Visualization of Time-Oriented Data
Dedication goes here...
Time is central to life. We are aware of time slipping way, being used well or poorly, or 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 could 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 for 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 displays 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.
Preface

Time is an exceptional data dimension due to its importance and unique influence. This implies that special methods are needed for dealing with time in data sets in order to take advantage of these specifics. Visual methods are a key part of this tool set allowing for presentation, analysis, and exploration of temporal data sets. Since these facts have been widely acknowledged in visualization research, a lot of more or less application specific techniques have been developed to support tasks in dealing with time-oriented data. In the meantime this number has reached a level that is no longer easily manageable. This is what our contribution aims for—to provide an overview of the field of visualization techniques for time-oriented data. But even more importantly we analyze the specifics and similarities of the investigated techniques in order to identify clusters for structuring and making the field manageable. We analyze the techniques uniformly along different sets of attributes making it easy to grasp their main characteristics and support researchers as well as developers in selecting appropriate techniques for their demands.

Additionally, the book contains several examples of early (pre-computer age) representation techniques, it explains the visualization process itself, specifies the characteristics of the dimension time, of time-oriented data, of related tasks, and introduces a conceptual framework for visually analyzing time-oriented data.

Vienna and Rostock, June 2010
Wolfgang Aigner
Silvia Mitsch
Heidrun Schumann
Christian Tominski
List of Recommendations

References

1. Introduction
2. Literature Review
3. Methodology
4. Results
5. Discussion
6. Conclusion
7. References

Other Notes

- Swara A. Anand, M.A.
- C6 INHIBITION
- Frontal lobe lesion
- Complete loss of object
- S.L. (4th Floor)
1.1 Introduction to Visualization

Visual representations have a long and venerable history in communicating facts and information. The dictionary explains the following [Spence, 2007]:

- visualization: to form a mental or mental image of something.

But only about twenty years ago, visualization became an independent self-contained research field. In 1987 the notion of visualization in scientific computing was introduced by McCormick, DeFanti, and Brown [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 the simulation and computations.

The goal of this new field of research was 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 [Schumann and Müller, 2000]:

- expressiveness,
- effectiveness, and
- appropriateness.

Expressiveness regards 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 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 the given task. While the value of a visual representation is so easy to determine [van Wijk, 2005], cost is often related to time efficiency (i.e., the needed computation time) and space efficiency (i.e., the exploited screen space).

Expressiveness, effectiveness, and appropriateness are criteria that any visual representations 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, the visualization process has to answer the two questions: "What has to be presented?" and "Why has it to be presented?". Both questions are discussed in more detail in the following.

What has to be presented – specification of the data. In recent years, different approaches to characterize the data – the central element of visualization – have been developed. In their overview article, Wong and Bergeron established the notion of multidimensional multivariate data as multivariate data that are given in a multidimensional domain [Wong and Bergeron, 1997]. 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 data set. 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 of [Mennis et al., 2000]. On the level of data, this framework is based on three perspectives: 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 with regard to the independent variables, including not only simple data values, but also objects and their relationships, where objects and relations may have associated with them arbitrary data attributes, i.e., the dependent variables as described above.

From an application point of view, it makes sense to visualize all described aspects. 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 axis; the aspect what to represent the values of a multivariate data set or to visualize objects as an abstraction of the data (e.g., to show the temporal behavior of a group of data values or to show the relationships between objects, which may also vary over time).

This book focuses on the aspect where. Our interest is in time and time-oriented data. The inherent properties of time and associated data as well as the specific implications for visualization will be discussed in more detail in Chapter 3.

Why has it to be presented – specification of the task. When it comes to answering the question why some data have to be visualized, we have to distinguish between three basic goals:

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

Explorative analysis can be seen as unstructured search. In this case, new a-priori knowledge about the data is given. It is the goal to get insight into the data, to begin extracting relevant information, and to come up with hypotheses about the data. 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.
The upper field of the document contains information about the recipient, the date, and the reference number. The text below is a continuation of a discussion, possibly regarding a partnership or agreement, and includes excerpts from a reference text or legal document.

"Dear [Recipient],

Thank you for your interest in our proposal. We appreciate the opportunity to discuss the terms and conditions of our partnership. As mentioned in our initial correspondence, we aim to collaborate on [specific project/venture].

Please find attached a copy of the reference text for your review. This text provides a comprehensive overview of our approach and the expected outcomes.

We look forward to your feedback and hope to proceed with the partnership in the near future.

Best regards,

[Your Name]"
1.2 Application Examples

Having introduced the very basics of interactive visualization, we now move on to some application examples. The goal is to illustrate concrete instantiations of visual representations and to demonstrate possible benefits of visual methods for data exploration and analysis.

Example 1: European Study of Cardiovascular Disease

data from a study on cardiovascular disease, which includes data on age, sex, smoking status, blood pressure, and cholesterol levels for a large number of patients. The goal is to identify patterns and correlations in the data that might help in understanding the disease and predicting risk factors.

Example 2: Weather Forecasting

data from a weather forecasting model, which includes temperature, humidity, wind speed, and precipitation data for different locations. The goal is to visualize the data in a way that makes it easy to see patterns and trends over time, and to determine the best location for a new weather station.

Example 3: Traffic Analysis

data from a traffic monitoring system, which includes data on traffic volume, speed, and flow for different times of day and days of the week. The goal is to identify congestion points and to optimize traffic flow.

Example 4: Financial Analysis

data from a financial analysis model, which includes data on stock prices, earnings, and other financial metrics for different companies. The goal is to visualize the data in a way that makes it easy to see trends and patterns over time, and to identify potential opportunities for investment.

Fig. 1.2: The data-state-reference model by [Chi, 2000].
Chapter 2
History of Graphing Time

Long before computers even appeared, visualization has been used to represent time-oriented data. To broaden the view beyond "scientifically formal" visualization techniques and provide background information on the history of visualization methods, we present historic and application-specific representations. They mostly consist of historical techniques of the pre-computer age as for example the works of William Playfair, Edouard-Jules Marey, or Charles Joseph Minard. Furthermore, we will take the reader on a path towards arts. Artists of all times were concerned with the questions of how to incorporate the dynamics of time and motion in their artworks. This chapter presents a few outstanding art directions and forms that are characterized by a strong focus on representing temporal concepts. We believe that art can be a valuable source of inspiration and concepts or methods found there might also be applicable to information visualization in order to improve existing techniques or create completely new ones.

2.1 Earliest Time-Series Representation

The probably oldest time-series representation to be found in literature is the illustration of planetary orbits created in the 10th or possibly 11th century (found in [Tufte, 1983], see Fig. 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.

2.2 Business Information Representations by William Playfair

Representing business information graphically is a broad application field with a long tradition. William Playfair can be seen as the founding protagonist. He published the first known time-series depicting economic data in 1786 [Tufte, 1983]. His works contain for example Silhouette Graphs, Bar Charts, and Line Graphs.

Fig. 2.1: Time-Series Plot depicting Planetary Orbits (10th/11th Century) [Tufte, 1983]. The illustration is part of a text from a monastery school and shows inclinations of the planetary orbits as a function of time.

