
Cluster and Calendar-Based Visualization	■		■		■	■	■		■		■	■	177
Tile Maps	■		■		■	■	■		■		■		178
Multi Scale Temporal Behavior	■		■		■	■	■		■		■		179
Recursive Pattern	■		■	■	■	■	■		■		■		180
GROOVE	■		■		■	■	■		■		■		181
SolarPlot	■		■		■	■	■		■		■		182
SpiraClock	■		■			■		■		■	■		183
Enhanced Interactive Spiral	■		■			■	■		■		■		184
Spiral Graph	■		■	■		■	■		■		■		185
Spiral Display	■		■	■		■	■	■	■		■	■	186
VizTree	■		■	■	■		■		■		■		187

continued on next page

Table 7.1: Overview and categorization of visualization techniques.

	data				time				vis				
	frame of reference		variables		arrangement		time primitives		mapping		dimensionality		
	abstract	spatial	univariate	multivariate	linear	cyclic	instant	interval	static	dynamic	2D	3D	page
TimeSearcher	■			■	■		■		■		■		188
TimeSearcher 3, River Plot	■			■	■		■		■		■		189
BinX	■			■	■		■		■		■		190
LiveRAC	■			■	■		■		■		■		191
LifeLines2	■			■	■		■		■		■		192
Similan	■			■	■		■		■		■		193
CareCruiser	■			■	■		■		■		■		194
Layer Area Graph	■			■	■		■		■		■		195
Braided Graph	■			■	■		■		■		■		196
ThemeRiver	■			■	■		■		■		■		197
3D ThemeRiver	■			■	■		■		■			■	198
Stacked Graphs	■			■	■		■		■		■		199
TimeWheel	■			■	■		■		■		■		200
MultiComb	■			■	■		■		■		■		201
VIE-VISU	■			■	■		■		■		■		202
Timeline Trees	■			■	■		■		■		■		203
Pixel-Oriented Network Visualization	■			■	■	■	■		■		■		204
CiteSpace II	■			■	■		■		■		■		205
history flow	■			■	■		■		■		■		206
PeopleGarden	■			■	■	■	■		■		■		207
PostHistory	■			■	■	■	■		■		■		208
MOSAN	■			■	■		■		■		■	■	209
Data Tube Technique	■			■	■		■		■			■	210
Kiviat Tube	■			■	■		■		■			■	211
Temporal Star	■			■	■		■		■			■	212
Time-tunnel	■			■	■		■		■			■	213
Parallel Glyphs	■			■	■	■	■		■			■	214
Worm Plots	■			■	■		■		■			■	215
Software Evolution Analysis	■			■	■		■		■			■	216
InfoBUG	■			■	■		■		■	■	■		217
Gravi++	■			■	■		■		■	■	■		218
CircleView	■			■	■		■		■	■	■		219
Trendalyzer, Animated Scatter Plot	■			■	■		■			■	■		220
TimeRider	■			■	■		■			■	■		221
Process Visualization	■			■	■		■			■	■		222
Flocking Boids	■			■	■		■			■		■	223

continued on next page

Table 7.1: Overview and categorization of visualization techniques.

	data				time				vis				
	frame of reference		variables		arrangement		time primitives		mapping		dimensionality		
	abstract	spatial	univariate	multivariate	linear	cyclic	instant	interval	static	dynamic	2D	3D	page
Time Line Browser	■			■	■		■	■	■		■		224
LifeLines	■			■	■		■	■	■		■		225
PatternFinder	■			■	■		■	■	■		■		226
Continuum	■			■	■		■	■	■		■		227
EventRiver	■			■	■		■	■	■		■		228
FacetZoom	■			■	■		■	■	■		■		229
Midgaard	■			■	■		■	■	■		■		230
VisuExplore	■			■	■		■	■	■		■		231
KNAVE II	■			■	■		■	■	■		■		232
Circos	■			■	■	■	■	■	■		■		233
Kaleidomaps	■			■		■	■		■		■		234
Intrusion Detection	■			■		■	■		■			■	235
Small Multiples	■	■	■	■	■	■	■		■		■	■	236
EventViewer	■	■	■	■	■	■	■	■	■		■		237
Ring Maps	■	■	■	■	■	■	■	■	■		■	■	238
Time-Oriented Polygons on Maps		■	■		■	■	■		■		■		239
Icons on Maps		■	■	■	■	■	■	■	■		■	■	240
Value Flow Map		■	■		■		■		■		■		241
Flow Map		■	■		■		■	■	■		■		242
Time-Varying Hierarchies on Maps		■	■		■		■		■			■	243
VIS-STAMP		■		■	■		■		■		■		244
Space-Time Cube		■	■	■	■		■	■	■			■	245
Spatio-Temporal Event Visualization		■	■	■	■		■		■			■	246
Space-Time Path		■	■	■	■		■	■	■			■	247
GeoTime		■		■	■		■		■	■		■	248
Pencil Icons		■		■	■		■	■	■			■	249
Data Vases		■		■	■		■	■	■		■	■	250
Wakame		■		■	■		■		■			■	251
Helix Icons		■		■		■	■	■	■			■	252
count	87	17	47	65	94	25	93	30	94	11	80	28	

Table 7.1: Overview and categorization of visualization techniques.

Let us now start the survey with the simple, but most widely used techniques for visualizing time-oriented data – the point plot and the line plot.

Point Plot

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

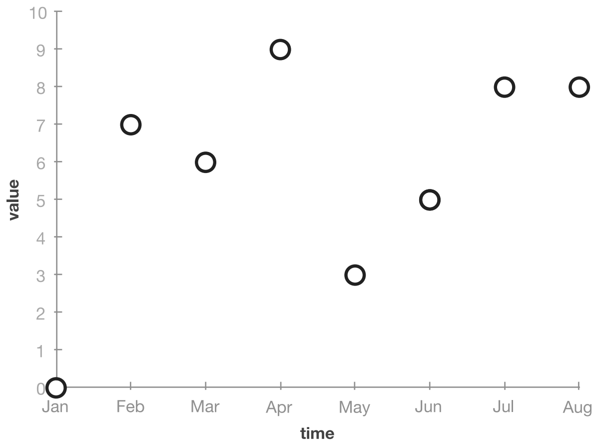


Fig. 7.1: Data are displayed as points in a Cartesian coordinate system where time and data are mapped to the horizontal axis and the vertical axis, respectively.
Source: Authors.

One of the most straightforward ways of depicting time-series data is using a Cartesian coordinate system with time on the horizontal axis and the corresponding value on the vertical axis. A point is plotted for every measured time-value pair. This kind of representation is called point plot, point graph, or scatter plot, respectively. Harris (1999) describes it as a 2-dimensional representation where quantitative data aspects are visualized by distance from the main axis. Many extensions of this basic form such as 3D techniques (layer graph) or techniques that use different symbols instead of points are known. This technique is particularly suited for emphasizing individual values. Moreover, depicting data using position along a common scale can be perceived most precisely by the human perceptual system.

References

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

Line Plot

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

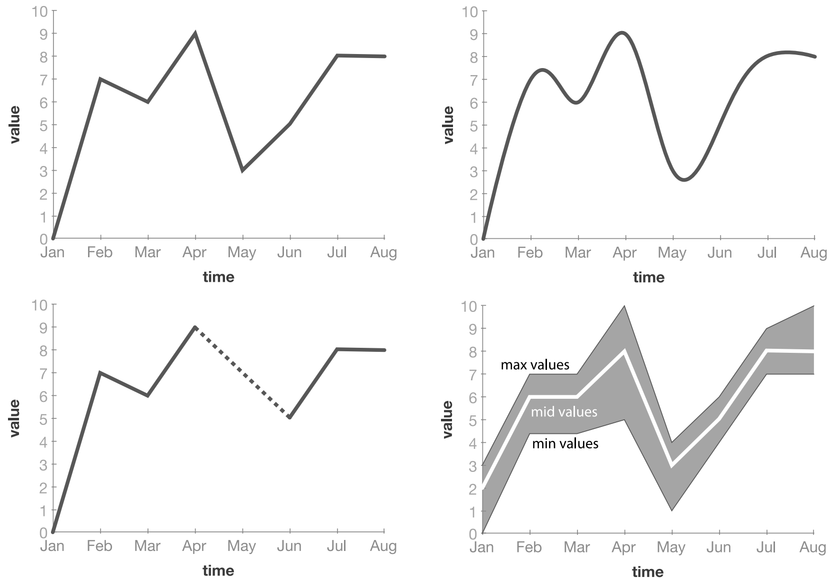


Fig. 7.2: Successive data points are connected with lines to visualize the overall change over time. (top-left: straight lines; top-right: Bézier curves; bottom-left: missing data; bottom-right: band graph).
Source: Authors.

The most common form of representing time-series are line plots. They extend point plots (↪ p. 152) by linking the data points with lines which emphasizes their temporal relation. Consequently, line plots focus on the overall shape of data over time. This is in contrast to point plots where individual data points are emphasized. As illustrated in Figure 7.2, different styles of connections between the data points such as straight lines, step lines (instant value changes), or Bezier curves can be used depending on the phenomenon under consideration. However, what has to be kept in mind is that one can not be sure in all cases about the data values in the time interval between two data points and that any kind of connection between data points reflects an approximation only. A further point of caution is missing data. Simply connecting subsequent data points might lead to false conclusions regarding the data. Therefore, this should be made visible to the viewer, for instance by using dotted lines (see Figure 7.2, bottom-left). There are many extensions or subtypes like fever graphs, band graphs (see Figure 7.2, bottom-right), layer line graphs, surface graphs, index graphs, or control graphs (see Harris, 1999).

References

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

Bar Graph, Spike Graph

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

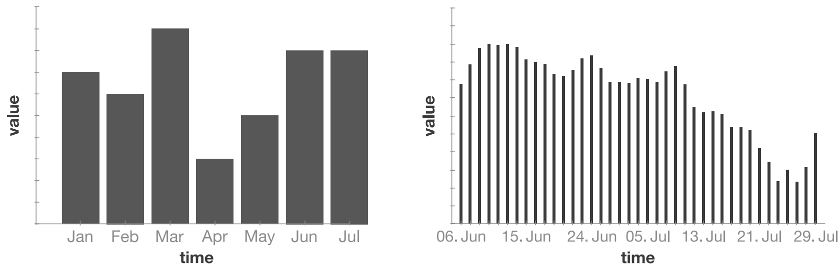


Fig. 7.3: Bar length is used to depict data values. Right: if bars are reduced to spikes the graph is also called a spike graph.
Source: Adapted from Harris (1999).

Bar graphs (see Harris, 1999) are a well known and widely used type of representation where bars are used to depict data values (see Figure 7.3, left). This makes comparisons easier than with point plots. As bar length is used to depict data values, only variables with a ratio scale (having a natural zero) can be represented. Consequently, the value scale also has to start with zero to allow for a fair visual comparison. In contrast to line plots, bar graphs emphasize individual values as do point plots. A variant of bar graphs often used for graphing larger time-series (e.g., stock market data such as price or volume) are spike graphs. As illustrated in Figure 7.3 (right), the vertical bars are reduced so as to appear as spikes, where spike height is again used to encode data values. This way, a good visual balance is achieved between focusing on individual values and showing overall development of a larger number of data values.

References

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

Sparklines

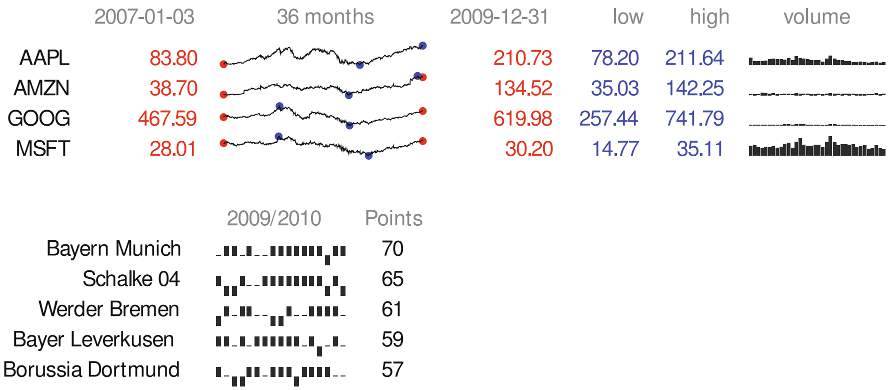


Fig. 7.4: Simple, word-like graphics intended to be integrated into text visualize stock market data (top). Bottom: Soccer season results using ticks (up=win, down=loss, base=draw).
Source: Generated with the sparklines package for LaTeX.

Tufte (2006) describes sparklines as simple, word-like graphics intended to be integrated into text. This adds richer information about the development of a variable over time that words themselves could hardly convey. The visualization method focuses mainly on giving an overview of the development of values for time-oriented data rather than on specific values or dates due to their small size and the omission of axes and labels. Sparklines can be integrated seamlessly into paragraphs of text, can be laid out as tables, or can be used for dashboards. They are increasingly adopted to present information on web pages (such as usage statistics) in newspapers (e.g., for sports statistics), or in finance (e.g., for stock market data). Usually, miniaturized versions of line plots (↪ p. 153) and bar graphs (↪ p. 154) are employed to represent data. For the special case of binary or three-valued data, special bar graphs can be applied that use ticks extending up and down a horizontal baseline . One use for this kind of data are wins and losses of sports teams where the history of a whole season can be presented using very little space. For line plots, the first and last value can be emphasized by colored dots (•) and printing the values themselves textually to the left and right of the sparkline. Moreover, the minimum and maximum values might also be marked by colored dots (•). Besides this, colored bands in the background of the plot can be used to show normal value ranges as for example here 4.8 8.3.

References

Tufte, E. R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, CT.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

SparkClouds

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

Fig. 7.5: Display of the 25 most important keywords in a series of twelve measurements. The bigger the font size is, the more important is a keyword. Keywords that are not among the current top 25 important keywords but have been among them at an earlier point in time are attenuated by using dimmed color and smaller font size.
Source: Lee et al. (2010). © 2010, IEEE. Used with permission.

Tag clouds visualize a set of keywords weighted by their importance. To this end, a layout of the keywords is computed. By varying font size, color, or other visual variables important keywords are emphasized over less-important keywords. Classic tag clouds, however, are incapable of representing the evolution of keywords. Lee et al. (2010) integrate sparklines (↪ p. 155) into tag clouds in order to visualize temporal trends in the development of keywords. The idea is to visually combine a keyword (or tag) and its temporal evolution. The keyword's importance is encoded with the font size used to render the text, where the size can correspond either to the overall importance of the keyword for the entire time-series or to the importance at a particular point in time. Attached to the keyword is a sparkline that represents the keyword's trend. A color gradient is shown in the background of each keyword-sparkline pair to make this design perceivable as a visual unit. Lee et al. (2010) conducted user studies with sparkclouds and could confirm that sparkclouds are useful and have advantages over alternative standard methods for visualizing text and temporal information.

References

Lee, B., Riche, N., Karlson, A., and Carpendale, S. (2010). SparkClouds: Visualizing Trends in Tag Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189.

Horizon Graph

data
frame of reference: **abstract**
variables: **univariate**

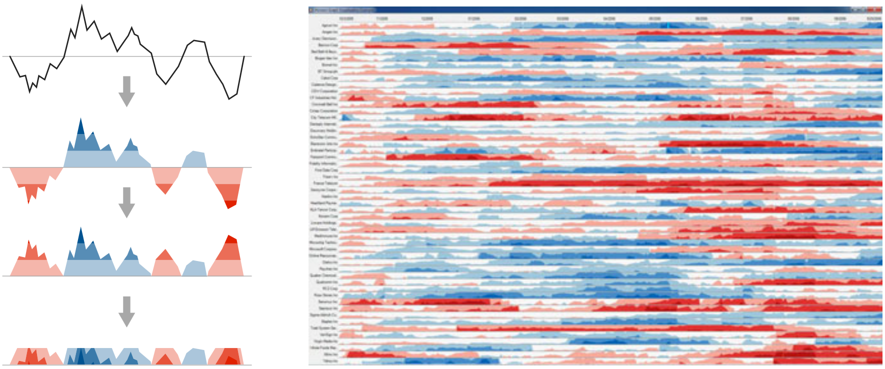


Fig. 7.6: The construction of a horizon graph from a line chart is illustrated on the left. Because horizon graphs require only little screen space they are very useful for comparing multiple time-dependent variables as shown to the right for stock market data.
Source: Left: Adapted from Reijner (2008). Right: Image courtesy of Hannes Reijner.

time
arrangement: **linear**
time primitives: **instant**

Reijner (2008) describes horizon graphs as a visualization technique for comparing a large number of time-dependent variables. Horizon graphs are based on the two-tone pseudo coloring technique by Saito et al. (2005). The left part of Figure 7.6 demonstrates the construction of horizon graphs (from top to bottom). Starting from a common line plot, the value range is divided into equally sized bands that are discriminated by increasing color intensity towards the maximum and minimum values while using different hues for positive and negative values. Then, negative values are mirrored horizontally at the zero line. Finally, the bands are layered on top of each other. This way, less vertical space is used, which means data density is increased while the resolution is preserved. A study by Heer et al. (2009) has shown that mirroring does not have negative effects and that layered bands are more effective than the standard line plot (↪ p. 153) for charts of small size, as for example in the right part of Figure 7.6.

vis
mapping: **static**
dimensionality: **2D**

References

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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1303–1312, New York, NY, USA. ACM Press.

Reijner, H. (2008). The Development of the Horizon Graph. In *Electronic Proceedings of the VisWeek Workshop From Theory to Practice: Design, Vision and Visualization*.

Saito, T., Miyamura, H., Yamamoto, M., Saito, H., Hoshiya, Y., and Kaseda, T. (2005). Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 173–180, Los Alamitos, CA, USA. IEEE Computer Society.

TrendDisplay

data

frame of reference: **abstract**
variables: **univariate**

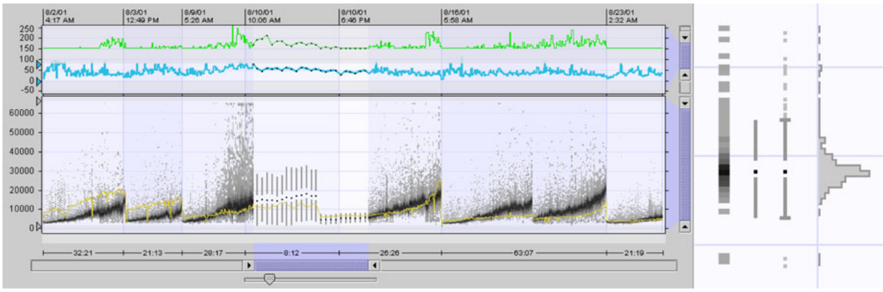


Fig. 7.7: The main panel shows the raw data (drug discovery data) and the top panel depicts derived statistical values. Depending on the available screen space, four different levels of visual abstraction are used: density distributions, thin box plots, box plots plus outliers, and bar histograms (as illustrated to the right).
Source: Brodbeck and Girardin (2003), © 2003 IEEE. Used with permission.

time

arrangement: **linear**
time primitives: **instant**

The TrendDisplay technique by Brodbeck and Girardin (2003) allows the analysis of trends in larger time-series. The technique is used for the drug discovery process and in quality control. Basically, the TrendDisplay window is composed of two panels. The main panel on the bottom shows the measured (raw) data and the top panel depicts derived statistical values (see Figure 7.7, left). Four different levels of detail are used in order to cope with large numbers of time points: density distributions, thin box plots, box plots plus outliers, and bar histograms (from low to high level of detail) (see Figure 7.7, right). In the temporal dimension, bifocal focus+context functionality is used for enlarging areas of interest without losing context information about neighboring data. The different levels of detail are chosen automatically depending on the available screen space. Moreover, brushing & linking as well as smooth transitions complete the highly interactive interface.

vis

mapping: **static**
dimensionality: **2D**

References

Brodbeck, D. and Girardin, L. (2003). Interactive Poster: Trend Analysis in Large Timeseries of High-Throughput Screening Data Using a Distortion-Oriented Lens with Semantic Zooming. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 74–75, Los Alamitos, CA, USA. IEEE Computer Society.

Decision Chart

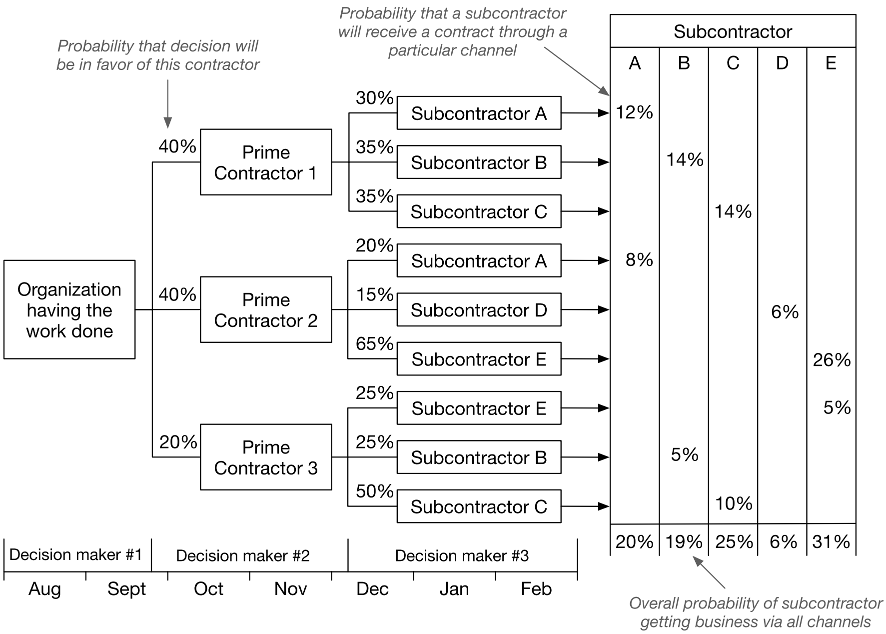


Fig. 7.8: Future decisions and corresponding alternative outcomes are depicted over time along with their probabilities.
Source: Adapted from Harris (1999).

Harris (1999) describes decision charts as a graphical representation for depicting future decisions and potential alternative outcomes along with their probabilities over time. It is one of very few techniques for time-oriented data that use the *branching time* model (see Section 3.1.1). Decision charts use a horizontal time axis along which information elements (decisions and probabilities) are aligned. Multiple decisions for a particular time interval are stacked on top of each other, indicating that they are possible alternatives for that interval. However, the temporal context itself is not of prime interest and is just indicated by a simple time scale on the bottom of the chart. The main advantage of the decision chart is that it allows planners to investigate possible outcomes and implications before decisions are made.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

References

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

TimeTree

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

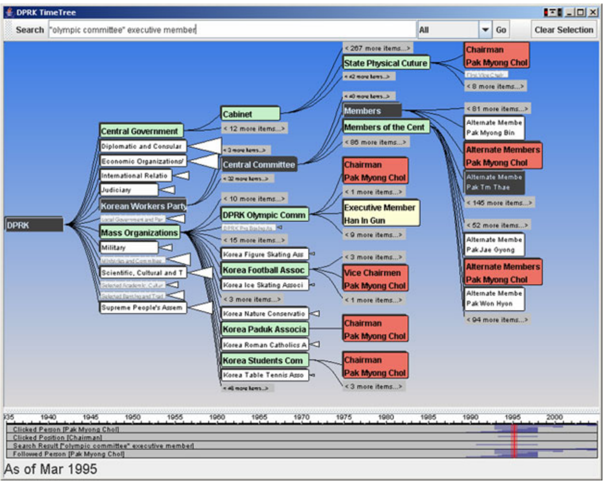


Fig. 7.9: The figure shows changes in the organizational structure of officials in the DPR Korea with focused and important nodes highlighted. The time slider (currently set to March 1995) shows selected personal events and serves for interactive navigation.
Source: Card et al. (2006), © 2006 IEEE. Used with permission.

TimeTree by Card et al. (2006) is a visualization technique to enable the exploration of changing hierarchical organizational structures and of individuals within such structures. The visualization consists of three parts: a time slider, a tree view, and a search interface (see bottom, center, and top of Figure 7.9, respectively). The time slider's main purpose is to allow users to navigate to any point in time. Additionally, it shows information strips with events for a selected set of individuals, and thus, provides insight into where in time interesting things have happened. The tree view shows the snapshot of the organizational structure corresponding to the selected time point. The tree visualization uses a degree-of-interest (DOI) approach to highlight important information. To this end, a specific color scheme is used to indicate certain data characteristics, as for instance, important nodes, nodes matching with search queries, or recently clicked nodes. Low-interest nodes, with regard to the user's current focus and search query, are ghosted, and entire subtrees may be represented as triangular abstractions in order to de-clutter the display and maintain the readability of important nodes. The search interface supports a textual search for individuals in the represented organization.

References

Card, S. K., Suh, B., Pendleton, B. A., Heer, J., and Bodnar, J. W. (2006). Time Tree: Exploring Time Changing Hierarchies. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10, Los Alamitos, CA, USA. IEEE Computer Society.

Arc Diagrams

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

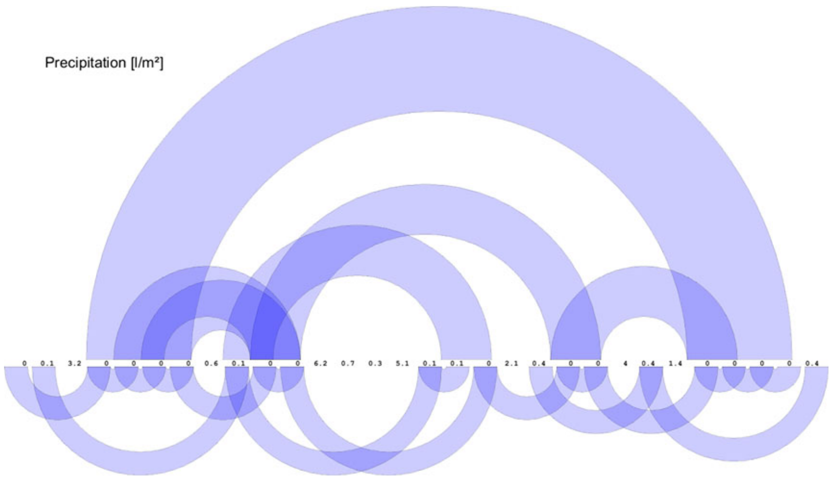


Fig. 7.10: A sequence of data values is shown along the horizontal axis. Matching subsequences are connected by arcs, where arc thickness and height encode subsequence size and occurrence distance, respectively.
Source: Image courtesy of Michael Zornow.

Patterns in sequences of data values can be visualized using arc diagrams. They were introduced as an interactive visualization technique by Wattenberg (2002). Given a sequence of values, the goal is to extract significant subsequences that occur multiple times in the original sequence. The visualization displays the sequence of data values in textual form along the horizontal (time) axis. Occurrences of significant subsequences are visually connected by spanning arcs. The arcs' thickness represents the size of the subsequence, that is, the number of data values in the subsequence. The height of an arc indicates the distance between two successive occurrences of the subsequence. To express data-specific aspects, one can separately use the space above or below the data sequence. This also helps to reduce overlap of arcs. Additionally, transparency is used to allow users to see through overlapping arcs. The visualization can be controlled interactively via several parameters (e.g., minimum size of subsequences or tolerance threshold for fuzzy pattern extraction) to keep the number of arcs at an interpretable level.

References

Wattenberg, M. (2002). Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 110–116, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

Interactive Parallel Bar Charts

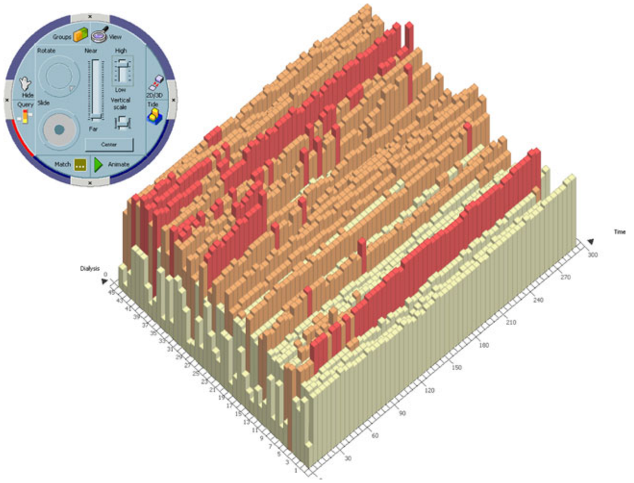


Fig. 7.11: Visualization of clinical time-dependent data where one axis represents different hemodialysis sessions and the other axis represents the series of time steps. One time-dependent variable (e.g., blood pressure) is encoded to the height of the bars.
Source: Chittaro et al. (2003), © 2003 Elsevier. Used with permission.

Chittaro et al. (2003) present a technique for visualizing time-dependent hemodialysis data. To keep the visualization of multiple hemodialysis sessions simple and easy to use for physicians, the design is based on common 3D bar charts, where the height of bars encodes individual data values. Multiple bar charts (one per hemodialysis session) are arranged on a regular grid in a parallel fashion. This visual display is easy to interpret despite the 3D projection. Visual exploration and analysis are facilitated through a wealth of interaction tools. Dynamic filtering combined with a color coding mechanism supports visual classification. To manage occlusions, interactive features are provided, such as flattening individual bars or groups of bars, or leaving only colored squares in the grid. The water level interaction is particularly helpful for comparison tasks: virtual water engulfs all bars with a height below a user-defined threshold. This eases the assessment of similarities and differences with respect to dialysis sessions and time. Additional visual cues support physicians in detecting anomalies or special events in the data and enable them to take necessary actions quickly. For multivariate analysis, an integration with parallel coordinate plots is supported.

References

Chittaro, L., Combi, C., and Trapasso, G. (2003). Data Mining on Temporal Data: A Visual Approach and its Clinical Application to Hemodialysis. *Journal of Visual Languages and Computing*, 14(6):591–620.

TimeHistogram 3D

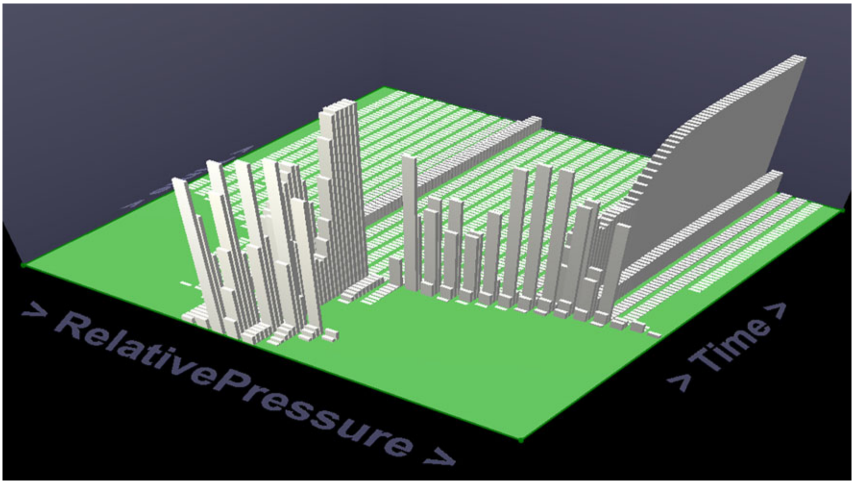


Fig. 7.12: Time is encoded along the x-axis, while a time-dependent variable (RelativePressure) is represented along the y-axis. For each time-value pair in the resulting grid in the x-y plane, the height of a cuboid represents the frequency of data items per grid cell.
Source: Kosara et al. (2004), © 2004 IEEE. Used with permission.

Kosara et al. (2004) proposed an interactive extension of well-known histograms called TimeHistogram 3D. The TimeHistogram is especially designed for time-oriented data. It has been developed to give an overview of complex data in the application context of computational fluid dynamics (CFD). A design goal of this technique was to show temporal information in static images while maintaining the easy readability of standard histograms. In the TimeHistograms in Figure 7.12, the x-axis encodes time and the y-axis encodes a time-dependent variable (RelativePressure), effectively creating a grid in the x-y plane, where each grid cell corresponds to a unique time-value pair. In order to visualize the number of data items (i.e., their frequency) per time-value pair, cuboids are shown for each cell, where the height of a cuboid encodes the frequency. This way, the user can see where in time and in which value range data items accumulate. Several interactive features such as brushing, scaling, and a 2D context display, which is shown in the background of the histogram, are part of this technique.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Kosara, R., Bendix, F., and Hauser, H. (2004). TimeHistograms for Large, Time-Dependent Data. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym)*, pages 45–54, Los Alamitos, CA, USA. IEEE Computer Society.

Intrusion Monitoring

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

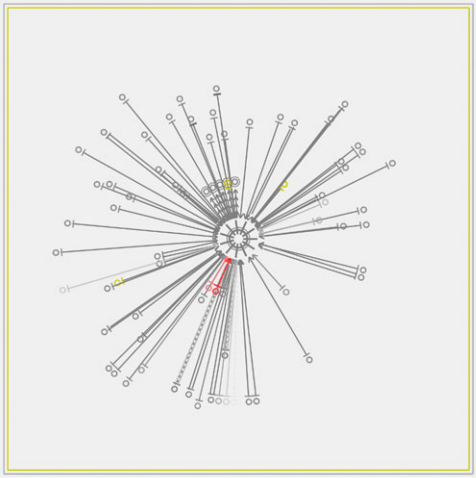


Fig. 7.13: The glyph in the center represents the monitored system. Connections to remote hosts are depicted as radially arranged lines. Critical and suspicious connections are visualized by red and yellow color, respectively.
Source: Image courtesy of Robert F. Erbacher.

Erbacher et al. (2002) describe a system that visualizes time-stamped network-related log messages that are dynamically generated by a monitored system. These messages correspond to events in a linear continuous time domain. The visualization shows the monitored server system as a central glyph encoding the number of users and the server's load (see Figure 7.13). Events are shown as radially arranged lines at whose end the remote host is shown as a smaller glyph. Regular network activities are drawn with a shade of gray. Unexpected or suspicious activities result in a change of color: Hosts that try to open privileged connections are colored in red, hosts that fail to respond turn yellow, lines representing timed-out connections or connections that failed the authentication procedure are shown in red, and connections that have been identified as intrusions are represented with even brighter red. To preserve a history of connections that have been terminated, the corresponding lines are faded out gradually. This kind of visual representation helps administrators in observing network communication. Presenting colored (red or yellow) lines among gray lines attracts the attention of administrators to suspicious activity and actions can be taken quickly to counter network attacks from remote hosts.

References

Erbacher, R. F., Walker, K. L., and Frincke, D. A. (2002). Intrusion and Misuse Detection in Large-Scale Systems. *IEEE Computer Graphics and Applications*, 22(1):38–48.

Anemone

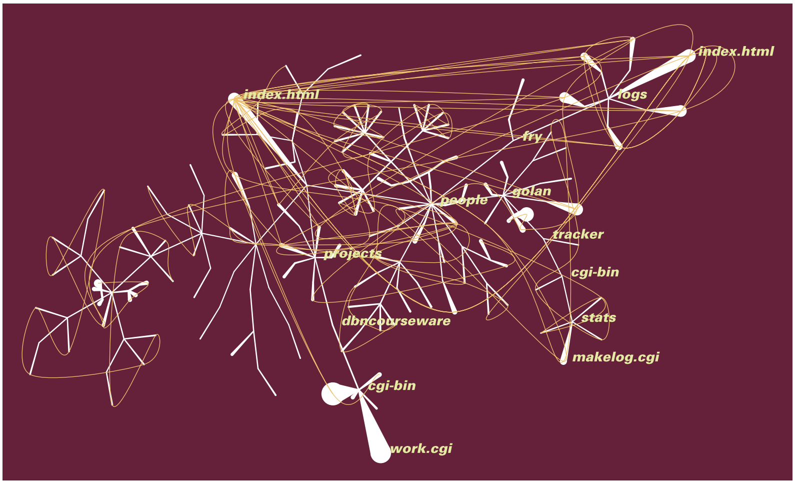


Fig. 7.14: Snapshot of Anemone showing traffic patterns of people visiting the web site of the Aesthetics & Computation Group at the M.I.T. Media Lab. The structure of the web site is shown as a node-link representation. Nodes vary in size, depending on how frequently a page is visited by users. Rarely-visited parts fade out slowly.
Source: Image courtesy of Ben Fry, MIT Media Laboratory, Aesthetics + Computation Group, © 1999-2005.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

Anemone by Fry (2000) is a technique related to the visualization of structured information. It is a dynamic, organic representation designed to reveal not only the static structure of a website, which is based on its organization into folders and files, but also to reveal dynamic usage patterns. To this end, a classic node-link representation is visually enriched with dynamically updated usage statistics to form a living representation that truly reflects the restless nature of a website. The static structure is shown as nodes that are connected via straight branches. At the tip of a branch resides the actual web page. Additional labels can be used to identify nodes by their corresponding page's name. Nodes dynamically change size depending on how often they are visited by users. When a user follows a link from one page to another, a thin curved line is drawn connecting both pages. If parts of a site have not been visited for a long time, they shrink in size and slowly fade out. To allow users to concentrate on particular items of interest, it is possible to select nodes and to lock them to a dedicated position. This is quite useful, because the dynamic character of the technique implies that visual representation constantly changes its appearance.

References

Fry, B. (2000). Organic Information Design, Master's thesis, Massachusetts Institute of Technology.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **interval**

vis

mapping: **static**
dimensionality: **2D**

Timeline

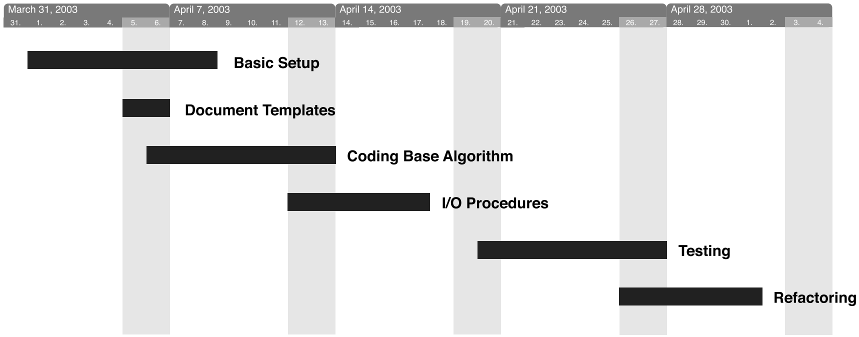


Fig. 7.15: Bars are arranged relative to a time axis to visualize both the location and duration of intervals. One can also see how intervals are related to each other.
Source: Generated by the authors.

If the time primitives of interest are not points but intervals, the visualization has to communicate not only where in time a primitive is located, but also how long it is. A simple and intuitive way of depicting incidents with a duration is by marking them visually along a time axis. This form of visualization is called timeline. Most commonly, a visual element such as a line or a bar represents an interval's starting point and duration (and consequently its end). Figure 7.15 shows an example with bars. If multiple intervals share a common time axis, as in this example, it is even possible to discern how the various intervals are related to each other (for possible relations of intervals see Section 3.1.2). Timelines are a very powerful visualization technique that, according to Tufte (1983), had been used long before computers even appeared (see also Chapter 2). Many different variants of timelines exist in diverse visualization tools. Additional interaction techniques often allow users not only to view time intervals, but also to create and edit them. Prominent examples are LifeLines (↪ p. 225) and Gantt charts (↪ p. 167).