(see Fig. 2.2 and 2.3). A quite unique representation of economic data is the so-called Phillips Curve – a 2D plot where time is neither mapped to the x- or the y-axis, but is rather shown sequentially as steps of the "curve". This way, the dimension of time is slightly de-emphasized in favor of showing the relationship of two time-dependent variables. Figure 2.4 for example shows "Unemployment rate" as the y-axis in relation to "Inflation rate" depicted at the y-axis. Each year's combination of those two values 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.

Fig. 2.2: Business Information Graphs by William Playfair (18th Century) [Tufte, 1983].
FIG. 3: The Water of Elanmore-June (Stage 3).

2.3 The Water of Elanmore-June (Stage 3).

A modification of the earlier information was taken into consideration. It shows the final stage of the project.

2.2 Business Environment: March 1999

(a) Total Costs

(b) Time Chart

(c) Project

Fig. 1: Phillips Curve (Timmerman, 1999) - The marginal cost of unemployment increases as the employment level increases.
and was a trained physician and physiologist. His interest in internal and external movements in humans and animals, like for example blood circulation, human walking, horse gaits, or dragonfly flight, lead to the decomposition of these movements via novel photography and representation methods (see Fig. 2.6). This photography method is called chronophotography and paved the way for the birth of modern film-making at the end of the nineteenth century. “Tirelessly, this brilliant visionary stopped the passage of time, accelerated it, slowed it down so “see the invisible,” and recreated life through images and machines”.

(a) A Person Walking.

(b) Chronophotography.

(c) Horse Gaits.

Fig. 2.6: Studies of Movement by E.J. Marey (19th Century) [La maison du cinema and Cinematheque Francaise, 2000, Tufte, 1983].

2.4 Graphical Representation of Historical Information

But E. J. Marey was not only fascinated by motion and movement in a physical sense. He was also interested in the graphical representation of history information utilizing timelines (see Figure 2.7). Another prominent example of graphically representing history information via annotated timelines is Deacon’s Synchronological Chart of Universal History which was originally published in 1900 and was drawn by Edmund Hull (see Figure 2.8). 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 literature (e.g., [Third Millennium Trust, 1999]).

Fig. 2.7: E.J. Marey’s Timeline (1885) [Tufte, 1983].

Even earlier than E.J. Marey, Charles Joseph Minard created a masterpiece of history information visualization in 1861. His graphical representation of Napoleon’s Russian Campaign of 1812 is extraordinary information rich, conveying no less than five different variables in two dimensions (see Fig. 2.9). E. Tufte comments on this representation “It may be the best statistical graphic ever drawn.” [Tufte, 1983]. 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 and 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 quite different approach of representing historical information is the illustration of the Cuban Missile Crisis during the cold war (see Fig. 2.10). The diagram shows decisions, possible decisions, and the outcomes thereof over time. This representation is similar to the Decision Chart presented in Section 3.1.1.4, p.37. The illustration of Rock ‘n’ Roll History shown in Figure 2.11 depicts protagonists and developments in the area as curved lines that are stacked according to their percentage of annual record sales. The ThemeRiverM technique (see Section 5.2, p.125) can be seen as further, more formal development of this idea.

Fig. 2.11: Rock Type History (Barley, 1989) - Depicts formations and geologic events over time.

Fig. 2.10: Clean Alluvial Delta (Broom and Deline, 1989) - Shows the evolution of the delta over time and the impact of geologic events.

Fig. 2.9: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Illustrates the distribution of depositional centers throughout geological history.

Fig. 2.8: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Depicts the geographical spread and distribution of depositional centers over time.

Fig. 2.7: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Highlights the evolution of depositional centers and their geographical spread.

Fig. 2.6: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Shows the complex interactions and movements of depositional centers over geological time.

Fig. 2.5: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Depicts the dynamic processes and geologic events that influenced the formation of depositional centers.

Fig. 2.4: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Illustrates the interplay between geologic events and the formation of depositional centers.

Fig. 2.3: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Highlights the geographical spread and distribution of depositional centers over time.

Fig. 2.2: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Depicts the evolution and movement of depositional centers over geological history.

Fig. 2.1: Archean Proterozoic Depositional Centers of the World (Hammond, 1989) - Shows the geographical spread and distribution of depositional centers over time.
2.5 Graphical Representation of Medical Information

In medicine, large amounts of information are generated and have to be processed mostly by humans. Graphical representations help to make this myriad of information graspable and are a crucial part of the workflow of healthcare personnel. These representations range from Fever Curves of the nineteenth century (see Fig. 2.12) and EEG Time-Series Plots (see Fig. 2.13) to information-rich patient status overviews (see Fig. 2.14). Especially the graphical summary of patient status by Edward R. Tufte and Seth-M. Powsser [Powsser and Tufte, 1994] makes use of concepts such as small multiples, focus+context, 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 it.

Fig. 2.12: Fever Chart created by Carl Reinhold Wunderlich in 1868 (adapted from [Plüger, 2001]).

Fig. 2.13: EEG Time-Series Plot [Tufte, 1983].

Fig. 2.14: Graphical Summary of Patient Status [Powsser and Tufte, 1994] — Concise summary of patient information. Uses small multiples, focus+context, and integrates textual as well as graphical information.
2.8 Visual Explanations, Visual Storytelling, and Comics

"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." [McCready, 1994].

Two disciplines that are also seldom connected to time-oriented information are visual explanations and visual storytelling. Although ubiquitous in various forms in daily life, they are rarely considered for visualizing abstract information. Visual explanations are often used in manuals for products like home electronics, furniture assembly, cars, and many more (see Fig. 2.18 and 2.19). 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 before is the craft of storytelling, especially visual storytelling, starting from caveman paintings and Egyptian hieroglyphs to picture books and comic strips (see Fig. 2.20 and 2.21). Time is the red line that ties everything together in visual storytelling. Many interesting techniques and paradigms exist that might be applicable to visualization in general [Gershon and Page, 2001] as well as to the representation of time-oriented information in particular.

Fig. 2.19: Life Cycle of the Japanese Beetle [Tufte, 1990].

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 give a hint that comics incorporate many concepts of time, while still retaining a static, 2-dimensional form. Scott McCloud analyzed many of the methods and paradigms of comics concluding that powerful concepts in terms of representing time, dynamics, movement are applied that are different from those applied in paintings or photography [McCready, 1994]. 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 juxtaposition of panels. But the individual panels portray more than single "frozen" moments in time and are more than photos put side by side. Single panels rather contain whole scenes whose temporal extent may span from milliseconds to arbitrary lengths (see Fig. 2.22). Not only the content of a panel sheds light on its length but also the shape of the panel itself can make a difference on our perception of time. Even more freedom in a temporal sense is given by the transition from one panel to the next or the space between panels, respectively (see Fig. 2.23). Here, time might be compressed, expanded, rewinded, 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 compared to paintings, photographs, and even film. Recently, research work on generating these comic-like effects from motion pictures has been conducted as for example in [Markovic and Gelaul, 2006].

Besides the pure temporal aspect, motion is another big topic in comics. Several visual techniques, such as motion lines or action lines plus additional effects as multiple images, streaking effects, or blurring, which are partly borrowed from photography, are applied.

Fig. 2.18: Visual explanation is used to illustrate stepwise processes [Aigner et al., 2004].
individual episodes begin at the same point in time and show different possible strands of events.

The movie *Pulp Fiction* \(^3\) 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 which closes the loop.

A further example for using interesting temporal constellations in film is the movie *Memento* \(^4\). The main character of the movie is a man who suffers from short-term memory loss and uses notes and tattoos to hunt his wife’s killer. What makes the story telling 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).

### 2.10 Renaissance

A very interesting approach of overcoming the limitations of time can be found in Renaissance paintings. Here, sequences of different temporal episodes are shown in a single composition. Figure 2.25 for example shows a painting by Masaccio and Masolino that presents two scenes in the life of St. Peter within a single scenery. This

![Fig. 2.25: Masaccio and Masolino, Scenes from the Life of St. Peter, c.1429-7, Brancacci Chapel, Florence — Shows different stages or episodes of a single person within a unifying scenery.](image)

---

\(^3\) *Pulp Fiction*, written by Quentin Tarantino et al., directed by Quentin Tarantino, 1994

\(^4\) *Memento*, written by J. and C. Nolan, directed by Christopher Nolan, 2000

### 2.11 Cubism

The beginning of the 20th century was characterized by new findings and breakthroughs in natural sciences, especially Mathematics and Physics, as for example Einstein’s theory of relativity. But not only the world of science was shaken by these developments, also artists were concerned about these topics in their own way. Foremost the protagonists of the Cubism art movement focused on incorporating time in their artworks. They coined the term *Four-dimensional Art.*

![Fig. 2.26: Marcel Duchamp, Nude Descending a Staircase, 1912 — Incorporates the dimension time by overlaying different stages of a person’s movement.](image)

![Fig. 2.27: Pablo Picasso, Portrait of Vollard, 1910 — Many different observations are composed and partly overlaid to form a single picture.](image)
The data transformation approaches that we present here are based on a combination of techniques and strategies. The selection of the right data transformation approach is crucial for achieving accurate and reliable results. In this section, we discuss various data transformation techniques, including data cleaning, data integration, data reduction, and data discretization. Each technique has its own strengths and weaknesses, and the choice of technique depends on the specific requirements of the data analysis.

2.1.1 Summary
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. In a 1991 literature survey by J.T. Fraser, the total number of entries judged to be potentially relevant to the systematic study of time reached about 65,000 [Whitrow et al., 2003]. This illustrates the breadth of the topic and the restless endeavor of man to uncover its secrets.

The most influential theories for natural sciences are probably Newton's concepts of absolute vs. relative time, and Einstein's four-dimensional spacetime. What can be extracted as bottom line across many theories is that time is unidirectional (arrow of time) and that time gives order to events.

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 is taking place in four-dimensional space, at a location that is defined by three spatial coordinates at a certain time as fourth coordinate [Lentz, 2005]. Both, Newton's notion of absolute time and Einstein's spacetime are concepts that describe time as 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.

3.1 Characteristics and Models of Time

First of all, it is important to make a clear distinction between the physical dimension and a model of time in information systems. When modeling time in information systems, absolute is not to perfectly emulate 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 [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 along with many references to fundamental publications is provided by Hájek [Hájek, 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 Coralwalla et al. [Coralwalla et al., 1998] note, most research focuses on the development of specialized models with different features for particular domains. But apart from the

It is interesting to note that much more precise calendars where known hundreds of years earlier in other cultures like the Mayans and in China.
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 later. 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 Fig. 3.7 and Fig. 3.8).

3.1.1.3 Arrangement: linear vs. cyclic

As the third design aspect, we take a look at the arrangement of the time domain. Corresponding to our natural perception of time, we mostly consider time as pro-
Figure 3.2: Finding the T-Lawrence corners for factor α.

![Diagram showing the factor α in a cycle diagram]

**Finding the T-Lawrence corners for factor α.**

1. Identify the starting point (α) on the cycle diagram.
2. Move clockwise around the cycle to find the next factor (β).
3. Continue moving clockwise until all factors have been identified.

**Key Points:**
- The cycle diagram shows the relationship between factors in a process.
- The T-Lawrence corners help in identifying critical points in the process.
- Understanding these points is crucial for process improvement.

---

3.2.2 Application of T-Lawrence in factor analysis:

By applying the T-Lawrence method, factors can be systematically analyzed to identify patterns and relationships within a dataset. This approach is particularly useful in fields such as psychology, economics, and social sciences, where complex data sets are common.

---

3.2.3 Consideration of Findings from the cycle diagram:

Upon examining the cycle diagram, it is evident that factor α plays a significant role in the overall process. Its position and interaction with other factors suggest a need for further investigation and potential intervention.

---

3.2.4 Further analysis of the cycle diagram for factor α:

Further analysis of the cycle diagram reveals that factor α is not only influenced by β but also impacts factors γ and δ. This interdependence highlights the complexity of the process and the necessity of a holistic approach to its management.

---

3.2.5 Conclusion:

The application of the T-Lawrence method in factor analysis has provided valuable insights into the process dynamics. By understanding the relationships between factors, stakeholders can make informed decisions to optimize performance and efficiency.
as the definition of concrete time elements used to relate data to time need to be specified. In the following subsection, we will discuss this facet in more detail.

### 3.1.2.1 Granularity and Calendars: none vs. single vs. multiple

To tame the complexity of time aspects, useful abstractions have been employed. These abstractions are used to provide different levels of granularity and relate data to time. Basically, granularities can be thought of as (human-made) abstractions of time in order to make it easier to deal with time in everyday life (like minutes, hours, days, weeks, months). More generally, granularities describe mappings from time values to larger or smaller conceptual units (see Figure 3.15 for an example of time granularities and their relationships). For example, 60 consecutive seconds are mapped to one minute or five time-steps in a discrete simulation model might relate one second to physical time. Specifically in simulation systems, the concept of granularity has to be considered when describing the relationship between simulation time and physical time. An overview and formalization of time granularity concepts is given by Bettini et al. (Bettini et al., 2000).

**Fig. 3.14:** Decision Chart [Harris, 1999] – 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.

**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.

A system of multiple granularities in lattice structures is referred to as "calendar" (see Fig. 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 (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 cycle as time passes (Jensen et al., 1998).

If a granularity and calendar system is supported by the data model, we consider this as multiple granularities. Besides this complex variant, there might be a single granularity only (e.g., every time value is given in terms of milliseconds) or such abstractions are not supported at all (none).