References

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

Gantt Chart

data

frame of reference: **abstract**
variables: **univariate**

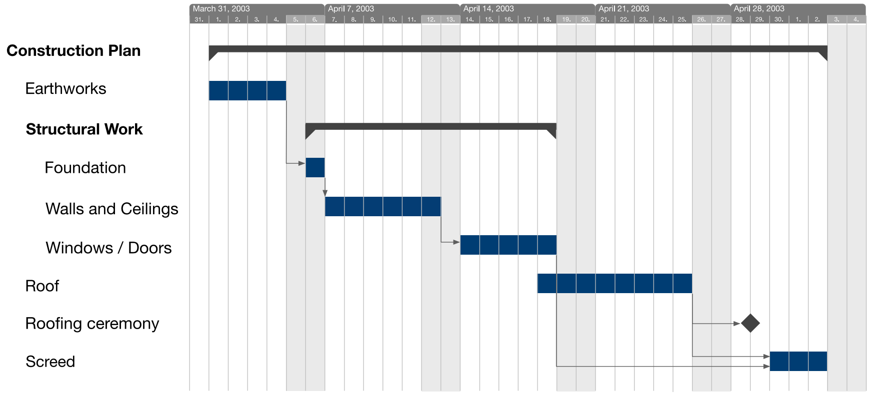


Fig. 7.16: The Gantt chart shows a project plan for construction works. To the left, the chart provides an indented list of tasks. In the main panel, timelines show position and duration of tasks in time, where black and blue bars stand for groups of tasks and individual tasks, respectively. Additionally, diamonds indicate milestones.
Source: Authors.

time

arrangement: **linear**
time primitives: **instant, interval**

Planning activities, people, and resources is a task that is particularly important in the field of project management. One of the common visualization techniques used for such tasks are Gantt charts. This kind of representation was originally invented by Gantt (1913) who studied the order of steps in work processes (see also Chapter 2, p. 25). Mainly work tasks with their temporal location and duration as well as milestones are depicted. The tasks are displayed as a textual list in the left part of the diagram and might be augmented by additional textual information such as resources, for example. Related tasks can be grouped to form a hierarchy, which is reflected by indentation in the task list. For displaying the position and duration of tasks in time, timelines (↪ p. 166) are drawn at the corresponding vertical position of the task list. This leads to an easily comprehensible representation of information from the past, present, and future. Hierarchically grouped tasks can be expanded and collapsed interactively. Summary lines are used to maintain an overview of larger plans. Sequence relationships are represented by arrows that connect tasks (e.g., an arrow from the end of task A to the beginning of task B shows that task B may start only after task A is finished). Milestones indicating important time points for synchronization within a project plan are visually represented by diamonds. The fact that tasks are mostly ordered chronologically, typically leads to a diagonal layout from the upper left to the lower right corner of the display.

vis

mapping: **static**
dimensionality: **2D**

References

Gantt, H. L. (1913). *Work, Wages, and Profits*. Engineering Magazine Co., New York, NY, USA.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

Perspective Wall

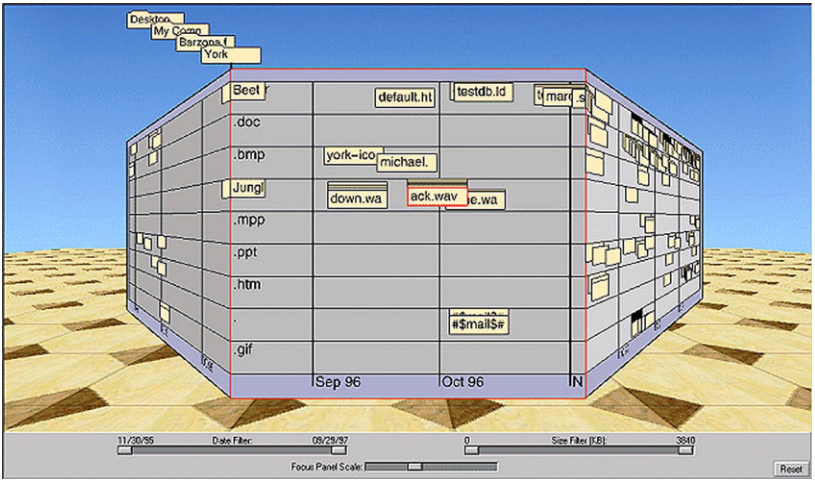


Fig. 7.17: A perspective wall representing time-related information of a file system for a period of several months. The focus (currently set to September/October 1996) shows detailed text labels for files, whereas the context regions only indicate files as yellow boxes.
Source: © Inxight Federal Systems.

Time-oriented data that are linked to a longer time axis (i.e., wide span in time or many time primitives) are usually difficult to represent visually because the image becomes very wide and exhibits an aspect ratio that is not suited for common displays. The perspective wall by Mackinlay et al. (1991) is a technique that addresses this problem by means of a focus+context approach. The key idea is to map time-oriented data to a 3D wall. For a user-selected focus, full detail is provided in the center of the display. Two context representations show the data in the past (to the left) and in the future (to the right) with regard to the current focus. The context is bent perspectively to reduce the display space occupied by these regions, effectively allowing for better space utilization in the focus (see center of Figure 7.17). Interaction methods are provided to enable users to navigate in time in order to bring different time spans into focus. The actual data representation on the wall may vary across applications; the only requirement is that time is mapped linearly from left to right. For example, one can use bars as in the figure or more advanced visual representations such as the ThemeRiver (↪ p. 197).

References

Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 173–179, New York, NY, USA. ACM Press.

DateLens

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **interval**

vis
mapping: **static**
dimensionality: **2D**

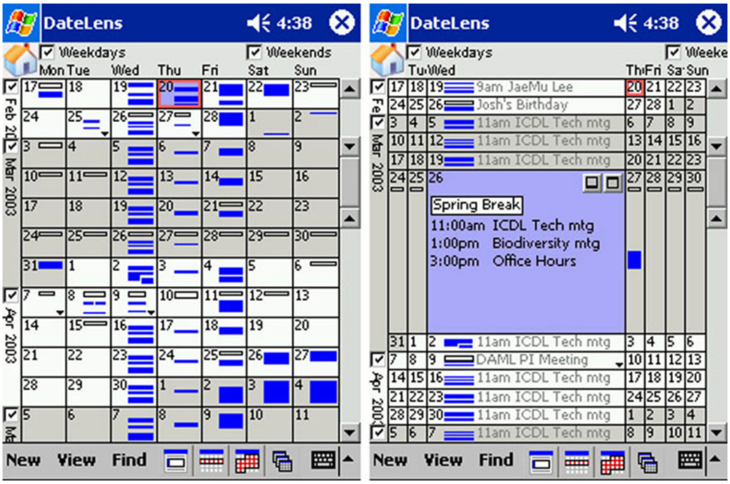


Fig. 7.18: A calendar grid shows the items of one's personal schedule as colored bars (left). Fisheye distortion is applied to show detailed textual information at the point the user is focusing on, and to maintain the context at less graphical detail (right).
Source: Images courtesy of Ben Bederson.

Most people use calendars to plan their daily life, for instance, to maintain a list of appointments or bookmark future events. Bederson et al. (2004) developed a tool to make it easier to work with a personal schedule on small handheld devices. Because display space is limited on such devices (compared to common desktop displays), focus+context mechanisms are applied to present temporal information at different levels of detail. Based on a common tabular representation of a calendar (see Figure 7.18, left), the DateLens magnifies table cells so as to provide more display space to important information that is currently in the user's focus (see Figure 7.18, right). The fisheye distortion magnifies the focus and reduces graphical detail in the context of the display. If sufficient display space is available, calendar entries are shown in textual form. Otherwise, temporal intervals of calendar entries are indicated by bars that visualize the starting point and temporal extent of appointments and events stored in the calendar. Various interaction mechanisms allow users to view the calendar at different temporal granularities and to navigate forward and backward in time.

References

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

TimeNets

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **interval**

vis

mapping: **static**
dimensionality: **2D**

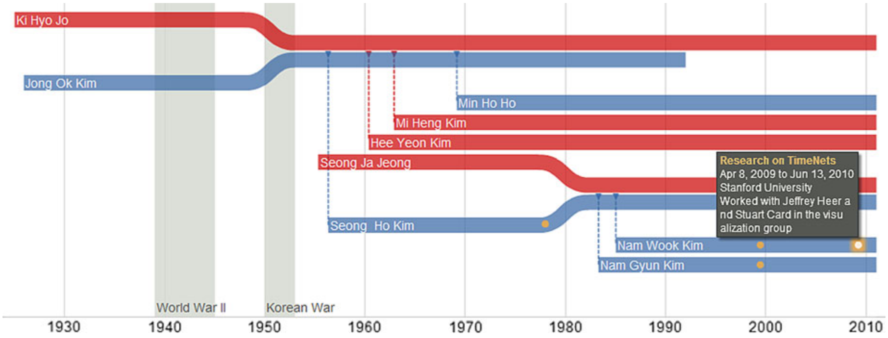


Fig. 7.19: TimeNets visualize temporal and structural aspects of genealogical data. Bands that extend along the horizontal time axis visualize individuals. Marriage and divorce are indicated by converging and diverging bands, respectively. Children are connected to their parents via drop lines. Labels are shown for the persons' names as well as for historical and personal events.
Source: Image courtesy of Jeffrey Heer and Nam Wook Kim.

Genealogical data are an interesting source of time-oriented information. In such data, not only family structures are of interest, but also temporal relationships. Kim et al. (2010) propose the TimeNets approach, which aims to visualize both of these aspects. TimeNets represent persons as individual bands that extend horizontally along a time axis from left to right. Each band shows a label of the person's name and different colors are used to encode sex: red is reserved for females, and males are shown in blue. Marriage of persons is visualized by converging the corresponding bands, while divorce is indicated by diverging bands. When a child is born, a new band is added to the display. A so-called drop line connects the band of the child to the parents' bands to convey the parent-child relationship. In order to allow users to focus on relevant parts of the data, a degree of interest (DOI) algorithm is applied. Bands below the DOI threshold are filtered out or smoothly fade in where they are linked to bands of relevant persons. Users can select multiple persons to focus on. On each change of the focus, the visualization shows a smooth transition of the display to keep users oriented.

References

Kim, N. W., Card, S. K., and Heer, J. (2010). Tracing Genealogical Data with TimeNets. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 241–248, New York, NY, USA. ACM Press.

Paint Strips

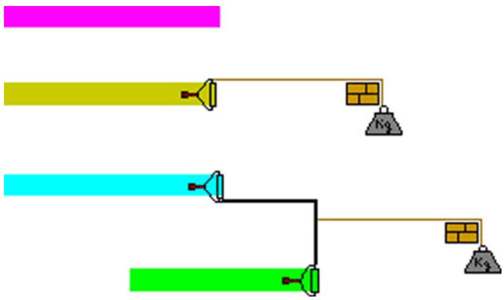


Fig. 7.20: Paint strips indicate the location and duration of time intervals, effectively allowing users to assess relationships of intervals. Temporal indeterminacy of intervals is indicated by paint rollers that can move flexibly within certain constraints, which are represented by wall elements. *Source: Image courtesy of Luca Chittaro.*

Chittaro and Combi (2003) designed paint strips to represent relations between time intervals for visualizing queries on medical databases. The technique is strongly related to timelines (↪ p. 166), but here paint strips are used as equivalents of bars to indicate time intervals, and optionally, the indeterminacy of intervals is communicated by placing paint rollers at either end of the paint strips. A paint roller with a weight attached to it means this interval can possibly extend in time. Graphical depictions of wall elements represent constraints on the extension. This way, the maximum duration and earliest start or latest end of intervals are defined, depending on which end of the painting strip the paint rollers are attached to. It is also possible to link strips, which means if one strip moves, the other one moves to the same extent as well. This relationship is indicated graphically by connecting the involved paint rollers before attaching them to the rope that holds the weight (see bottom of Figure 7.20). Paint strips were especially developed for medical applications but can be used anywhere where indeterminate time intervals have to be visualized. Thanks to the simplicity of the paint strip metaphor, there is room for application-dependent enhancements, such as textual annotations for start and end points as well as for durations of intervals.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **interval**

vis
mapping: **static**
dimensionality: **2D**

References

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.

PlanningLines

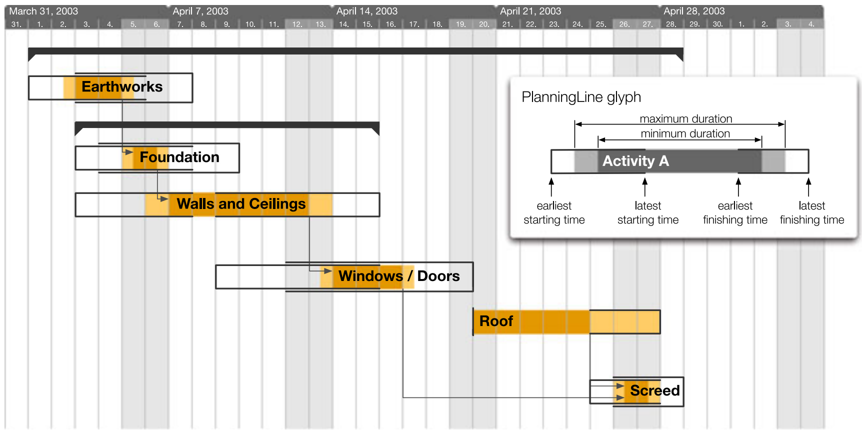


Fig. 7.21: Project plan of construction works that represents temporal uncertainties via PlanningLines. A PlanningLine glyph consists of two encapsulated bars, which represent minimum and maximum duration. The bars are bounded by two caps encoding the start and end intervals.
Source: Adapted from Aigner et al. (2005).

Since the future is always inherently connected with possible uncertainties, delays, and the unforeseen, these issues need to be dealt with in many domains like project management or medical treatment planning. PlanningLines by Aigner et al. (2005) allow the representation of temporal uncertainties, thus supporting project managers in their difficult planning and controlling tasks. PlanningLines have been designed to be easily integrated into well-known timeline-based visualization techniques such as Gantt charts (↪ p. 167). A single glyph (see glyph explanation in Figure 7.21) provides a visual representation of the temporal indeterminacies of a single activity, facilitates the identification of (un)defined attributes, supports in maintaining logical constraints (e.g., bars may not extend caps), and gives a visual impression of the individual and overall uncertainties. Uncertainties might be introduced by explicit specifications, usually connected with future planning (e.g., "The meeting will start between 12 p.m. and 2 p.m.." – which might be any point in time between noon and 2 p.m.) or is implicitly present in cases where data are given with respect to multiple temporal granularities (e.g., data given on a granularity of days and shown on an hourly scale).

References

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.

Time Annotation Glyph

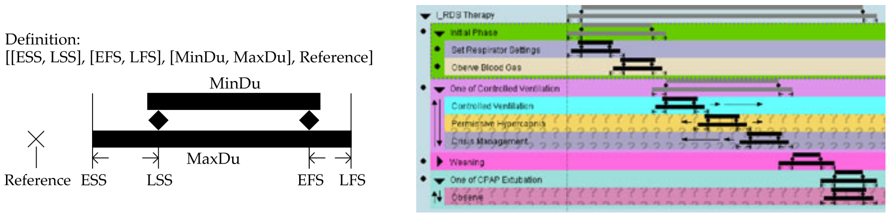


Fig. 7.22: The time annotation glyph was designed to represent the temporal constraints of medical treatment plans. It uses the metaphor of bars that lie on pillars. Left: Single glyph and associated parameters. Right: Usage in a tool for representing the temporal and hierarchical aspects of a medical treatment plan as well as the execution order of individual parts.
Source: Images courtesy of Robert Kosara.

The time annotation glyph by Kosara and Miksch (2001) uses the simple metaphor of bars that lie on pillars to represent a complex set of time attributes. Four vertical lines on the base specify the earliest and the latest starting and ending times. Supported by these pillars lies a bar that is as long as the maximum duration. On top of the maximum duration bar, a bar that represents the minimum duration lies upon two diamonds indicating the latest start and the earliest end. Furthermore, undefined parts and different granularities are indicated visually. Because of this metaphor, a few simple time parameter constraints can be understood intuitively. For example, the minimum duration cannot be shorter than the interval between latest start and earliest end – if it was, the minimum duration bar would fall down between its supports. All of the parameters might be defined relative to a reference point that is also represented graphically. In summary, the following parameters are shown: earliest starting shift (ESS), latest starting shift (LSS), earliest finishing shift (EFS), latest finishing shift (LFS), minimum duration (MinDu), and maximum duration (MaxDu). The technique is used to represent the time annotations of medical treatment plans within the AsbruView application (see Figure 7.22, right) as described in Kosara and Miksch (2001).

References

Kosara, R. and Miksch, S. (2001). Metaphors of Movement - A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131.

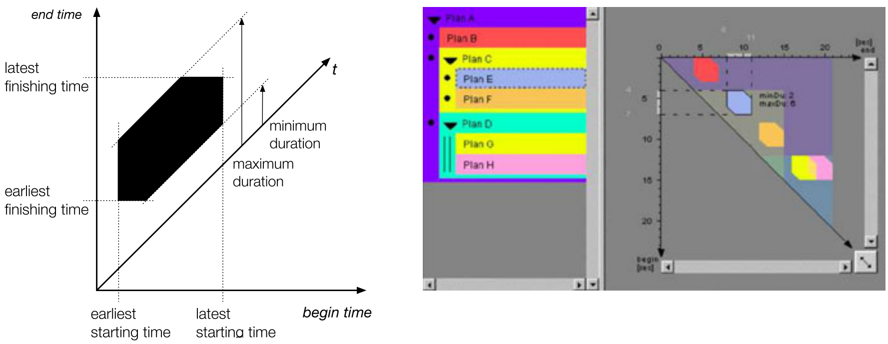
data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **interval**

vis
mapping: **static**
dimensionality: **2D**

SOPO Diagram

data
frame of reference: **abstract**
variables: **univariate**



time
arrangement: **linear**
time primitives: **interval**

Fig. 7.23: A SOPO diagram shows the possible configurations of the begin and end times of an event via a constrained polygonal shape. Right: SOPOView – an interactive visualization tool for working with SOPOs applied for medical treatment plans.
Source: Images courtesy of Robert Kosara.

For planning and scheduling, the temporal extents of events can be characterized by sets of possible occurrences (SOPOs), i.e., a set of possible begin and end times during which an event may happen. Rit (1986) defined a theoretical model for the definition and propagation of temporal constraints for scheduling problems. A graphical representation of SOPOs was introduced as a visual aid for understanding and solving such problems. In this representation, the extent of temporal uncertainty is expressed via a polygonal shape. The axes of a SOPO diagram represent begin time (x-axis) and end time (y-axis). Points in this diagram do not represent points in time, but complete intervals specified by their begin (x-coordinate) and end time (y-coordinate). Hence, the extent of an interval is represented by its position, not its visual extent. The area an item covers reflects all intervals that fit the specification given by means of earliest start, latest start, earliest end, latest end, minimum, and maximum duration (see Figure 7.23, left). Moreover, the exact occurrence of an event may be constrained by other related events which further modify the sets of possible occurrences. The propagation of such constraints is aided graphically, e.g., via overlaps of individual SOPOs. Later, this idea was interactively enhanced and further developed to be applied for visualizing medical treatment plans in the tool SOPOView (see Figure 7.23, right and Kosara and Miksch, 2002).

vis
mapping: **static**
dimensionality: **2D**

References

Kosara, R. and Miksch, S. (2002). Visualization Methods for Data Analysis and Planning. *International Journal of Medical Informatics*, 68(1–3):141–153.

Rit, J.-F. (1986). Propagating Temporal Constraints for Scheduling. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 383–388, Los Altos, CA, USA. Morgan Kaufmann.

Silhouette Graph, Circular Silhouette Graph

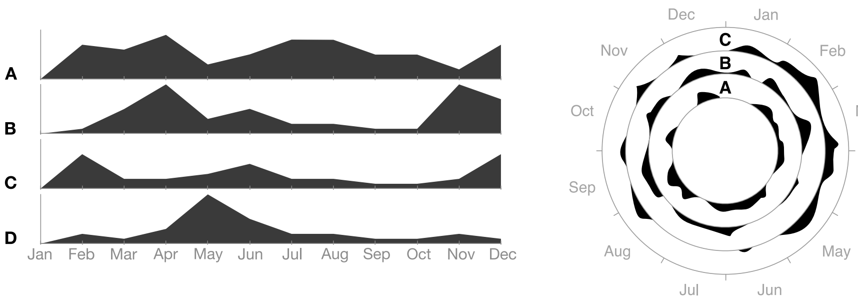


Fig. 7.24: Silhouette graphs are filled line plots that can be used for enhancing the comparison of multiple time-series that are put side-by-side.
Source: Adapted from Harris (1999).

Silhouette graphs emphasize the visual impression of time-series by filling the area below the plotted lines (see Harris, 1999). This leads to distinct silhouettes that enhance perception at wide aspect ratios of long time-series compared to line plots (↪ p. 153) and allow an easier comparison of multiple time-series. On the left of Figure 7.24, time is mapped to the horizontal axes and multiple time-series are stacked upon each other. Other layouts of the axes might be used for reflecting different time characteristics. One example are circular silhouette graphs (see Figure 7.24, right) that represent silhouette graphs on concentric circles in order to emphasize periodicities in time.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

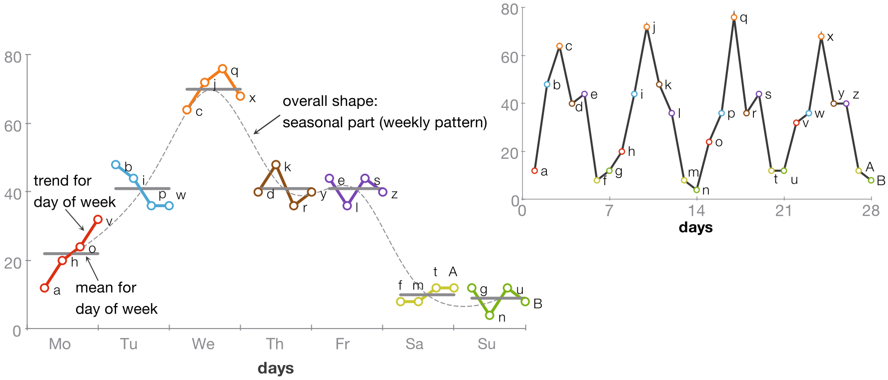
References

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

Cycle Plot

data

frame of reference: **abstract**
variables: **univariate**



time

arrangement: **linear, cyclic**
time primitives: **instant**

Fig. 7.25: Both seasonal and trend components of a time-series can be discerned from cycle plots (left), which is hardly possible when using line plots (right).
Source: Adapted from Cleveland (1993) with permission of William Cleveland.

Time-series data may contain a seasonal as well as a trend component, which is also reflected in many statistical models. Cleveland (1993) describes cycle plots as a technique to make seasonal and trend components visually discernable. This is achieved by showing individual trends as line plots embedded within a plot that shows the seasonal pattern. For constructing a cycle plot, one has to define the time primitives to be considered for the seasonal component. The horizontal axis of the cycle plot is then subdivided accordingly. Figure 7.25 demonstrates this using weekdays. We are interested in the trend for each weekday and the general weekly pattern. The data for a particular weekday are visualized as a separate line plot (e.g., data of the 1st, 2nd, 3rd, and 4th Monday). This allows the identification of individual trends for each day of the week. In the figure, we see an increasing trend for Mondays, but a decreasing trend for Tuesdays. Additionally, the cycle plot shows the mean value for each weekday (depicted as gray lines). Connecting the mean values as a line plot (dashed line in the figure) reveals the seasonal pattern, which in this case is a weekly pattern that clearly shows a peak on Wednesday.

vis

mapping: **static**
dimensionality: **2D**

References

Cleveland, W. (1993). *Visualizing Data*. Hobart Press, Summit, NJ, USA.

Cluster and Calendar-Based Visualization

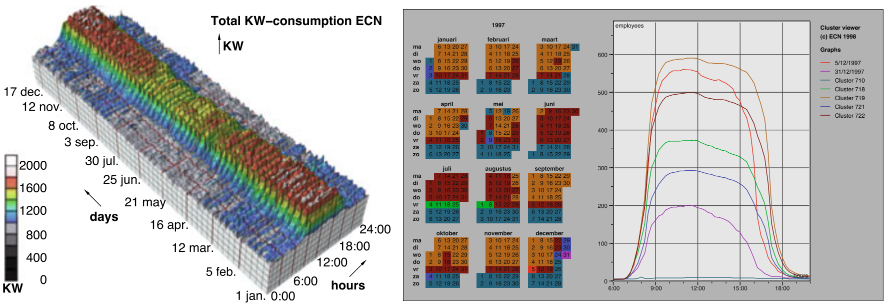


Fig. 7.26: The 3D visualization on the left represents daily power consumption patterns for several weeks of a year. The calendar representation on the right shows the cluster affiliation of daily patterns and allows users to discern days with uncommon patterns, in this case office hours of employees.
Source: Van Wijk and Van Selow (1999), © 1999 IEEE. Used with permission.

Temporal patterns can indicate at which time of the day certain resources are highly stressed. Relevant applications can be found in computing centers, traffic networks, or power supply networks. An approach that allows for finding temporal patterns at different temporal granularities has been proposed by Van Wijk and Van Selow (1999). The starting point of the approach is to consider the course of a day as a line plot (↪ p. 153) covering the 24 hours of a day. Multiple daily courses of this kind are visualized as a three-dimensional height field (see left part of Figure 7.26), where the hours of the day are encoded along one axis, individual days are encoded along the second axis, and data values are encoded as height (along the third axis). This allows users to detect short term daily patterns and long term trends of the data at higher temporal granularity. To assist in the analysis of the data, Van Wijk and Van Selow further suggest grouping similar daily courses into clusters (see also Section 6.2, p. 130). The data belonging to a particular cluster are aggregated to define the representative for that cluster, i.e., the representative again forms a daily course. Cluster affiliation of individual dates is then color-coded into a calendar as depicted in the right part of Figure 7.26. The user can adjust the number of clusters to be shown so as to find the level of abstraction that suits the data and the task at hand. The combination of analytical and visual methods as applied here is useful for identifying days of common and exceptional daily behavior.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D, 3D**

References

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 4–9, Los Alamitos, CA, USA. IEEE Computer Society.

Tile Maps

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

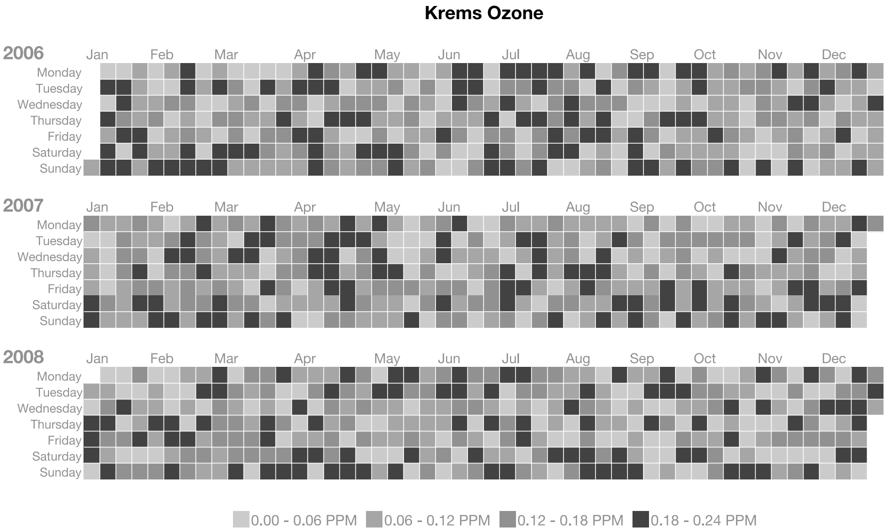


Fig. 7.27: The matrices show ozone measurements made in the course of three years, where data values for individual days are encoded to the brightness of matrix cells.
Source: Adapted from Mintz et al. (1997) with permission of David Mintz.

Tile maps as described by Mintz et al. (1997) represent a series of data values along a calendar division. The idea behind this technique is to arrange data values according to different temporal granularities. For example, data values measured on a daily basis are displayed in a matrix where each cell (or tile) corresponds to a distinct day, a column represents a week, and a row represents all data values for a particular weekday (see Figure 7.27). One additional level of granularity can be integrated by stacking multiple matrices as shown in the figure. Data values are visualized by varying the lightness of individual tiles. A visual representation constructed this way can be interpreted quite easily, because it corresponds to our experience of looking at calendars. The arrangement as a two-dimensional matrix allows users to identify long-term trends by considering the matrix as a whole, to discern individual trends for Mondays, Tuesdays, and so forth by looking at the matrix rows, and to derive weekly patterns by investigating matrix columns. For example, the U.S. Environmental Protection Agency provides a web tool that generates tile maps automatically for specified air pollutants and locations.

References

Mintz, D., Fitz-Simons, T., and Wayland, M. (1997). Tracking Air Quality Trends with SAS/GRAPH. In *Proceedings of the 22nd Annual SAS User Group International Conference (SUGI97)*, pages 807–812, Cary, NC, USA. SAS.

Multi Scale Temporal Behavior

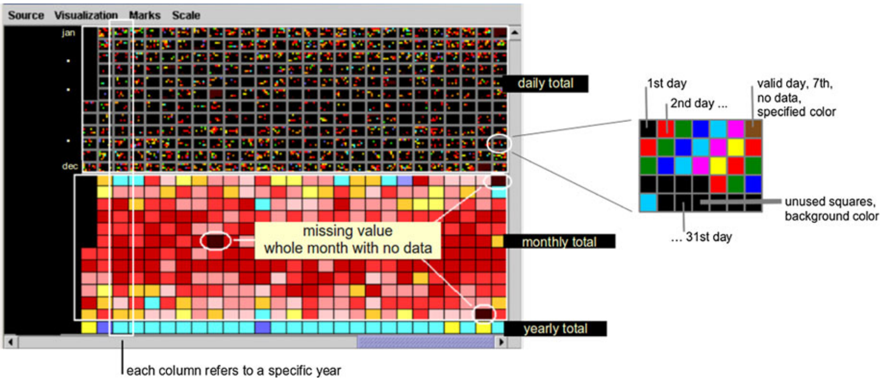


Fig. 7.28: Different levels of granularity are shown simultaneously in one display to allow users to explore patterns in precipitation data from Brazil at different temporal levels.
Source: Image courtesy of Milton Hirokazu Shimabukuro.

The Multi Scale Temporal Behavior technique by Shimabukuro et al. (2004) comprises different levels of granularity and aggregation to explore patterns at different temporal levels. The basis for the visualization is a matrix that is divided vertically into three regions, one for each of the three scale levels: daily data, monthly data, and yearly data (see Figure 7.28, top to bottom). Each column of the matrix represents a year worth of data. The cells in the topmost region represent months. They show full detail by color-coding individual pixels within a cell according to daily values. The middle region shows aggregated data. Here cells are no longer subdivided into pixels, but are colored uniformly, where the color represents the aggregation of daily values to a single monthly value. The same principle is applied for the bottom region (in fact, the bottom row). Twelve monthly values are aggregated into a single value for the year, which can again be represented by color. A significant and non-trivial problem in dealing with real world datasets are missing data values. This issue is tackled by the authors by preprocessing the data and marking missing values as such in the display.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

References

Shimabukuro, M., Flores, E., de Oliveira, M., and Levkowitz, H. (2004). Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. In *Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pages 107–117, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **abstract**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

Recursive Pattern

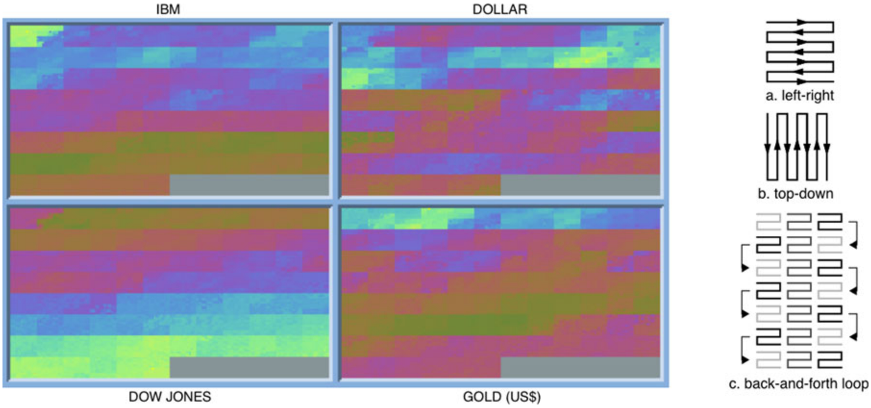


Fig. 7.29: Four time-series are shown as recursive patterns, where the color of a pixel represents stock price. Pixels are arranged to semantically match the hierarchical structure of the data: A row represents a year with its 12 months, each month is further subdivided into weeks, which in turn consist of five workdays, each of which represents nine data values for a single day. Right: Examples of possible pixel arrangements.
Source: Keim et al. (1995), © 1995 IEEE. Used with permission.

The most space-efficient way of visualizing data is to represent them on a per-pixel basis. Keim et al. (1995) suggest a variety of pixel-based visualization approaches of which the recursive pattern technique is particularly suited to display large time-series. The key idea behind the recursive pattern technique is to construct an arrangement of pixels that corresponds to the inherently hierarchical structure of time-oriented data given at multiple granularities. Figure 7.29 shows financial data as a pixel-based visualization. The initial step is to map nine data values collected per day to a 3x3 pixel group. This group is then used to form a larger group for a week of workdays containing 5x1 day groups. Recursively, groups for months, years, and decades can be created by arranging groups of the next lower granularity in a semantically meaningful way (e.g., 12 months are grouped into a year). In the resulting pattern, each pixel is color-coded with regard to a single data value in the time-series. Multiple dense pixel displays of this kind can be combined to get an overview of large multivariate datasets.

References

Keim, D., Kriegel, H.-P., and Ankerst, M. (1995). Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data. In *Proceedings of IEEE Visualization (Vis)*, pages 279–286, Los Alamitos, CA, USA. IEEE Computer Society.

GROOVE

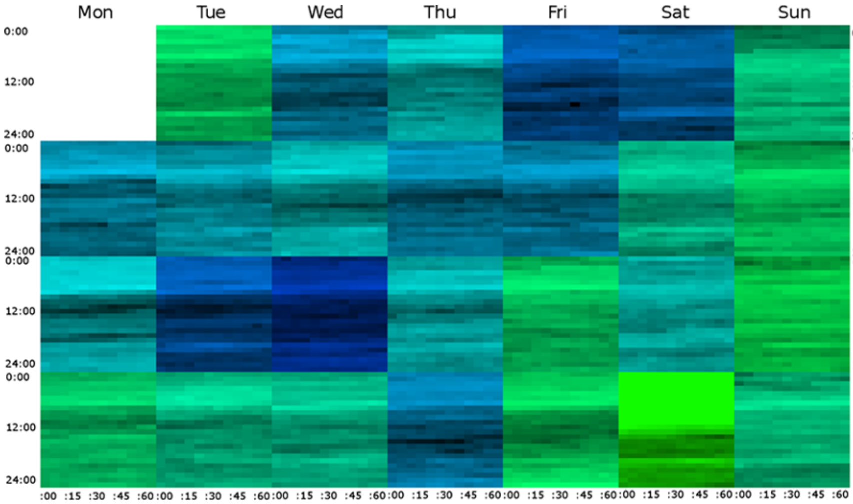


Fig. 7.30: GROOVE visualizations combine detail and overview readings and use regular layouts based on time granularities. Here, police assignments are shown over the course of one month. For each 5-minute interval the number of units assigned to a duty is given. Each block shows a day, rows of blocks represent weeks, columns of blocks show the day of the week, and within each block a row of pixels displays one hour. Color components are used to show detail values for pixels (lightness – bright: low, dark: high) and average values per block (hue – green: low, blue: high).
Source: Generated with the GROOVE software.

GROOVE (Granularity Overview OVErlay) visualizations as presented by Lammarsch et al. (2009) utilize a user-configurable set of four time granularities to partition a dataset in a regular manner. That is, a recursive layout is achieved that shows columns and rows of larger blocks and a pixel arrangement within blocks for the detail structure. Following the concept of recursive patterns (↪ p. 180) different arrangements might be chosen (e.g., row-by-row or back-and-forth). The specific of GROOVE visualizations is the combination of overview (aggregated values) and details in one place using one of three kinds of overlays. This allows micro and macro readings and avoids eye movements between the overview and detail representations. First, color components can be employed with color-based overlay (see Figure 7.30). Second, opacity overlay applies interactive crossfading between the overview and the detail display. Third, spatial overlay can be used for viewing the data selectively at different levels of aggregation by expanding and collapsing areas.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

References

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.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

SolarPlot

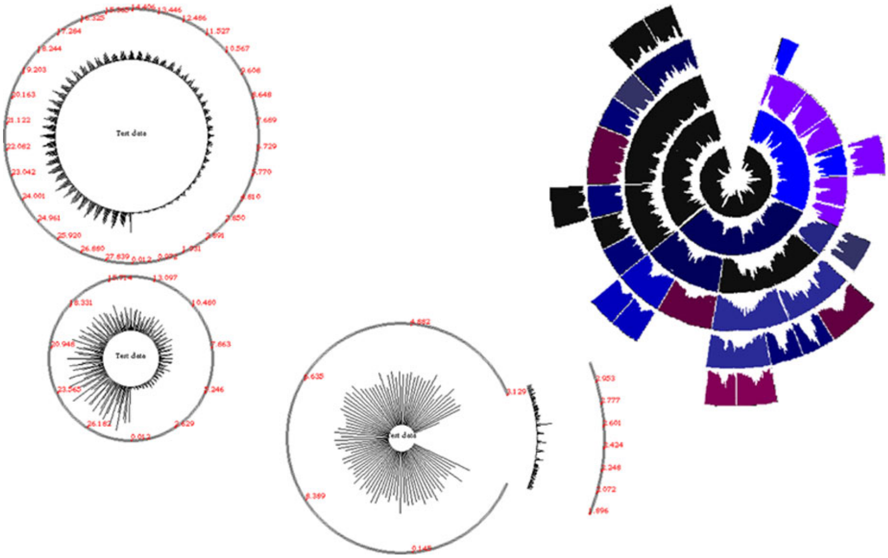


Fig. 7.31: Daily values of ticket sales data over a period of 30 years are plotted around the circumference of a resizable circle (left). For a selected interval, a more detailed arc can be shown (center). The technique can be enhanced so as to combine the representation of data and a hierarchical structure, in this case email traffic and company organization (right).
Source: Chuah (1998), © 1998 IEEE. Used with permission.

With the SolarPlot technique introduced by Chuah (1998), values are plotted around the circumference of a circle as shown left in Figure 7.31. Much like in a circular histogram, the first step is to partition the data series into a number of bins. Each bin is represented by a sunbeam whose length encodes the frequency of data items in the corresponding bin. The SolarPlot determines the number of bins dynamically depending on the size of the circle. Users are allowed to expand or contract the circle in order to get more or fewer bins, or in other words, to get a more or less detailed representation of the data. This way it is possible to explore the data at different levels of abstractions and to discern patterns globally across aggregation levels. The SolarPlot also supports locally switching to a more detailed plot for a user-selected focus interval as shown in the center of Figure 7.31. Chuah (1998) further suggests a variation of the SolarPlot called SolarPlot + Aggregate TreeMap, where the display of data is combined with a visual representation of a hierarchical structure (see Figure 7.31, right).

References

Chuah, M. C. (1998). Dynamic Aggregation with Circular Visual Designs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 35–43, Los Alamitos, CA, USA. IEEE Computer Society.

SpiraClock

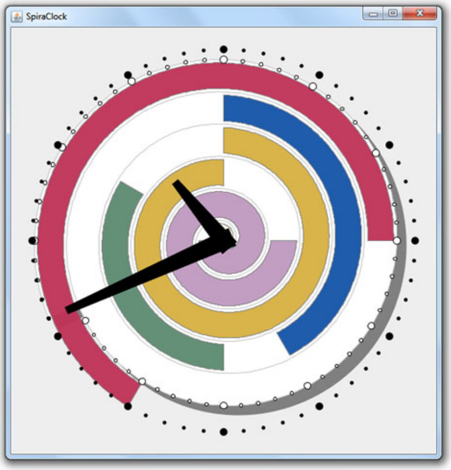


Fig. 7.32: The SpiraClock builds upon the clock display. The minute hand currently points to a meeting that has already started. Future appointments are aligned along a spiral on the clock face. *Source: Adapted from Dragicevic and Huot (2002) with permission of Pierre Dragicevic.*

The SpiraClock invented by Dragicevic and Huot (2002) visualizes time by using the clock metaphor. The visual representation consists of a clock face and two hands indicating hour and minute. The interior of the clock shows a spiral that extends from the clock's circumference toward its center. Each cycle of the spiral represents 12 hours, with the current hour shown at the outermost cycle and future hours displayed in the center (about nine future hours in Figure 7.32). Time intervals (e.g., meetings) are represented as thick segments along the spiral shape. These segments show when intervals start and end. Users can also see if certain appointments are in conflict because they *overlap*, or if the agenda is too tight, because many appointments *meet* (for further interval relations see Section 3.1.2). As time advances, the spiral is constantly updated and future intervals gradually move outward until they are current. Past intervals gradually fade out. In this sense, the SpiraClock enhances classic clocks with a preview of the near future and a brief view to the past. The SpiraClock allows users to drag the clock handles to visit different points in time, and intervals of interest can be highlighted and corresponding textual annotations can be displayed.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **cyclic**
time primitives: **interval**

vis
mapping: **dynamic**
dimensionality: **2D**

References

Dragicevic, P. and Huot, S. (2002). SpiraClock: A Continuous and Non-Intrusive Display for Upcoming Events. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 604–605, New York, NY, USA. ACM Press. Extended Abstracts.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **cyclic**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

Enhanced Interactive Spiral

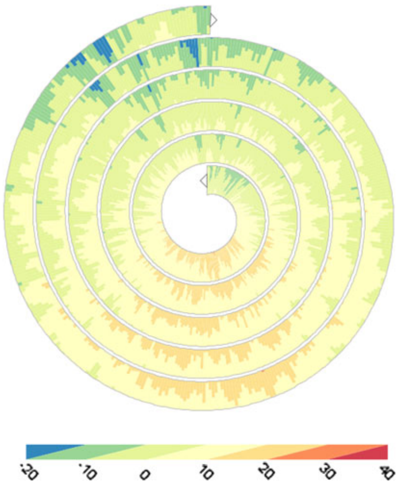


Fig. 7.33: The spiral shows the daily temperature measured in Rostock from 2006 to 2010. The blue and green colors in the upper left part of the spiral represent the colder winters of 2009 and 2010.
Source: Generated with the enhanced interactive spiral display tool.

Tominski and Schumann (2008) apply the enhanced two-tone color-coding by Saito et al. (2005) to visualize time-dependent data along a spiral. Each time primitive is mapped to a unique segment of the spiral. Every segment is subdivided into two parts that are colored according to the two-tone coloring method. The advantage of using the two-tone approach is that it realizes the overview+detail concept by design. The two colors used per spiral segment allow users to quickly recognize the value range of that segment (overview). If the value range is of interest, the proportion of the two colors indicates the particular data value more precisely (detail). The enhanced spiral can be adjusted interactively in various ways. The number of time primitives, the number of cycles, and additional geometrical parameters influence the shape of the spiral and thus the mapping of the time domain. The data representation is mainly controlled by the color scales applied and parameters such as the number of colors, the direction of the mapping, and the mapping function (linear vs. logarithmic). Navigation in time is possible via direct manipulation of the spiral.

References

Saito, T., Miyamura, H., Yamamoto, M., Saito, H., Hoshiya, Y., and Kaseda, T. (2005). Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 173–180, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C. and Schumann, H. (2008). Enhanced Interactive Spiral Display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity*, pages 53–56, Linköping, Sweden. Linköping University Electronic Press.

Spiral Graph

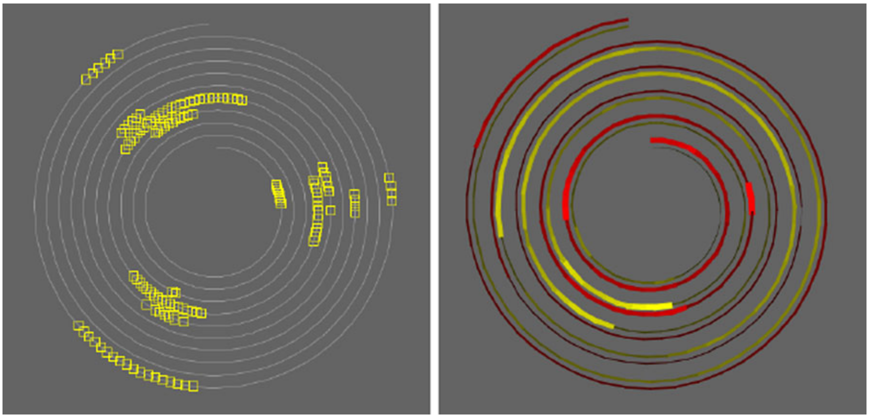


Fig. 7.34: A spiral graph encodes time-series data along a spiral, thus emphasizing the cyclic character of the time domain. Actual data values are visualized using symbols for nominal data (left) as well as color and line thickness for quantitative data (right).
Source: Weber et al. (2001), © 2001 IEEE. Used with permission.

The spiral graph developed by Weber et al. (2001) is a visualization technique that focuses on cyclic characteristics of time-oriented data. To this end, the time axis is represented by a spiral. Time-oriented data are then mapped along the spiral path. While nominal data are represented by simple icons (see Figure 7.34, left), quantitative data can be visualized by color, line thickness, or texture. One can also visualize multivariate time-series by intertwining several spirals as shown for two variables in the right part of Figure 7.34. In this case, a distinct hue is used per spiral so that individual variables can be discerned. Weber et al. (2001) further envision extending the spiral to a three-dimensional helix, in order to cope with larger time-series. The main purpose of the spiral graph is the detection of previously unknown periodic behavior of the data. The user can interactively adjust the spiral's cycle length (i.e., the number of data values mapped per spiral cycle) to explore the data for cyclic patterns. As an alternative, it is also possible to smoothly animate through possible cycle lengths. In this 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 (see also Section 4.2.1, p. 84).

data

frame of reference: **abstract**
variables: **uni-, multivariate**

time

arrangement: **cyclic**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

References

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing Time-Series on Spirals. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 7–14, Los Alamitos, CA, USA. IEEE Computer Society.

Spiral Display

data
frame of reference: **abstract**
variables: **uni-, multivariate**

time
arrangement: **cyclic**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D, 3D**

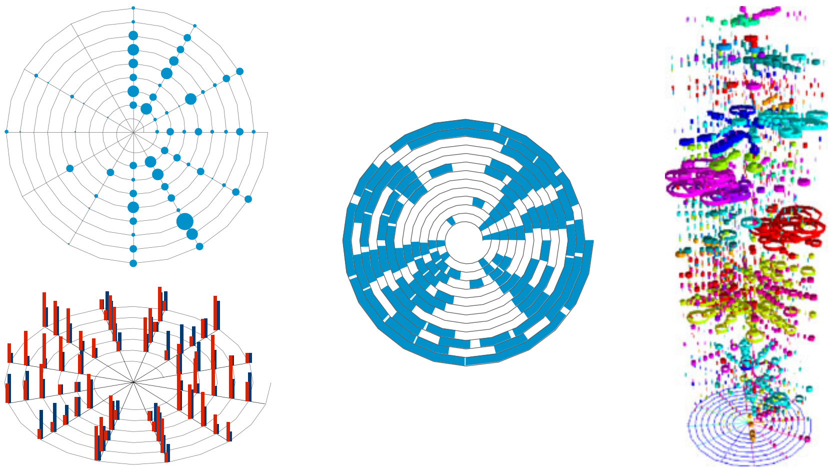


Fig. 7.35: Data can be visualized along a spiral in different ways: by the area of circular elements (top-left), by the sizes of multiple spikes (bottom-left), as bars marking start and end of intervals (center), or by the volume of hollow cans aligned at different layers along the vertical axis (right). *Source: Carlis and Konstan (1998), © 1998 ACM. Used with permission.*

The interactive spiral display by Carlis and Konstan (1998) uses Archimedean spirals to represent the time domain. Data values at particular time points are visualized as filled circular elements whose area is proportional to the data value (see Figure 7.35, top-left). In the case of interval-based data, filled bars are aligned with the spiral shape to indicate start and end of intervals (see Figure 7.35, center). If multivariate data are given at time points, the spiral is tilted and data values are visualized as differently colored spikes, where spike color indicates variable affiliation and spike height encodes the corresponding data value (see Figure 7.35, bottom-left). Alternatively, one can use the vertical z-axis to separate the display of multiple variables (see Figure 7.35, right). In this case, each time-dependent variable has its own layer along the z-axis and is represented with a unique color. Within a layer, data values are encoded to the volume of cans, which are lidless and hollow to prevent occlusion. The system implemented by Carlis and Konstan (1998) allows users to display multiple linked spirals to perform comparison tasks. The cycle lengths of spirals can be adjusted interactively and can also be animated automatically for discovering periodic patterns.

References

Carlis, J. V. and Konstan, J. A. (1998). Interactive Visualization of Serial Periodic Data. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 29–38, New York, NY, USA. ACM Press.

VizTree

data

frame of reference: **abstract**
variables: **uni-, multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

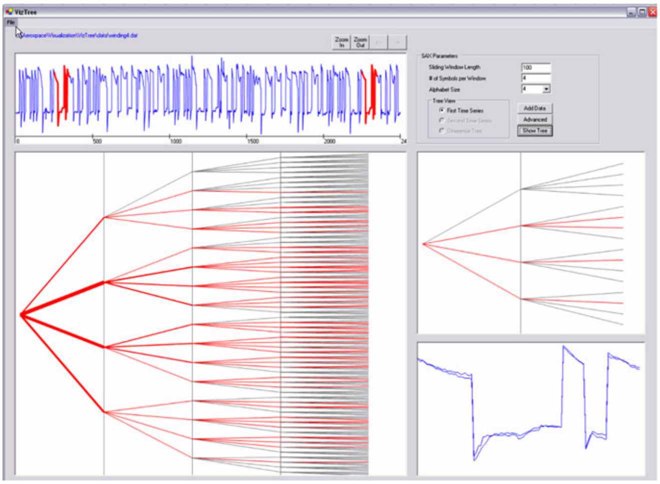


Fig. 7.36: Time-series of arbitrary length are represented as fixed-length subsequence trees. The figure shows a winding dataset that records the angular speed of a reel. Top-left: input time-series; top-right: parameter setting area for discretization and subsequence length; left: subsequence tree for the time-series; center-right: detail view of the tree shown in the left panel; lower-right: subsequences matching a particular string representation (e.g., subsequences starting with 'ab') whereas positions of matched subsequences are highlighted in the top-left panel.
Source: Image courtesy of Eamonn Keogh.

VizTree by Lin et al. (2005) is a time-series pattern discovery and visualization system for massive time-series datasets. It uses the time-series discretization method SAX (symbolic aggregate approximation) developed earlier by Lin et al. (2007). SAX discretizes time-series into a sequence of symbols (e.g., 'abacacc'). Subsequences (patterns) are generated by moving a sliding window along the sequence. These subsequences are combined and represented by a horizontal tree visualization where the frequency of a pattern is encoded by the thickness of a branch (or light gray when frequency is zero). The VizTree interface consists of multiple coordinated views that show the input time-series along with the subsequence tree as well as control and detail-on-demand panels. VizTree can be used to accomplish different pattern discovery tasks interactively: finding frequently occurring patterns (i.e., motif discovery) and surprising patterns (i.e., anomaly detection), query by content, and the comparison of two time-series by calculating a difference tree.

References

Lin, J., Keogh, E. J., 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. 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.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

TimeSearcher

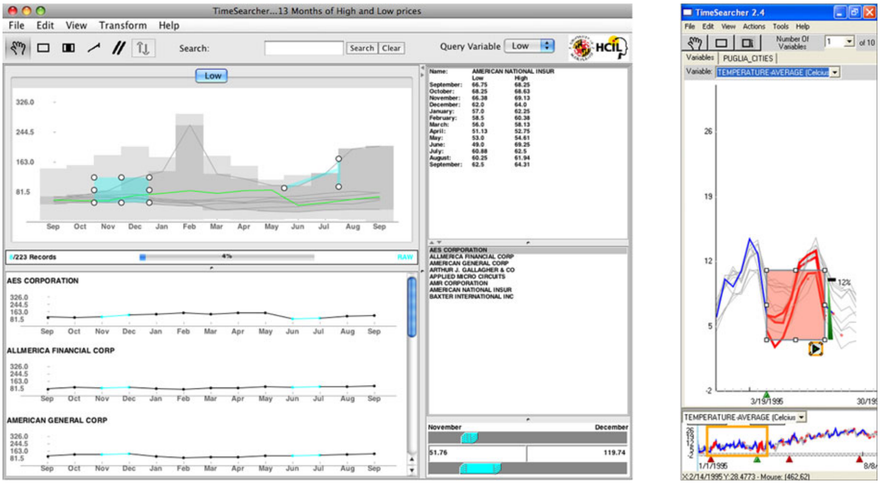


Fig. 7.37: Left: TimeSearcher 1 showing stock price data (top-left: Multiple line plots and query boxes; bottom-left: line plots for individual stocks matching the query). Right: TimeSearcher 2 with query-by-example using a searchbox (light red background) and matching items in red.
Source: Generated with the TimeSearcher software with permission of University of Maryland Human-Computer Interaction Lab.

Hochheiser and Shneiderman (2004) implemented TimeSearcher as a visual exploration tool for multiple time-series. While employing a straightforward visual representation using line plots, its main objective is to enable users to identify and find patterns in the investigated data. To this end, the so-called *timebox query model* has been developed. It allows the specification of a rectangular query region that defines both a time interval and a value range of interest. Those time-series that comply with a query (i.e., overlap with the timebox) are displayed, whereas all others are filtered out. Users can combine multiple timeboxes to refine the query further and other query functionalities such as leaders and laggers, angular queries, and variable timeboxes are also part of TimeSearcher. To provide contextual information, the data envelope and the query envelope can be displayed. Buono et al. (2005) extended these features in TimeSearcher 2 by allowing the representation of heterogeneous datasets and providing a *searchbox query model* that effectively implements a query-by-example functionality. Here, occurrences of a brushed portion of the time-series are searched, whereas the similarity threshold of matches can be adjusted.

References

Buono, P., Aris, A., Plaisant, C., Khella, A., and Shneiderman, B. (2005). Interactive Pattern Search in Time Series. In *Proceedings of the Conference on Visualization and Data Analysis (VDA)*, pages 175–186. SPIE.

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.

TimeSearcher 3, River Plot

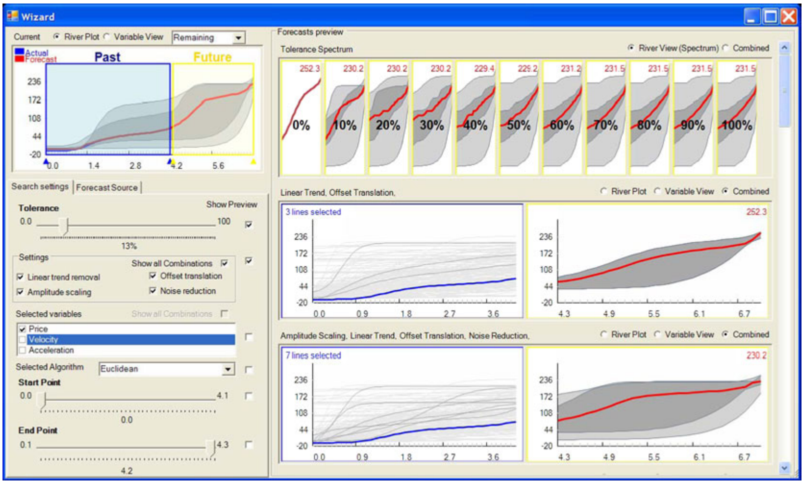


Fig. 7.38: Forecasting of online auction data. Top-left: selection of time interval for similarity search; bottom-left: selection of variables to consider and parameters to vary in the previews; right: preview area that assists users in understanding the impact of parameters – varying tolerance levels as river plots and different combinations of applied transformations as line plots and river plots.
Source: Image courtesy of Paolo Buono.

Buono et al. (2007) developed TimeSearcher 3 as a tool to support similarity-based forecasting of multivariate time-series. Similarity-based forecasting is a data-driven method using the similarity to a set of historical data for predicting future behavior. The outcome of the algorithm is affected by a number of options and parameters, for instance, the transformations applied or the tolerance threshold used for matching. As results, the median of the matched subsets becomes the forecast and descriptive statistics measures reflect the uncertainty associated with the forecast. This is displayed graphically as a simplified, continuous box plot, called a river plot. It uses superimposed, colored regions, for which light gray indicates the range between the minimum and maximum and dark gray the range between the 25% and 75% percentiles, and a line in the center, where red indicates the forecast, brown shows the median during the matching period, and black is the median before this period. TimeSearcher 3 builds upon TimeSearcher (↪ p. 188) and adds a preview interface to allow users to interactively explore the effects of adjusting algorithm parameters and to see multiple forecasts simultaneously.

References

Buono, P., Plaisant, C., Simeone, A., Aris, A., Shneiderman, B., Shmueli, G., and Jank, W. (2007). Similarity-Based Forecasting with Simultaneous Previews: A River Plot Interface for Time Series Forecasting. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 191–196, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

BinX

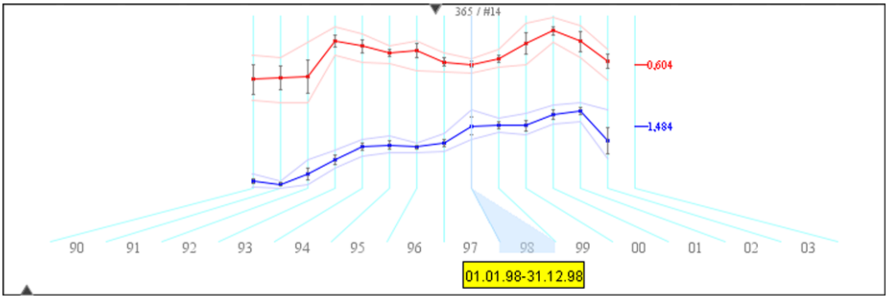


Fig. 7.39: Exchange rates for two currencies are compared using the BinX tool. Each bin aggregates the daily rates for a whole year. A selected bin is highlighted and its position on the global time scale is marked accordingly.
Source: Generated with the BinX tool with permission of Tamara Munzner.

Large time-series require the application of abstraction methods in order to reduce the number of time points to be displayed, thus keeping visualization costs at a manageable level. Finding a suitable degree of abstraction, however, is not an easy task. The BinX tool developed by Berry and Munzner (2004) is interesting in that it supports the exploration of different aggregations of a time-series. The aggregation is based on constructing bins, each of which holds a user-defined number of time points. Easy-to-use interaction is offered to quickly try out differently sized bins. BinX visualizes one or two quantitative time-dependent variables using common chart elements. An overview of the time axis is preserved at all times at the bottom of the BinX representation. The central chart view displays the two time-series in an aggregated fashion according to currently chosen bin size. In order to faithfully represent aggregated information, line plot (↪ p. 153), box-plots, and a min-max band are used in combination. The correspondence between a point in the chart and a time span (bin) on the time axis is represented upon user request. BinX supports clustering of bins as an additional mechanism for analytic abstraction. In this case, cluster affiliation of bins is encoded via color.

References

Berry, L. and Munzner, T. (2004). BinX: Dynamic Exploration of Time Series Datasets Across Aggregation Levels. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 5–6, Los Alamitos, CA, USA. IEEE Computer Society.

LiveRAC

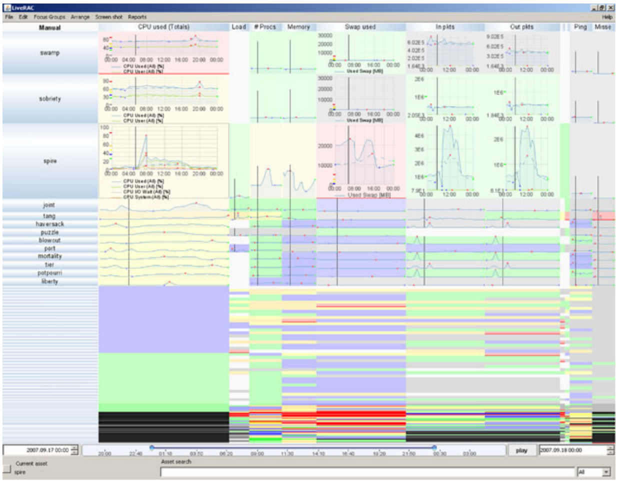


Fig. 7.40: A full day of system management time-series data showing more than 4000 devices in rows and 11 columns representing groups of monitored parameters. Representations within cells adapt to the available screen space by using different representations with more or less detail. *Source: McLachlan et al. (2008), © 2008 ACM. Used with permission.*

McLachlan et al. (2008) developed LiveRAC, a system for analyzing system management time-series data. LiveRAC scales to dozens of parameters collected from thousands of network devices. Familiar representations such as line plots (↪ p. 153), bar graphs (↪ p. 154), and sparklines (↪ p. 155) appear as the cells of a spreadsheet-like matrix. Rows and columns of the matrix are associated with monitored network devices and monitored parameters, respectively. Each cell contains an area-aware chart showing time on the horizontal axis and parameters on the vertical axis. To ensure that all cells remain visible at all times (i.e., to avoid scrolling), LiveRAC uses a so-called *stretch and squish* layout, which dynamically compresses and expands cells according to user interaction. Moreover, the individual charts adapt to the available screen space. This semantic zoom functionality ranges from charts with detailed labels, to smaller charts with fewer curves and less labeling, and ultimately to colored blocks for the smallest view. The cell background color represents changeable thresholds of minimum, maximum, or average values of the displayed parameters. Aggregation is applied if cells would overlap due to space restrictions, which is reflected in color intensity.

References

McLachlan, P., Munzner, T., Koutsofios, E., and North, S. (2008). LiveRAC: Interactive Visual Exploration of System Management Time-Series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, CHI '08, pages 1483–1492, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

LifeLines2

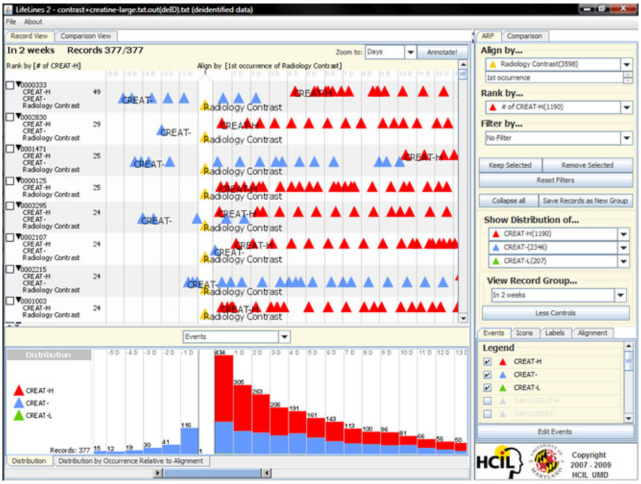


Fig. 7.41: LifeLines2 show patient records in stacked rows, where triangles indicate health-related events. A histogram view visualizes the number of events over time. Interaction operators (i.e., align, rank, filter) support visual exploration.
Source: Image courtesy of Taowei David Wang.

LifeLines2 by Wang et al. (2009) is an interactive visual exploration interface for instantaneous events based on categorical, health-related data (e.g., high, normal, or low body temperature). Events are displayed as triangles along a horizontal time axis, where color indicates event categories and data of different patient records are stacked vertically. An aggregation of events is represented as a histogram showing the number of occurrences over time. LifeLines2 introduces three powerful operators for interactive exploration: align, rank, and filter. The align operator can be used to arrange all records along a specific event type in temporal order, for example, to align a group of patients with regard to their first heart attack. Additionally, the time axis switches from an absolute time representation to relative time originating from the specified event (e.g., one week before, or two weeks after the first heart attack). The rank operator is useful for ordering records according to the number of occurrences of a specified event type. The filter operator allows searching of particular sequences of events including both the presence of events and the absence of events (e.g., patients having had a heart attack but no stroke following it).

References

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 Transactions on Visualization and Computer Graphics*, 15:1049–1056.

Similan

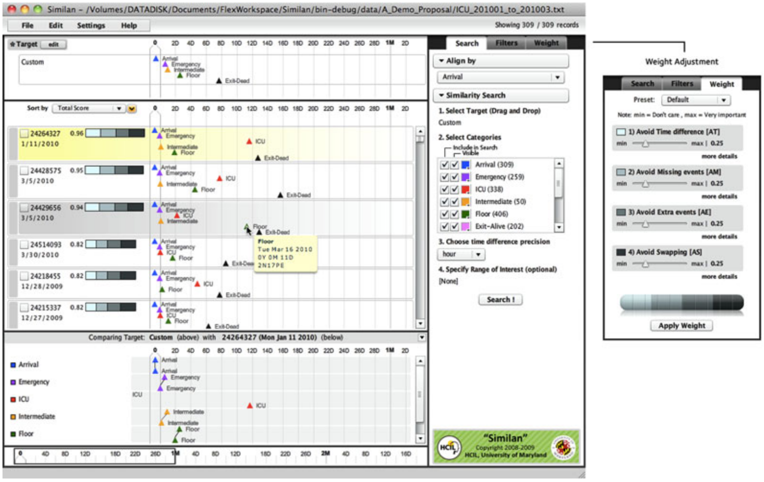


Fig. 7.42: Similan ranks patient records (center) according to their similarity to a target record (top). Individual records can be compared directly with the target (bottom). Various interaction operators, including adjustment of similarity weights (right), can be used to refine the visualization. *Source: Image courtesy of Krist Wongsuphasawat.*

Wongsuphasawat and Shneiderman (2009) describe Similan as a system for exploring patient records. Patient records are stacked upon each other and show health-related events as triangles, where color indicates event categories (e.g., arrival, emergency, ICU). Similan uses the same visual representation as LifeLines2 (↪ p. 192) but provides a different approach to data exploration. Instead of interactive filtering, records are ranked according to their similarity to a given event sequence (query-by-example). In Figure 7.42, the topmost record is the record that is most similar to the user-specified event sequence. This can be used to search for groups of patients who share similar temporal patterns. A dedicated view is provided to allow a direct comparison of the target query record with any particular record in the data. Another scenario is to search for an event sequence that the user is not certain whether it exists in the data; in this way the tool can give the most similar results if the exact event sequence does not exist. For determining the similarity of event sequences a similarity measure (M&M measure) has been developed. The weights of factors that determine the similarity measure can be adjusted interactively by the user.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

References

Wongsuphasawat, K. and Shneiderman, B. (2009). Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 27–34, Los Alamitos, CA, USA. IEEE Computer Society.

CareCruiser

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

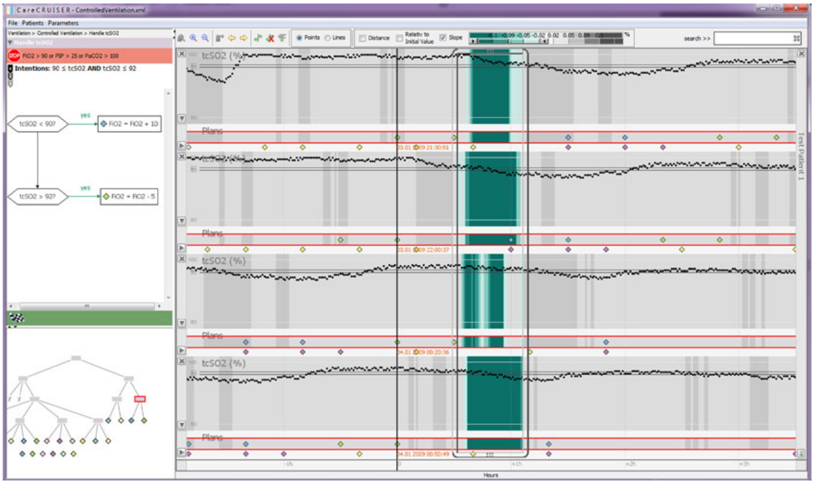


Fig. 7.43: A patient's parameters are displayed together with the applied clinical actions. In the selected area on the right, a delayed drop of the patient's $tcSO_2$ values after applying a specific clinical action is revealed. Contextual views are shown on the left – top-left: flow-chart like representation of the treatment plan logic; bottom-left: hierarchical decomposition of treatment plan. *Source: Generated with the CareCruiser software.*

CareCruiser by Gschwandtner et al. (2011) is a visualization system for exploring the effects of clinical actions on a patient's condition. It supports exploration via aligning, color-highlighting, filtering, and providing focus and context information. Aligning clinical treatment plans vertically supports the comparison of the effects of different treatments or the comparison of different effects of one treatment plan applied on different patients. Three different color-schemes are provided to highlight interesting portions of the development of a parameter: highlighting the distance of the actual values to the intended value helps to identify critical values; highlighting the progress of the actual values relative to the initial values shows to what extent the applied treatment plan has the intended effect; and highlighting the slope of a value helps to explore the immediate effects of applied clinical actions. A range slider is provided to filter the color-highlighting for selected events (see Figure 7.43, top) and a focus window which grays out the color-information outside its borders is used to support a focused investigation of a region of specific interest.

References

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

Layer Area Graph

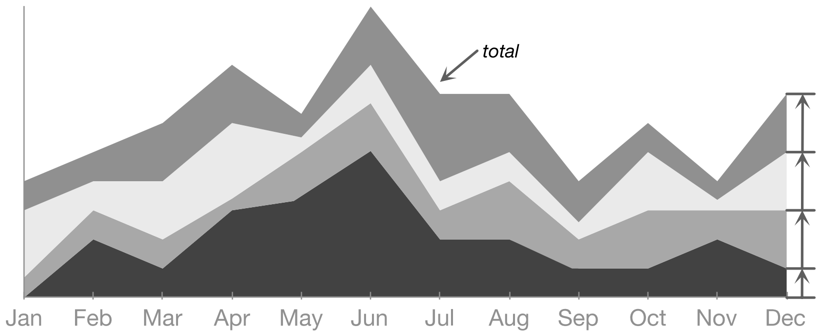


Fig. 7.44: Shows the developments of four variables as layered bands while emphasizing the total sum.
Source: Adapted from Harris (1999).

Layer area graphs might be used when comparing time-series that share the same unit and can be summed up (see Harris, 1999). A layer area graph is a stacked visualization where time-series plots are drawn upon each other as layered bands. Caution needs to be exercised for this kind of representation because it is sensitive to the order of the layers. Different orders influence the visual appearance of the individual layers because only the bottommost layer has a straight baseline. All subsequent layers are drawn relative to the layers below. An advantage of layer area graphs is the fact that they emphasize the total sum of values while providing information about the parts that constitute it. More advanced visualization techniques such as the ThemeRiver (↪ p. 197) or stacked graphs (↪ p. 199) build upon the basic principle of layer area graphs.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

References

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

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

Braided Graph

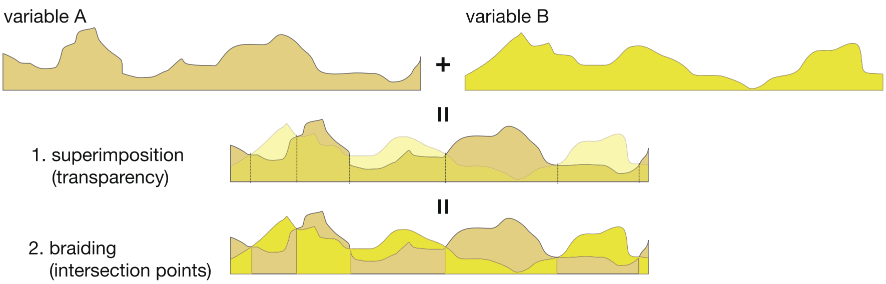


Fig. 7.45: Construction scheme of a braided graph for two variables. Top: individual variables as silhouette graphs; center: superimposed silhouette graphs using transparency (dashed lines show intersection points); bottom: braided graph (segments between intersection points are sorted to ensure visibility of all fragments).
Source: Adapted from Javed et al. (2010). © 2010 IEEE. Used with permission.

Braided graphs allow for superimposing silhouette graphs to show multivariate data. They were developed in order to take advantage of the enhanced perception of silhouette graphs (↪ p. 175) and at the same time avoiding the disadvantage of varying baselines of layered graphs (↪ p. 195). Simply drawing silhouette graphs on top of each other would lead to occlusion problems where a silhouette for larger data values occludes silhouettes for smaller values. The solution to this problem is to identify the points at which silhouettes intersect and to adapt the drawing order in between two intersections individually so that smaller silhouettes are always in front of larger ones. This ensures that all segments of all variables remain visible for the complete time-series. In a user study, Javed et al. (2010) compared line plots (↪ p. 153), silhouette graphs (↪ p. 175), horizon graphs (↪ p. 157), and braided graphs along the three tasks of determining local maxima, comparing global slopes, as well as locating and comparing values at specific time points. Besides the visualization type, the number of displayed time-series, and the height of the representation was varied. Interestingly, the type of visualization was not found to have a significant effect on task correctness in all conditions. However, subjects using line plots and braided graphs were significantly faster when searching for local maxima. For value and slope comparison tasks this was not the case. In general, higher numbers of time-series caused decreased correctness and increased completion time. Decreasing display space had a negative effect on correctness but little impact on completion time.

References

Javed, W., McDonnel, B., and Elmqvist, N. (2010). Graphical Perception of Multiple Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–34.

ThemeRiver

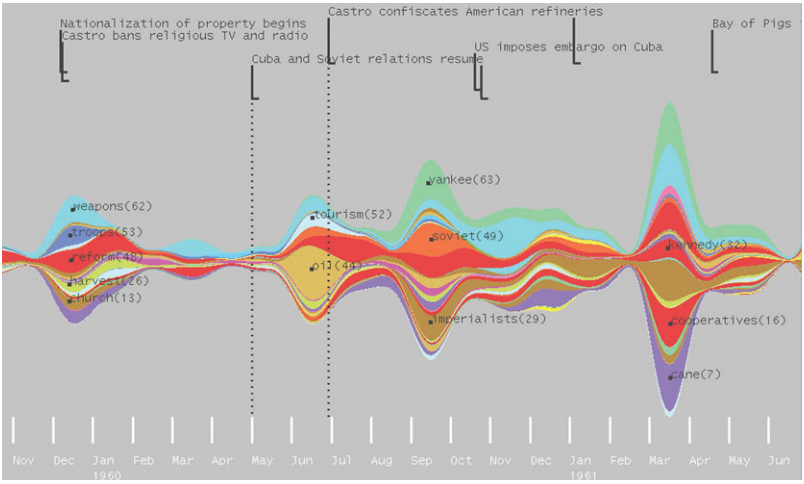


Fig. 7.46: The ThemeRiver representation uses the metaphor of a river that flows through time. Colored currents within the river reflect thematic changes in a document collection, where the width of a current represents the relevance of its associated theme.
Source: Havre et al. (2002), © 2002 IEEE. Used with permission.

The ThemeRiver technique developed by Havre et al. (2000) represents changes of news topics in the media. Each topic is displayed as a colored current whose width varies continuously as it flows through time. The overall image is a river that comprises all of the topics considered. The ThemeRiver provides an overview of the topics that were important at certain points in time. Hence, the main focus is directed towards establishing a picture of an easy to follow evolution over time using interpolation and approximation. Moreover, ThemeRiver representations can be annotated, e.g., with related major historical events, and raw data points with exact values can be shown. Even though the ThemeRiver was originally invented to visualize thematic changes in document collections, it is also suited to represent other multivariate, quantitative data. Because perception of data differs depending on where in the river individual variables are shown, it is important to provide interaction techniques to allow users to rearrange the horizontal position of variables.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

References

Havre, S., Hetzler, E., and Nowell, L. (2000). ThemeRiver: Visualizing Theme Changes Over Time. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 115–124, Los Alamitos, CA, USA. IEEE Computer Society.

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.

3D ThemeRiver

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

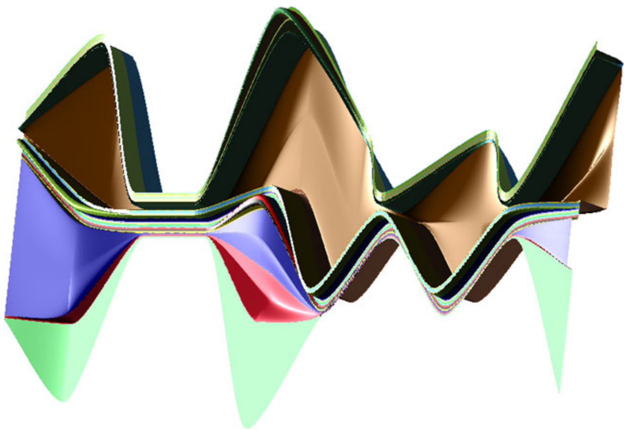


Fig. 7.47: Distinctly colored currents form the overall shape of the 3D ThemeRiver. The width, and additionally the height of currents is varied to visualize time-oriented data. In this figure, width encodes the overall distribution of 17 clusters of aerosol data and height indicates the incidence of zinc.
Source: Imrich et al. (2003), © 2003 IEEE. Used with permission.

Imrich et al. (2003) propose a 3D variant of the ThemeRiver technique (↪ p. 197). The 3D approach inherits the basic visual design from its 2D counterpart: multiple time-oriented variables are encoded to the widths of individually colored currents that form a river flowing through time along a horizontal time-axis. In the 2D variant, only one data variable can be visualized per current, namely by varying the current's width. Imrich et al.'s extension addresses this limitation. By extending the design to the third dimension it is possible to use an additional visual encoding: the height (in 3D) of a current can be varied to encode further information. This design is particularly suited to visualizing ternary covariate trends in the data. Imrich et al. conducted user tests to evaluate the usefulness of the 3D encoding, and indeed got positive results that indicate that the 3D variant has advantages over the 2D variant. Specifically, the availability of appropriate interactive 3D navigation tools is highlighted as an important factor contributing to the success of the 3D ThemeRiver.

References

Imrich, P., Mueller, K., Imre, D., Zelenyuk, D., and Zhu, W. (2003). Interactive Poster: 3D ThemeRiver. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA. IEEE Computer Society.

Stacked Graphs

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

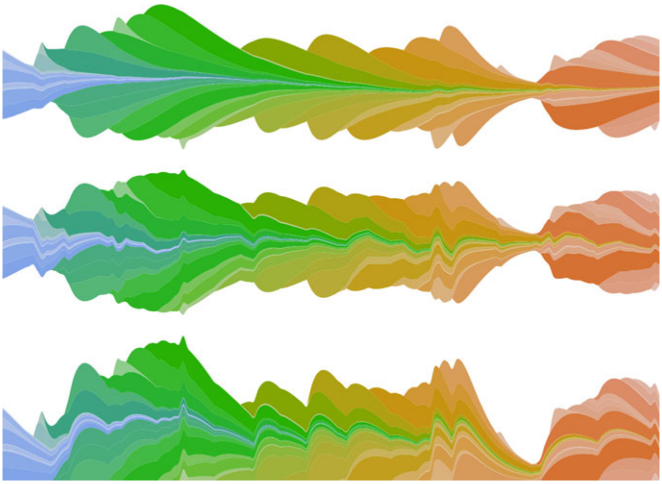


Fig. 7.48: Multivariate time-series visualized as stacked graphs with different designs: Streamgraph design (top), ThemeRiver layout (center), and traditional stacking (bottom).
Source: Generated with the streamgraph_generator code base.