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 smallest granularity). Hence, the underlying time domain can be described as a sequence of non-decomposable, consecutive time intervals of identical duration (Jensen et al., 1998). These smallest units are termed "chronons". This allows for a simple representation of a point in time as number of chronons relative to a reference point (e.g., milliseconds <chronon> since January 1, 1970 00:00:00 GMT). Based on this, a granularity
Figure 3.16: Artificial Intelligence and Environmental Engineering

In recent years, the role of artificial intelligence (AI) in environmental engineering has become increasingly important. AI technologies are being used to improve processes, monitor environmental conditions, and predict and mitigate the effects of pollution. This figure illustrates the potential applications of AI in environmental engineering, including

- Monitoring and modeling of environmental data
- Predictive analytics for pollution control
- Optimization of resource utilization
- Decision support systems for environmental management

These applications are supported by advancements in AI and machine learning algorithms, making it possible to handle complex environmental challenges more effectively.
3.1 Characteristics and Models of Time

3.1.2.2 Time Primitives: instant vs. interval vs. span

Second, we present a set of basic elements often used to relate data to time, so-called time primitives. These time primitives can be seen as intermediary layers between data elements and the time domain on the bottom or already abstractions upon it in the form of granularities as presented in the previous section. Basically, time primitives can be divided into anchored (absolute) and unanchored (relative) primitives. Primitives that are located absolutely at points of the point- or interval-based time domain are instants and intervals.

An instant (also known as time point) is a single point in time, e.g., May 23, 1977. Depending on whether a point-based or interval-based scope is used (see Sect. 3.1.1.2) time model is used, an instant might also have a duration (see Fig. 3.19 and Fig. 3.20). Furthermore, in a discrete time domain (scale, see Sect. 3.1.1.1), an instant is specified relative to a granularity and represents a certain granule within a granularity. The smallest possible temporal extent of an instant in a discrete time domain is that of a single chronos (granularity, see Sect. 3.1.2.1). Examples for instants are the date of birth "May 23, 1977" and the beginning of a presentation at "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".

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

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

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 Fig. 3.21). Alternatively, intervals can be modeled as begin instant + duration (positive span), or as duration (positive span) + end instant. An interval that is defined in terms of begin and end is modeled as a closed interval including the begin as well as the end instant. In a discrete, interval-based time model an interval can be represented as a set of (possibly non-contiguous) granules or chronons. The only unanchored primitive is the span. It represents a directed duration of time, e.g., 4 days (see Fig. 3.22). A time span is defined as directed, unanchored primitive that represents a directed amount of time in terms of a number of granules in a given granularity. Examples for spans are the length of a vacation of "10 days" and the duration of a lecture of "150 minutes". Fig. 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 in time [Jensen 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).

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

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

Relations between the given time primitives can be specified in different ways [Peque, 1994]. We will discuss these relations in terms of topology, i.e., relative locations of time elements. Depending on the time primitives used, different relations between elements are possible (see Fig. 3.23, 3.24, 3.25). Especially when reasoning about time, these relations are important concepts. Allen defined a set of basic interval relations [Allen, 1983] that are very common in time modeling (see Fig. 3.24). The set of possible relations is determined by further design aspects. Most of the previously given visualization examples are suited for representing instants. Gantt charts are an example for visualization techniques that show interval data (see Fig. 3.26).
A "department" is depicted as a rectangle with a "section" within it. The sections are labeled with numbers and are connected by lines. The text mentions "Figure 3.24: Internal relations. Departmental relations as defined by..." and includes a list of possible relations between departments. The diagram shows a network of these relations, indicating how different departments interact with each other.

Figures 3.25 and 3.26 depict various internal relations within departments. Figure 3.25 focuses on specific sections labeled with numbers, while Figure 3.26 shows a more detailed view with additional labels and connections.

The text also refers to "Figure 3.24: Internal relations. Departmental relations as defined by..." and "Figure 3.25: Internal relations..." and "Figure 3.26: Internal relations..." which seem to be part of a larger discussion on internal relations within departments.
3.1 Characteristics and Models of Time

actuated as “don’t know when” information, or more precisely, “don’t know exactly when” information [Jensen et al., 1998]. When only having incomplete or uncertain knowledge of time-related information, we need an indeterminate time model. Examples for 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”). These are examples of temporally indeterminate information. Notice that temporal indeterminacy as well as 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 begin and latest begin of an interval) or is implicitly present in 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 begin instant “June 14, 2009” and the end instant “June 17, 2009” both at 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 to between 6 a.m. and 12 p.m. of the specified day (see Fig. 3.27). Examples for time models that consider temporal indeterminacy are HMAP2 (Combi and Pozzi, 2001) and the time model underlying the time annotations used in the medical treatment plan specification language Asbru (Mitsch, 1999). A visualization technique capable of depicting temporal indeterminacy is for example PlanningLines (see Fig. 3.28).

<table>
<thead>
<tr>
<th>[June 13, 2009; June 19, 2009]</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum duration</td>
</tr>
<tr>
<td>maximum duration</td>
</tr>
<tr>
<td>possible beginning</td>
</tr>
<tr>
<td>possible ending</td>
</tr>
<tr>
<td>Hours</td>
</tr>
<tr>
<td>Days</td>
</tr>
</tbody>
</table>

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”.

2 The word “HMAP” is not an abbreviation, but it is the transliteration of the ancient Greek poetical word “day”.

3 Time & Time-Oriented Data

Fig. 3.28: PlanningLines [Aigner et al., 2005] – 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.

Granularities, time primitives, and determinacy are aspects that allow for useful abstractions and simplifications in dealing with time as it has been laid out in the design space before. After discussing the question of modeling the time domain itself, we now move on to the question of modeling time-oriented data.

3.2 Modeling 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. Many different types of data are related to time. One can think of stock exchange data, census data, simulation data, and much more. This diversity requires an appropriate specification.

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

Puny is an example of static non-temporal data. This kind of data is not addressed in this book.

Static Temporal Data (see Fig. 3.30(b)): If the internal time contains more than one temporal primitive, while the external time contains only one element, then the data can be considered time dependent. Since the values stored in the data depend on the internal time, static temporal data can be understood as a historic view of how the real world or some model looked like 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 (e.g., the TimeWheel [Timmis et al., 2004]).

Dynamic Non-temporal Data (see Fig. 3.30(c)): If the internal time contains only one, but the external time is composed of multiple temporal primitives, then the data depend on the external time. Clearly speaking, the data change over time, i.e., they are dynamic. Since the internal time is not considered, only the current state of the data is preserved; a historic 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 [Makovicić et al., 2003]. However, since internal time and external time can usually be mapped from one to the other, some of the know visualization techniques for static temporal data can be applied for dynamic non-temporal data as well.

Dynamic Temporal Data (see Fig. 3.30(d)): If both, internal and external time, are composed of multiple temporal elements, 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 for 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 not made by current visualization approaches. This is due to the fact that integrating both temporal dimensions into the visualization is a challenging task. Therefore, visualization techniques for dynamic temporal data are beyond the scope of this book.

3.3 Discussion

In this section, we structured and specified the characteristics of time and time-oriented data. We approached this from two perspectives: first, we characterized time and time models specifying the design space and abstractions. Secondly, we defined data values in connection to time elements, which we called modeling of time-oriented data. Table 3.1 and 3.2 summarize these two perspectives and the corresponding aspects.

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

The second perspective focuses on the data variables and their associations with time elements using the understanding of time models explained above, which resulted in the definitions of scale, frame of reference, kind of data, and number of variables as well as in the distinction of internal and external time.

All these aspects need to be considered when visualizing and analyzing data variables over time. We mainly focused on the temporal relations to data variables, but neglected the relationship between different data variables due to the importance of the time dimensions in this book. 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. Many useful modeling alternatives and reference models have been developed and can be adopted, as for example continuous models using scalars, vectors, or tensors, etc. [Wright, 2007] or discrete models using structures like trees, graphs, etc. [Shneiderman, 1996].