Stacking multiple graphs on top of each other is a suitable approach to visualizing multiple time-dependent variables (see Harris, 1999). Elaborate variants of stacked graphs have been investigated in detail by Byron and Wattenberg (2008). To visualize the evolution of an individual variable, data values are encoded to the height of a so-called layer that extends along the horizontal time axis. A special color map is applied to visualize additional data variables and to make individual layers distinguishable. Several layers are then stacked on top of each other, effectively creating an overall graph that represents the visual sum of the entire dataset. Layout and sorting of layers can be done in various ways, resulting in quite different designs such as the Streamgraph design, the ThemeRiver layout (↪ p. 197), or traditional layer area graphs (↪ p. 195). The Streamgraph design (Figure 7.48, top) is notable because it received quite positive feedback when it appeared on the New York Times web site as a visual representation of box office revenues. In that version, individual layers were also outfitted with text labels.

References

Byron, L. and Wattenberg, M. (2008). Stacked Graphs – Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252.

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

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

TimeWheel

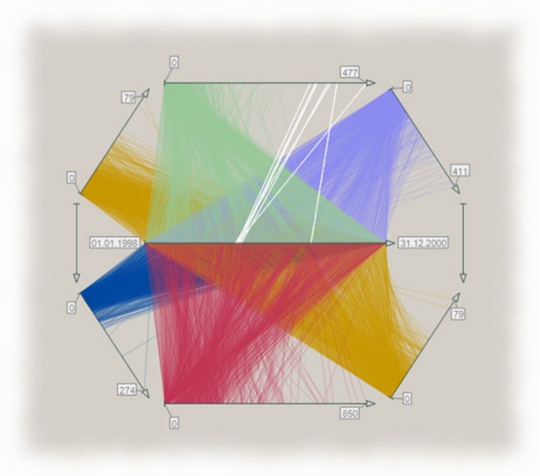


Fig. 7.49: The TimeWheel's central axis represents time. The axes in the periphery represent time-dependent variables; here we see the number of cases for eight diagnoses. Days with particularly high numbers of influenza cases are highlighted.
Source: Generated with the VisAxes software.

Tominski et al. (2004) describe the TimeWheel as a technique for visualizing multiple time-dependent variables. The TimeWheel consists of a single time axis and multiple data axes for the data variables. The time axis is placed in the center of the display to emphasize the temporal character of the data. The data axes are associated with individual colors and are arranged circularly around the time axis. In order to visualize data, lines emanate from the time axis to each of the data axes to establish a visual connection between points in time and associated data values. These lines form visual patterns that allow users to identify positive or negative correlations with the time axis, trends, and outliers. Such patterns can be best discerned for those data axes that are parallel to the time axis. To bring data axes of interest into this focus, users can rotate the TimeWheel. Focused data axes are further emphasized by stretching them, effectively providing them with more drawing space. Data axes that are perpendicular to the time axis are more difficult to interpret and are, therefore, attenuated using color fading and shrinking. Interactive exploration, including navigation in time, is supported through different types of interactive axes.

References

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247, New York, NY, USA. ACM Press.

MultiComb

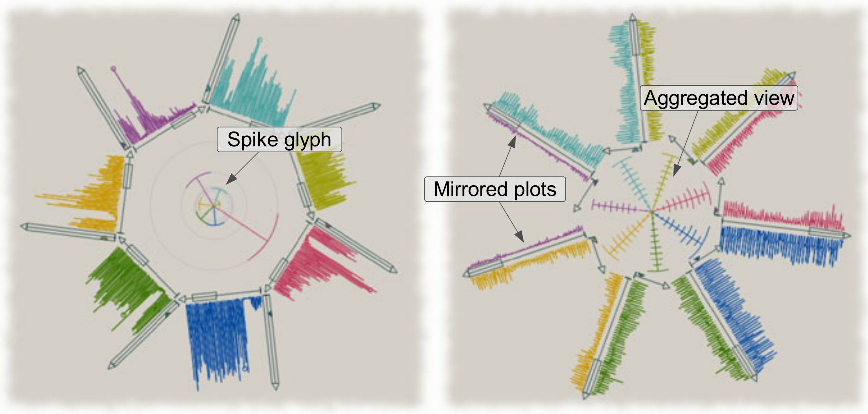


Fig. 7.50: Two MultiComb representations visualize seven time-dependent variables. In the left MultiComb, line plots are arranged around the display center, in the right one, they extend outwards. The very center of the MultiCombs can be used to display additional information via a spike glyph or an aggregated view.
Source: Generated with the VisAxes software.

Line plots (↪ p. 153) are expressive visual representations for univariate data. The rationale behind the MultiComb visualization is to utilize this expressiveness for representing multiple time-dependent variables. Tominski et al. (2004) describe the MultiComb as a visual representation that consists of multiple radially arranged line plots. Two alternative designs exist: time axes are arranged around the display center (see Figure 7.50, left) or time axes extend outwards from the MultiComb's center (see Figure 7.50, right). In the latter case, optional mirror plots duplicate plots of neighbor variables to ease visual comparison. To maintain a certain aspect ratio for the separate plots, the axes do not start in the very center of the MultiComb. The screen space in the center can therefore be used to provide additional views: a spike glyph can be shown to allow a detailed comparison of data values for a selected time point, or an aggregated view might display the history of a temporal data stream in an aggregated fashion. Various possibilities for interaction allow users to browse in time, to zoom into details of the time axes, as well as to add, remove, and reorder plots, and to rotate the MultiComb.

References

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

data
frame of reference: **abstract**
variables: **multivariate**

VIE-VISU

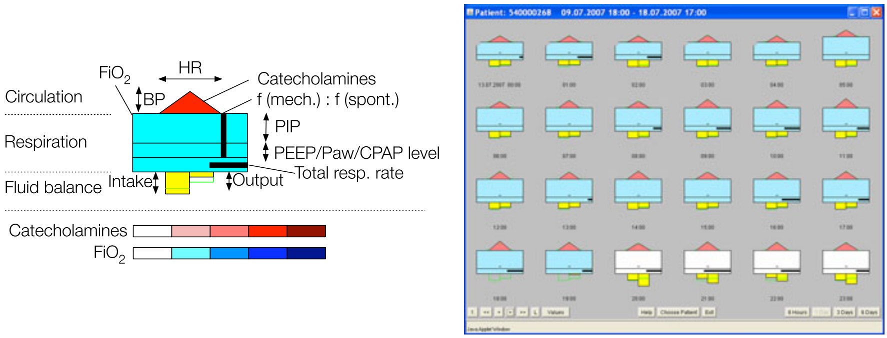


Fig. 7.51: VIE-VISU encodes fifteen health-related patient parameters to different visual attributes of a glyph (left). The example on the right shows neonatal patient information on an hourly basis for the course of the day.
Source: Left: Adapted from Horn et al. (2001). Right: Image courtesy of Werner Horn.

time
arrangement: **linear**
time primitives: **instant**

Paper-based analysis of patient records is hard to conduct because many parameters are involved and an overall assessment of the patient's situation is difficult. Therefore, Horn et al. (2001) developed VIE-VISU, an interactive glyph-based visualization technique for time-oriented patient records. The glyph consists of three parts that represent circulation, respiration, and fluid balance parameters. All in all, 15 parameters are visualized using different visual attributes (i.e., length, width, color) as illustrated in the left part of Figure 7.51. For example, the circulation parameter heart rate (HR) is encoded to the width of the triangle on top of the glyph and the triangle's color encodes catecholamines (color legend is given at the bottom). Each glyph represents a one hour period and 24 glyphs are combined in a small multiples display (↪ p. 236) as shown in the right part of Figure 7.51. Interaction controls support navigation in time and switching to different periods for the small multiples view. VIE-VISU helps users to combine different measurements, maintain their relationships, show their development over time, and make specific, possibly life threatening situations easy to spot.

vis
mapping: **static**
dimensionality: **2D**

References

Horn, W., Popow, C., and Unterasinger, L. (2001). Support for Fast Comprehension of ICU Data: Visualization using Metaphor Graphics. *Methods of Information in Medicine*, 40(5):421–424.

Timeline Trees

data
frame of reference: **abstract**
variables: **multivariate**

Day	**Market basket and money spent**
Mon:	milk \$1, bananas \$3
Tue:	cheese \$1, apples \$3
Wed:	milk \$1, bananas \$1, grapes \$2
Thu:	milk \$1
Fri:	milk \$1, cheese \$3

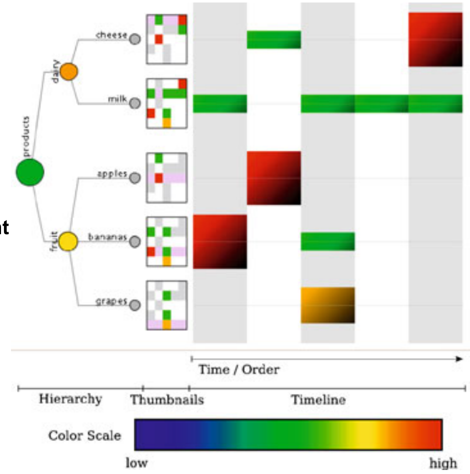


Fig. 7.52: A smaller set of products in a market basket is visualized using timeline trees. One can see that milk is bought regularly (green boxes for all but one day), and that cheese, apples, and bananas are more expensive (higher red-colored boxes).
Source: Image courtesy of Michael Burch.

time
arrangement: **linear**
time primitives: **instant**

Data that describe items which are related to each other are quite common. An example of such data are transactions in on-line shopping systems where products being bought together are considered to be related. Burch et al. (2008) visualize temporal sequences of transactions by means of so-called timeline trees. The visual representation consists of three parts: a display of an information hierarchy, a timeline representation of temporal sequences, and thumbnail pictures. The information hierarchy is a static hierarchical categorization of data items (e.g., a system of product groups), where groups can be expanded or collapsed interactively to view the data at different levels of detail. The timeline view shows multiple sequences of boxes for the current level of detail, where color and box size are used to encode data values (e.g., product price) of an item (or group) at a particular point in time. Thumbnails for each leaf of the information hierarchy show an overview of transactions masked by the corresponding leaf node. Enhanced with several interaction facilities, timeline trees help users to understand trends in the data and to find relations between different levels of abstractions (e.g., different product groups, or product groups and specific products).

vis
mapping: **static**
dimensionality: **2D**

References

Burch, M., Beck, F., and Diehl, S. (2008). Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 75–82, New York, NY, USA. ACM Press.

Pixel-Oriented Network Visualization

data
frame of reference: **abstract**
variables: **multivariate**

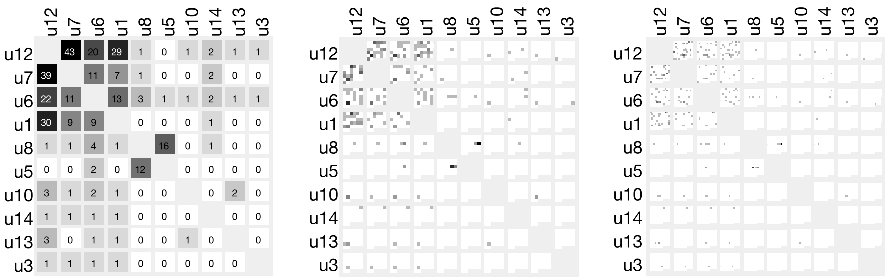


Fig. 7.53: Wiki collaboration patterns. Left: adjacency matrix showing users as rows and columns and collaboration intensity by color brightness of cells that connect two users (darker means more collaboration); center: pixel-oriented view where collaboration dynamics are shown in a 6x6 pixel array laid out row by row and each pixel represents a four week period; right: more fine-grained configuration that shows weekly steps.
Source: Images courtesy of Klaus Stein.

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

Social networks consist of actors and relationships between them. Unlike most static node-link representations of graph-like structures would suggest, these networks are dynamically changing over time. The two most common forms of visualizing time-varying networks are applying animation to node-link diagrams or applying the concept of small multiples (↪ p. 236) by showing snapshots of different points in time. An alternative display is suggested by Stein et al. (2010). They developed a pixel-oriented visualization of networks (PONV) that reveals interaction patterns between actors by integrating pixel-based representations (↪ p. 180) within the cells of an adjacency matrix. An adjacency matrix can be represented visually as a matrix table whose rows and columns represent the nodes of the network. The table cell at the intersection of a particular column and row visualizes information about the relationship between the corresponding nodes. Figure 7.53 (left) shows an example of an adjacency matrix where the darkness of a cell represents the collaboration intensity of two individuals (the value 0 and a white cell background indicates that there is no relationship between two individuals). In order to show the temporal evolution of the relationships, Figure 7.53 (center) uses an alternative representation. Now each cell contains a 6x6 pixel glyph, where each pixel represents the aggregated collaboration intensity of a four-week period. The user can interactively control the visualization, including the color scale, the pixel pattern arrangement, and the time period to be covered by each pixel (e.g., daily values as in Figure 7.53, right).

References

Stein, K., Wegener, R., and Schlieder, C. (2010). Pixel-Oriented Visualization of Change in Social Networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 233–240, Los Alamitos, CA, USA. IEEE Computer Society.

CiteSpace II

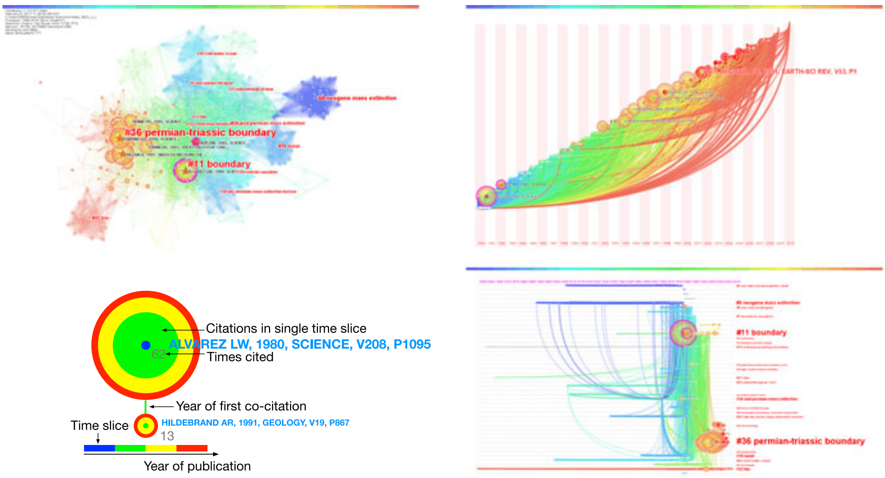


Fig. 7.54: A network of 750 most cited articles on mass extinction (1980–2010). Top-left: cluster view; bottom-left: legend – node size reflects overall amount of citations and colored rings show citations per time slice; top-right: time-zone view; bottom-right: timeline view.
Source: Images courtesy of Chaomei Chen.

CiteSpace II by Chen (2006) is a system that supports visual exploration of bibliographic databases. It combines rich analytic capabilities to analyze emerging trends in a knowledge domain with interactive visualization of co-citation networks. Three complementary views are provided for the visual representation: a cluster view, a time-zone view, and a timeline view. The cluster view represents a network as a node-link diagram using a force-directed layout. Node size shows how often an article or cluster was cited overall and citation tree rings of a node display the citation history from the center outward. The color of a ring represents a time period and its thickness is proportional to the number of citations in this period. Colors of links represent the time slice of the first co-citation. The time-zone view displays a network by arranging its nodes along vertical strips representing time zones using a modified spring-embedder layout that controls only the vertical positions of nodes freely. In the timeline view, time is mapped to the horizontal position and clusters are arranged along horizontal lines. Users can adjust a complex set of parameters to control the analysis process as well as interact and manipulate the visualization of a knowledge domain. CiteSpace II also provides clustering and labeling functions to help the user to interpret various structural and temporal patterns.

References

Chen, C. (2006). CiteSpace II: Detecting and Visualizing Emerging Trends and Transient Patterns in Scientific Literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

history flow

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

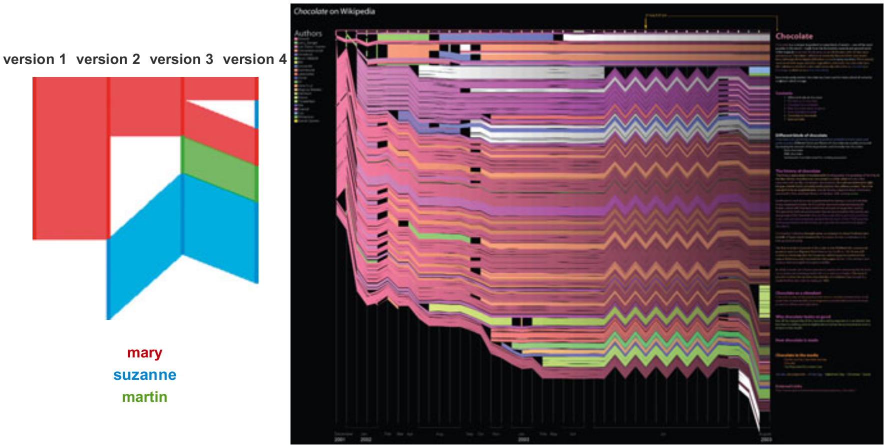


Fig. 7.55: The history flow shows vertical revision lines, one for each revision, where colored sections reflect the different authors of a document as schematically depicted on the left. This method is applied to the visualization of the Wikipedia entry on chocolate as shown on the right. *Source: Images courtesy of Fernanda B. Viégas.*

Viégas et al. (2004) designed history flow to be an exploratory wiki article analysis tool for finding author collaboration patterns, showing relations between document versions, revealing patterns of cooperation and conflict, as well as making broad trends immediately visible. The basis for the representation are so-called revision lines. These top-aligned, vertical lines are displayed for every version of a document. The length of revision lines is proportional to the document length. Individual sections of a revision line are colored differently to visualize which authors worked on which parts of a document. The sections associated with a particular author are visually connected from one revision to the next. One can discern stable sections and splits of sections. Gaps in connections clearly indicate deletions and insertions. Two different layouts can be used for spacing revision lines: uniform spacing (space by occurrence / event-based) or spacing according to time (space by date / time-based). The first layout shows each document change equally spaced without showing time intervals between versions proportionally. The second alternative additionally gives information about the exact timing.

References

Viégas, F. B., Wattenberg, M., and Dave, K. (2004). Studying Cooperation and Conflict between Authors with history flow Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 575–582, New York, NY, USA. ACM Press.

PeopleGarden

Fig. 7.56: This PeopleGarden shows about 1200 posts entered to a discussion board during a period of two months. Users are represented by flowers whose petals, in turn, represent individual messages posted by a user.
Source: Xiong and Donath (1999), © 1999 ACM. Used with permission.

Xiong and Donath (1999) developed PeopleGarden as a graphical representation of users' interaction histories in discussion groups. PeopleGarden visualizes data about users and messages posted to an online interaction environment. It integrates information on the time of posting, amount of response, and whether a post starts a new conversation. For intuitive understanding, PeopleGarden uses the metaphor of a garden of flowers. The garden represents the whole environment and flowers represent individual users within the environment. The petals of a flower stand for the messages posted by a user, the time of posting is mapped to the ordering and saturation of the petals, the amount of response is represented by circles that are stacked on top of petals, and color is used to depict whether a post starts a new conversation. Furthermore, the height of the flower gives information about how long a specific user has been a member of the discussion group. Using these visual representations, one can easily spot dominant voices, long time participants, or very active groups.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

References

Xiong, R. and Donath, J. (1999). PeopleGarden: Creating Data Portraits for Users. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 37–44, New York, NY, USA. ACM Press.

PostHistory

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

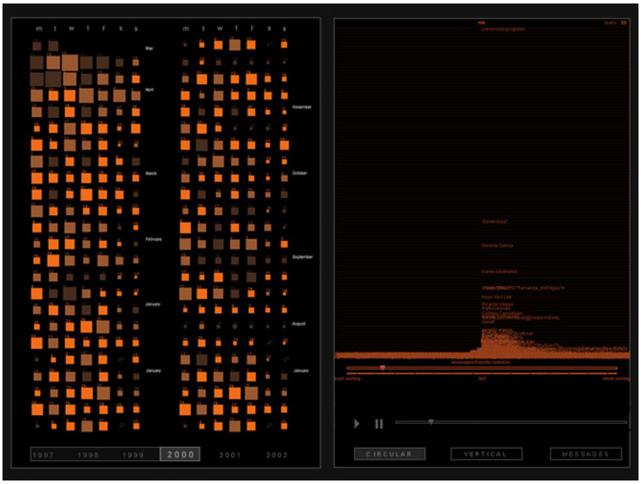


Fig. 7.57: The calendar panel on the left shows e-mail activity on a daily basis, where the number of emails and their average directedness are mapped to box size and color, respectively. The contacts panel on the right displays the names of people who sent messages to the user.
Source: Image courtesy of Fernanda B. Viégas.

Viégas et al. (2004) developed PostHistory with the goal of visually uncovering different patterns of e-mail activity (e.g., social networks, e-mail exchange rhythms) and the role of time in these patterns. PostHistory is user-centric and focuses on a single user's direct interactions with other people through e-mail. The social patterns are derived from analyzing e-mail header information. So, not the content of messages, but the tracked traffic is used as the basis for the analysis of people's e-mail conversations over time. Basically, the user interface visualizes a full year of e-mail activity and is divided into two main panels: a calendar panel on the left and a contacts panel on the right. The calendar panel shows the intensity of e-mail activity on a daily basis whereas a square represents a single day and each row of squares represents a week. The size of a square is determined by the quantity of e-mail received on that day and its color represents the average directedness of messages, i.e., whether a mail was received via TO, CC, or BCC. The brighter the color, the more directed the messages are that are present on that day. The contacts panel is used for displaying the names of people who sent messages to the user.

References

Viégas, F., Boyd, D., Nguyen, D., Potter, J., and Donath, J. (2004). Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. In *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, pages 109–118, Los Alamitos, CA, USA. IEEE Computer Society.

MOSAN

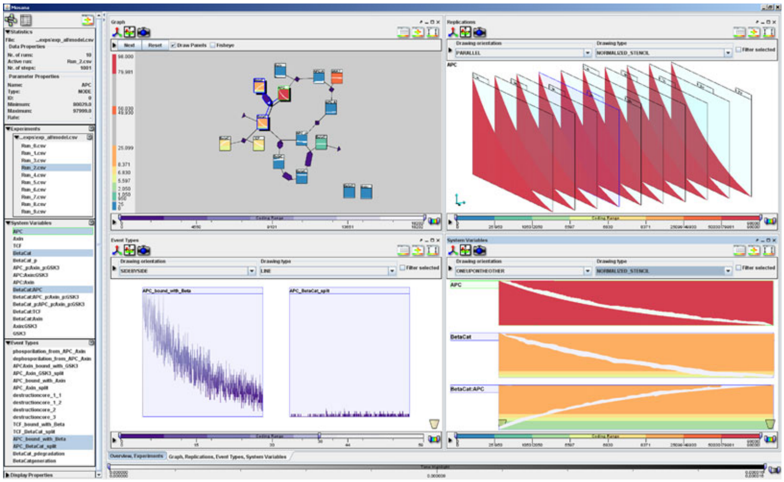


Fig. 7.58: The top-left view shows a simulated reaction network. An overview of the time-dependent simulation data is given by the small line plots in the boxes of the network. Three coordinated linked views are provided for the comparison of simulation runs and variables.
Source: Image courtesy of Andrea Unger.

MOSAN is a tool for visualizing multivariate time-oriented data that result from simulation of reaction networks. Due to the stochastic multi-run simulation, each variable comprises multiple time-series. In order to facilitate the understanding of the complex dependencies in the data it is necessary to jointly visualize structural information and stochastic simulation data together. To this end, Unger and Schumann (2009) combine different views within a single interactive interface. In an overview, time-oriented data are shown along with the structural relations among the variables in the reaction network. The structural relations are shown by a graph layout, where boxes correspond to variables, and simulation data are visualized by small line plots within the boxes (see Figure 7.58, top-left). The small line plots provide a highly aggregated view of the stochastic simulation data, thus focusing on the communication of the general temporal trends. Furthermore, advanced color-coding is applied to the plots to support the comparison of heterogeneous value ranges among variables. In addition to the overview, coordinated linked views support the inspection of individual time-series of the same variable (see Figure 7.58, top-right) as well as the detailed inspection and comparison of temporal developments of different variables selected from the overview (see Figure 7.58, bottom).

References

Unger, A. and Schumann, H. (2009). Visual Support for the Understanding of Simulation Processes. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

Data Tube Technique

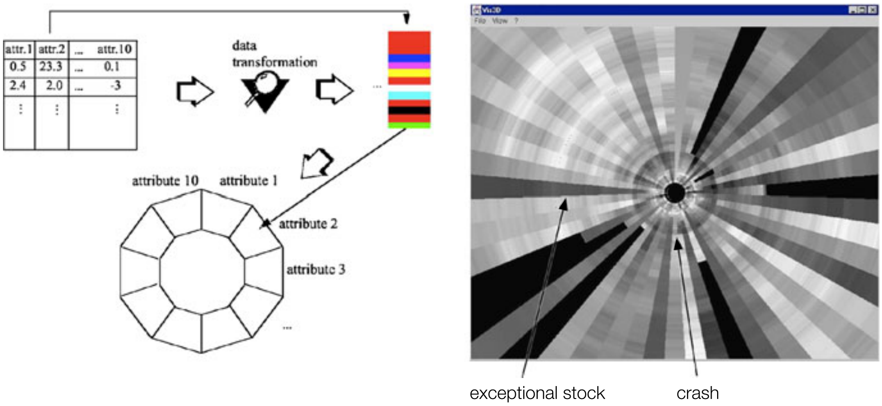


Fig. 7.59: Data are mapped onto the inside of a 3D tube using a tabular layout. Each slice of the tube represents an instant and each cell represents a data parameter by color. Left: visual mapping schema; right: exploring 50 different stocks.
Source: Images courtesy of Mihael Ankerst.

In the data tube technique by Ankerst (2001) multiple time-oriented variables are mapped to bands that follow the inside of a 3D tube (see Figure 7.59, left). Each slice of the tube represents an instant and each cell represents a data value by color. The tube is viewed from above and time is flowing to or from the center of the tube. The user is able to explore the data by interactively moving through the 3D tube. Because of the 3D perspective distortion, cells that are further away appear to be smaller in size, much like in a focus+context display. As a result of this, the number of displayed parameters and the number of displayable records can be quite large. Later, Ankerst et al. (2008) also developed a comprehensive temporal data mining architecture called DataJewel that is closely integrated with pixel-oriented visualization techniques.

References

Ankerst, M. (2001). Visual Data Mining with Pixel-oriented Visualization Techniques. In *Proceedings of ACM SIGKDD Workshop on Visual Data Mining*, New York, NY, USA. ACM Press.

Ankerst, M., Kao, A., Tjoelker, R., and Wang, C. (2008). DataJewel: Integrating Visualization with Temporal Data Mining. In Simoff, S., Böhlen, M., and Mazeika, A., editors, *Visual Data Mining*, volume 4404 of *Lecture Notes in Computer Science*, pages 312–330. Springer, Berlin, Germany.

Kiviat Tube

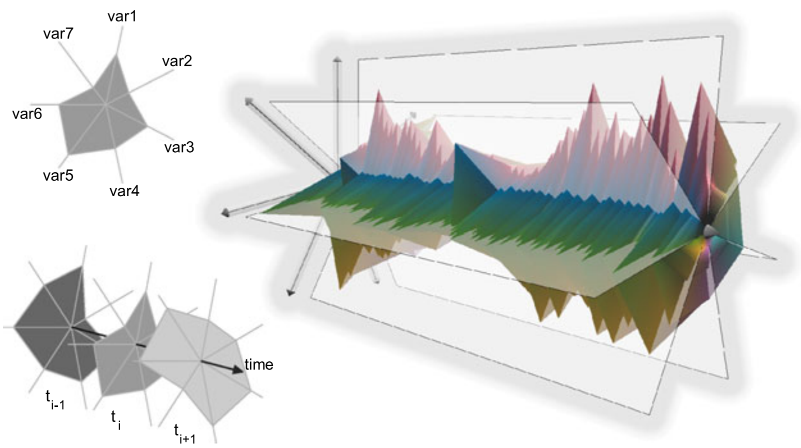


Fig. 7.60: Construction of a three-dimensional Kiviat tube representing seven time-dependent variables. Peaks and valleys indicate ups and downs in the evolution of the data over time. Wings assist in associating features of the Kiviat tube to particular variables in the data.
Source: Generated the VisAxes3D tool.

The Kiviat tube by Tominski et al. (2005) visualizes multiple time-dependent variables. The construction of a Kiviat tube is as simple as stacking multiple Kiviat graphs (see Kolence and Kiviat, 1973) along a shared time axis. Each Kiviat graph represents the data for multiple variables for a specific point in time. But instead of drawing individual Kiviat graphs, a three-dimensional surface is constructed. This way, multiple, otherwise separated time points are combined to form a single 3D body that represents the dataset as a whole. The spatial characteristics of a Kiviat tube can be recognized easily, as it allows users to identify peaks or valleys in the data over time. Additional semitransparent wings assist in relating identified patterns to particular variables. Common interaction methods can be used for zooming and rotation around arbitrary axes. Rotation specifically around the time axis enables users to quickly access variables on all sides of the Kiviat tube. Interactive axes allow users to navigate back and forth in time to visit different intervals of a possibly large time-series.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Kolence, K. W. and Kiviat, P. J. (1973). Software Unit Profiles & Kiviat Figures. *SIGMETRICS Performance Evaluation Review*, 2:2–12.

Tominski, C., Abello, J., and Schumann, H. (2005). Interactive Poster: 3D Axes-Based Visualizations for Time Series Data. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 49–50, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

Temporal Star

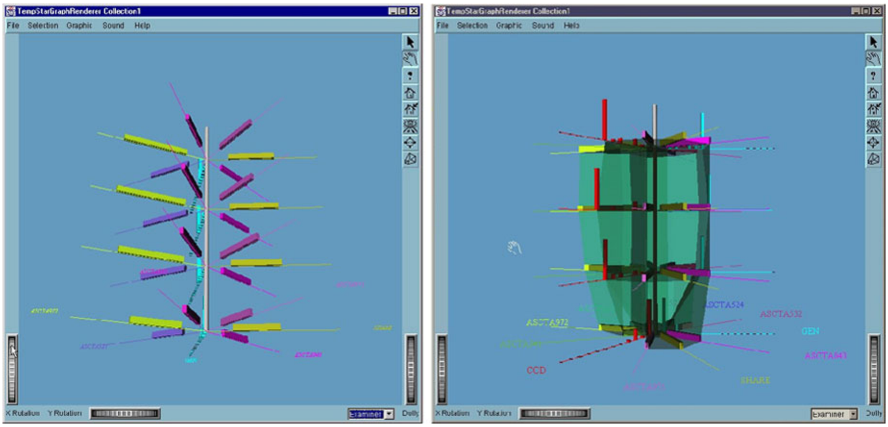


Fig. 7.61: 3D representation of circular column graphs that are arranged in a row to represent each time step. A transparent veil can be displayed to enhance the perception of the dataset's evolution. *Source: Images courtesy of Monique Noirhomme.*

The temporal star technique by Noirhomme-Fraiture (2002) visualizes multivariate data structures in 3D. For each point in time, a circular column graph is drawn that represents each variable's value as a bar length in a circular arrangement. These graphs are aligned in a row to represent the development of the dataset over time (see Figure 7.61, left). A unique color is assigned to each variable to aid recognition of variables across time. Moreover, a transparent veil can be displayed to enhance the perception of the dataset's evolution as a whole (see Figure 7.61, right). The concept used is similar to that of the Kiviat tube (↪ p. 211), which uses Kiviat graphs instead of circular column graphs. In the temporal star technique, difference plots are also integrated, showing the relative differences between variables rather than absolute values. The rendering parameters, the shown time intervals, and the configuration of the axes can be adjusted interactively. Furthermore, the temporal star technique is integrated with a data warehousing application that provides rich data manipulation features.

References

Noirhomme-Fraiture, M. (2002). Visualization of Large Data Sets: The Zoom Star Solution. *Journal of Symbolic Data Analysis*, 0(0).

Time-tunnel

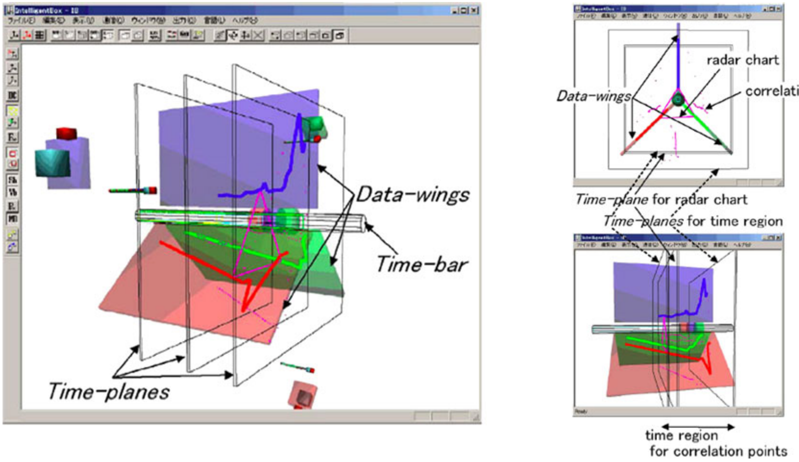


Fig. 7.62: Left: individual plots are put onto semi-transparent planes (data-wings) that are positioned around a central time-bar. Right: selected time intervals can be viewed as radar charts.
Source: Akaishi and Okada (2004), © 2004 IEEE. Used with permission.

Akaishi and Okada (2004) developed time-tunnel as a data analysis technique for visualizing a number of time-series plots in a 3D virtual space. The individual plots are put onto semi-transparent planes (data-wings) that are positioned around a central time-bar in a fan-like manner (see Figure 7.62, left). In the example above, line plots are used for visualizing data but any other linear time visualization might also be used. Multiple planes can be overlapped and compared due to their transparency. Furthermore, single time slices or selected time intervals can be viewed as radar charts (see Figure 7.62, right) and provide a multivariate point of view. Additionally, not only time-series, but any kind of information can be put onto a plane, making it a multimedia presentation tool.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Akaishi, M. and Okada, Y. (2004). Time-tunnel : Visual Analysis Tool for Time-series Numerical Data and its Aspects as Multimedia Presentation Tool. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 456–461, Los Alamitos, CA, USA. IEEE Computer Society.

Parallel Glyphs

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

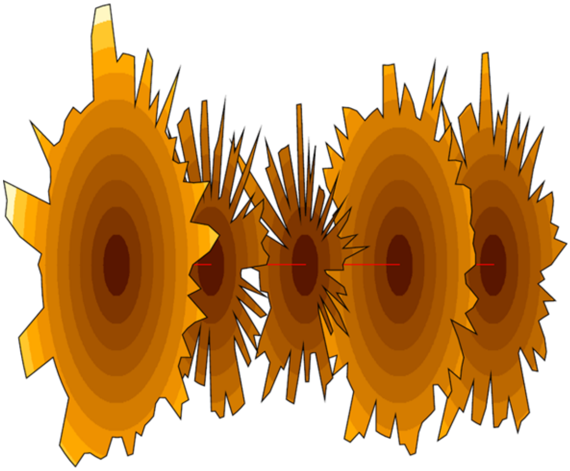


Fig. 7.63: Parallel glyphs are used to visualize five different variables over time. Each radial glyph shows a single variable over time whereas each point on the outside of a glyph corresponds to a data value measured at a specific point in time. Differently colored rings assist in comparing data values.
Source: Fanea et al. (2005), © 2005 IEEE. Used with permission.

Multivariate time-series can be visualized as parallel glyphs. Fanea et al. (2005) synthesized this technique as a combination of parallel coordinates and star glyphs. The visualization uses multiple star glyphs, each of which consists of as many radially arranged spikes as there are time points in the data. The length of a spike corresponds to the data value measured at the spike's associated time point. The tips of subsequent spikes are connected via a polyline, effectively creating a polygonal shape, which visualizes the data of one variable in a radial fashion. As an alternative representation to the spikes, the shape can be filled with differently colored rings to make the data values easier to compare. Multiple such star glyphs are generated, one for each variable of the dataset. These glyphs are then arranged in a three-dimensional space along a shared axis in a parallel fashion. To assist users in identifying correlations among variables, polylines can be used to connect the same time step along the glyphs. In order to avoid clutter, it is possible to restrict this feature to a user-selected number of time steps. The technique offers various ways of manipulating the display, including switching the role of variables and data records, rotation and zooming in the 3D presentation space, and adjustment of colors.

References

Fanea, E., Carpendale, M. S. T., and Isenberg, T. (2005). An Interactive 3D Integration of Parallel Coordinates and Star Glyphs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 149–156, Los Alamitos, CA, USA. IEEE Computer Society.

Worm Plots

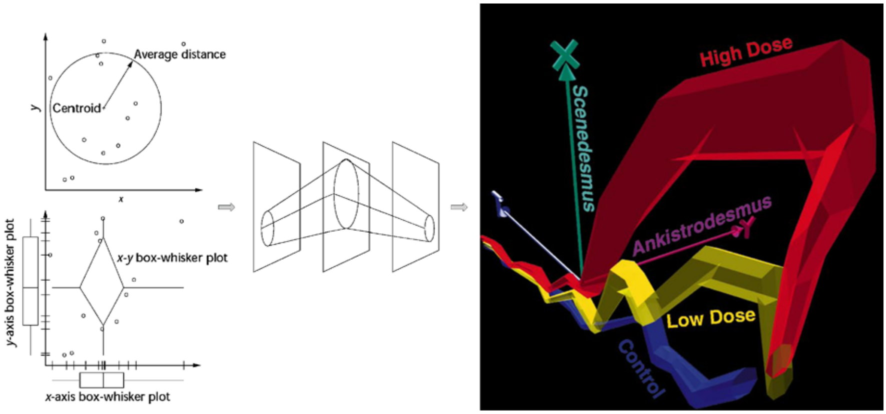


Fig. 7.64: Worm plots are generated by creating visual abstractions of point groups at multiple time steps and by assembling a 3D surface that resembles a worm. The worm plot on the right shows three groups (control, high dose, low dose) of data points of toxicology experiments plotted against the two variables Scenedesmus and Ankistrodesmus.
Source: Matthews and Roze (1997), © 1997 IEEE. Used with permission.

Worm plots have been developed by Matthews and Roze (1997) to help scientists gain qualitative insights into the temporal development of groups of points in scatter plots. The initial step necessary to construct a worm plot is generating a visual abstraction of multiple points. One way to do this is to compute the centroid of a group of points and the average distance of points to the centroid. The visual abstraction is then a circle with a radius equal to the average distance and located at the centroid. Alternatively, a 2D generalization of box-whisker plots can be used to form a diamond-shaped visual abstraction. Such abstractions are computed for each time step (i.e., each scatter plot). Subsequently, a three-dimensional surface (worm) is assembled from the visual abstractions of each group. This procedure is illustrated in the left part of Figure 7.64. Presented in an interactively manipulatable virtual world, worm plots allow users not only to see where in the variable space point groups are located, but also to discern the compactness of point groups, and to understand the development of these characteristics over time.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Matthews, G. and Roze, M. (1997). Worm Plots. *IEEE Computer Graphics and Applications*, 17(6):17–20.

Software Evolution Analysis

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

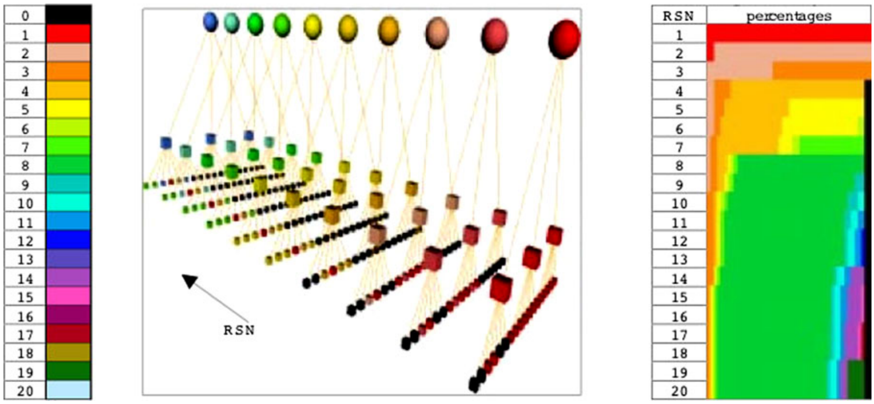


Fig. 7.65: 3D visualization to analyze software systems and product families. Left: color legend for different versions; center: hierarchical decomposition by modules, packages, and files (color represents current version); right: individual file where each row represents a single version colored by percentage of code originating from a particular (previous) version.
Source: Gall et al. (1999), © 1999 IEEE. Used with permission.

The software evolution analysis technique by Gall et al. (1999) uses 3D visualization to analyze software systems or product families respectively. The information is decomposed hierarchically into modules, packages, and files or similar concepts. This hierarchy is depicted as a three dimensional tree structure in which the leaf nodes represent individual files. Multiple such trees are aligned in layers in the 3D space, with one layer for each revision of the software. Color is used to distinguish different versions and to show changes over time. Furthermore, individual files might be inspected in more detail to explore the evolution of changes over time using proportionally colored version lines (see Figure 7.65, right). This way, patterns are formed that can be used to identify, for example, stable parts of a system, frequently changed parts, similarities, and more.

References

Gall, H., Jazayeri, M., and Riva, C. (1999). Visualizing Software Release Histories: The Use of Color and Third Dimension. In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 99–108, Los Alamitos, CA, USA. IEEE Computer Society.

InfoBUG

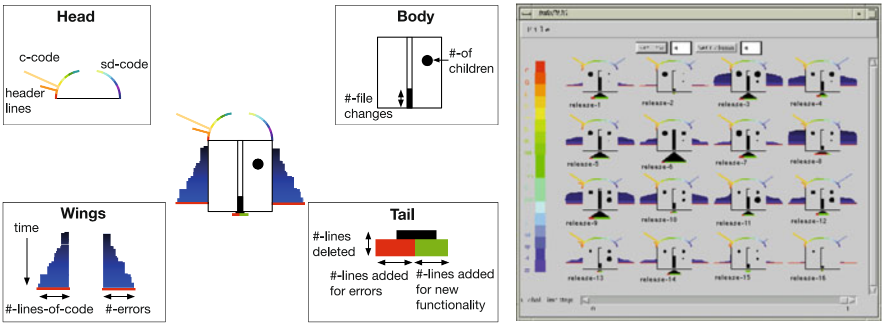


Fig. 7.66: The InfoBUG encodes data about software to the wings, head, tail, and body of a glyph (left). The wings show lines of code and number of errors over time, whereas body, head, and tail show further data for a selected time point. A small multiples view can be used to compare different software releases over time (right).
Source: Chuah and Eick (1998), © 1998 IEEE. Used with permission.

Chuah and Eick (1998) developed InfoBUG for visualizing changes in software projects. The InfoBUG is an information-rich graphic that combines a multitude of different heterogeneous data values. The glyph resembles an insect with wings, head, tail, and body. The different parts of the glyph are used to represent four different classes of information about software projects (see Figure 7.66, left): code lines and errors (wings), types of code (head), added and deleted lines of code (tail), and number of file changes and children (body). The wings represent the lines of code (left wing) and the number of errors (right wing) over time as vertical silhouette graphs. While the wings show data over time, the other parts of the glyph show only the data of a user-selected time point, which is indicated as a red line at the wings. Antennas on the InfoBUG's head represent different types of code, where color indicates the type of code and the relative sizes of different types are encoded by antenna length. The bug's tail represents the number of deleted and added lines. Finally, the InfoBUG's body visualizes information about the number of altered files via a bar in the middle of the body and the number of child objects via filled circles. Small multiples (↪ p. 236) can be used to compare the different releases of a software product over time (see Figure 7.66, right). Furthermore, the representation can be animated to follow the course of time.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static, dynamic**
dimensionality: **2D**

References

Chuah, M. C. and Eick, S. G. (1998). Information Rich Glyphs for Software Management Data. *IEEE Computer Graphics and Applications*, 18(4):24–29.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static, dynamic**
dimensionality: **2D**

Gravi++

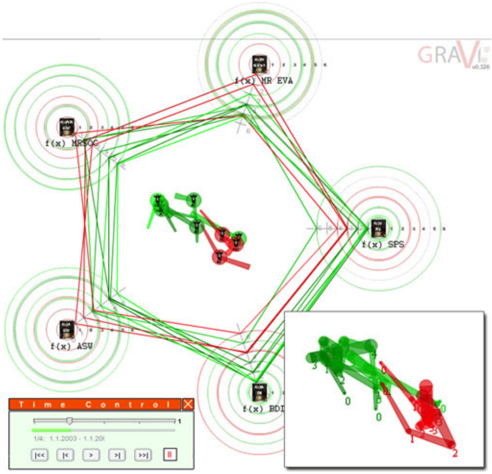


Fig. 7.67: Patient icons in the middle of the display are positioned relative to the surrounding parameters (in this case items of a questionnaire) following a spring-based model. Individual answers to questionnaire items are shown as concentric rings and star glyphs that show a set of answers as polygonal line might be displayed. The user can step through time manually or can use animation, which can be steered via the control panel on the lower left. Furthermore, traces might be displayed that convey information about the evolution of values over time as shown in detail on the lower right.
Source: Image courtesy of Klaus Hinum.

Hinum et al. (2005) designed Gravi++ to find predictors for the treatment planning of anorexic girls. It represents patients and data gathered from questionnaires during treatment over the course of several weeks or months. Patients are represented by icons that are laid according to a spring-based model relative to the surrounding icons that represent items of a questionnaire. This leads to the formation of clusters of persons who gave similar answers. To visualize the changing values over time, animation is used. The position of each person's icon changes over time, making it possible to trace, compare, and analyze the changing values. Alternatively, the change over time can be represented by traces. The size and path of the person's icon is shown corresponding to all time steps or only to a restricted subset like the previous and the next time step. To visualize the exact values of each question, rings around the question's icon can be drawn and star glyphs might be shown.

References

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.

CircleView

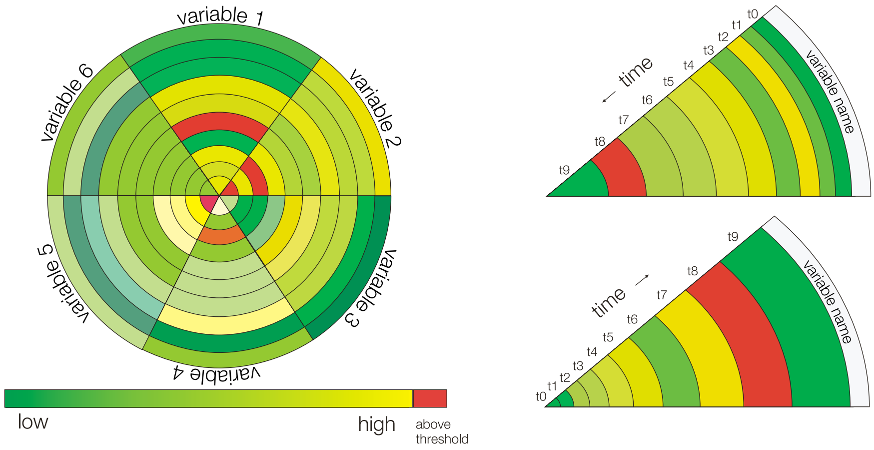


Fig. 7.68: Multiple variables are shown as segments of a circle. Each segment is further subdivided into time slots that represent data values using color. Left: six variables over ten time steps; right: different time axes arrangements (outside–in vs. inside–out) and space assignment that emphasizes more recent values.
Source: Adapted from Keim et al. (2004).

Keim et al. (2004) developed CircleView for visualizing multivariate streaming data as well as static historical data. Its basic idea is to divide a circle into a number of segments, each representing one variable. The segments are further divided into slots covering periods of time, and color shows the (aggregated) data value for the corresponding interval. Thus, time is mapped linearly along the segments. The user can interactively adjust the number of time slots, the time span per slot, different layouts for the time axis, and might emphasize more recent slots by assigning increasingly more space to them (see Figure 7.68, right). Since the order of segments is important for the visual appearance and comparison, it can be adjusted by the user or set automatically using similarity measures. For streaming data the segments of the circle are shifted automatically from the center to the edge (or vice versa). Keim and Schneidewind (2005) also presented a multi-resolution approach on top of CircleView, where time slots for coarser granularities are shown besides detail values.

References

Keim, D. A. and Schneidewind, J. (2005). Scalable Visual Data Exploration of Large Data Sets via MultiResolution. *Journal of Universal Computer Science*, 11(11):1766–1779.

Keim, D. A., Schneidewind, J., and Sips, M. (2004). CircleView: A New Approach for Visualizing Time-Related Multidimensional Data Sets. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 179–182, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static, dynamic**
dimensionality: **2D**

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **dynamic**
dimensionality: **2D**

Trendalyzer, Animated Scatter Plot

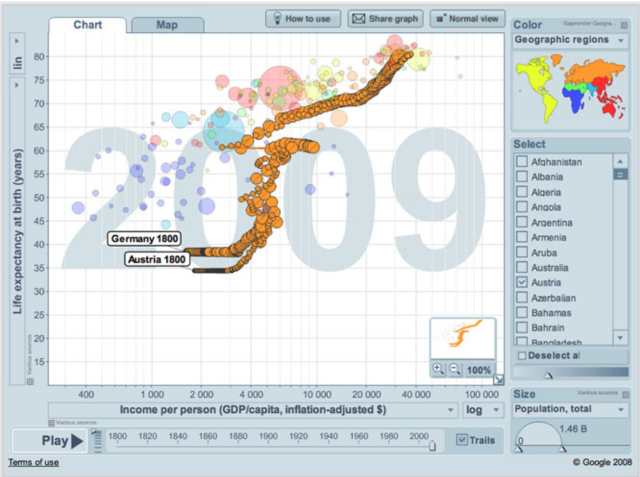


Fig. 7.69: Two data variables are mapped to the horizontal and vertical axes, symbol size represents a third variable, and animation is used to step through time. Additionally, trails are activated for the selected countries, Austria and Germany, which help to preserve the path of a variable through time.
Source: Generated with Trendalyzer with permission of the Gapminder Foundation.

Trendalyzer by Gapminder Foundation (2010) is an interactive visualization and presentation tool that is based on scatter plots. In contrast to point plots (↪ p. 152) where time is mapped on the horizontal or vertical axis, animation is used to represent time. Hence, two data variables are mapped onto the axes of the Cartesian coordinate system and animation is used to step through time. The size of a dot represents a third variable and color is used for distinguishing groups. The animation can be controlled via a time slider, a play/pause button, and a slider for adjusting animation speed. Furthermore, trails might be displayed, which help to preserve the path of a variable through time. This means that dots stay visible and are connected over time. Robertson et al. (2008) evaluated the use of animation in conveying trends over time and compared Trendalyzer with a modified version of trails, and small multiples (↪ p. 236). The results show that animation is both slower and less accurate than the other representations but is well suited as a presentation aid.

References

Gapminder Foundation (2010). Gapminder Trendalyzer. URL, http://www.gapminder.org/world/. Retrieved Feb., 2011.

Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J. (2008). Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14:1325–1332.

TimeRider

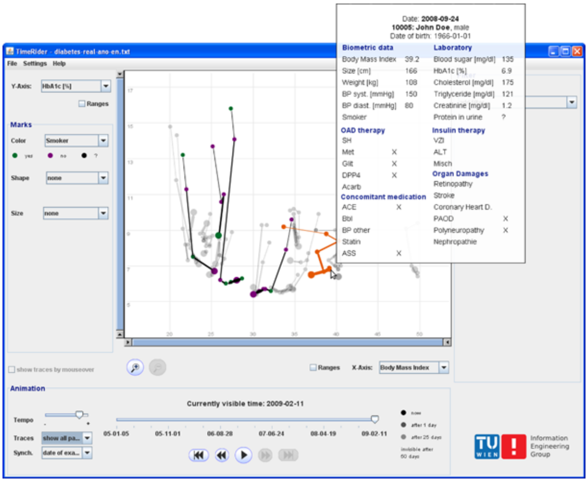


Fig. 7.70: Patients are represented as marks in a scatter plot that can be animated over time. Body mass index is mapped to the horizontal axis, HbA1c to the vertical axis, and mark color shows whether a patient smoked. Time controls for animation and synchronization settings are visible at the bottom. Additionally, traces are displayed that connect values over time. A detail-on-demand window showing further patient data is displayed when hovering over a patient mark.
Source: Generated with the TimeRider software.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

TimeRider by Rind et al. (2011) is an enhanced animated scatter plot (↪ p. 220) for exploring multivariate trends in cohorts of diabetes patients. The enhancements tackle three challenges of medical data: irregular sampling, data wear (i.e., decreasing validity over time), and patient records covering different portions of time. Animation of irregularly sampled data is achieved via interpolation of individual values along a linear trajectory. To account for data wear and to maintain temporal context, transparency and traces are used to enrich the visual encoding of time. For comparing patient histories that cover different portions of time, TimeRider provides four synchronization modes: by calendar date, patient age, start of treatment, and end of treatment. To take better advantage of animation, TimeRider is highly interactive; apart from common interactions to select, pan, zoom, filter, and show details on demand, the user can change the visual mapping of axes, color, shape, and size (see Figure 7.70, left). Other task-specific features are value ranges that can be highlighted in the background of the scatter plot and dynamic queries on data variables.

References

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 *Proceedings of the International Conference on Human-Computer Interaction (HCI-I)*, Berlin, Germany. Springer. To appear.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

Process Visualization

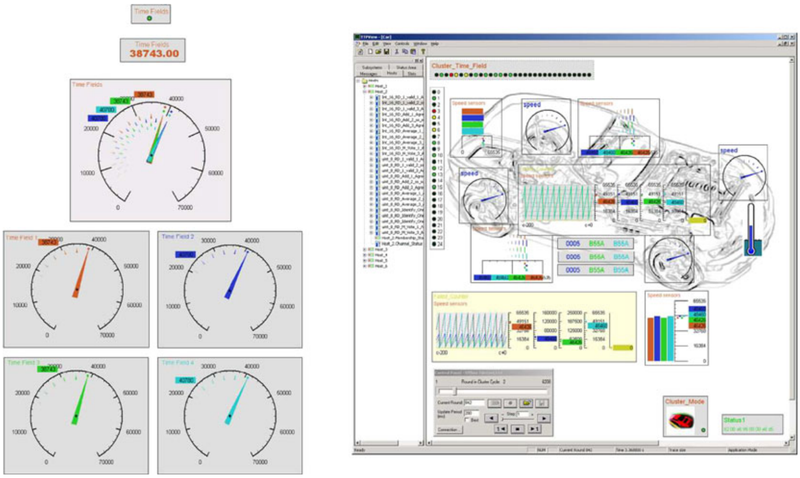


Fig. 7.71: Process visualization can be supported by providing virtual instruments at different levels of detail (left). This helps users to stay focused on important variables of an automotive process, while less relevant information is presented at a higher level of abstraction only (right). *Source: Matković et al. (2002), © 2002 IEEE. Used with permission.*

Process visualization, for instance in automotive environments, has to deal with a multitude of time-varying input variables to be monitored. Matković et al. (2002) suggest a focus+context approach to help users keep track of the important changes of a process. The key idea is to provide virtual instruments that represent monitored variables at different levels of detail. Instruments representing focused variables provide more detailed information, for example, a brief view on a variable's history, which is not possible with classic gauges. On the other hand, less relevant variables are visualized using heavily abstracted virtual instruments that might show just the numeric value or even only a colored dot. Multiple such instruments are arranged in a virtual environment that is used as visual reference for the monitoring scenario. Focus and context within the environment can change dynamically during monitoring, either upon detection of certain events in the data or via user interaction. The approach of Matković et al. (2002) is an excellent example of visualization of dynamic temporal data (see Section 3.3).

References

Matković, K., Hauser, H., Sainitzer, R., and Gröller, E. (2002). Process Visualization with Levels of Detail. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 67–70, Los Alamitos, CA, USA. IEEE Computer Society.

Flocking Boids

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **3D**

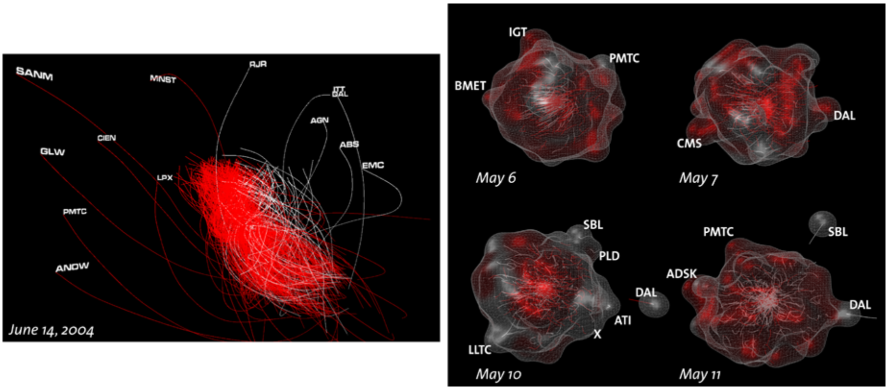


Fig. 7.72: Stock market data are represented as flocking boids that move in a three-dimensional presentation space. Left: boids leaving the flock indicate that the corresponding stock price behaves differently than the majority of prices; right: implicit surfaces surrounding boids help users to recognize the spatial structure of the flock.
Source: Vande Moere (2004), © 2004 IEEE. Used with permission.

Stock market data change dynamically during the day as prices are constantly updated. Vande Moere (2004) proposes to visualize such data by means of information flocking boids. The term boids borrows from the simulation of birds (bird objects = boids) in flocks. In order to visualize stock market prices, each stock is considered to be a boid with an initially random position in a 3D presentation space. Upon arrival of new data, boid positions are updated dynamically according to several rules. These rules attempt to avoid collisions of boids, to move boids at the same speed as their neighbors in the flock, to move boids toward the flock's center, to keep similar boids close to each other, and to let boids stay away from boids that are dissimilar. The visual representation is inherently dynamic and aims at the users' capability to perceive emergence of patterns as the visualization updates. To this end, boids and corresponding traces are visualized as animated curves, as shown on the left in Figure 7.72. This 3D visual representation is enhanced by enclosing boids within implicit surfaces, which help users recognize the spatial structure of the flock (see Figure 7.72, right). The flocking boids visualization can be useful for detecting various patterns in the data such as the emergence of clusters, the separation of boids from the main flock, or a general chaotic behavior of boids.

References

Vande Moere, A. (2004). Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 97–104, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **multivariate**

Time Line Browser

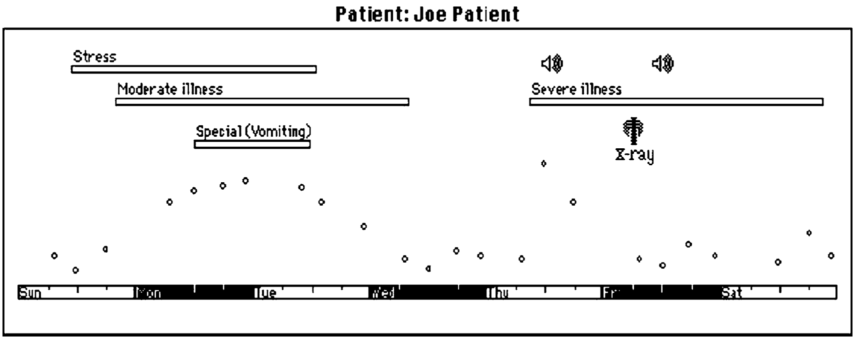


Fig. 7.73: Heterogeneous patient information is visualized along a common horizontal time axis. Intervals are displayed as labeled bars and events are displayed as icons. The small circles form a point plot that shows the patient's blood glucose over time.
Source: Cousins and Kahn (1991), © 1991 Elsevier. Used with permission.

time

arrangement: **linear**
time primitives: **instant, interval**

Cousins and Kahn (1991) developed the time line browser for visualizing heterogeneous time-oriented data. The time line browser integrates qualitative and quantitative data as well as instant and interval data into a single coherent view. To this end, Cousins and Kahn (1991) distinguish simple events, complex events, and intervals. Simple events are represented as small circles, whereas complex events are shown as icons. Bars are used to indicate location and duration of intervals. These depictions are aligned with respect to a common horizontal time axis (↪ p. 166), where textual labels might be used to display further details. In addition to the visualization, a formal system for timeline elements and timeline operations has been developed. It defines five basic operations (i.e, slice, filter, overlay, add, new) for manipulating timelines and also supports composite operations. These operations are useful for addressing the issues of different temporal granularities and the calendar mapping problem.

vis

mapping: **static**
dimensionality: **2D**

References

Cousins, S. B. and Kahn, M. G. (1991). The Visual Display of Temporal Information. *Artificial Intelligence in Medicine*, 3(6):341–357.

LifeLines

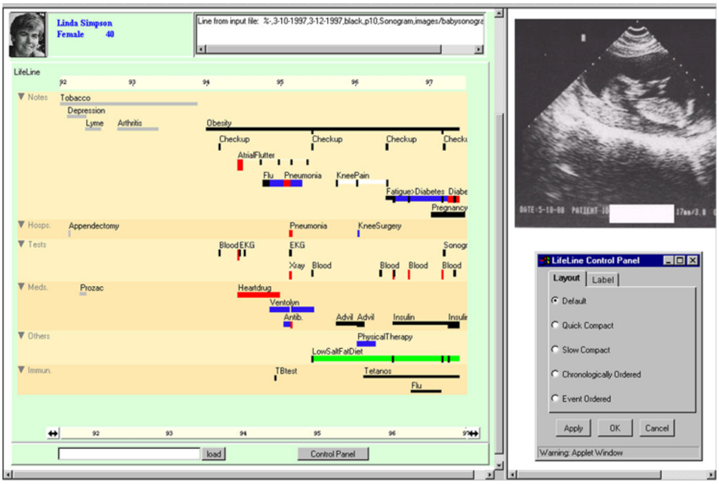


Fig. 7.74: Horizontal bars are used to show the temporal location and duration of health-related incidents. The example shows several facets of patient information and an additional linked sonogram on the right.
Source: Image courtesy of Catherine Plaisant and University of Maryland Human-Computer Interaction Lab.

A simple and intuitive way of depicting incidents is by drawing a horizontal line on a time scale for the time span the incident took. This form of visualization is called timeline (↪ p. 166). Plaisant et al. (1998) apply and extend this concept for visualizing health-related incidents in personal histories and patient records. Consequently, they call their approach LifeLines. Horizontal bars are used to show the temporal location and duration of incidents, treatments, or rehabilitation. Additional information can be encoded via the height as well as the color of individual bars. In order to structure the displayed information in groups, so-called facets are introduced. Multiple such facets are stacked vertically. Depending on the information sought by the user, facets can be expanded and collapsed. When collapsed, only a very small and geometrically as well as semantically down-scaled visual representation without textual labels is shown. When expanded, a facet shows full detail. External information related to certain incidents might be provided on demand in a linked view, as for example x-ray images or sonograms.

References

Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., and Shneiderman, B. (1998). LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, pages 76–80, Bethesda, MD, USA. American Medical Informatic Association (AMIA).

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

PatternFinder

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

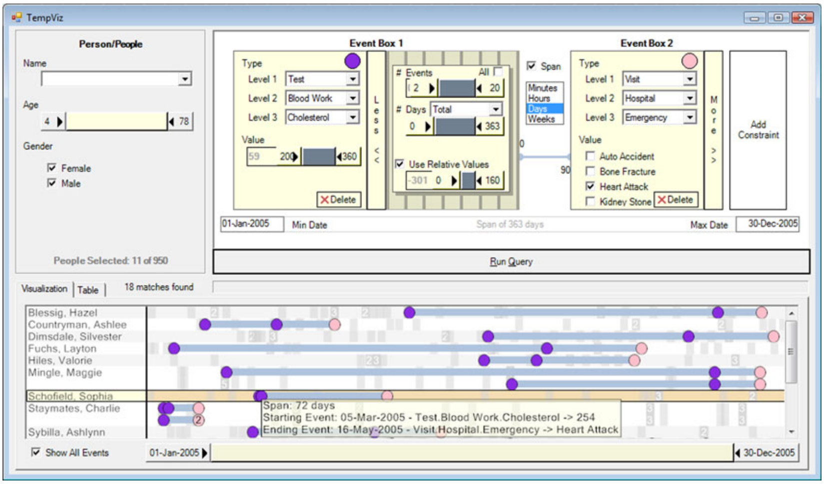


Fig. 7.75: The query formulated in the visual interface (top) relates a cholesterol test to a subsequent emergency visit in a hospital. The resulting visualization (bottom) shows several patient records that match with the query as vertically stacked ball-and-chain representations.
Source: Image courtesy of Jerry Alan Fails.

PatternFinder by Fails et al. (2006) is used for constructing queries to find temporal patterns in medical record databases. The temporal patterns consist of events that are associated with data, and time spans that separate events. Users formulate queries by imposing constraints on events and time spans. Events can be selected from a hierarchically structured vocabulary and constraints for associated variables can be specified in a visual interface along with temporal constraints. This way, users can build queries for the existence of events (e.g., persons with heart attack), temporally ordered events (e.g., heart attack followed by stroke), temporally ordered value changes (e.g., BMI of 25 or higher followed by BMI of 20 or lower), and trends over time (e.g., BMI decreasing). Event sequences might not only be specified in terms of temporal order, but also in terms of temporal distance (e.g., time span of 28 days or less between heart attack and stroke). Moreover, all of the mentioned query types can also be combined. For visualizing query results, a so-called ball-and-chain representation is used: results are shown as vertically stacked timelines, where colored circles represent matched events and bars stand for matched time spans.

References

Fails, J., 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 the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 167–174, Los Alamitos, CA, USA. IEEE Computer Society.

Continuum

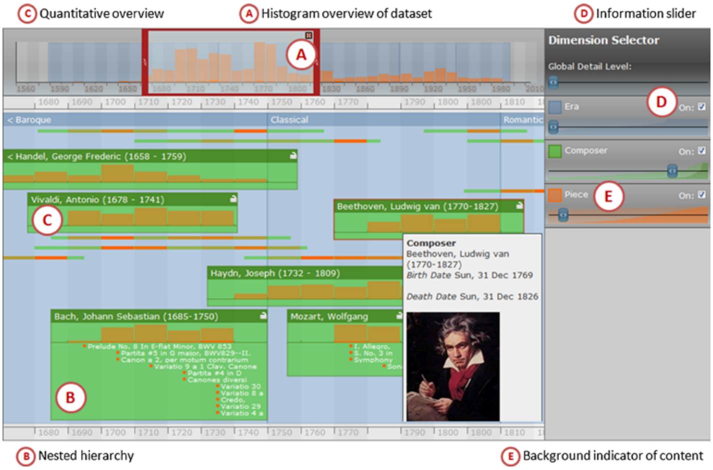


Fig. 7.76: Continuum showing a music dataset with the three variables era, composer, and piece. Scalable histograms provide a complete representation of the data and quantify the focal data item. Top-left: limeline overview; bottom-left: timeline detail view; right: dimension selector panel. *Source: Image courtesy of Paul André.*

Collections of small events often constitute larger, more complex events, like for example talks at conferences or legs of a race. Moreover, events might also be related to other events at other points in time (e.g., a paper written at some point in time and referenced later). Continuum by André et al. (2007) is a timeline visualization tool to represent large amounts of hierarchically structured temporal data and their relationships. It addresses the three problems of scale, hierarchy, and relationships by using scalable histogram overviews, flattening high-dimensional data into dynamically adjustable hierarchies, and arching connection lines for representing non-hierarchical relationships. The interface consists of three main panels that show overview, detail, and the dimension configuration. The timeline overview always represents the complete timespan of the dataset using scalable histograms where the vertical axis quantifies the user-selected focal data item. The timeline detail view shows hierarchical relationships as nested elements and applies semantic zooming depending on the amount of information to be displayed. With the dimension selector, users can interactively control the hierarchical buildup and the level of detail to be shown.

References

André, P., Wilson, M. L., Russell, A., Smith, D. A., Owens, A., and schraefel, m.c. (2007). Continuum: Designing Timelines for Hierarchies, Relationships and Scale. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 101–110, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **2D**

EventRiver

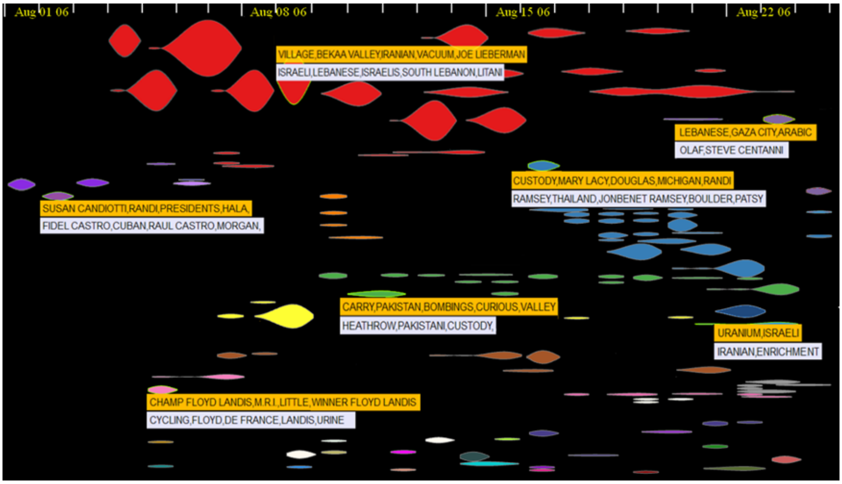


Fig. 7.77: CNN news data from August 2006. Event bubbles flow in a horizontal river of time, where important events are highlighted in red in the top part of the river.
Source: Luo et al. (2011). © 2011, IEEE. Used with permission.

Text collections such as news corpora or email archives often contain temporal references, which embed the text's information into a temporal context. Luo et al. (2011) describe a technique, called EventRiver, for exploring such text collections interactively in terms of important events and the stories that these events constitute. In a first phase, events are extracted from the data using a number of analytical steps, including keyword identification and temporal locality clustering. This phase yields a set of events which are characterized by their position in time, by their duration, and by several other measures (e.g., temporal influence, strength, co-strength). The visual design of EventRiver is based on so-called event bubbles that flow in a horizontal river of time (i.e., along a horizontal time axis). The bubbles are placed horizontally where events are located in time. A bubble's shape illustrates how an event has emerged and disappeared over time. Colors and the bubbles' vertical positions in the river are chosen so as to highlight important interconnected events that constitute long term stories in the text documents. While tooltip labels show the important keywords of events, document details are provided on demand in separate views. Analysts can adjust the EventRiver by using various interaction techniques including dynamic filtering, semantic and temporal zooming, and manual relocation of event bubbles.

References

Luo, D., Yang, J., Krstajic, M., Ribarsky, W., and Keim, D. (2011). EventRiver: Visually Exploring Text Collections With Temporal References. *IEEE Transactions on Visualization and Computer Graphics*. To appear.

FacetZoom

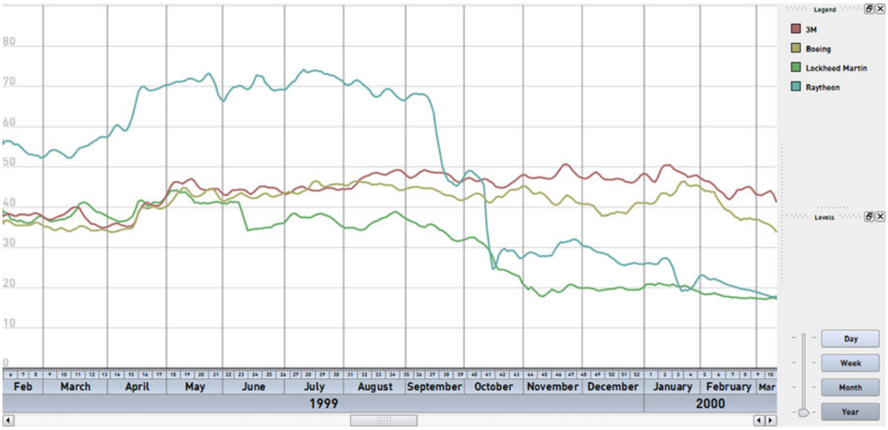


Fig. 7.78: The hierarchical structure of time is shown as an interactive horizontal time axis widget that has a data view attached to it, in this case a visualization of stock market data.
Source: Image courtesy of Raimund Dachselt.

FacetZoom is a technique that enables users to navigate hierarchically structured information spaces (see Dachselt et al., 2008). The hierarchical structure of time is a natural match for this technique. What Dachselt and Weiland (2006) originally called TimeZoom is a visual navigation aid for time-oriented data. The basic idea is to display a horizontal time axis that represents different levels of temporal granularity as stacked bars (e.g., decades, years, months, weeks, days). The time axis is an interactive widget that can be used to access data from different parts of the time domain at different levels of abstraction. In addition to continuous zooming and panning via mouse, it is also possible to simply select discrete intervals from the time axis. Depending on the user's selection, the time axis display is altered to accommodate the selected part of the time axis with more display space. Accordingly, the data view, which is attached to the time axis, can use the extra space to represent more data items in greater detail. While the actual mapping of time is static, the navigation steps of the user, including the visual adjustment of the time axis, are smoothly animated.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

References

Dachselt, R., Frisch, M., and Weiland, M. (2008). FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1353–1356, New York, NY, USA. ACM Press.

Dachselt, R. and Weiland, M. (2006). TimeZoom: A Flexible Detail and Context Timeline. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 682–687, New York, NY, USA. ACM Press. Extended Abstracts.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **2D**

Midgaard

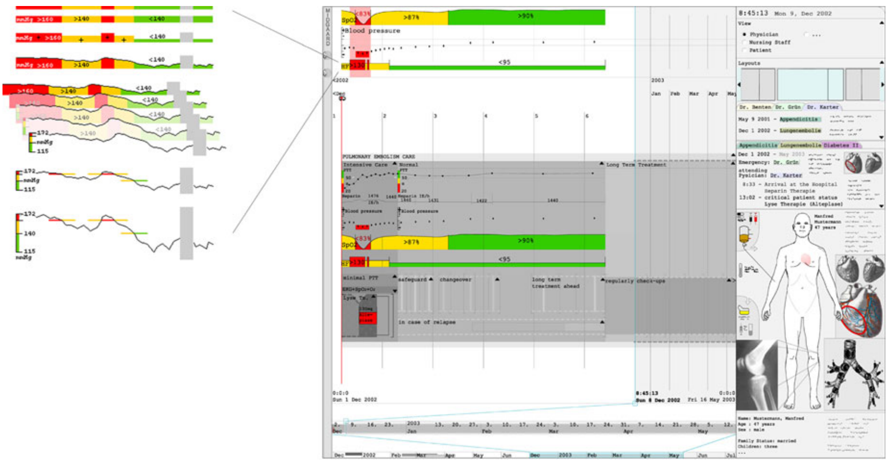


Fig. 7.79: Midgaard integrates the display of time-oriented patient data and treatment plans. Right: main user interface showing different measurements (e.g., blood gas measurements, blood pressure) using line plots, their corresponding temporal abstractions using color, and treatment plans as well as additional patient information on the right, and an interactive multi-scale time axis at the bottom; left: different steps of semantic zooming of a time-series from a broad overview (top) to a detailed view with details of fine structures (bottom).
Source: Authors.

Several tightly integrated visualization techniques have been developed in the Midgaard project by Bade et al. (2004) to enhance the understanding of heterogeneous patient data. To support the user in exploring the data and to capture as much qualitative and quantitative information as possible on a limited display space, Midgaard supports different levels of abstractions for time-oriented data (see Section 6.3). Switching between these levels is achieved via a smoothly integrated semantic zoom functionality (see Figure 7.79, left). These methods were designed to allow users to interact with data and time. Navigation in time is done using three linked time axes (see Figure 7.79, bottom-right). The first one (bottom) provides a fixed overview of the underlying time interval covering its full range. Selecting a subrange in that time axis defines the temporal bounds for the main display area and the second (middle) time axis. Selecting a further subrange in the middle time axis defines detail and surrounding context areas in time. By interactively adjusting the subranges, users can easily zoom and pan in time.

References

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 105–112, New York, NY, USA. ACM Press.

VisuExplore

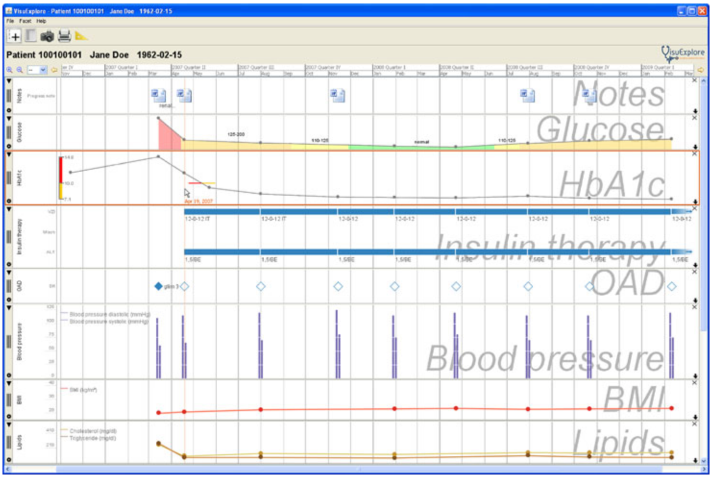


Fig. 7.80: Visualization of heterogeneous medical parameters of a diabetes patient. Beneath a panel that shows patient master data, eight visualization views show progress notes in a document browser, glucose and HbA1c as line plots with semantic zoom, insulin therapy as timelines, OAD as event chart, blood pressure as bar graphs, and BMI as well as lipids as line plots (top to bottom). *Source: Generated with the VisuExplore software.*

VisuExplore by Rind et al. (2010) is an interactive visualization system for exploring a heterogeneous set of medical parameters over time. It uses multiple views along a common horizontal time axis to convey the different medical parameters involved. VisuExplore provides an extensible environment of pluggable visualization techniques and its primary visualization techniques are deliberately kept simple to make them easily usable in medical practice: line plots (↪ p. 153), timeline charts (↪ p. 166), bar graphs (↪ p. 154), event charts, line plots with semantic zoom (see p. 112), and document browsers (see Figure 7.80, top). Furthermore, data might also be presented as textual tables to augment the visual representations. VisuExplore's interactive features allow physicians to get an overview of multiple medical parameters and focus on parts of the data. Users may add, remove, resize, and rearrange visualization views. Additionally, a measurement tool is integrated that makes it possible to determine time spans between user selected points of interest and this works not only within one but also across different views.