We took this hard road along many characterizations and modeling concerns, because only if we have a clear understanding of how the data look like can we develop visualization methods that are successful in supporting analysts in solving analytic tasks effortlessly. In the next chapter, we explore the visualization design problem in more depth.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time:</td>
<td>Recommended practice to manage time efficiently</td>
</tr>
<tr>
<td>Risk:</td>
<td>Identify potential risks and develop mitigation strategies</td>
</tr>
<tr>
<td>Data:</td>
<td>Collect and analyze relevant data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Identifying Time-Critical Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identify time-critical tasks</td>
</tr>
<tr>
<td>2. Prioritize tasks by urgency and importance</td>
</tr>
<tr>
<td>3. Schedule tasks based on priority</td>
</tr>
<tr>
<td>4. Monitor progress and adjust schedule as needed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attention:</th>
<th>[Attention: Suggested focus and concentration techniques]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design:</td>
<td>Create a well-organized workspace</td>
</tr>
<tr>
<td>Space:</td>
<td>Ensure adequate lighting and temperature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3: Enhancing Concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Eliminate distractions</td>
</tr>
<tr>
<td>2. Set specific goals</td>
</tr>
<tr>
<td>3. Use a timer or stop watch</td>
</tr>
<tr>
<td>4. Take short breaks</td>
</tr>
</tbody>
</table>

3.3 Engagement
Chapter 4
The Visualization Problem

Many different types of data are related to time. One can think of stock exchange data, census data, simulation data, and much more. But also news articles, photo collections, or project plans can contain temporal information. All these data have in common that they are time-oriented and, in theory, they should be representable with one and the same visualization approach. In practice, however, the data exhibit specific characteristics and hence each example requires a dedicated visualization. For instance, stock exchange data can be visualized with Flocking Boids [Moore, 2004], census data can be represented as described in [Shaumbat et al., 2003], SimVis [Doletich et al., 2004] is efficient for visualizing simulation data. News articles (or contained keywords) can be analyzed with ThemeRiver [Havre et al., 2002], photo collections can be mapped via MyLifeBits [Gemmell et al., 2006], and project plans can be made comprehensible with PlanningLinen [Alger, 2005].

Apparently, this list of techniques is not exhaustive. The aforementioned approaches are just a few examples out of many that recognize the special role of the dimension of time. We shall complete this list in a rich survey of such visualization techniques in Chapter 5.

Besides using dedicated techniques, time-oriented data can also be visualized using generic approaches. Since time is most seen as a quantitative dimension (or at least can be mapped to a quantitative domain), common visualization frameworks like the Xmdv-Tool [Ward, 1994], Visage [Koolejchick et al., 1997], or Polaris [Stoltte et al., 2002] as well as standard visualization techniques like Parallel Coordinates [Inselberg and Dimsdale, 1987] or more or less sophisticated diagrams and charts [Harris, 1999] have their eligibility for visualizing time-oriented data. For simple data and basic analysis tasks, these approaches outperform specialized techniques, because they are easy to learn and understand (e.g., common time diagram). However, in many cases, time is treated as one quantitative variable among many others, as for instance in Parallel Coordinates – not more, not less. Therefore, generic approaches usually do not support establishing a direct visual connection between multiple variables and the time axis, they do not communicate the specific aspects of time (e.g., the different levels of temporal aggregation), and they are limited in their capabilities to enable direct interactive exploration and browsing of time-oriented data, which is essential for a successful visual analysis.

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 [Silva and Cunace, 2002, Müller and Schumann, 2003, Alger et al., 2005]. 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 tasks and goals. What is required is a systematic view on visualization of time-oriented data.

4.1 Visualization Aspects

Instead of using formal or theoretical constructs, we decided to present a systematic view 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 visualized (time and data)?
2. Why is it visualized (user tasks)?
3. How is it visualized (visual representation)?

Because any visualization originates from some data, the first question addresses the data and the time axis to which the data are tied. The motivation for generating a visualization is reflected in the second question. It relates to the aim of the visualization and examines the tasks of the users. How the data are represented is covered by the third question. The following sections will provide more detailed explanations of each criterion, including sub-criteria, and respective categories.

4.1.1 What: Time & Data

Self-evidently, the temporal dimension itself is an interesting aspect that any approach to temporal visual analysis has to consider. It is virtually impossible to design effective analysis methods, without knowledge about the characteristics of the time axis as well as the given data. The complete design space of time itself has been presented in detail in Section 3.1, the characteristics of the given data have been introduced in Section 3.2. Here, we will only briefly summarize these aspects.

Characteristics of the time axis The following list briefly reiterates the key criteria of the time axis:
4.1 Visualization Aspects

- **Rate of change**: How fast is a data element changing or how much difference is there from one data element to another over time?
  - Starting point: data element
  - Search for: magnitude of change over time
  - Example: "How fast did the price of gasoline change last year?"
- **Sequence**: 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 accident?"
- **Synchronization**: Do data elements exist together?
  - Starting point: data elements
  - Searching for: occurrence at the same point or interval in time
  - Example: "Is Jill's birthday on Easter Monday this year?"

In principle, this list of tasks covers two major cases. First, given one or more data elements the user is searching for time primitives that exhibit the given data values, and second, given one or more primitives the user seeks to discern their associated data values. This reflects the well-established distinction between identification (looking for data values) and localization (looking for points in time and space).

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

Elementary tasks are further divided into lookup, comparison, and relation seeking. The lookup task is divided into direct and inverse lookup and in this way includes both searching for data values and searching for points in space and time. **Relation seeking** tasks search for occurrences of relations specified between data characteristics or references, for example, looking for courses that take place on Monday. 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 relate characteristics, whereas inverse comparison tasks search for relations between references.

Synoptic tasks are 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 find 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 of high relevance.

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 the time axis and the corresponding data are to be represented. Chapter 5 will give evidence 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: time to space vs. time to 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, data variables are mapped to some geometry within a spatial
4.1 Visualization Aspects

time axis is exactly represented by an animated visualization. In practice, this is only rarely the case. 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.

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. Data that contain only few snapshots of the underlying phenomenon should preferably be represented via 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. This way the underlying dynamics in the data are reflected best by the visualization.

The distinction between the mappings time to space and 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 of and the major trends in the analyzed data. However, there are also critical voices on animations used for the purpose of visualization (e.g., [Tversky et al., 2002, Simmons and Renninger, 2005]). Especially when larger multivariate time series have to be visualized, animation-based approaches reach their limits. In such cases, users cannot follow all changes in the visual representation or animations simply take too long and users face an indigestible flood of information. This problem aggravates when using animations in multiple views visualization environments. On the other hand, if animations are well designed and can be steered interactively by the user (e.g., slow motion or fast forward) mapping the dimension of time to the physical time is beneficial. Not only from the perception point of view, but also because using physical time for the visual mapping implies that the spatial dimensions of the presentation space can be used exclusively to visualize the data depending on time.

This is not the case 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 manner. The fact that static representations show all information on one screen is advantageous to 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 representation. 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 data sets, analytic methods and interaction are mandatory to avoid visual clutter.