References

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, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **2D**

KNAVE II

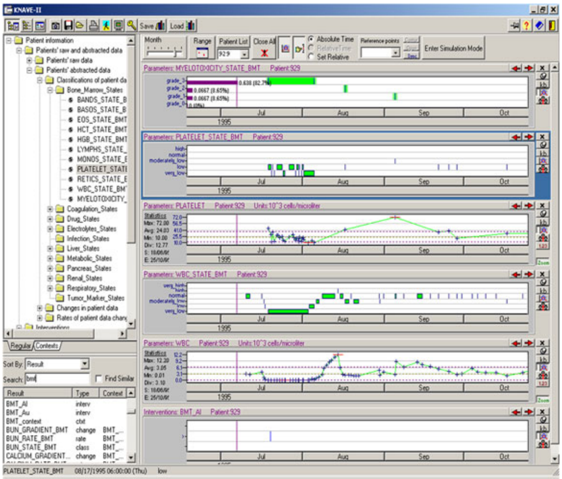


Fig. 7.81: KNAVE II supports the visualization and knowledge-based navigation of patient data and abstractions thereof. Left: tree view of clinical domain ontology for navigation; right: visualization panels for raw data and abstractions; bottom-left: search panel.
Source: Shahar et al. (2006), © 2006 Elsevier. Used with permission.

KNAVE II by Shahar et al. (2006) enables visual browsing and exploring of patient's data (raw measured values and external interventions such as medications). The system focuses mainly on the visual display of temporal abstractions of the data (see Section 6.3) and shows domain-specific concepts and patterns. In order to abstract the raw data, a predefined knowledge base is used that defines three types of interpretations: classification of data (e.g., low – normal – high), change of data (e.g., increasing – decreasing), and rate of change (e.g., slow – fast). Colored timelines (↪ p. 166) depict abstracted intervals as bars where a bar's vertical position within a panel encodes its qualitative value. For example, the second panel (blue frame) in Figure 7.81 shows the platelet state abstracted to the qualitative values: very low, low, moderately low, normal, high (from bottom to top). KNAVE II also allows users to view the raw data as line plots (↪ p. 153). Moreover, statistics for parameters and corresponding abstractions can be superimposed as bar graphs (↪ p. 154) as shown in the topmost panel, or can be given as text labels, as shown in the third and fifth panels. A granularity-based zoom allows users to quickly navigate in time simply by clicking individual granules, for instance the month *Aug*.

References

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(2):115–135.

Circos

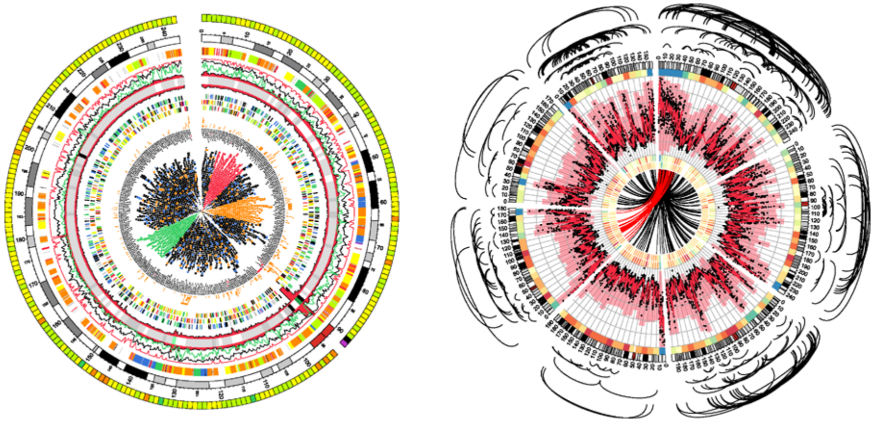


Fig. 7.82: Display of multivariate data using data tracks in a radial layout. Images show human chromosome data using point plots, line plots, tiles, histograms, heatmaps, and connectors that link points on the circle (right).
Source: Images courtesy of Martin Krzywinski.

Circos by Krzywinski et al. (2009) uses a circular design to generate multivariate displays. It uses concentric bands (data tracks) as display areas and is capable of displaying data as point plots (↪ p. 152), line plots (↪ p. 153), histograms, heat maps, tiles, connectors, and text. In this sense, time is mapped circularly to the circumference of data tracks. The configuration of a visual representation is handled via plain-text files where rules that can be defined for each data track filter and format data elements based on position, value or previous formatting. For communicating quantitative data via color, perceptually uniform color schemes based on the work of Brewer (1999) are used. Circos was initially developed for genomics and bioinformatics data to visualize alignments, conservation, and intra- and inter-chromosomal relationships. Relationships between pairs of positions are represented by the use of ribbons that connect elements. In the same way, relational data encoded in tabular formats can be shown. Due to its flexible approach, Circos has also been applied to numerous other application areas, such as urban planning, and has been used for infographics in newspapers and ads to display complex relationships.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

References

Brewer, C. A. (1999). Color Use Guidelines for Data Representation. In *Proceedings of the Section on Statistical Graphics*, pages 55–60, Baltimore, MD, USA. American Statistical Association.

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.

Kaleidomaps

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

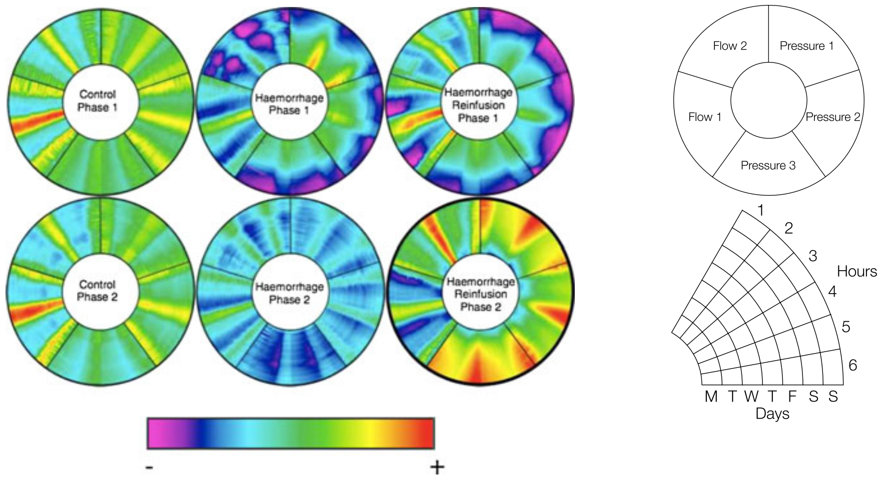


Fig. 7.83: On the left, six kaleidomaps show the morphology of blood pressure and flow waves over two experimental phases. Top-right: illustration of the layout of variables within a kaleidomap; bottom-right: layout of time within a segment.
Source: Bale et al. (2006), © 2006 IEEE. Used with permission.

Kaleidomaps by Bale et al. (2007) visualize multivariate time-series data and the results of wave decomposition techniques using the curvature of a line to alter the detection of possible periodic patterns. The overall idea of kaleidomaps is similar to the rendered output of a kaleidoscope for children, from whence the name comes. A base circle is broken into segments of equal angles for different variables. Each circle segment has two axes representing time, one along the radius and one along the arc of the segment. The data values and categories are represented using color. Due to the circular nature of kaleidomaps, the number of variables in one circle is limited to a maximum of six to eight. Interaction techniques within the kaleidomaps allow an analyst to drill down both in time and frequency domains in order to uncover potential relationships between time, space, and waveform morphologies. Kaleidomaps were developed in the domain of critical care medicine, but case studies have shown their usefulness in other domains as well, like in environment analysis.

References

Bale, K., Chapman, P., Barraclough, N., Purdy, J., Aydin, N., and Dark, P. (2007). Kaleidomaps: A New Technique for the Visualization of Multivariate Time-Series Data. *Information Visualization*, 6(2):155–167.

Bale, K., Chapman, P., Purdy, J., Aydin, N., and Dark, P. (2006). Kaleidomap Visualizations of Cardiovascular Function in Critical Care Medicine. In *Proceedings of International Conference on Medical Information Visualisation - BioMedical Visualisation (MediVis)*, pages 51–58, Los Alamitos, CA, USA. IEEE Computer Society.

Intrusion Detection

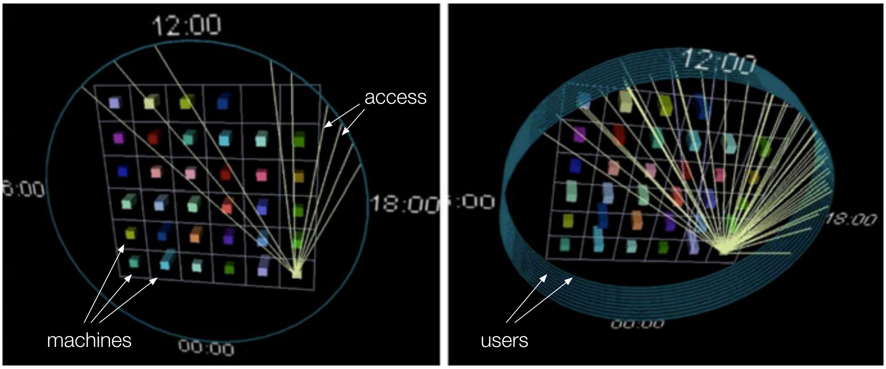


Fig. 7.84: Machines in a network are represented by a matrix of 3D cubes in the center of the display. Time is mapped to the circumference of a circle enclosing the matrix of machines. When a particular machine is accessed, a line is drawn that links the particular machine with a point in time on the circular time axis.
Source: Images courtesy of Kovalan Muniandy.

A 3D visualization technique by Muniandy (2001) helps to analyze user access to computers in a network over time for intrusion detection. The different parameters time, users, machines, and access are mapped onto a 3D cylinder. In Figure 7.84, time is mapped onto the circumference of a circle showing the 24 hours of a day. The units along the circle can be configured to represent either hours, months, or years. Different users are represented by individual cylinder slices that are stacked upon each other and machines are represented as cubes that are arranged in a matrix. Access to a machine by a user is visualized by a line connecting the user slice at the corresponding access time with the accessed machine. This way, certain patterns of network access can easily be spotted visually and suspicious behavior can be revealed. Details-on-demand are displayed when hovering with the mouse over an element of the visualization. To mitigate occlusion, the representation can be zoomed and rotated freely by the user. Moreover, filtering can be applied to remove clutter.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Muniandy, K. (2001). Visualizing Time-Related Events for Intrusion Detection. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA. IEEE Computer Society. Late Breaking Hot Topics.

data

frame of reference: **abstract, spatial**
variables: **uni-, multivariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D, 3D**

Small Multiples

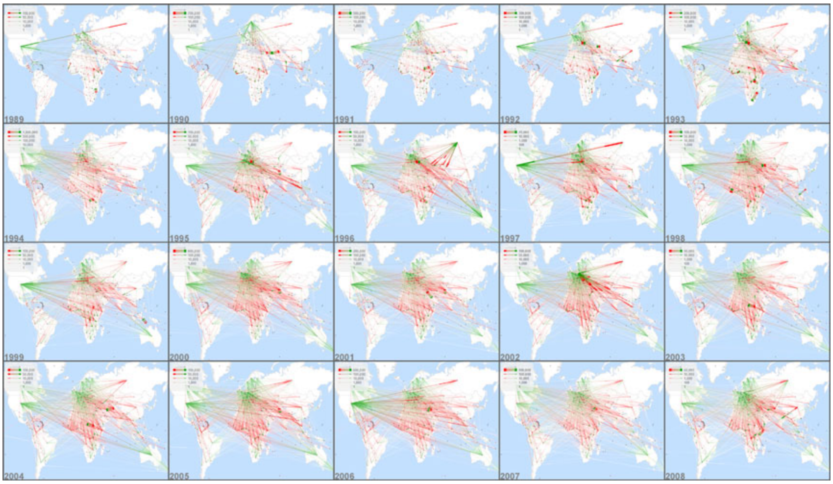


Fig. 7.85: Small multiples showing migration data. Each miniature map visualizes migration of people as links between countries. Green color indicates the origin, red color the destination, and line width the volume of migration.
Source: Generated with the JFlowMap software with permission of Ilya Boyandin.

Small multiples are more a general concept than a specific technique. Tufte (1983, 1990) describes small multiples as a set of miniature visual representations. For time-oriented data, each miniature visualizes a selected time point. The concrete depiction may show a single variable or multiple variables in an abstract or spatial context using a 2D or 3D presentation space. Particularly relevant is the arrangement of the small multiples as it dictates how the time axis is perceived. Linear or circular arrangements can be used, or specific arrangement patterns can be applied to account for different granularities of the time axes. Small multiples provide an overview of the data and allow users to visually compare the data at different time points. Another advantage of small multiples is that the concept can be applied to virtually any existing visualization technique; the only thing to do is to create a thumbnail from an existing visual representation for each time step. Depending on the amount of screen space occupied by each thumbnail, however, the number of representable time steps could be rather moderate. Or, if the images are shrunk to fit more time steps, less details are visible.

References

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.
Tufte, E. R. (1990). *Envisioning Information*. Graphics Press, Cheshire, CT.

EventViewer

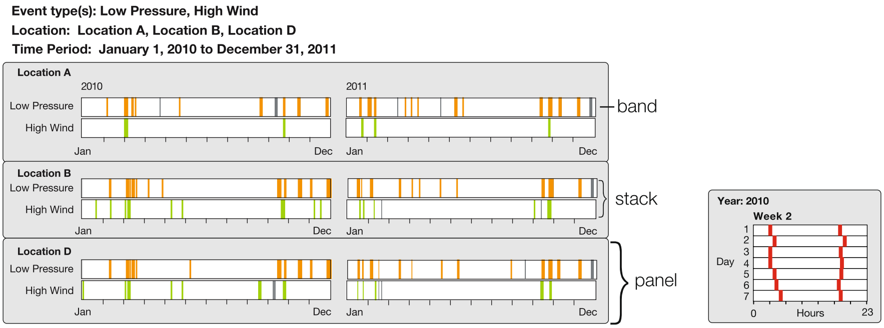


Fig. 7.86: Visual exploration of event data. Spatial, temporal, and thematic dimensions of events can flexibly be assigned to configurations of bands, stacks, and panels. Left: low pressure and high wind events are shown along three locations and two years; right: display configuration to reveal temporal patterns along hours of a day.
Source: Adapted from Beard et al. (2008).

EventViewer by Beard et al. (2008) is a framework that has been developed to visualize and explore spatial, temporal, and thematic dimensions of sensor data. The system supports queries on events that have been extracted from such data and are stored in an events database. The spatial, temporal, and thematic categories of selected events can flexibly be assigned to three kinds of nested display elements called bands, stacks, and panels. Bands are the primary graphic object and act as display container for a set of events. The horizontal dimension of a band represents time and bars within a band represent instances of events. The length of a bar corresponds to the event's duration and color can be used to encode other data values. Furthermore, missing data is shown by using gray bars to make a clear visual distinction to areas without events (shown as empty areas). Stacks consist of event bands that are placed on top of each other and panels are collections of stacks. Each of the three data dimensions space, time, and theme can be modeled along hierarchies or lattices. For time, calendric systems consisting of time granularities like hours, days, weeks, and years are used (see Section 3.1.2). The configuration of display elements are broken down along these hierarchical and lattice structures and form small multiples (↪ p. 236). The assignments can be changed interactively by the user via direct manipulation, thus revealing different kind of patterns as for example periodic patterns, spatial and temporal trends, or event-event relationships.

data
frame of reference: **abstract, spatial**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

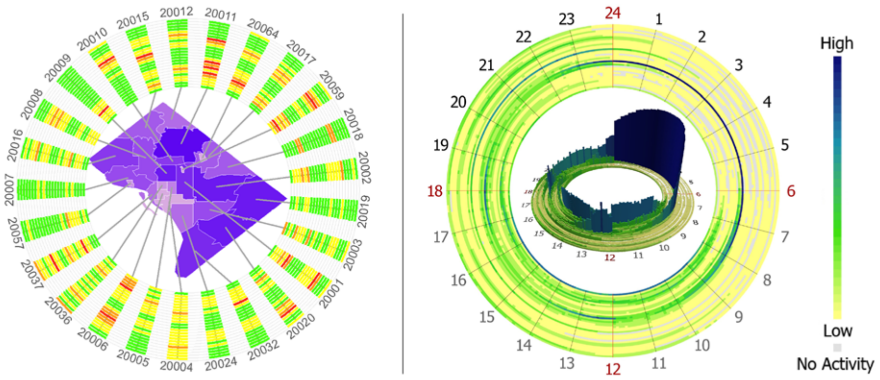
vis
mapping: **static**
dimensionality: **2D**

References

Beard, K., Deese, H., and Pettigrew, N. R. (2008). A Framework for Visualization and Exploration of Events. *Information Visualization*, 7:133–151.

Ring Maps

data
frame of reference: **abstract, spatial**
variables: **uni-, multivariate**



time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

Fig. 7.87: Ring maps representing health related alert levels (green, yellow, orange, red) for various zip code regions during a period of 24 weeks (left) and degree of activity during the course of a day for 96 human activities, one shown per ring (right).
Source: Left: Image courtesy of Guilan Huang. Right: Image courtesy of Jinfeng Zhao.

The basic idea of ring maps is to create multiple differently sized rings, each of which is subdivided into an equal number of ring segments (see Zhao et al., 2008; Huang et al., 2008). The rings and their segments as well as the center area of the overall visual representation can be used in various ways. One can utilize ring maps to visualize spatio-temporal data. To this end, a map is shown in the center and the ring segments of a particular angle are associated with a specific area of the map. This is depicted in the left part of Figure 7.87, where different angles show the data for different zip code regions. A time-series for each region can then be represented by the rings, for instance, by assigning the first series entry to the inner-most ring and the last one to the outer-most ring. The actual data visualization is done by color coding. There are other ways of mapping information to rings and segments. The right part of Figure 7.87 shows an application of ring maps where the hours of the day are mapped to the ring segments and the rings represent different activities a person can be busy with during the course of a day. The degree of activity is encoded by color. This time the center of the display is used to show a complementary 3D representation to assist users in spotting highly active regions.

vis
mapping: **static**
dimensionality: **2D, 3D**

References

Huang, G., Govoni, S., Choi, J., Hartley, D. M., and Wilson, J. M. (2008). Geovisualizing Data With Ring Maps. *ArcUser*, Winter 2008.

Zhao, J., Forer, P., and Harvey, A. S. (2008). Activities, Ringmaps and Geovisualization of Large Human Movement Fields. *Information Visualization*, 7(3):198–209.

Time-Oriented Polygons on Maps

data
frame of reference: **spatial**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D**

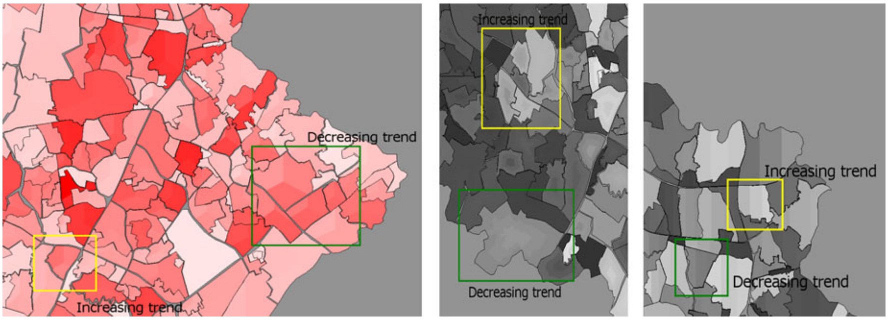


Fig. 7.88: Development of high school population in school districts over the years 2005, 2006, and 2007. Three different layouts to represent change over time of values associated with areas of a map. Left: wedges; center: rings; right: time slices. Data values are shown using color. *Source: Shanbhag et al. (2005), © 2005 IEEE. Used with permission.*

Three time-oriented visualization methods are presented by Shanbhag et al. (2005) to analyze and support effective allocation of resources in a spatio-temporal context. Wedges, rings, and time slices are the three basic layouts used to display changes of data values over time on a map. For all three variants, data values and categories are represented using color components (hue, saturation, and brightness). In the layout of the wedges the area of a polygon is partitioned in a clock-like manner into radical sectors equal to the number of time points (see Figure 7.88, left). The ring layout is based on the idea of the concentric rings of a tree trunk where the innermost ring corresponds to the earliest time point and the outermost ring corresponds to the latest time point (see Figure 7.88, center). The time slices layout divides a polygon into vertical slices that are ordered from left to right according the progress in time (see Figure 7.88, right). The wedges, rings, and time slice layouts are applied to polygonal areas of a map. These visualizations were used for example to repartition school districts (seen as planning polygons) where variables such as student population by grade, number of students requiring free meals, and test scores needed to be analyzed to plan the future allocation of resources.

References

Shanbhag, P., Rheingans, P., and desJardins, M. (2005). Temporal Visualization of Planning Polygons for Effcient Partitioning of Geo-Spatial Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 211–218, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D, 3D**

Icons on Maps

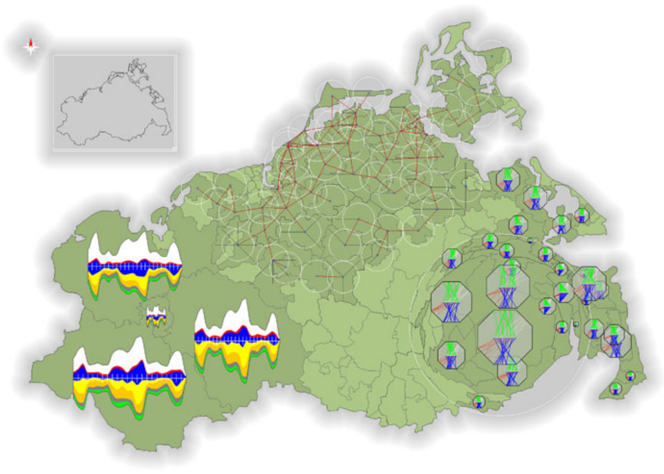


Fig. 7.89: The figure illustrates the embedding of time-representing icons into a map in order to visualize spatio-temporal data. This illustration shows ThemeRiver icons and TimeWheel icons in the left and the right part of the map, respectively. The northern part of the map illustrates a conflict graph as used for local optimization of icon positions.
Source: Generated with the LandVis system.

When time-oriented data additionally contain spatial dependencies, it is necessary to visualize both the temporal aspects and the spatial aspects. A sensible approach to achieving this is to adapt existing solutions. Maps are commonly applied to represent the spatial context of the data. In order to apply existing visualization techniques to represent the temporal context, they must be made compatible with the map display. First and foremost, this implies a reduction in size, which effectively means creating icons from otherwise full-window visual representations. In a second step, it is then possible to place multiple icons on the map, where the data are anchored in space. If there are too many icons on a map, they most likely occlude each other. Therefore, additional methods are applied to resolve occlusions by the global or local optimization of icon positions. The problem of finding suitable icon positions is very much related to the cartographic map labeling problem. Tominski et al. (2003) and Fuchs and Schumann (2004) demonstrate the integration of the ThemeRiver (↪ p. 197) and the TimeWheel (↪ p. 200) into a map display.

References

Fuchs, G. and Schumann, H. (2004). Visualizing Abstract Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 139–144, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2003). Visualisierung zeitlicher Verläufe auf geografischen Karten. In *Kartographische Schriften, Band 7: Visualisierung und Erschließung von Geodaten*, pages 47–57. Kirschbaum Verlag, Bonn, Germany.

Value Flow Map

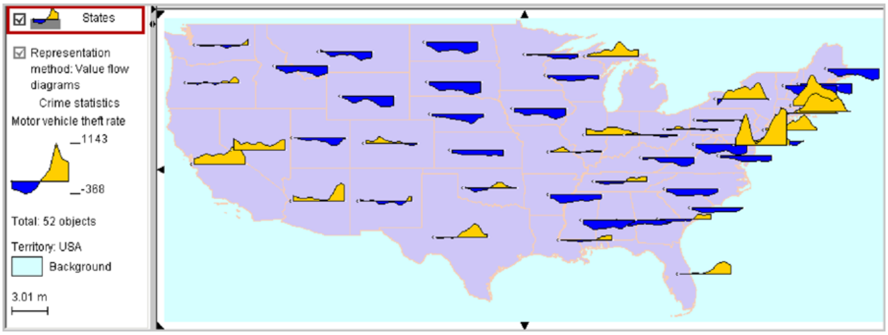


Fig. 7.90: Univariate spatio-temporal data are represented by embedding multiple miniature silhouette graphs into regions of a map. The graphs use a specific encoding where yellow color corresponds to a positive deviation of the variable from the data's mean and blue color indicates a negative deviation.
Source: Image courtesy of Gennady Andrienko.

What Andrienko and Andrienko (2004) call value flow map is a technique to visualize variation in spatio-temporal data. A value flow map shows one miniature silhouette graph (↪ p. 175) for each area of a cartographic map to represent the temporal behavior of one data variable per area. Typically, temporal smoothing is carried out by replacing the values of a point-based time scale with the mean values of an interval-based time scale. In this way, small fluctuations are disregarded and major trends become visible. Moreover, a number of data transformations can be applied to define the mapping of the graphs. An example of such a transformation is to show the variable's deviation from the data's mean, rather than the raw data, that is, data values are replaced by their differences to the mean in order to represent positive and negative variations. This way, the silhouette graphs visualize quite well how the data values flow in time and space (hence value flow map). This is a necessary requirement to enable analysts to detect patterns, and thus to support exploring spatial distributions, comparing data evolution at different locations, as well as finding similarities and outliers.

data

frame of reference: **spatial**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

References

Andrienko, N. and Andrienko, G. (2004). Interactive Visual Tools to Explore Spatio-Temporal Variation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 417–420, New York, NY, USA. ACM Press.

data
frame of reference: **spatial**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D**

Flow Map

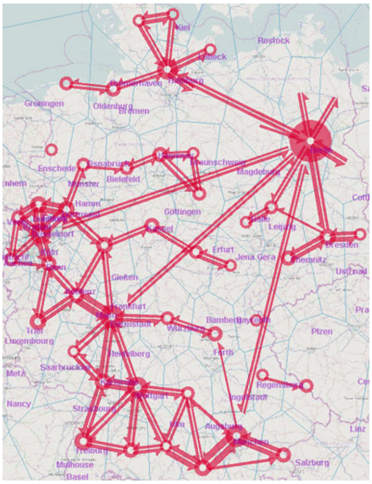


Fig. 7.91: The flow map shows characteristic movements of photographers over time extracted from metadata of more than 590,000 geo-referenced and time-stamped photographs.
Source: Image courtesy of Gennady Andrienko.

Flow maps show movements of objects over time, that is, they show a change of positions over time, rather than a change of data values. Usually, such movements form directed (optionally segmented) trajectories connecting the starting point of a movement and its end point. Such trajectories can be represented visually as more or less complex arrows or curves, where width, color, and other attributes can be used to encode additional information (see Kraak and Ormeling, 2003). A famous example is Minard's flow map of Napoleon's Russian campaign (see Figure 2.8 on page 21). A high number of flows, however, leads to overlapping trajectories and thus to visually cluttered flow maps. In order to represent massive data, flow maps can show abstractions of movements, rather than individual movements (see Andrienko and Andrienko, 2011). To faithfully communicate the underlying data, characteristic movements need to be extracted. First, time points are aggregated to larger time intervals and individual places are substituted with larger regions so as to arrive at abstracted trajectories that show mean trends. Secondly, the trajectories are grouped based on a similarity search, e.g., by applying cluster analysis or self organizing maps (SOM). In this way, places with similar dynamics are merged, and individual trajectories are replaced by trajectories associated with the groups.

References

Andrienko, N. and Andrienko, G. (2011). Spatial Generalization and Aggregation of Massive Movement Data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219.

Kraak, M.-J. and Ormeling, F. (2003). *Cartography: Visualization of Geospatial Data*. Pearson Education, Harlow, England, 2nd edition.

Time-Varying Hierarchies on Maps

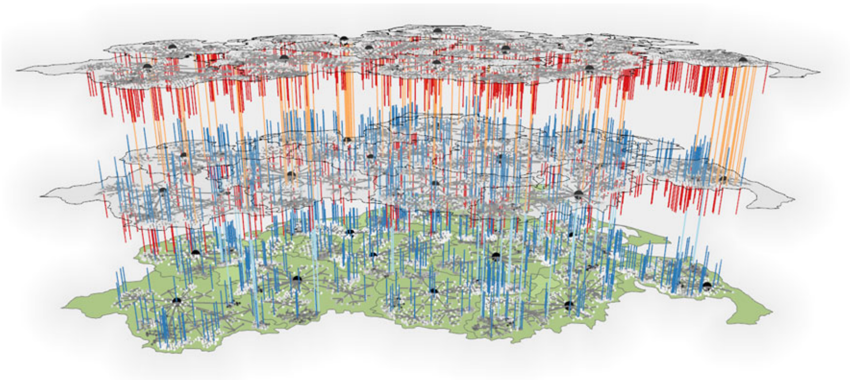


Fig. 7.92: Hierarchy layouts are embedded into areas of a map, where each map layer corresponds to one time step. Colored links and spikes between layers indicate significant changes from one time step to the other.
Source: Generated with the LandVis system.

Hierarchical structures can be found in many application areas. A technique for visualizing hierarchies that change over time in a geo-spatial context is described by Hadlak et al. (2010). This technique follows the idea of using the third dimension of the presentation space to represent the dimension of time, which is analog to the space-time cube approach (↪ p. 245). For a series of time steps, individual map layers are constructed, where each map region shows an embedded hierarchy layout and each node's color visualizes a data value. To facilitate the identification of changes between two layers, visual cues are added. Differently colored links between subsequent layers are used to indicate nodes that have moved or whose attribute values have changed significantly. Significance is determined by a user-selectable threshold. Positive attribute changes are shown as red links and negative changes are shown in blue. Links representing node movements are colored with a shade of gray. Addition or deletion of nodes and edges is indicated by spikes. Spikes that represent deletion leave a layer in the direction of the time axis and are shown in blue. Those that mark addition enter a layer and are shown in red. The layering approach in combination with the described visual cues allows users to compare successive time steps more closely.

data

frame of reference: **spatial**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

References

Hadlak, S., Tominski, C., and Schumann, H. (2010). Visualization of Attributed Hierarchical Structures in a Spatio-Temporal Context. *International Journal of Geographical Information Science*, 24(10):1497–1513.

VIS-STAMP

data

frame of reference: **spatial**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D**

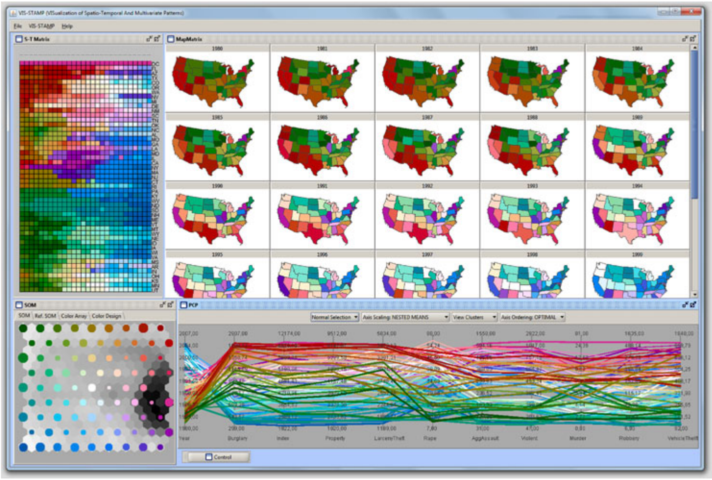


Fig. 7.93: Multiple views show multivariate spatio-temporal crime data that have been clustered by a self-organizing map (SOM). An enhanced color coding schema is used consistently among all views to visualize cluster affiliation.
Source: Generated with the VIS-STAMP system.

Spatio-temporal data can be complex and multi-faceted. Guo et al. (2006) developed a system called VIS-STAMP that integrates computational, visual, and cartographic methods for visual analysis and exploration of such data. At the heart of the system is a self-organizing map (SOM) that is used for multivariate clustering, sorting, and coloring. The visual ensemble comprises a matrix view (top-left in Figure 7.93), a map view (top-right), a parallel coordinates view (bottom-right), and a SOM view (bottom-left). The matrix view's columns represent time points and its rows stand for geographic regions. Cluster affiliation of the matrix cells is visualized by means of an enhanced color coding schema. The color coding is consistent across all views. The map view follows the small multiples approach (↪ p. 236) and shows color-coded map thumbnails, one for each time point. The parallel coordinates view addresses the multivariate character of the data. Finally, the SOM view offers a detailed view and control interface of the underlying self-organizing map. A number of automatic and interactive manipulation techniques (e.g., reordering and sorting) facilitate the data analysis.

References

Guo, D., Chen, J., MacEachren, A. M., and Liao, K. (2006). A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474.

Space-Time Cube

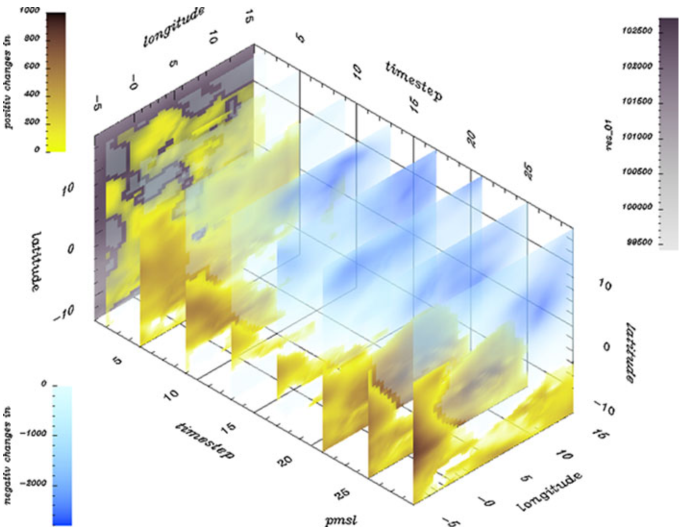


Fig. 7.94: This space-time cube represents two spatial dimensions (latitude and longitude) along the y-axis and the z-axis, and time along the x-axis. Multiple color-coded layers are embedded into the cube to visualize spatio-temporal climate data.
Source: Image courtesy of Thomas Nocke.

A classic concept that combines the visualization of space and time is the space-time cube, which is attributed to the pioneer work of Hägerstrand (1970). The basic idea is to map two spatial dimensions to two axes of a virtual three-dimensional cube and to use the third axis for the mapping of time. The spatial context is often represented as a map that constitutes one face of the space-time cube. The three-dimensional space inside the cube is used to represent spatio-temporal data, where possible visual encodings are manifold. One can place graphical objects in the cube in order to mark points of interest, or one can construct trajectories that illustrate paths of objects (↪ p. 247). Associated data can be encoded to the properties of graphical objects and trajectories, where color and size are common candidates. Another technique is to place multiple layers along the time axis, each of which encodes the data for a specific time point. Space-time cubes usually rely on appropriate interaction to allow users to view the data from different perspectives. A contemporary review of the concept can be found in the work by Kraak (2003).

References

Hägerstrand, T. (1970). What About People in Regional Science? *Papers of the Regional Science Association*, 24:7–21.

Kraak, M.-J. (2003). The Space-Time Cube Revisited from a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages 1988–1995, Newcastle, UK. The International Cartographic Association (ICA).

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

Spatio-Temporal Event Visualization

data

frame of reference: **spatial**
variables: **uni-, multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

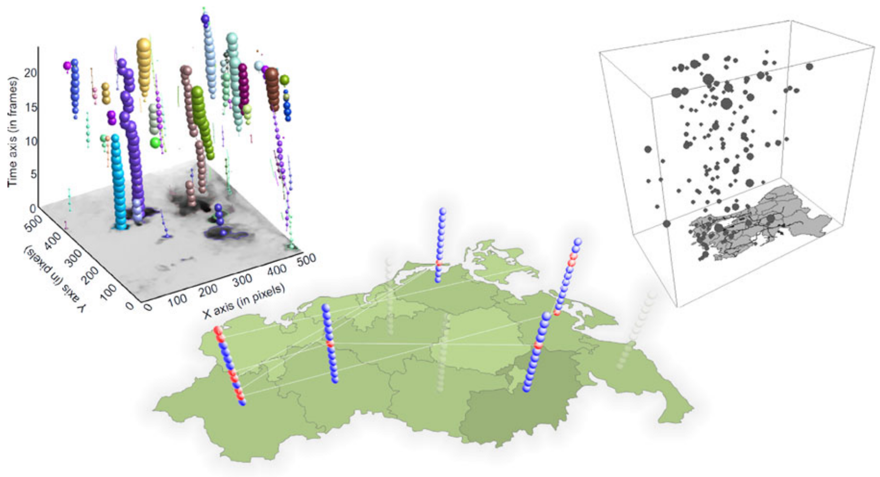


Fig. 7.95: Events in space and time are visualized by embedding graphical objects of varying size and color into space-time cubes. From left to right, the cubes show events related to convective clouds, human health data, and earthquakes.
Source: Left: Turdukulov et al. (2007), © 2007 Elsevier. Used with permission. Center: Generated with the LandVis system. Right: Gatalsky et al. (2004), © 2004 IEEE. Used with permission.

Events usually describe happenings of interest. In order to analyze events in their spatial and temporal context, one can make use of the space-time cube concept (↪ p. 245). The actual events are visualized by placing graphical objects in the space-time cube at those positions where events are located in time and space. Attributes associated with events can be encoded, for example, by varying size, color, shape, or texture of the graphical objects. Marking events in a space-time cube is a general concept with a wide range of applications: Turdukulov et al. (2007) explore events related to the development of convective clouds, Tominski et al. (2005) consider maxima in human health data as events of interest, and Gatalsky et al. (2004) visualize earthquake events.

References

Gatalsky, P., Andrienko, N., and Andrienko, G. (2004). Interactive Analysis of Event Data Using Space-Time Cube. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 145–152, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

Turdukulov, U. D., Kraak, M.-J., and Blok, C. A. (2007). Designing a Visual Environment for Exploration of Time Series of Remote Sensing Data: In Search for Convective Clouds. *Computers & Graphics*, 31(3):370–379.

Space-Time Path

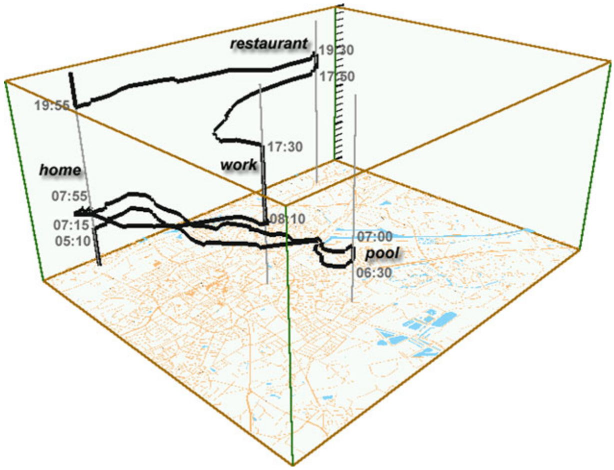


Fig. 7.96: A space-time path is embedded into a space-time cube and shows a person's movement For better orientation, important places are marked by vertical lines and annotations.
Source: Kraak (2003), © 2003 International Cartographic Association (ICA). Used with permission.

The space-time path is a specific representation of data in a space-time cube (↪ p. 245). The roots of the concept of space-time paths can be found in the work by Lenntorp (1976). Kwan (2009) describes contemporary visual representations that are based on the classic concept. A space-time path is constructed by considering the location of an object as a three-dimensional point in space and time. Multiple such points ordered by time describe the path that an object has taken. The path can be rendered as a polyline that connects successive points. In order to encode data along a space-time path, one can vary the line's color, use differently dashed line segments, or employ other visual attributes. Alternatively, a space-time path can be represented as a three-dimensional tube, where the tube's radius can be varied to encode additional data values. Today's implementations usually offer interaction to allow for virtual movements through space and time, or for rotation and zoom.

References

Kraak, M.-J. (2003). The Space-Time Cube Revisited from a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages 1988–1995, Newcastle, UK. The International Cartographic Association (ICA).

Kwan, M.-P. (2009). Space-Time Paths. In Madden, M., editor, *Manual of Geographic Information Systems*, chapter 25, pages 427–442. American Society for Photogrammetry and Remote Sensing, Bethesda, MD, USA.

Lenntorp, B. (1976). Paths in Space-Time Environments: A Time Geographic Study of Movement Possibilities of Individuals. In *Lund Studies in Geography*, number 44 in *Series B: Human Geography*. Royal University of Lund, Lund, Sweden.

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static, dynamic**
dimensionality: **3D**

GeoTime

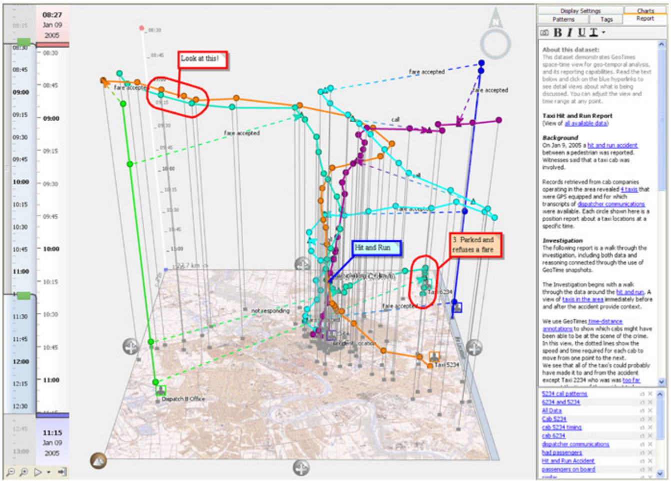


Fig. 7.97: The visualization shows taxis involved in a hit and run accident as colored points and paths. Patterns of interest are annotated in the scene, and a narrative can be authored on the right. *Source: Image courtesy of William Wright. GeoTime is a registered trademark of Oculus Info Inc.*

Kapler and Wright (2005) describe GeoTime® as a system to visualize data items (e.g., objects, events, transactions, flows) in their spatial and temporal context. It provides a dynamic, interactive version of the space-time cube concept (↪ p. 245), where a map plane illustrates the spatial context and time is mapped vertically along the third display dimension. Items and tracks are placed in the space-time cube at their spatial and temporal coordinates. GeoTime provides a variety of visual and interactive capabilities. Time intervals of interest can be selected by the user and events are smoothly animated along the time axis. Alternative projections of the display allow users to focus more on either temporal or spatial aspects. Notable about GeoTime are its annotation, storytelling, and pattern recognition features (see Eccles et al., 2008). They enable automatic as well as user annotation of the representation with findings, as well as the creation of stories about the data for analytic exploration and communication. Additional functionality allows the analysis of events and transactions in time above a network diagram (see Kapler et al., 2008).

References

Eccles, R., Kapler, T., Harper, R., and Wright, W. (2008). Stories in GeoTime. *Information Visualization*, 7(1):3–17.

Kapler, T., Eccles, R., Harper, R., and Wright, W. (2008). Configurable Spaces: Temporal Analysis in Diagrammatic Contexts. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 43–50, Los Alamitos, CA, USA. IEEE Computer Society.

Kapler, T. and Wright, W. (2005). GeoTime Information Visualization. *Information Visualization*, 4(2):136–146.

Pencil Icons

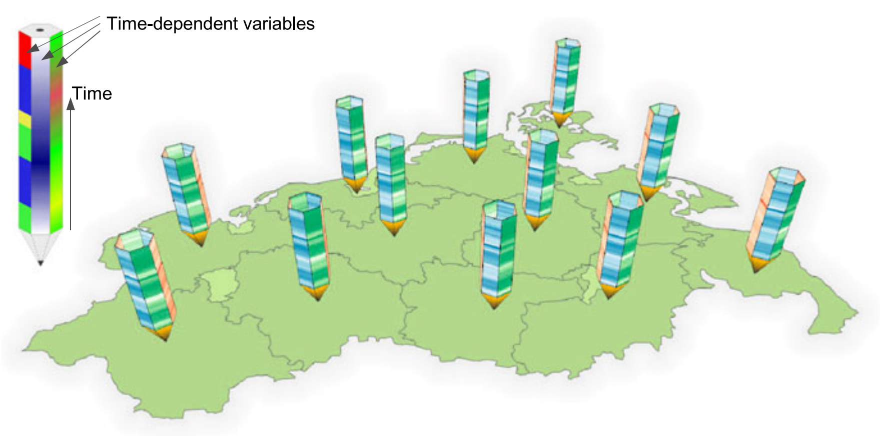


Fig. 7.98: Multiple time-dependent variables are mapped onto the faces of pencil icons to visualize temporal dependencies in the data. By placing the icons on a map, the spatial dependencies are communicated.
Source: Generated with the LandVis system.

Pencil icons have been developed by Tominski et al. (2005) to visualize multivariate spatio-temporal data. The technique is based on the space-time cube concept (↪ p. 245), where the spatial frame of reference is represented as a map in the x-y plane of a virtual three-dimensional cube. The dimension of time is mapped along the cube's z-axis. Within the cube, pencil icons are positioned where data is available. This way, the spatial context is communicated. Each pencil icon represents the temporal context and multiple time-dependent variables by mapping time along the pencil, starting at the tip, and by associating each face of the pencil with an individual time-dependent variable. Color coding is applied to visualize the data. Color lightness is varied according to data values, and different hues are used to help users identify particular variables. The linear shape of the pencil is suited to represent linear characteristics of the underlying time axis. Heterogeneous data can be depicted by using appropriate color scales. In order to deal with occlusion and information displayed at the pencils' back faces, several interaction techniques are provided, including navigation in the virtual world as well as the individual and linked rotation of pencils.

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

References

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **spatial**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **2D, 3D**

Data Vases

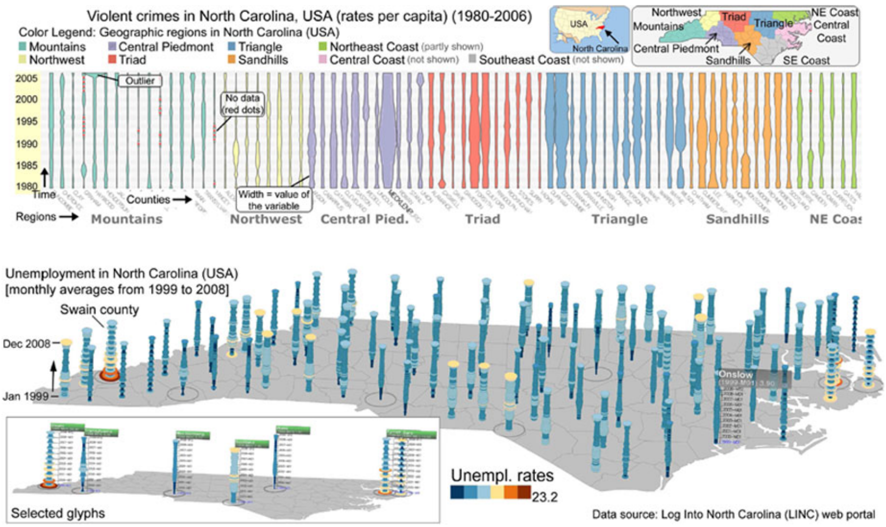


Fig. 7.99: Shape and color of a data vase encode time-varying data values. The spatial component of the data can be communicated by a vertical alignment of 2D data vases (top), or by embedding 3D data vases into a space-time cube (bottom).
Source: Thakur and Hanson (2010), © 2010 IEEE. Used with permission.

The data vases technique has been designed to visualize multiple time-varying variables. Thakur and Rhyne (2009) describe two alternative designs: a 2D and a 3D variant. A 2D data vase is basically a graph constructed by mirroring a line plot (↪ p. 153) against the time axis, effectively creating a symmetric shape that can be filled (segment-wise) with a data-specific color. Such data vases can then be arranged on the screen to create a meaningful visualization. For spatio-temporal data, one can use multiple vertically aligned data vases, each of which represents an individual geographic region. Thakur and Hanson (2010) further elaborate the idea of extending data vases to the third dimension. In 3D, data vases are constructed by stacking discs along a vertical time axis, where each disc maps the data for a particular time primitive to disc size and color. Such 3D data vases can then be embedded into a space-time cube (↪ p. 245), i.e., a virtual 3D world where two dimensions are used to show a geographic map and the third dimension encodes time.

References

Thakur, S. and Hanson, A. J. (2010). A 3D Visualization of Multiple Time Series on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 336–343, Los Alamitos, CA, USA. IEEE Computer Society.

Thakur, S. and Rhyne, T.-M. (2009). Data Vases: 2D and 3D Plots for Visualizing Multiple Time Series. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 929–938, Berlin, Germany. Springer.

Wakame

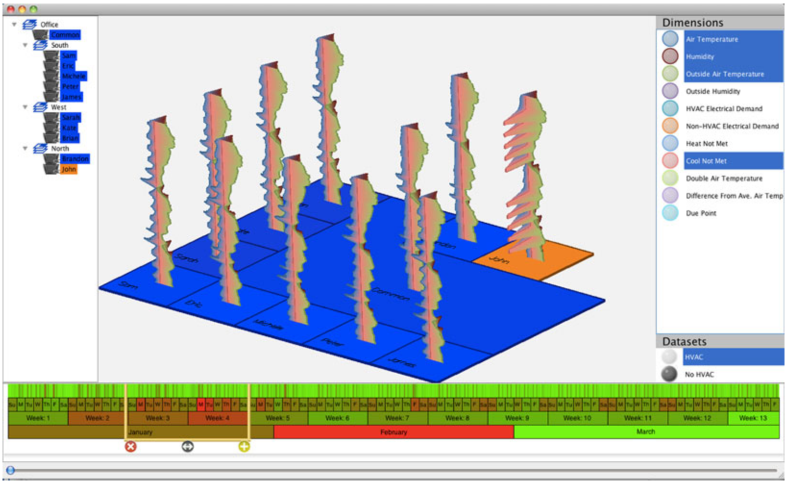


Fig. 7.100: Three-dimensional visualization objects, so-called Wakame, show multiple variables along time. Placing multiple Wakame on a map allows analysts to make sense of spatio-temporal data. A time scale widget on the bottom can be used to select intervals of interest.
Source: Image courtesy of Clifton Forlines.

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

Forlines and Wittenburg (2010) describe an interactive system for visualizing multivariate spatio-temporal data. The temporal aspects are encoded to multivariate glyphs, so-called Wakame. A single Wakame basically corresponds to a radar chart that has been extruded along the third dimension. In a radar chart, different variables are represented on radially arranged axes that are connected to form a polyline. Wakame are constructed as solid three-dimensional objects whose shape indicate temporal trends and relations among time-dependent variables (also see Kiviat tube, ↪ p. 211). Embedding multiple Wakame into a map display facilitates the understanding of spatial aspects. What is noteworthy about the Wakame system are its interaction and animation facilities. An intelligent camera positioning mechanism supports users in finding perspectives on the Wakame that most likely bear interesting information. A hierarchical time axis widget (↪ p. 229) denotes by color how "different" each time primitive is from its neighbors. This allows users to pick interesting time primitives easily. Upon interaction, animation is used to smoothly interpolate between views. Additional animation schemes are offered to switch between the three-dimensional Wakame view and traditional two-dimensional radar charts and line plots (↪ p. 153), which might be better suited for certain analysis tasks.

References

Forlines, C. and Wittenburg, K. (2010). Wakame: Sense Making of Multi-Dimensional Spatial-Temporal Data. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 33–40, New York, NY, USA. ACM Press.

data

frame of reference: **spatial**
variables: **multivariate**

time

arrangement: **cyclic**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **3D**

Helix Icons

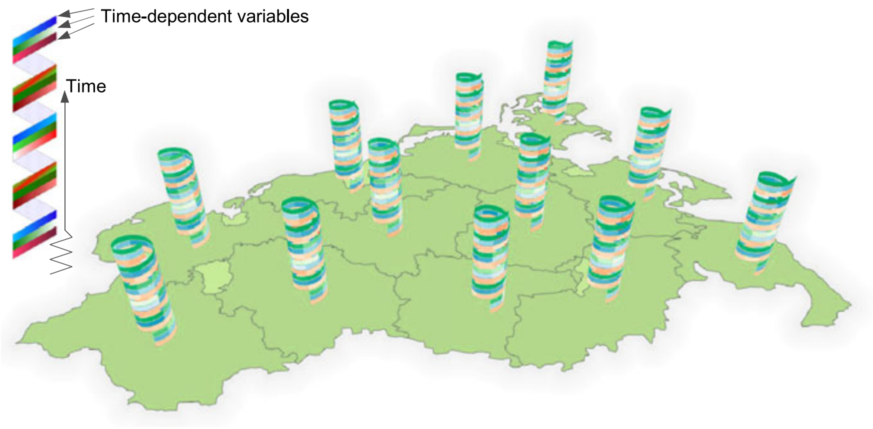


Fig. 7.101: Helix icons use color-coding to visualize multivariate spatio-temporal data along helix ribbons, which emphasize the data's cyclic temporal character. The spatial aspect of the data is illustrated by embedding helix icons in a space-time cube.
Source: Generated with the LandVis system.

Helix icons by Tominski et al. (2005) are useful for emphasizing the cyclic character of spatio-temporal data. The underlying model of this technique is the space-time cube (↪ p. 245), which maps the spatial context to the x-axis and the y-axis, and the dimension of time to the z-axis of a virtual three-dimensional cube. The actual data visualization is embedded into the cube. To this end, a helix ribbon is constructed to unroll the time domain along the z-axis. Each segment of the helix ribbon visualizes a specific instant (or interval) in time by means of color-coding. Multiple time-dependent variables can be visualized by subdividing the helix ribbon into narrower sub-ribbons, each of which represents a different variable. Using unique hues for each sub-ribbon helps the user distinguish variables. As for spiral graphs (↪ p. 185), interaction techniques help users in finding an appropriate number of segments per cycle so that periodic patterns in the data are revealed. The inherent 3D representation problems (i.e., information displayed on helix back faces or inter-icon occlusion) are dealt with by offering 3D navigation through the space-time cube and rotation of helix icons.

References

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

7.2 Summary

This chapter reviewed 101 existing techniques for visualizing time and time-oriented data. Since there are observable balances and imbalances with regard to our categorization, it is worth taking a closer look at possible explanations.

Data – frame of reference In this book we mainly focus on abstract data, which is also reflected by our survey. Showing time-oriented data in a spatial frame of reference significantly increases the design efforts because more information has to be packed into the visual mapping. Particularly, the disciplines of cartography and geo-visualization, which are established, independent fields of research, have developed approaches to combining the visualization of temporal and spatial aspects of data (see Kraak and Ormeling, 2003; Andrienko and Andrienko, 2006).

Data – variables The number of techniques for univariate and multivariate data are almost balanced. While classic techniques often consider simpler univariate data, modern approaches take on the challenge of dealing with multiple variables. The survey also contains several techniques that cope with multiple variables simply by the repetition of a basic visualization design that only addresses univariate data (↪ p. 180).

Time – arrangement Most of the techniques in the survey support linear time; the approaches with cyclic time are significantly outnumbered. Reasons for this might be that users are usually interested in trends evolving from past, to present, to future, rather than in finding cycles in the data. The latter aspect, however, is important to fully understand the data, and therefore, expert data analysts need effective cyclic representations as well (↪ p. 186).

Time – time primitives Instants in time are the most commonly used time primitive in our survey. This seems natural because data are often measured at a particular point in time. Intervals occur less often, for example, in planning scenarios, where it is important to know how long certain activities will take (↪ p. 172). And as soon as it becomes necessary to abstract from individual instants to intervals in order to deal with bigger and bigger datasets, we have to prepare the visual representation accordingly (↪ p. 230).

Vis – mapping Apparently, the static pages of a book are better suited for showing static techniques. In this sense, our survey is a bit biased in that it contains mostly static approaches. However, dynamic animation is equally important and often it is the first solution offered when time-oriented data have to be visualized. Animation can also be an option in combination with static methods to extend the capacity of a technique in terms of the data that can be handled (↪ p. 248).

Vis – dimensionality Two-dimensional visual representations are often preferred over three-dimensional ones, because they are more abstract and thus easier to understand. Especially techniques developed in the early days of computer graphics tend to stick with two dimensions simply due to the limited computing power available then. However, modern technologies have made it easier both for visualization designers to implement three-dimensional visualization, and for visualization users to navigate and explore virtual 3D visualization spaces. This is particularly useful when data with spatial references have to be visualized (↪ p. 245).

There is another significant fact that can be derived from the survey: Most approaches address the model of an *ordered* time domain, while only a few of them explicitly consider the visualization of *branching* alternative strings of time, and none of them is capable of visualizing data that are based on the model of *multiple perspectives*. Therefore, particularly branching time and time with multiple perspectives deserve more research attention in the future.

From the review we can also see that some general concepts reoccur in several instantiations, as for instance the general application of line plots as the most basic visualization of time-dependent data, the utilization of the third display dimension to encode time, or the mapping of time to spiral shapes in order to visualize cyclic aspects. We also see that quite a number of publications is specific to a particular *what* and *why*, and as a consequence represent tailored solutions in terms of *how* the data are visualized. On the one hand, specific solutions are highly adapted and fine-tuned to be successful in supporting a specific set of users that try to solve a particular problem. On the other hand, however, these solutions are hard to adapt and reuse for other visualization problems, even when a new problem is similar to the original one and differs only in one aspect of our categorization schema. Therefore, existing techniques often lack broader applicability.

From the application perspective – a perspective that many users share as their day-to-day work is to make sense of constantly changing data – a general framework would be favorable. What other challenges need to be addressed in the future and what such a framework could look like will be discussed in the next chapter.

References

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

Kraak, M.-J. and Ormeling, F. (2003). *Cartography: Visualization of Geospatial Data*. Pearson Education, Harlow, England, 2nd edition.

Norman, D. A. (1993). *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Chapter 8
Conclusion

> 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)

This book has dealt with concepts and methods for visualizing time and time-oriented data. In this chapter, we will briefly summarize the key aspects that have been elaborated so far. We will further consider the current state of visualization of time-oriented data and identify application issues and research challenges. We conclude with the description of a blueprint of concepts that may serve as a basis for tackling the identified problems in the future.

8.1 Summary

Most of the existing techniques for computational analysis and visualization have been developed to deal with numbers. In practice, these numbers are often anchored in space and time. Depictions of a spatial frame of reference lie at the core of cartography and geo-visualization, and these disciplines have yielded many powerful visualization techniques. However, until now, no independent research field which focuses on the visualization of time-oriented data has been established. Nevertheless, visual representations of time and time-oriented data have a long and venerable history. This fact has been illustrated by means of several classic examples from the pre-computer era in Chapter 2.

Designing appropriate visual representations for time-oriented data requires responding to the questions:

- *What* is presented?
- *Why* is it presented?
- *How* is it presented?

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_8,

So, in order to adequately address the *what* aspect, one needs to consider the specific characteristics of the time domain as well as the characteristics of the data that are related to time. In Chapter 3, we used four principal criteria to categorize the characteristics of time: the scale (ordinal vs. discrete vs. continuous), the scope (linear vs. cyclic), the arrangement (point-based vs. interval-based), and the viewpoint (linear vs. branching vs. multiple perspectives). To characterize the data, four major criteria have been used: the scale of variables (qualitative vs. quantitative), the frame of reference (abstract vs. spatial), the kind of data (events vs. states), and the dimensionality (univariate vs. multivariate). We further defined time primitives that act as a kind of glue between time and data. Time primitives are often organized in hierarchically structured calendar systems to accommodate different levels of temporal granularity (e.g., seconds, minutes, hours).

With the *why* question user tasks come into play. We addressed this in Chapter 4. At the task level we distinguished lookup and comparison tasks, direct and indirect search, as well as elementary (performed on individual values) and synoptic tasks (performed on sets of values). 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.

General principles of *how* time and time-oriented data can be visualized were presented in Chapter 4 as well. Two basic distinctions were made: the dimension of time can be represented using the display space (i.e., static representation) or physical time (i.e., dynamic representation), where the presentation space can be either two-dimensional or three-dimensional. We discussed these categories in detail and gave various examples of visualization designs addressing specific aspects of the data level, the task level, and the presentation level.

Visual exploration and analysis of time-oriented data also requires interaction methods that allow users to manipulate the visual representation in a variety of ways, including navigation of time and data, adjustment of graphical encoding and spatial arrangement, selection of data of interest, filtering out irrelevant data, and many more. Chapter 5 provided a compact overview of interaction.

Moreover, analytical methods have to be provided for supporting the generation of expressive visual representations. Among other purposes, analytical methods are useful for computing data abstractions that may serve to cope with large volumes of data or to allow visual analysis at different levels of granularity. Chapter 6 was dedicated to the aspect of analytical support.

As diverse and manifold as time and data characteristics and visualization design choices, interaction concepts, and analytical methods are, the body of available visualization techniques for time and time-oriented data is equally diverse. In Chapter 7, we summarized many classic and state-of-the-art techniques, some of a general nature, others very specific, and some well-established, and others more progressive or alternative. We categorized all of these techniques according to six major criteria, described briefly the main idea behind the techniques, and provided illustration examples. The result is a compact overview of the variety of existing visual representations for time and time-oriented data.

In conclusion, the survey gives evidence that time is indeed an important dimension that deserves special treatment in visual, interactive, and analytical methods. However, tools or systems that provide the broad functionality demanded specifically for time-oriented data are not available. Moreover, there are several open issues to be addressed in the future. In the next sections, we will take a look at these issues from an application perspective as well as from a research point of view.

8.2 Application Issues

A main concern from an application perspective is to bridge the gap between the development of powerful visual methods on the one hand, and their integration into the real-life workflows in different application scenarios on the other hand. This requires research which addresses the problems in terms of both software issues and user issues.

Software issues – systems & formats

There are a variety of commercial and open source visualization systems, as for instance, Tableau[1], Spotfire[2], MagnaView[3], or vtk[4]. Many of the available systems provide excellent support for visual exploration and analysis of multivariate data. However, the specifics of time are not always easy to handle, because time is treated as just another quantitative variable or because the software lacks support for the wide range of characteristics which are relevant when dealing with time (e.g., support for cyclic time or for different time primitives). As a result, in order to visually explore temporal dependencies, the user has to manually find an appropriate visual mapping that emphasizes the time axis. Moreover, this means it is difficult or actually not feasible for users to apply a particular system for all the different characteristics of time discussed before.

On the other hand, visualization research has yielded powerful research prototypes that provide dedicated support for the time aspect. A prominent example in this regard is the TimeSearcher[5] project for visual exploration of time-series data (↪ p. 188). However, the integration of such prototypes into the infrastructure of day-to-day business is usually problematic and requires additional effort. Furthermore, research prototypes are usually not designed to cover all aspects of time, but instead to address only particular cases – mostly the visualization of linear and ordered time domains.

---

1 http://www.tableausoftware.com/; Retrieved Feb., 2011.

2 http://spotfire.tibco.com/; Retrieved Feb., 2011.

3 http://www.magnaview.nl/; Retrieved Feb., 2011.

4 http://www.vtk.org/; Retrieved Feb., 2011.

5 http://www.cs.umd.edu/hcil/timesearcher/; Retrieved Feb., 2011.

Another significant problem to be solved is caused by the diversity of existing data formats and interfaces. Processes that generate or collect data and tools that manipulate or analyze the data often use specific databases and data formats that meet the requirements of the particular application scenario. Software tools for visualizing the data and interacting with them often use different formats. This circumstance requires individual and possibly complex data transformations, which represent a substantial obstacle. To this end, researchers who develop more comprehensible and simplified data interfaces will be rewarded with greatly expanded user communities.

User issues – knowledge & support

Besides improving the technical basis, it is important to take the needs of the users into account. In this regard, an important point is to improve the awareness about new visualization and interaction methods. Nowadays, users mostly apply traditional visualization techniques such as line plots or bar graphs. These techniques are well-established and have proven to be useful. However, new innovative visualization methods allow the representation of a larger number of variables and data values, provide comprehensive interaction functionality, and take the specific aspects of time into account. These new possibilities can lead to new findings.

Moreover, in many application domains, visual methods are primarily used to present results. This is an adequate strategy for all those analytical problems whose solution can be computed automatically and that do not require input from the user to generate analysis results. However, many analytical problems do not have a single closed solution, but rather require an interactive visual exploration process that integrates the user tightly throughout the analysis workflow.

Another point to mention is that typically users in specific application domains are the ones to create visual representations. This implies that they have to know which visual representation should be used for which task. However, although these users have a strong domain-specific background, it can not be assumed that they are also experts in visualization design. If users were better supported by the visualization systems in choosing expressive, effective, and appropriate visualization techniques, the quality of information display and analysis results could greatly improve. To this end, facilities for specifying data characteristics and analytic tasks are mandatory.

Moreover, cumbersome data transformations and extensive switching between application and visualization systems are substantial obstacles for widespread use. In fact, interactive visualization methods for time-oriented data have to be integrated into application portals and systems in order to allow domain experts to use these techniques effortlessly and seamlessly.

To summarize, bridging the gap between research on interactive visualization methods and their application requires both imparting an awareness of the variety of possibilities and providing means to effectively use them within a given application infrastructure.

8.3 Research Challenges

Designing appropriate visual representations and tools for time-oriented data as well as making them applicable in real world problem solving scenarios requires further scientific investigation. In the following, we will discuss several aspects in this regard.

Problem specification & user guidance

In the field of software engineering it is generally acknowledged that the first step in developing tools and user interfaces should be a sound analysis of the given problem domain (see Hackos and Redish, 1998; Courage and Baxter, 2005). The same applies for designing visual representations. However, most of today's visualization systems do not provide any means of describing the visualization problem. In the case of visually analyzing time-oriented data, this means (1) the characteristics of time and associated time-oriented data as well as (2) the intentions and tasks of users have to be specified. If appropriate descriptions are provided to store this knowledge, visualizations can be realized that suit the data and the tasks. Although first approaches for automatic visualization design have been developed (see Mackinlay, 1986; Senay and Ignatius, 1994; Wills and Wilkinson, 2010), further research is needed to allow an easy-to-use specification of data and tasks and the provision of adequate descriptors, in this way enabling the automatic suggestion and computation of appropriate visual representations. The aim is to guide the users, rather than burden them with technical details. Thus, a significant shift could be realized from a technique-centered view to a user-centered view, i.e., a view that puts the user into the focus (see Kerren et al., 2007).

New visualization methods

Choosing adequate visualizations is one point, but providing a set of suitable techniques that cover all the different aspects of time is another concern. Although a large diversity of powerful visualization techniques for time-oriented data have been developed, most of them support only certain parts of the introduced time and data categorization. In the particular case of visualizing multivariate data, usually linear, point-based, and ordered time domains are assumed. Further investigations are required, including the development of techniques for interval-based time, branching time, and multiple perspectives, for simultaneously displaying raw data and data abstractions, and for showing the time-oriented data in their spatial frame of reference.

Another important challenge to be considered originates from the hierarchical nature of time, which leads to the situation that data may be given at multiple levels of granularity. New interactive visualization techniques are required to allow analysts to combine different levels of data and time and to switch between the levels.

This specific challenge for time-oriented data is related to the more general open research topic of handling variables given at multiple scales (see Keim et al., 2010).

Data quality & data provenance

We have primarily considered visual representations that show the data themselves, rather than information on data quality or data provenance. However, taking uncertainties and provenance into account will significantly improve the expressiveness of visual representations, and also strengthen the user's confidence in the findings made. In the case of time-oriented data, uncertainties of the data and uncertainties of the temporal frame of reference have to be communicated. Currently, visualization methods communicating both time and data uncertainties are not available. Therefore, new visualization strategies have to be developed. The concern of representing the quality of data has also been acknowledged as an open research topic in visualization research in general (see Keim et al., 2010).

Scalability

Apart from issues concerning data quality, the fast growing quantities of data also pose a considerable challenge to visualization. Not too long ago researchers and practitioners were exploring and analyzing acquired or simulated datasets of modest size (e.g., from kilobytes to a few megabytes). Nowadays, it is common to focus on problems involving enormous amounts of data in gigabytes, terabytes, petabytes, or more (see Ward et al., 2010). This calls for particular characteristics and functionalities of data management as well as of the visual and algorithmic design (see Keim et al., 2010). With regard to time-oriented data, we need to be able to deal with both very long time-series containing vast amounts of time primitives (e.g., covering large time spans and/or at very fine-grained time scales) and large numbers of time-dependent variables in parallel. If the visualization and interaction techniques as well as the analytical methods are able to cope with such large quantities of time-oriented data, we will be able to extend the frontiers of complex and demanding research areas, such as biomedicine or climate research.

Novel interaction methods

Interaction methods are essential to allow users to explore time-oriented data as well as the space of possible visual encodings. Obviously, the features of time-oriented data influence the interaction techniques and tasks carried out on such data. Navigating in time and switching between different levels of temporal granularity are mandatory when interacting with time-oriented data, but are rather uncommon when interacting with abstract quantitative variables. The visualization of larger datasets benefits from visual overviews and the ability to drill down into areas of interest while preserving orientation within the information space. Interacting directly with visual representations and analytical methods provides more control and tighter feedback for the human analyst.

However, interaction techniques for time-oriented data mostly consider linear, point-based time domains. More research is needed to enable users to interact with visual representation of cyclic time, branching time, or multiple perspectives. This also includes reasoning about what it means to move in cycles, along branches, or in multiple perspectives. Moreover, support is needed to structure the interactive exploration process and to provide guidance in terms of where to go next or which encoding to choose. It is important that these investigations be made in accordance with the users' demands. A recent workshop on "Interacting with Temporal Data" organized at the ACM CHI Conference 2009 gives evidence of the growing interest in this research topic (see Mackay et al., 2009).

Advanced analytical methods

Most analytical methods for time-oriented data available today treat time as a flat, ordered sequence of events. Thus, these methods are lacking information about the time intervals between events or about after how much time a particular pattern will reoccur. But, the natural structures of time, such as years and seasons, as well as social structures of time, such as weeks and business days, can strongly influence what findings can be extracted from time-oriented data. For example, the pattern of monthly sales may vary largely due to differences in the arrangements of workdays, weekends, and holidays. However, only few existing analytical methods, like for example the seasonally adjusted autoregressive integrated moving average (SARIMA), model cyclic temporal behavior adequately. As a consequence, better support for dealing with the hierarchical and cyclical structures of time as well as their semantics (i.e., accounting for the specifics of time) is needed.

Moreover, many analytical methods are like black-boxes which accept some time-oriented data as input and generate some analytic result as output. However, it remains largely unexplored how to parameterize analytical methods appropriately to adapt to the given data and tasks, or how to combine multiple analytical methods to generate better analysis results. To solve these problems, the black-boxes must be made transparent and steerable for the user, where sufficient support by the visualization system should be taken for granted.

Evaluation

In order to be able to automatically suggest techniques to the user or to automatically adapt chosen techniques to data and tasks, we need to know which techniques are *good*. This requires evaluation. Evaluation has to be conducted in terms of the three criteria expressiveness, effectiveness, and appropriateness (see Chapter 1). Expressiveness and effectiveness are related to the data level and the task level, respectively. They require testing whether the characteristics of time and data are sufficiently communicated, and whether the visual representation matches the tasks, expectations, and perceptual capabilities of users. With the appropriateness criterion, resources come into play. Moreover, the application domain has to be taken into account to address the appropriateness criterion.

Because thorough evaluation requires a combined consideration of multiple criteria, Munzner (2009) introduced a nested model for visualization design and validation. She proposed to subdivide the generation of visual representations into four nested levels (i.e., characterization of data and tasks, abstraction into operation and data types, design of encoding and interaction, and development of algorithms) and argues that distinct evaluation methodologies should be used for each level of the model. Plaisant (2004) gives an overview of currently applied evaluation methods and points out challenges specific to the evaluation of information visualization. These challenges also apply to evaluating time-oriented data visualizations. Although general evaluation methods (see Lazar et al., 2010) and methods tailored to visualization, for example, to measure effectiveness (see Zhu, 2007), are readily available, more research is needed in terms of specifically evaluating the encoding of and the interaction with time and time-oriented data. To this end, it is necessary to develop new evaluation strategies that are tailored to the requirements of visualizing time-oriented data.

Multi-user & heterogeneous display environments

Typically, visualization and interaction techniques are designed to be used in single user and single display scenarios. However, modern technologies and infrastructures enable analysts to use multi-touch displays with tangible interaction, to operate large-display environments with virtual interaction, or to work in smart environments, which are heterogeneous conglomerates of dynamically linked devices. On the one hand, technological progress opens up new possibilities. For example, one can imagine multi-user and multi-display problem solving scenarios, where a large-scale display shows an overview of a large time-oriented dataset, while multiple user groups discuss selected details from different parts of the dataset shown on multiple linked tabletop displays. On the other hand, new research questions and challenges have to be addressed. From a technical point of view, there is a need to scale visualization, interaction, and analytical facilities to the given resources (display, interaction, and computation devices) and to the intended audience (single user, small or large user groups). From a user perspective, research is necessary to investigate how multiple users can work together to analyze time-oriented data cooperatively. On a more general level, these issues are related to the challenges of display scalability and human scalability as described by Thomas and Cook (2005).

Non-visual mapping and accessibility

Apart from the numerous options of visually representing time-oriented data we have seen throughout the book, other forms of representing time are possible. Time could for example be mapped to sound or to haptic sensations as with braille interfaces. Smell and flavor might also be candidates for such a mapping. Despite the fact that these mappings are in principle imaginable, their feasibility and usefulness have to be investigated. Particularly, this also addresses accessibility by users with disabilities, especially blind users. Speeth (1961) already showed how seismographic data can be presented in an auditory display. An example of a recent attempt in this direction is a system for data sonification by Zhao et al. (2008) to explore spatial data for users with visual impairments. Apart from being an essential issue for government agencies this is also a fruitful research topic that can lead to novel insights.

Intertwining visual, interactive, and analytical methods

Finally, there is a challenge that touches – or better, that encompasses all previously mentioned concerns: In order to facilitate exploration and analysis of time-oriented data, we should not consider visual, interactive, and analytical methods in isolation, but instead should strive for a tight intertwining of them, effectively utilizing their strengths and compensating their weak spots. In the next section, we give an outlook in this direction.

8.4 Visual Analytics

The development of new methods is only one side of the coin; combining them efficiently and integrating them into the workflow of the user is the other side. The new research field of *visual analytics* aims to combine visual, interactive, and analytical methods within real scenarios to solve real application problems (see Thomas and Cook, 2005; Keim et al., 2010).

A basic precondition for achieving this goal is to provide an open, extensible framework that:

- comes with interchangeable building blocks for visualization, interaction, and analysis of time-oriented data,
- uses commonly agreed-upon interfaces for flexible combination of methods,
- provides descriptors for specifying available techniques as well as data and user tasks, and
- offers an easy-to-use user interface for interactive exploration, including on-demand guidance.

Designing such a comprehensive framework is a formidable research challenge, not to mention the effort required to actually implement the framework's broad functionality. To make a very first step in this direction, we sketch a basic design concept in Figure 8.1.

Data component The data component encapsulates all data-related aspects, including data import from heterogeneous sources, flexible data management, and efficient data search and retrieval. Such functionality is typically realized by database management systems (DBMS), where the underlying paradigm can be manifold (e.g., relational model, temporal database, key-value store, column store, graph database, etc.).

The data component contains time-oriented data, which typically come from various external data sources in different idiosyncratic formats. DBMS functionality can be used to consolidate, unify, and combine the different data sources.

In order to support the seamless interplay of all of the components of the framework, appropriate specifications are needed. This is reflected by the descriptor database. This database includes metadata about the data under investigation such as characteristics of the time domain and characteristics of associated time-oriented data (see Chapter 3).

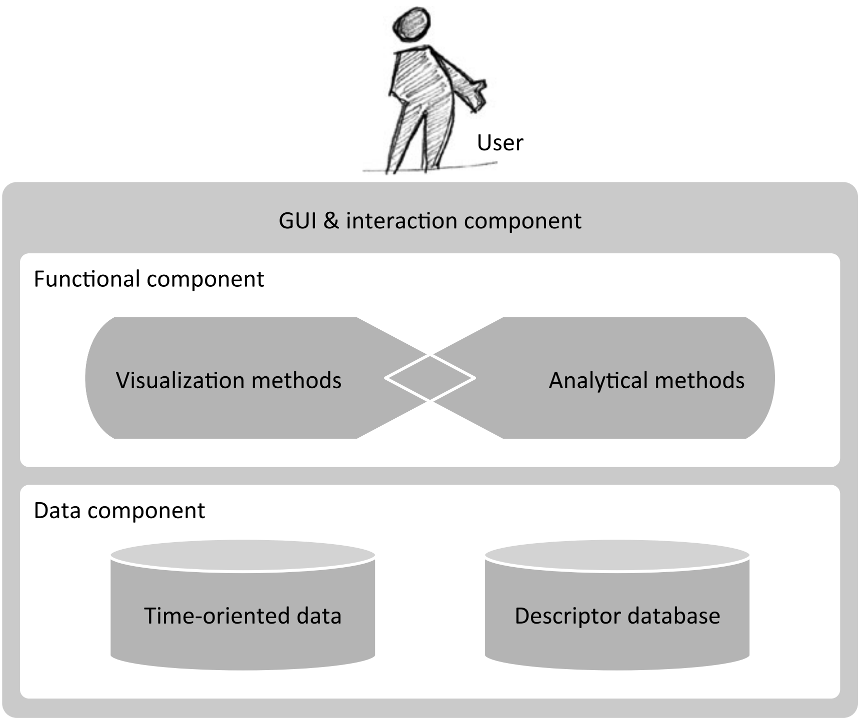


Fig. 8.1: Conceptual framework for visual analytics of time-oriented data.

Besides the mandatory data specification, the descriptor database should provide further useful information such as a description of user tasks and a specification of the capabilities of visual, interactive, and analytical methods. Unfortunately, these latter aspects are rarely supported by current systems. But only if data, tasks, and capabilities are properly specified is it possible to automatically suggest suitable (combinations of) visualization, interaction, and analysis methods for any particular combination of data and tasks. An example for the potential of automatic adaptation according to some specification is the concept of event-based visualization as described in Section 5.4.

An interesting question is how to collect the information to fill the descriptor database. Although some information can be generated automatically, for example, by computing statistics such as minimum, maximum, or data distribution, in most cases it is necessary to resort to manual specification. Therefore, the framework should contain support for both automatic processing to create descriptors as well as interactive specification and refinement to complete the descriptors. Furthermore, there is a need to change descriptors dynamically during the visual analysis process. For example, analysis results may lead to new hypotheses about the data, which in turn manifest themselves in the formulation of new user tasks, in a switch of interests, or in the need to access additional data, which affects the generation of metadata.

Functional component The heart of the framework is given by the functional component that includes visualization methods and analytical methods. The set of visualization methods can be filled with the techniques described in Chapter 7. The set of analytical methods includes computational approaches for analyzing and mining time-oriented data as described in Chapter 6.

GUI & interaction component The third and very important component consists of an all-encompassing graphical user interface and corresponding interaction methods. This component provides the means for embedding visual analytics in day-to-day workflows and for allowing for interactive exploration and analysis of the time-oriented data as discussed in Chapter 5.