Dimensionality of the presentation space: 2D vs. 3D

The presentation space of a visualization can be either two dimensional or three dimensional, or short 2D or 3D. 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 for visualizing 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 exists for a computer display, projection is required before rendering 3D visualizations. Usually the projection is transparent to the user, commonly realized through standard computer graphics methods, which require no additional efforts.

Fig. 4.4: Mapping to 2D: Spiral geometry is used to represent the time axis and data is 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 4.4) 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 the dimension of time, the possibilities of encoding the data depending on time are restricted. One variable can be encoded to the remaining spatial dimension of the presentation space, as for instance in a bar chart, 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
4.2 Visualization Design

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, are 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. 4.6: Appropriate Depending on whether linear or cyclic character of the data is mapped, different insights can be gained from visual representations.

The example illustrates that in addition to using the right kind of representation (linear vs. cyclic), it is also necessary to find an appropriate parameterization of the visual representation. The difficulty is to find suitable parameter settings to uncover patterns in unknown data sets. Automatically animating through possible parameter values – cycle lengths in our example – is one option to assist users in finding such patterns. During the course of the animation, visual patterns emerge as the cycle length is approaching cycles in the data that match in length. Analytic methods can help in narrowing down the search space, which in our example means finding promising candidates with adequate cycle length [Yung et al., 2000]. Additionally, interactive exploration is useful to allow users to experiment with different cycle lengths. Less sharp or uncommon patterns, which are hard to distill using analytic methods and easily overseen during an animation, can be found this way. It is the interplay of visual means, analytic methods, and interaction that allows us to generate visualizations like the spiral in Figure 4.6 (right), and hence, to take full advantage of the capabilities of the human perceptual system, e.g., in recognizing patterns and motion. We will come back to this point in Chapter 6.

Data characteristics: continuous data streams TODO

4.2.2 Task Level

We already 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. Designing adequate color scales is an intricate step that influences the expressiveness of visual representations significantly. In order to arrive at expressive color coding, it is necessary to provide flexible color schemes that can be adapted not only to the data, but also to the task at hand.

In the following, we will use color coding as an example to demonstrate how aspects of the task level influence the visualization design process. We will explain how color scales can be generated in a task-dependent manner, and how they can be applied to the visualization of cyclic time-oriented data. But let us first briefly review general aspects of color coding.

Color Coding. A color coding scheme can be characterized by a color mapping function \( f : D \to C \) that maps values of a data set \( D \) 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 such colors with distinct data values (or groups of values). Moreover, color coding 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 imply that these colors represent data values that are similar. Figure 4.7 demonstrates a basic mapping strategy.

Fig. 4.7: Simple strategy for mapping data to color.

Toles describes further factors that are relevant for color coding [Toles, 2007]. We adapted his statements for the case of visualization of time-oriented data:

- **Characteristics of the data**: We already identified characteristics of time and time-oriented that should be considered when designing visual representations. In the particular case of color coding, 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
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 color values with concrete data values. In order to support comparison tasks, all variables involved in the comparison must be represented by a unified common color scale, which can be problematic when variables exhibit quite different value ranges. The next paragraphs will provide more details on how efficient color scales for lookup and comparison tasks can be designed.

Color Scales for the Lookup Task: Lookup tasks are basically a search for specific data characteristics or for references in time that exhibit such specific characteristics (inverse lookup). The inverse lookup task is relatively easy to handle because relevant data values or subsets are known beforehand and are thus can be accentuated using appropriate colors. On the other hand, the design of color scales for direct lookup is intricate as the whole range of data values must be easily identifiable. One way to arrive at suitable color scales is to extract statistical meta data from the underlying value range to drive the adjustment of a predefined color scale. Let’s take a look at three possible ways of adaptation:

- Expansion of the value range: The intervals displayed in the color scale legend are a key to easy and correct interpretation of a color-coded visualization. However, usually the color legend linearly samples intermediate labels between the data’s minimum and maximum value, which often results in odd and difficult to interpret values such as 2.357 or 355.87. Consider for example the left color scale shown in Figure 4.10(a). 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 range of values mapped to the color scale. This is done in a way so as to arrive at color mapping that is more easy to interpret. The right color scale in Figure 4.10(a) demonstrates this positive effect.

- Adjustment of control points: A color map is defined by several control points each of which 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 4.10(b) shows an example where control points are equally distributed (interpolation is not applied for this segmented color scale). While equally distributed control points are generally 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 4.10(b), where control points have been shifted according the the data distribution. This allows users to easily associate data values with certain ranges of the data distribution.

- Skewing of the color mapping function: This last adaptation strategy strives for appropriate handling of data ranges with specific value distributions. Uneven distributions 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 mapping functions can be applied to visualize data with skewed value distributions. In cases where the underlying data distribution cannot be described by an analytical function, equalization can be applied to design a more effective color scale. The net effect of equalization is that colors of a color scale are in ac-
4.2 Visualization Design

Each attribute. 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 (b) allows for visual comparison, but data values of the first and third attribute are no longer distinguishable because their value ranges are rather small compared to the one of the second attribute. Figure (c) illustrates that adapting to color scale to the global value distribution is beneficial. Figure (d) shows the visualization outcome when applying the color scale construction as described before: 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.

Fig. 4.13: Different color scales for visual comparison of three variables. (a) Individual color scales; (b) one global color scale; (c) adapted color scale based on Box-Whisker plot; (d) optimized color scale for comparison tasks.

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 Representation Level

Finally, there are design issues at the presentation level. Communicating the time-dependence of data first and foremost 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 for a design decision to be made at the presentation level. Visualization approaches using a 2D presentation space have to take care that the time axis is emphasized, because time and data dimensions usually 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 by 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 data sets in 2D. The basic idea is that for each dimension of the n-dimensional data space, a coordinate axis on the plane is constructed and scaled with respect to the corresponding value range. In this way, a lossless projection of the n-dimensional data space onto the 2-dimensional screen space is accomplished. Parallel Coordinates [Inselberg and Dimsdale, 1990] are a well-known example for 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

Fig. 4.14: In a parallel coordinates plot, the time axis (left-most axis) is just one of many axes and it is not treated in any particular way to emphasize the importance of time.
4.3 Summary

Solving the visualization problem primarily requires answering the three questions: (1) What has to be visualized? (2) Why it has to be visualized? (3) How should it be visualized? The answers to the first two questions decide about the answer of the third question.

In the case of visualizing time-oriented data answering the what-question requires both specifying the characteristics of the time axis as well as specifying the characteristics of the data associated with the time axis. In Chapter 3, we have shown that many different aspects characterize time and time-oriented data. It is virtually impossible to simultaneously check all of them within a single visualization process. On top of this, there exists no visualization technique that is able to handle all of the different aspects simultaneously and to present 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 task at hand have to be communicated by the visual representation, others can be diminished or even omitted. However, this is a crucial problem, and most of today's visualization systems do not support the process of generating appropriate task-specific visual representations. Thus, our primary aim can only be to communicate the problem, but also to demonstrate the need and potential of considering the interrelation between the what, why, and how aspects by example as done in Section 4.2.

Table 4.1 again summarizes the key characteristics of the three aspects. The what-aspect has been detailed in Tables 3.1 and 3.2. For describing the why-aspect, an
Bar Graph, Spike Graph

Bar Graphs (see Fig. 5.3, left) [Harris, 1999] are a well-known and widely used type of representation where bars are used to depict data values. This makes comparisons easier as with Point Plots. As bar length is used to depict data values, only variables with a ratio scale (having a natural zero) might 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 it is the case with Point Plots.

A subtype of Bar Graphs that is often used for graphing stock market data (volume) are Spike Graphs (see Fig. 5.3, right) [Harris, 1999]. Vertical bars that are reduced to lines (spikes) for each data value. This way, a good visual balance is achieved between focusing on individual values and overall development.

References
null
Cluster and Calendar-Based Visualization

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. Van Wijk and van Selow presented an approach that allows for finding temporal patterns at different temporal granularities (van Wijk and van Selow, 1999). The starting point of their approach is to represent the course of a day as a line chart covering the 24 hours of a day. Two alternative methods to visualize these charts are proposed. The first is to present charts for all days in a dense series effectively generating a 3D surface that encodes the daily courses as well as the development of the courses at the higher temporal granularity (see Figure 5.7 (left)). This allows users to detect short term daily patterns and long term trends. To assist in the detection, van Wijk and van Selow suggest grouping daily courses into clusters. The process generates clusters whose aggregated daily course are representative for all days belonging to a cluster. Cluster affiliation of dates is then color coded into a calendar as depicted in Figure 5.7 (right). The number of clusters shown can be chosen by the users to find the level of abstraction that suits data and task at hand. The combination of analytical and visual methods as applied here is useful to identify days of common and exceptional daily behavior.

References

Fig. 5.11: The DateLens applies fisheye distortion to emphasize the user's focus and to have enough display space to show textual information. The temporal context is maintained at all times, by providing a tabular calendar grid and by rendering calendar entries as colored bars.

Most people use calendars to plan their daily life, for instance, to maintain a list of appointments or bookmark future events. Bederson et al. have developed a tool to ease planning of the personal schedule and its analysis on small handheld devices (Bederson et al., 2004). As 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, the DateLens magnifies table cells in a way so as to provide important information currently in the user's focus with more display space (see Figure 5.11). The fisheye distortion used to magnify the focus preserves contextual information at all times in the context of the display. If sufficient space is available calendar entries are displayed in textual form. Otherwise, temporal intervals of calendar entries are indicated by bars that visualize 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

Fig. 5.12: Decision Chart [Harris, 1989] - Future decisions and potential alternative outcomes are depicted over time along with their probabilities.

A Decision Chart [Harris, 1999] allows for depicting future decisions and potential alternative outcomes along with their probabilities over time (see Fig. 5.12). The main advantage of this visualization is that it allows for investigating possible outcomes and implications before decisions are made. It is one of very few techniques that support the branching time characteristic (see Section 3.1.1.4). Visually, the horizontal position of the represented information elements (decisions and probabilities) are determined by time within the 2D representational space. However, the temporal context itself is not of prime interest and given as orientation by a simple time scale on the bottom of the representation.

References
TimeHistogram 3D

Fig. 5.15: Time-Histogram 3D [Kosara et al., 2004] – Uses third dimension to represent time. For each time step a histogram is drawn as a row of cuboids. Example shows CFD (computational fluid dynamics) data.

TimeHistogram 3D [Kosara et al., 2004] is an interactive extension of well known Histograms especially designed for temporal data. It has been developed in order to give an overview of complex time-varying data in the application context of computational fluid dynamics (CFD). A design goal of these techniques was to show temporal information in static images while maintaining the easy readability of standard histograms. In TimeHistograms 3D, the third dimension is used to represent time. Hence, for each time step a histogram is drawn as a row of cuboids (see Fig. 5.15). Several interactive features such as brushing, scaling, and a 2D context display that is shown on the background of the histogram are part of this technique.

References

Intrusion Monitoring

Fig. 5.16: The glyph in the center represents the monitored system. Connections to remote hosts are depicted by radially arranged lines. Critical and suspicious connections are visualized by red and yellow color, respectively.

Erbacher et al. describe a system that visualizes time-stamped network-related log messages dynamically generated by a monitored system [Erbacher et al., 2002]. These messages correspond to events on a linear continuous time axis. The visualization shows the monitored server system as a central glyph encoding the number of users and the server's load (see Figure 5.16). Events are shown as radially arranged lines at whose end the remote host is shown as a small 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
A Technique — Visual element that represents an incident at a given time using a line. A line can be horizontal, vertical, or an angled line. Each endpoint of the line is labeled with the time of the incident.

Fig. 5.15: Timeline — Visual element that represents an incident at a given time using a line. A line can be horizontal, vertical, or an angled line. Each endpoint of the line is labeled with the time of the incident.

Fig. 5.17: Anomera — Anomera is a technique for visualizing network structure and usage statistics. It is a powerful visualization technique that allows users to easily identify patterns and anomalies in network traffic data. The technique uses a directed graph representation of the network, with nodes representing network devices and edges representing communication flows. Anomera's key feature is its ability to highlight unusual or unexpected traffic behavior, such as sudden spikes or drops in traffic volume, which can indicate potential security threats or operational issues. The technique is particularly useful for network administrators who need to monitor and analyze large-scale network data in real-time.

References

Fig. 5.19: A perspective wall representing temporal 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 just indicate files as yellow boxes.

When time-oriented data are linked to a longer time axis (i.e., wide span in time or many temporal primitives), the visual representation is usually difficult as it becomes very wide and exhibits an aspect ratio not suited for common displays. The perspective wall [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 past (to the left) and the future (to the right) of the data. The context is bent perspective to reduce the display space occupation of these regions, effectively allowing for better space utilization in the focus (see Figure 5.19). Interaction methods are provided to enable users to navigate the time axis in order to bring different time spans into the focus. The actual data representation on the wall may vary among applications. It is, for instance, possible to use simple bar charts or more advanced visual representations such as the ThemRiver. The only requirement to be fulfilled by the data mapping is that it uses a linear time axis that extends from left to right.

References
Fig. 5.23: (SOPO) Sets of Possible Occurrences (Kosara and Miliusch, 2002) – Left: single SOPO, Right: SOPOView – Part of ActusView to represent medical treatment plans.

The interactive 2D technique SOPO View (Kosara and Miliusch, 2002) utilizes Rir’s Sets of Possible Occurrences (SOPOs) (Rit, 1986). The axes of the diagram are used to depict start interval (x-axis) and end interval (y-axis) (see Fig. 5.23, left). Minimum and maximum duration are the constraining borders parallel to the 45° time flow axis. The area which a SOPO covers contains all intervals that fit the specification given by means of earliest start, latest start, earliest end, latest end, minimum, and maximum duration. Hence, any point in this diagram represents a complete interval, specified by its start (x-coordinate) and end time (y-coordinate). SOPOs were designed for the easy graphical propagation of temporal constraints, not for making a complex notion of time easy to understand. Specifically, parallel plans and hierarchical decomposition are very hard to depict and would work. Moreover, a notation for undefined parts is missing in the original design (Kosara and Miliusch, 2002).