In the previous two paragraphs we used only a few lines to discuss the functional component (visualization and analysis) and the GUI & interaction component, because these aspects have been detailed in separate chapters in this book. Notably, there are many different techniques available, but they are considered in isolation. The big challenge of visual analytics is to facilitate smooth interoperability among different approaches to the exploration and analysis of time-oriented data.

If we arrive at commonly agreed-upon interfaces, the building blocks and methods of the framework can be integrated into application portals and thus into the workflows of users. If we succeed in providing appropriate descriptors, automatic selection of suitable techniques and corresponding configurations can be accomplished, effectively guiding the user throughout the visual analysis process. This leads to the new paradigm of making the users and their tasks the focal point, where a central aspect is to assemble visual analytics tools automatically as illustrated in

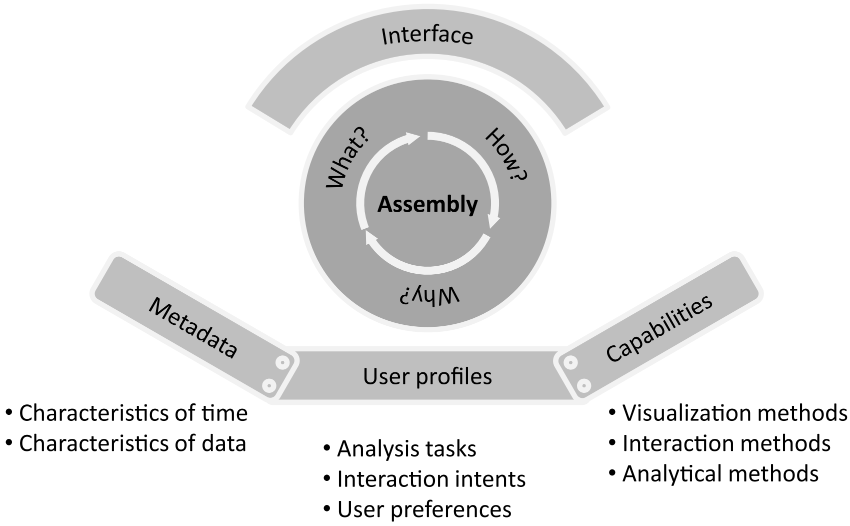


Fig. 8.2: Automatic assembly of visual analytics components.

Figure 8.2. This automatic approach aims to relieve users of the burden of compiling solutions manually. As a result, users can focus on accomplishing tasks on the data by utilizing the power of analytical methods to verify hypotheses and the flexibility of interactive exploration to discover unexpected findings.

Implementing such a user-centric view is one of the most challenging problems identified by the current research agenda of visual analytics (see Keim et al., 2010). First steps have been taken to solve this problem and contributing to an improved analysis of time-oriented data is one of them.

References

Courage, C. and Baxter, K. (2005). *Understanding Your Users*. Morgan Kaufmann, San Francisco, CA, USA.

Hackos, J. T. and Redish, J. C. (1998). *User and Task Analysis for Interface Design*. John Wiley & Sons, Inc., New York, NY, USA.

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.

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

Mackay, W. E., Van Kleek, M. G., and Tabard, A. (2009). Interacting with Temporal Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 4783–4786, New York, NY, USA. ACM Press. Extended Abstracts.

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

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

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

Senay, H. and Ignatius, E. (1994). A Knowledge-Based System for Visualization Design. *Computer Graphics and Applications*, 14(6):36–47.

Speeth, S. D. (1961). Seismometer Sounds. *The Journal of the Acoustical Society of America*, 33(7):909–916.

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.

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

Ward, M., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters Ltd, Natick, MA, USA.

Wills, G. and Wilkinson, L. (2010). AutoVis: Automatic Visualization. *Information Visualization*, 9(1):47–69.

Zhao, H., Plaisant, C., Shneiderman, B., and Lazar, J. (2008). Data Sonification for Users with Visual Impairment: A Case Study with Georeferenced Data. *ACM Transactions on Computer-Human Interaction*, 15(1):4:1–4:28.

Zhu, Y. (2007). Measuring Effective Data Visualization. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 652–661, Berlin, Germany. Springer.

References

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.

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.

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.

Akaishi, M. and Okada, Y. (2004). Time-tunnel: Visual Analysis Tool for Time-series Numerical Data and its Aspects as Multimedia Presentation Tool. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 456–461, Los Alamitos, CA, USA. IEEE Computer Society.

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

André, P., Wilson, M. L., Russell, A., Smith, D. A., Owens, A., and schraefel, m.c. (2007). Continuum: Designing Timelines for Hierarchies, Relationships and Scale. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 101–110, New York, NY, USA. ACM Press.

Andrienko, N. and Andrienko, G. (2004). Interactive Visual Tools to Explore Spatio-Temporal Variation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 417–420, New York, NY, USA. ACM Press.

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

Andrienko, N. and Andrienko, G. (2011). Spatial Generalization and Aggregation of Massive Movement Data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219.

Ankerst, M. (2001). Visual Data Mining with Pixel-oriented Visualization Techniques. In *Proceedings of ACM SIGKDD Workshop on Visual Data Mining*, New York, NY, USA. ACM Press.

Ankerst, M., Kao, A., Tjoelker, R., and Wang, C. (2008). DataJewel: Integrating Visualization with Temporal Data Mining. In Simoff, S., Böhlen, M., and Mazeika, A., editors, *Visual Data Mining*, volume 4404 of *Lecture Notes in Computer Science*, pages 312–330. Springer, Berlin, Germany.

Antunes, C. M. and Oliveira, A. L. (2001). Temporal Data Mining: An Overview. Workshop on Temporal Data Mining at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3,

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 105–112, New York, NY, USA. ACM Press.

Baldonado, M. Q. W., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for Using Multiple Views in Information Visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 110–119, New York, NY, USA. ACM Press.

Bale, K., Chapman, P., Barraclough, N., Purdy, J., Aydin, N., and Dark, P. (2007). Kaleidomaps: A New Technique for the Visualization of Multivariate Time-Series Data. *Information Visualization*, 6(2):155–167.

Bale, K., Chapman, P., Purdy, J., Aydin, N., and Dark, P. (2006). Kaleidomap Visualizations of Cardiovascular Function in Critical Care Medicine. In *Proceedings of International Conference on Medical Information Visualisation - BioMedical Visualisation (MediVis)*, pages 51–58, Los Alamitos, CA, USA. IEEE Computer Society.

Barbeu-Dubourg, J. (1753). Carte chronographique. Paper roll in scroll case, 40 cm x 16.5 m. Princeton University Library.

Beard, K., Deese, H., and Pettigrew, N. R. (2008). A Framework for Visualization and Exploration of Events. *Information Visualization*, 7:133–151.

Becker, R. A. and Cleveland, W. S. (1987). Brushing Scatterplots. *Technometrics*, 29:127–142.

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

Been, K., Daiches, E., and Yap, C.-K. (2006). Dynamic Map Labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780.

Bergman, L., Rogowitz, B. E., and Treinish, L. A. (1995). A Rule-based Tool for Assisting Colormap Selection. In *Proceedings of IEEE Visualization (Vis)*, pages 118–125, Washington, DC, USA. IEEE Computer Society.

Berry, L. and Munzner, T. (2004). BinX: Dynamic Exploration of Time Series Datasets Across Aggregation Levels. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 5–6, Los Alamitos, CA, USA. IEEE Computer Society.

Bertin, J. (1981). *Graphics and Graphic Information-Processing*. de Gruyter, Berlin, Germany. translated by William J. Berg and Paul Scott.

Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, WI, USA. translated by William J. Berg.

Bettini, C., Jajodia, S., and Wang, X. S. (2000). *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer, Secaucus, NJ, USA, 1st edition.

Borland, D. and Taylor, R. (2007). Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17.

Brewer, C. A. (1999). Color Use Guidelines for Data Representation. In *Proceedings of the Section on Statistical Graphics*, pages 55–60, Baltimore, MD, USA. American Statistical Association.

Brinton, W. C. (1914). *Graphic Methods for Presenting Facts*. The Engineering Magazine Company, New York, NY, USA.

Brinton, W. C. (1939). *Graphic Presentation*. Brinton Associates, New York, NY, USA.

Brockwell, P. J. and Davis, R. A. (2009). *Time Series: Theory and Methods*. Springer, New York, USA, 2nd edition.

Brodbeck, D. and Girardin, L. (2003). Interactive Poster: Trend Analysis in Large Timeseries of High-Throughput Screening Data Using a Distortion-Oriented Lens with Semantic Zooming. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 74–75, Los Alamitos, CA, USA. IEEE Computer Society.

Buono, P., Aris, A., Plaisant, C., Khella, A., and Shneiderman, B. (2005). Interactive Pattern Search in Time Series. In *Proceedings of the Conference on Visualization and Data Analysis (VDA)*, pages 175–186, SPIE.

Buono, P., Plaisant, C., Simeone, A., Aris, A., Shneiderman, B., Shmueli, G., and Jank, W. (2007). Similarity-Based Forecasting with Simultaneous Previews: A River Plot Interface for Time

Series Forecasting. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 191–196, Los Alamitos, CA, USA. IEEE Computer Society.

Burch, M., Beck, F., and Diehl, S. (2008). Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 75–82, New York, NY, USA. ACM Press.

Byron, L. and Wattenberg, M. (2008). Stacked Graphs – Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252.

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

Card, S. K., Suh, B., Pendleton, B. A., Heer, J., and Bodnar, J. W. (2006). Time Tree: Exploring Time Changing Hierarchies. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10, Los Alamitos, CA, USA. IEEE Computer Society.

Carlis, J. V. and Konstan, J. A. (1998). Interactive Visualization of Serial Periodic Data. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 29–38, New York, NY, USA. ACM Press.

Chapple, S. and Garofalo, R. (1977). *Rock 'N' Roll is Here to Pay: The History and Politics of the Music Industry*. Burnham Inc Pub, Chicago, IL, USA.

Chen, C. (2006). CiteSpace II: Detecting and Visualizing Emerging Trends and Transient Patterns in Scientific Literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377.

Chen, H. (2004). Compound Brushing Explained. *Information Visualization*, 3(2):96–108.

Chi, E. H. (2000). A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 69–76, Washington, DC, USA. IEEE Computer Society.

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.

Chittaro, L., Combi, C., and Trapasso, G. (2003). Data Mining on Temporal Data: A Visual Approach and its Clinical Application to Hemodialysis. *Journal of Visual Languages and Computing*, 14(6):591–620.

Chuah, M. C. (1998). Dynamic Aggregation with Circular Visual Designs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 35–43, Los Alamitos, CA, USA. IEEE Computer Society.

Chuah, M. C. and Eick, S. G. (1998). Information Rich Glyphs for Software Management Data. *IEEE Computer Graphics and Applications*, 18(4):24–29.

Clancey, W. J. (1985). Heuristic Classification. *Artificial Intelligence*, 27(3):289–350.

Cleveland, W. (1993). *Visualizing Data*. Hobart Press, Summit, NJ, USA.

Cockburn, A., Karlson, A., and Bederson, B. B. (2009). A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, 41(1):2:1–2:31.

Combi, C., Keravnou-Papailiou, E., and Shahar, Y. (2010). *Temporal Information Systems in Medicine*. Springer, Berlin, Germany.

Combi, C. and Pozzi, G. (2001). HMAP - A Temporal Data Model Managing Intervals with Different Granularities and Indeterminacy from Natural Language Sentences. *The VLDB Journal*, 9(4):294–311.

Constantine, L. L. (2003). Canonical Abstract Prototypes for Abstract Visual and Interaction Design. In Jorge, J., Nunes, N. J., and e Cunha, J. F., editors, *Interactive Systems: Design, Specification, and Verification*, volume 2844 of *Lecture Notes in Computer Science*. Springer, Berlin, Germany.

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

Courage, C. and Baxter, K. (2005). *Understanding Your Users*. Morgan Kaufmann, San Francisco, CA, USA.

Cousins, S. B. and Kahn, M. G. (1991). The Visual Display of Temporal Information. *Artificial Intelligence in Medicine*, 3(6):341–357.

Crabbe, A. (2003). Different Pictures of Time. URL, http://davinci.ntu.ac.uk/rws/researchers/design/tonycrabbe/different.htm. Retrieved Nov. 12, 2003.

Dachselt, R., Frisch, M., and Weiland, M. (2008). FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1353–1356, New York, NY, USA. ACM Press.

Dachselt, R. and Weiland, M. (2006). TimeZoom: A Flexible Detail and Context Timeline. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 682–687, New York, NY, USA. ACM Press. Extended Abstracts.

Doleisch, H. and Hauser, H. (2002). Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 147–154, Plzen, Czech Republic. University of West Bohemia.

dos Santos, S. and Brodlie, K. (2004). Gaining Understanding of Multivariate and Multidimensional Data through Visualization. *Computers & Graphics*, 28:311–325.

Dragicevic, P. and Huot, S. (2002). SpiraClock: A Continuous and Non-Intrusive Display for Upcoming Events. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 604–605, New York, NY, USA. ACM Press. Extended Abstracts.

Dyreson, C. E., Evans, W. S., Lin, H., and Snodgrass, R. T. (2000). Efficiently Supporting Temporal Granularities. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):568–587.

Eccles, R., Kapler, T., Harper, R., and Wright, W. (2008). Stories in GeoTime. *Information Visualization*, 7(1):3–17.

Elmqvist, N. and Tsigas, P. (2007). A Taxonomy of 3D Occlusion Management Techniques. In *Proceedings of the IEEE Conference on Virtual Reality (VR)*, pages 51–58, Los Alamitos, CA, USA. IEEE Computer Society.

Erbacher, R. F., Walker, K. L., and Frincke, D. A. (2002). Intrusion and Misuse Detection in Large-Scale Systems. *IEEE Computer Graphics and Applications*, 22(1):38–48.

Fails, J., 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 the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 167–174, Los Alamitos, CA, USA. IEEE Computer Society.

Fanea, E., Carpendale, M. S. T., and Isenberg, T. (2005). An Interactive 3D Integration of Parallel Coordinates and Star Glyphs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 149–156, Los Alamitos, CA, USA. IEEE Computer Society.

Farquhar, A. B. and Farquhar, H. (1891). *Economic and Industrial Solutions*. G. B. Putnam's Sons, New York, NY.

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.

Ferguson, S. (1991). The 1753 Carte chronographique of Jacques Barbeu-Dubourg. *Princeton University Library Chronicle*, 52(2):190–230.

Forlines, C. and Wittenburg, K. (2010). Wakame: Sense Making of Multi-Dimensional Spatial-Temporal Data. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 33–40, New York, NY, USA. ACM Press.

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*, pages 40–62. Oxford University Press, New York, NY, USA.

Friendly, M. (2008). A Brief History of Data Visualization. In Chen, C.-h., Härdle, W., and Unwin, A., editors, *Handbook of Data Visualization*, pages 15–56. Springer, Berlin, Heidelberg, Germany.

Fry, B. (2000). Organic Information Design, Master's thesis, Massachusetts Institute of Technology.

Fuchs, G. and Schumann, H. (2004a). Intelligent Icon Positioning for Interactive Map-Based Information Systems. In *Proceedings of the International Conference of the Information Resources Management Association (IRMA)*, pages 261–264, Hershey, PA, USA. Idea Group Inc.

Fuchs, G. and Schumann, H. (2004b). Visualizing Abstract Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 139–144, Los Alamitos, CA, USA. IEEE Computer Society.

Funkhouser, H. G. (1936). A Note on a Tenth Century Graph. *Osiris*, 1(1):260–262.

Furia, C. A., Mandrioli, D., Morzenti, A., and Rossi, M. (2010). Modeling Time in Computing: A Taxonomy and a Comparative Survey. *ACM Computing Surveys*, 42:6:1–6:59.

Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. (2006). Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 201–208, New York, NY, USA. ACM Press.

Gall, H., Jazayeri, M., and Riva, C. (1999). Visualizing Software Release Histories: The Use of Color and Third Dimension. In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 99–108, Los Alamitos, CA, USA. IEEE Computer Society.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*, ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, PA, USA.

Gantt, H. L. (1913). *Work, Wages, and Profits*. Engineering Magazine Co., New York, NY, USA.

Gapminder Foundation (2010). Gapminder Trendalyzer. URL, http://www.gapminder.org/world/. Retrieved Feb., 2011.

Gatalsky, P., Andrienko, N., and Andrienko, G. (2004). Interactive Analysis of Event Data Using Space-Time Cube. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 145–152, Los Alamitos, CA, USA. IEEE Computer Society.

Gershon, N. and Page, W. (2001). What Storytelling Can Do for Information Visualization. *Communications of the ACM*, 44(8):31–37.

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

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

Guo, D., Chen, J., MacEachren, A. M., and Liao, K. (2006). A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474.

Haber, R. B. and McNabb, D. A. (1990). Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society, Los Alamitos, CA, USA.

Hackos, J. T. and Redish, J. C. (1998). *User and Task Analysis for Interface Design*. John Wiley & Sons, Inc., New York, NY, USA.

Hadlak, S., Tominski, C., and Schumann, H. (2010). Visualization of Attributed Hierarchical Structures in a Spatio-Temporal Context. *International Journal of Geographical Information Science*, 24(10):1497–1513.

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

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

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

Harrower, M. A. and Brewer, C. A. (2003). ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps. *The Cartographic Journal*, 40(1):27–37.

Hauser, H., Ledermann, F., and Doleisch, H. (2002). Angular Brushing of Extended Parallel Coordinates. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 127–130, Los Alamitos, CA, USA. IEEE Computer Society.

Havre, S., Hetzler, E., and Nowell, L. (2000). ThemeRiver: Visualizing Theme Changes Over Time. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 115–124, Los Alamitos, CA, USA. IEEE Computer Society.

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.

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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1303–1312, New York, NY, USA. ACM Press.

Heer, J. and Robertson, G. (2007). Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247.

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

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.

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.

Holz, C. and Feiner, S. (2009). Relaxed Selection Techniques for Querying Time-Series Graphs. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 213–222, New York, NY, USA. ACM Press.

Horn, W., Popow, C., and Unterasinger, L. (2001). Support for Fast Comprehension of ICU Data: Visualization using Metaphor Graphics. *Methods of Information in Medicine*, 40(5):421–424.

Hsu, W., Lee, M. L., and Wang, J. (2008). *Temporal and Spatio-Temporal Data Mining*. IGI Global, Hershey, PA, USA.

Huang, G., Govoni, S., Choi, J., Hartley, D. M., and Wilson, J. M. (2008). Geovisualizing Data With Ring Maps. *ArcUser*, Winter 2008.

Hägerstrand, T. (1970). What About People in Regional Science? *Papers of the Regional Science Association*, 24:7–21.

Imrich, P., Mueller, K., Imre, D., Zelenyuk, D., and Zhu, W. (2003). Interactive Poster: 3D ThemeRiver. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA. IEEE Computer Society.

Inselberg, A. and Dimsdale, B. (1990). Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. In *Proceedings of IEEE Visualization (Vis)*, pages 361–378, Los Alamitos, CA, USA. IEEE Computer Society.

Jackson, J. E. (2003). *A User's Guide to Principal Components*, Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York, NY, USA.

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

Jankun-Kelly, T., Ma, K.-L., and Gertz, M. (2007). A Model and Framework for Visualization Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):357–369.

Javed, W., McDonnel, B., and Elmqvist, N. (2010). Graphical Perception of Multiple Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–34.

Jensen, C. S., Dyreson, C. E., Böhlen, M. H., Clifford, J., Elmasri, R., Gadia, S. K., Grandi, F., Hayes, P. J., Jajodia, S., Käfer, W., Kline, N., Lorentzos, N. A., Mitsopoulos, Y. G., Montanari, A., Nonen, D. A., Peressi, E., Pernici, B., Roddick, J. F., Sarda, N. L., Scalas, M. R., Segev, A., Snodgrass, R. T., Soo, M. D., Tansel, A. U., Tiberio, P., and Wiederhold, G. (1998). The Consensus Glossary of Temporal Database Concepts – February 1998 Version. In Etzion, O., Jajodia, S., and Sripada, S., editors, *Temporal Databases: Research and Practice*, volume 1399 of *Lecture Notes in Computer Science*, pages 367–405. Springer, Berlin, Germany.

Jeong, D. H., Ziemkiewicz, C., Fisher, B., Ribarsky, W., and Chang, R. (2009). iPCA: An Interactive System for PCA-based Visual Analytics. *Computer Graphics Forum*, 28(3):767–774.

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

Kapler, T., Eccles, R., Harper, R., and Wright, W. (2008). Configurable Spaces: Temporal Analysis in Diagrammatic Contexts. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 43–50, Los Alamitos, CA, USA. IEEE Computer Society.

Kapler, T. and Wright, W. (2005). GeoTime Information Visualization. *Information Visualization*, 4(2):136–146.

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

Keim, D., Kriegel, H.-P., and Ankerst, M. (1995). Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data. In *Proceedings of IEEE Visualization (Vis)*, pages 279–286, Los Alamitos, CA, USA. IEEE Computer Society.

Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in Visual Data Analysis. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 9–16, Los Alamitos, CA, USA. IEEE Computer Society.

Keim, D. A. and Schneidewind, J. (2005). Scalable Visual Data Exploration of Large Data Sets via MultiResolution. *Journal of Universal Computer Science*, 11(11):1766–1779.

Keim, D. A., Schneidewind, J., and Sips, M. (2004). CircleView: A New Approach for Visualizing Time-Related Multidimensional Data Sets. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 179–182, New York, NY, USA. ACM Press.

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

Kim, N. W., Card, S. K., and Heer, J. (2010). Tracing Genealogical Data with TimeNets. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 241–248, New York, NY, USA. ACM Press.

Kolence, K. W. and Kiviat, P. J. (1973). Software Unit Profiles & Kiviat Figures. *SIGMETRICS Performance Evaluation Review*, 2:2–12.

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 the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym)*, pages 45–54, Los Alamitos, CA, USA. IEEE Computer Society.

Kosara, R. and Miksch, S. (2001). Metaphors of Movement - A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131.

Kosara, R. and Miksch, S. (2002). Visualization Methods for Data Analysis and Planning. *International Journal of Medical Informatics*, 68(1–3):141–153.

Kraak, M.-J. (2003). The Space-Time Cube Revisited from a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages 1988–1995, Newcastle, UK. The International Cartographic Association (ICA).

Kraak, M.-J. and Ormeling, F. (2003). *Cartography: Visualization of Geospatial Data*. Pearson Education, Harlow, England, 2nd edition.

Krasner, G. E. and Pope, S. T. (1988). A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49.

Kristensson, P., Dahlback, N., Anundi, D., Bjornstad, M., Gillberg, H., Haraldsson, J., Martensson, I., Nordvall, M., and Stahl, J. (2009). An Evaluation of Space Time Cube Representation of Spatiotemporal Patterns. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):696–702.

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.

Kwan, M.-P. (2009). Space-Time Paths. In Madden, M., editor, *Manual of Geographic Information Systems*, chapter 25, pages 427–442. American Society for Photogrammetry and Remote Sensing, Bethesda, MD, USA.

La maison du cinema and Cinematheque Francaise (2000). Étienne-Jules Marey: Movement in Light. URL, http://www.expo-marey.com/ANGLAIS/home.html. Retrieved Nov. 18, 2004.

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.

Laxman, S. and Sastry, P. (2006). A Survey of Temporal Data Mining. *Sādhanā*, 31:173–198.

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

Lee, B., Riche, N., Karlson, A., and Carpendale, S. (2010). SparkClouds: Visualizing Trends in Tag Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189.

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.

Lenntorp, B. (1976). Paths in Space-Time Environments: A Time Geographic Study of Movement Possibilities of Individuals. In *Lund Studies in Geography*, number 44 in *Series B: Human Geography*. Royal University of Lund, Lund, Sweden.

Lenz, H. (2005). *Universalgeschichte der Zeit*. Marixverlag, Wiesbaden, Germany.

Lin, J., Keogh, E. J., 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. 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.

Liu, L. and Özsu, M. (2009). *Encyclopedia of Database Systems*. Springer, Berlin, Heidelberg, Germany.

Luboschik, M., Schumann, H., and Cords, H. (2008). Particle-Based Labeling: Fast Point-feature Labeling Without Obscuring Other Visual Features. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1237–1244.

Luo, D., Yang, J., Krstajic, M., Ribarsky, W., and Keim, D. (2011). EventRiver: Visually Exploring Text Collections With Temporal References. *IEEE Transactions on Visualization and Computer Graphics*. To appear.

Mackay, W. E., Van Kleek, M. G., and Tabard, A. (2009). Interacting with Temporal Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 4783–4786, New York, NY, USA. ACM Press. Extended Abstracts.

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

Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 173–179, New York, NY, USA. ACM Press.

Marey, E.-J. (1875). *La Méthode Graphique dans les Sciences Expérimentales (Suite)*, volume 1, pages 255–278. Masson, G., Paris, France.

Marey, E.-J. (1894). *Le mouvement*. Masson, G., Paris, France.

Markovic, D. and Gelautz, M. (2006). Comics-Like Motion Depiction from Stereo. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 155–160, Plzen, Czech Republic. University of West Bohemia.

Matković, K., Hauser, H., Sainitzer, R., and Gröller, E. (2002). Process Visualization with Levels of Detail. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 67–70, Los Alamitos, CA, USA. IEEE Computer Society.

Matthews, G. and Roze, M. (1997). Worm Plots. *IEEE Computer Graphics and Applications*, 17(6):17–20.

McCloud, S. (1994). *Understanding Comics*. HarperPerennial, New York, NY, USA.

McCormick, B. H., DeFanti, T. A., and Brown, M. D. (1987). Visualization in Scientific Computing. *Computer Graphics*, 21(6).

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

McLachlan, P., Munzner, T., Koutsofios, E., and North, S. (2008). LiveRAC: Interactive Visual Exploration of System Management Time-Series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, CHI ’08, pages 1483–1492, New York, NY, USA. ACM Press.

Mennis, J. L., Peuquet, D., and Qian, L. (2000). A Conceptual Framework for Incorporating Cognitive Principles into Geographical Database Representation. *International Journal of Geographical Information Science*, 14(6):501–520.

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. *Artificial Intelligence 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 *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM)*, pages 281–290, Berlin, Germany. Springer.

Mintz, D., Fitz-Simons, T., and Wayland, M. (1997). Tracking Air Quality Trends with SAS/GRAPH. In *Proceedings of the 22nd Annual SAS User Group International Conference (SUGI97)*, pages 807–812, Cary, NC, USA. SAS.

Mitsa, T. (2010). *Temporal Data Mining*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Boca Raton, FL, USA.

Muniandy, K. (2001). Visualizing Time-Related Events for Intrusion Detection. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA. IEEE Computer Society. Late Breaking Hot Topics.

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

Müller, W. and Schumann, H. (2003). Visualization Methods for Time-Dependent Data - An Overview. In *Proceedings of Winter Simulation Conference (WSC)*, pages 737–745, Los Alamitos, CA, USA. IEEE Computer Society.

Nelson, A. (2008). Travel Time to Major Cities: A Global Map of Accessibility. Office for Official Publications of the European Communities, Luxembourg.

Newman, L. H. (1965). *Man and Insects*. Aldus Books, London, UK.

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

Noirhomme-Fraiture, M. (2002). Visualization of Large Data Sets: The Zoom Star Solution. *Journal of Symbolic Data Analysis*, 0(0).

Norman, D. A. (1993). *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Norman, D. A. (2002). *The Design of Everyday Things*. Basic Books (Perseus), New York, NY, USA.

Paternò, F., Mancini, C., and Meniconi, S. (1997). ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In *Proceedings of IFIP TC13 Interantional Conference on Human-Computer Interaction (INTERACT)*, pages 362–369, Boston, MA, USA. Kluwer Academic Publishers.

Paternò, F. and Santoro, C. (2002). One Model, Many Interfaces. In *Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces (CADUI)*, pages 143–154, Boston, MA, USA. Kluwer Academic Publishers.

Petzold, I. (2003). *Beschriftung von Bildschirmkarten in Echtzeit*, PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn.

Peuquet, D. J. (1994). It’s about Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. *Annals of the Association of American Geographers*, 84(3):441–461.

Peuquet, D. J. (2002). *Representations of Space and Time*. The Guilford Press, New York, NY, USA.

Piringer, H., Tominski, C., Muigg, P., and Berger, W. (2009). A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120.

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

Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., and Shneiderman, B. (1998). LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, pages 76–80, Bethesda, MD, USA. American Medical Informatic Association (AMIA).

Playfair, W. (1805). *An Inquiry into the Permanent Causes of the Decline and Fall of Powerful and Wealthy Nations*. Greenland and Norris, London, UK.

Playfair, W. (1821). *A Letter on our Agricultural Distresses, their Causes and Remedies*. William Sams, London, UK.

Playfair, W. and Corry, J. (1786). *The Commercial and Political Atlas: Representing, by Means of Stained Copper-Plate Charts, the Progress of the Commerce, Revenues, Expenditure and Debts of England during the Whole of the Eighteenth Century*. printed for J. Debrett; G. G. and J. Robinson; J. Sewell; the engraver, S. J. Neele; W. Creech and C. Elliot, Edinburgh; and L. White, London, UK.

Powsner, S. M. and Tufte, E. R. (1994). Graphical Summary of Patient Status. *The Lancet*, 344(8919):386–389.

Priestley, J. (1765). *A Chart of Biography*. Johnson, J., London, UK.

Pulo, K. (2007). Navani: Navigating Large-Scale Visualisations with Animated Transitions. In *Proceedings of the International Conference Information Visualisation (IV)*, Los Alamitos, CA, USA. IEEE Computer Society.

Reijner, H. (2008). The Development of the Horizon Graph. In *Electronic Proceedings of the VisWeek Workshop From Theory to Practice: Design, Vision and Visualization*.

Reinders, F., Post, F. H., and Spoelder, H. J. (2001). Visualization of Time-Dependent Data with Feature Tracking and Event Detection. *The Visual Computer*, 17(1):55–71.

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 *Proceedings of the International Conference on Human-Computer Interaction (HCI-I)*, Berlin, Germany. Springer. To appear.

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, New York, NY, USA. ACM Press.

Rit, J.-F. (1986). Propagating Temporal Constraints for Scheduling. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 383–388, Los Altos, CA, USA. Morgan Kaufmann.

Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J. (2008). Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14:1325–1332.

Robertson, P. K. (1991). A Methodology for Choosing Data Representations. *IEEE Computer Graphics and Applications*, 11(3):56–67.

Rosenberg, D. and Grafton, A. (2010). *Cartographies of Time: A History of the Timeline*. Princeton Architectural Press, New York, NY, USA.

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.

Saito, T., Miyamura, H., Yamamoto, M., Saito, H., Hoshiya, Y., and Kaseda, T. (2005). Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 173–180, Los Alamitos, CA, USA. IEEE Computer Society.

Schulze-Wollgast, P., Tominski, C., and Schumann, H. (2005). Enhancing Visual Exploration by Appropriate Color Coding. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 203–210, Plzen, Czech Republic. University of West Bohemia.

Schumann, H. and Müller, W. (2000). *Visualisierung – Grundlagen und allgemeine Methoden*. Springer, Berlin, Germany.

Senay, H. and Ignatius, E. (1994). A Knowledge-Based System for Visualization Design. *Computer Graphics and Applications*, 14(6):36–47.

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(2):115–135.

Shahar, Y., Miksch, S., and Johnson, P. (1998). The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. *Artificial Intelligence in Medicine*, 14(1-2):29–51.

Shanbhag, P., Rheingans, P., and desJardins, M. (2005). Temporal Visualization of Planning Polygons for Effcient Partitioning of Geo-Spatial Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 211–218, Los Alamitos, CA, USA. IEEE Computer Society.

Shimabukuro, M., Flores, E., de Oliveira, M., and Levkowitz, H. (2004). Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. In *Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pages 107–117, Los Alamitos, CA, USA. IEEE Computer Society.

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

Shneiderman, B. (1994). Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77.

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.

Silva, S., Madeira, J., and Santos, B. S. (2007). There is More to Color Scales than Meets the Eye: A Review on the Use of Color in Visualization. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 943–950, Los Alamitos, CA, USA. IEEE Computer Society.

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.

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

Speeth, S. D. (1961). Seismometer Sounds. *The Journal of the Acoustical Society of America*, 33(7):909–916.

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

Stacey, M. and McGregor, C. (2007). Temporal Abstraction in Intelligent Clinical Data Analysis: A Survey. *Artificial Intelligence in Medicine*, 39(1):1–24.

Stein, K., Wegener, R., and Schlieder, C. (2010). Pixel-Oriented Visualization of Change in Social Networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 233–240, Los Alamitos, CA, USA. IEEE Computer Society.

Steiner, A. (1998). *A Generalisation Approach to Temporal Data Models and their Implementations*, PhD thesis, Swiss Federal Institute of Technology.

Stolte, C., Tang, D., and Hanrahan, P. (2002). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65.

Telea, A. C. (2007). *Data Visualization: Principles and Practice*. A K Peters, Ltd., Natick, MA, USA.

Thakur, S. and Hanson, A. J. (2010). A 3D Visualization of Multiple Time Series on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 336–343, Los Alamitos, CA, USA. IEEE Computer Society.

Thakur, S. and Rhyne, T.-M. (2009). Data Vases: 2D and 3D Plots for Visualizing Multiple Time Series. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 929–938, Berlin, Germany. Springer.

Third Millennium Press (2001). *Zeittafel der Weltgeschichte. Den letzen 6000 Jahren auf der Spur.* Könemann Verlagsgesellschaft mbH, Cologne, Germany.

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.

Tominski, C. (2011). Event-Based Concepts for User-Driven Visualization. *Information Visualization*, 10(1):65–81.

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247, New York, NY, USA. ACM Press.

Tominski, C., Abello, J., and Schumann, H. (2005a). Interactive Poster: 3D Axes-Based Visualizations for Time Series Data. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 49–50, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Fuchs, G., and Schumann, H. (2008). Task-Driven Color Coding. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 373–380, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2003). Visualisierung zeitlicher Verläufe auf geografischen Karten. In *Kartographische Schriften, Band 7: Visualisierung und Erschließung von Geodaten*, pages 47–57. Kirschbaum Verlag, Bonn, Germany.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005b). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C. and Schumann, H. (2008). Enhanced Interactive Spiral Display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity*, pages 53–56, Linköping, Sweden. Linköping University Electronic Press.

Tomitsch, M., Aigner, W., and Grechenig, T. (2007). A Concept to Support Seamless Spectator Participation in Sports Events Based on Wearable Motion Sensors. In *Proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA07)*, pages 209–214, Los Alamitos, CA, USA. IEEE Computer Society.

Tory, M. and Möller, T. (2004). Rethinking Visualization: A High-Level Taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 151–158, Los Alamitos, CA, USA. IEEE Computer Society.

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

Tufte, E. R. (1990). *Envisioning Information*. Graphics Press, Cheshire, CT.

Tufte, E. R. (1997). *Visual Explanations*. Graphics Press, Cheshire, CT.

Tufte, E. R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, CT.

Turdukulov, U. D., Kraak, M.-J., and Blok, C. A. (2007). Designing a Visual Environment for Exploration of Time Series of Remote Sensing Data: In Search for Convective Clouds. *Computers & Graphics*, 31(3):370–379.

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

Udell, J. (2004). Space, Time, and Data. *InfoWorld*, 26(26):32.

Unger, A. and Schumann, H. (2009). Visual Support for the Understanding of Simulation Processes. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64, Los Alamitos, CA, USA. IEEE Computer Society.

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

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 4–9, Los Alamitos, CA, USA. IEEE Computer Society.

Vande Moere, A. (2004). Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 97–104, Los Alamitos, CA, USA. IEEE Computer Society.

Viégas, F., Boyd, D., Nguyen, D., Potter, J., and Donath, J. (2004a). Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. In *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, pages 109–118, Los Alamitos, CA, USA. IEEE Computer Society.

Viégas, F. B., Wattenberg, M., and Dave, K. (2004b). Studying Cooperation and Conflict between Authors with history flow Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 575–582, New York, NY, USA. ACM Press.

Volaric, A. (2003). Four-Dimensional Art. URL, http://www.mathsci.appstate.edu/~sjg/class/3510/final/geoartws.pdf. Retrieved Feb., 2011.

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

Wainer, H. (2005). *Graphic Discovery: A Trout in the Milk and Other Visual Adventures*. Princeton University Press, Princeton, NJ, USA.

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 Transactions on Visualization and Computer Graphics*, 15:1049–1056.

Ward, M., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters Ltd, Natick, MA, USA.

Ward, M. O. (1994). XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of IEEE Visualization (Vis)*, pages 326–333, Los Alamitos, CA, USA. IEEE Computer Society.

Ware, C. (2008). *Visual Thinking for Design*. Morgan Kaufmann, Burlington, MA, USA.

Wattenberg, M. (2002). Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 110–116, Los Alamitos, CA, USA. IEEE Computer Society.

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing Time-Series on Spirals. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 7–14, Los Alamitos, CA, USA. IEEE Computer Society.

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

Wehrend, S. and Lewis, C. (1990). A Problem-Oriented Classification of Visualization Techniques. In *Proceedings of IEEE Visualization (Vis)*, pages 139–143, Los Alamitos, CA, USA. IEEE Computer Society.

Whitrow, G. J., Fraser, J. T., and Soulsby, M. P. (2003). *What is Time?: The Classic Account of the Nature of Time*. Oxford University Press, New York, NY, USA.

Wills, G. and Wilkinson, L. (2010). AutoVis: Automatic Visualization. *Information Visualization*, 9(1):47–69.

Winckler, M. A., Palanque, P., and Freitas, C. M. D. S. (2004). Tasks and Scenario-Based Evaluation of Information Visualization Techniques. In *Proceedings of the Annual Conference on Task Models and Diagrams (TAMODIA)*, pages 165–172, New York, NY, USA. ACM Press.

Wolter, M., Assenmacher, I., Hentschel, B., Schirski, M., and Kuhlen, T. (2009). A Time Model for Time-Varying Visualization. *Computer Graphics Forum*, 28(6):1561–1571.

Wong, P. C. and Bergeron, R. (1997). 30 Years of Multidimensional Multivariate Visualization. In Nielson, G. M., Hagen, H., and Müller, H., editors, *Scientific Visualization*, pages 40–62. IEEE Computer Society, Los Alamitos, CA, USA.

Wongsuphasawat, K. and Shneiderman, B. (2009). Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization. In *Proceedings of the IEEE*

Symposium on Visual Analytics Science and Technology (VAST), pages 27–34, Los Alamitos, CA, USA. IEEE Computer Society.

Wright, H. (2007). *Introduction to Scientific Visualization*. Springer, Berlin, Germany.

Wunderlich, C. A. (1870). *Das Verhalten der Eigenwärme in Krankheiten*. Otto Wigand, Leipzig, Germany, 2nd edition.

Xiong, R. and Donath, J. (1999). PeopleGarden: Creating Data Portraits for Users. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 37–44, New York, NY, USA. ACM Press.

Xu, R. and Wunsch II, D. C. (2009). *Clustering*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Yang, J., Wang, W., and Yu, P. S. (2000). Mining Asynchronous Periodic Patterns in Time Series Data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 275–279, New York, NY, USA. ACM Press.

Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. (2007). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231.

Zhao, H., Plaisant, C., Shneiderman, B., and Lazar, J. (2008a). Data Sonification for Users with Visual Impairment: A Case Study with Georeferenced Data. *ACM Transactions on Computer-Human Interaction*, 15(1):4:1–4:28.

Zhao, J., Forer, P., and Harvey, A. S. (2008b). Activities, Ringmaps and Geovisualization of Large Human Movement Fields. *Information Visualization*, 7(3):198–209.

Zhu, Y. (2007). Measuring Effective Data Visualization. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 652–661, Berlin, Germany. Springer.

Index

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3,