References


Fig. 5.24: SolarPlot [Chua, 1998] – Values are plotted around the circumference of a circle. Example shows ticket sales data. Left: simple SolarPlot, Center: SolarPlot with a focus interval, Right: SolarPlot + Aggregate TreeMap Hybrid – Adds information on hierarchical decomposition.

With the SolarPlot technique (Chua, 1998), values are plotted around the circumference of a circle. It is particularly useful for viewing global trends and relationships in large data sets. Automatic aggregation is performed depending on the size of the base display circle which makes it a circular histogram technique. Figure 5.24 shows ticket sales data using the SolarPlot representation. Continuous expansion and contraction is supported via user interaction in order to view different levels of aggregation. Furthermore, zoom, magic lenses, and spatial distortion support focusing on a particular interval in more detail while preserving contextual information (see Fig. 5.24, center). A variation of this technique are SolarPlot + Aggregate TreeMap Hybrids where hierarchy information can be shown additionally (see Fig. 5.24, right).

References

Spiral Display

Fig. 5.27: Spiral Display (Carlisle and Konstan, 1998) – Cycle length can be adjusted interactively and also animated automatically for discovering period lengths.

Most line graph and circle-based representations have the disadvantage of discontinuities at the transition from the end of one period to the beginning of the subsequent one. A spiral layout tries to avoid this by "smoothly connecting" the periodical data in order to form a single, continuous set. The interactive Spiral Display (Carlisle and Konstan, 1998) uses so-called "blobs" for representing quantitative values at discrete time steps. Blobs are filled circular elements whose area is proportional to the data value (see Fig. 5.27, left). Additionally, interval data can be visualized by using filled bars (see Fig. 5.27, center). The period length of the spiral can be adjusted interactively and also animated automatically for discovering period lengths. For working with several data sets simultaneously, either Bar Charts on the spiral (see Fig. 5.27, right) or multiple linked spirals might be used.

The 3D Spiral Display is a 3D extension of the previous technique allowing for the display of multiple data sets by using the z-axis and aligning them on top of each other. Therefore, the z-axis has no quantitative meaning and the different data sets are additionally color coded differently. Furthermore, hollow "cans", whose volume is proportional to the data value, are used for representation instead of "blobs" to prevent occlusion.

References

Ring Maps

Fig. 5.31: Ring maps representing health-related alert levels (green, yellow, orange, red) for various zip code regions during a period of 24 weeks (left) and human activities during the course of a day (right).

The basic idea of ring maps is to create multiple rings each of which subdivided into an equal number of ring segments [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 multiple ways. One can utilize ring maps to visualize spatio-temporal data. To this end, a map is shown the center and the segments at a particular angle are associated with a unique area of the map as depicted in Figure 5.31 (left). A time-series for each area 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. Figure 5.31 (right) shows an application of ring maps where the hours of the day are mapped to the ring segments and rings represent different activities a person can be busy with during the course of a day. The degree of activity is color coded. This time the center of the display is used to show a complementary 3D representation to assist users in spotting highly active activities.

References

Silhouette Graph, Layer Area Graph, Circular Silhouette Graph

Fig. 5.32: Top left: Silhouette Graph; Bottom left: Layer Area Graph; Top right: Circular Silhouette Graph.

Silhouette Graphs [Harris, 1999] (see Fig. 5.32, top left) for example emphasize the visual impression of a time-series by filling the area below the plotted line in order to create distinct silhouettes that are easier to compare when put side by side. The Circular Silhouette Graph [Harris, 1999] (see Fig. 5.32, top right) is a circular version of the linear Silhouette Graph presented for emphasizing periodicities in time.

When time-series are compared that share the same unit and can be summed up, also other techniques such as a Layer Area Graph [Harris, 1999] (see Fig. 5.32, bottom left) might be used. This is a stacked visualization where time-series plots are drawn upon each other. This kind of representation is sensitive to data series ordering which influences the visual appearance of the individual layers. It mainly emphasizes the total sum of values.

References


Horizon Graph

Fig. 5.33: Horizon Graph [Reijzer, 2008] – Allows for the comparison of large amounts of time series at once by mirroring and layering bands. Left: Construction scheme. Right: Example for comparing stock market data.

Horizon Graphs where developed for comparing a large number of variables side by side over time [Reijzer, 2008]. By applying mirroring and layering bands to Lineplots, less vertical space is used which means that data density is increased while the data resolution is preserved. In doing so, the readability of the graph is more effective and efficient than by decreasing the size of a Lineplot only. Fig. 5.33 (left) demonstrates the principle of Horizon Graphs: Starting from a common Lineplot (a), positive and negative values are colored (b) and mirrored horizontally subsequently (c). Furthermore, the value range is divided into equally sized bands (d) that are discriminated by increasing color intensity towards the maximum and minimum values and are layered upon each other (e).

Horizon Graphs have been investigated empirically whereas value comparison tasks have been studied and different configurations of chart height and number of bands were compared [Heer et al., 2009]. Results are that mirroring of negative values does not have negative effects and using layered bands has been found to be more effective than the standard plot when the chart sizes get quite small.

References


MultiComb

Chart plots are expressive visual representations for bivariate data. The rationale behind the MultiComb visualization is to utilize this expressiveness for representing multiple time-dependent variables. The MultiComb consists of multiple radially arranged plots (see Tominski et al., 2004). Two alternative designs exist: time axes are arranged around the display center (see Figure 5.36 left) or time axes extend outward from the MultiComb's center (see Figure 5.36 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 chart 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 for a detailed comparison of data values for a selected time point, or an aggregated view might display a temporal history of a temporal data stream in an aggregated fashion. Various possibilities for interaction allow users to browse in time, zoom into details of the time axes, as well as to add, remove, and reorder plots, and to rotate the MultiComb.

References


Recursive Pattern

As datasets become larger, it is getting more and more difficult to use screen space efficiently. The most space-efficient way of visualizing data is to represent them on a per-pixel basis. Keim et al. suggest a variety of pixel-based visualization approaches of which the recursive pattern techniques limited to display large multivariate time series (Keim et al., 1995). The idea is to recursively arrange pixels such that multiple time-dependent variables can be analyzed in their inherently hierarchical temporal context. Figure 5.37 shows financial data as a pixel-based visualization. The initial step is to map data of the day to a 3 by 3 pixel group. This group is then used to form a larger group for a week containing 5 by 1 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., 4 weeks comprise a month, 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. The dense pixel display offers a good overview of large data sets. Details can be discerned using interactive zooming.

References

References


Theorem

Let \( f(x) \) be a continuous function on \( [a, b] \). Then, for any \( \epsilon > 0 \), there exists a \( \delta > 0 \) such that for all \( x, y \) in \( [a, b] \), if \( |x - y| < \delta \), then \( |f(x) - f(y)| < \epsilon \).

Fig. 5.32: Temporally averaged squared difference between the input and output of the network.
LifeLines2 [Wang et al., 2009] is an interactive visual exploration interface for instantaneous events based on categorical data (e.g., high/normal/low body temperature). Data records are composed of triangles that represent event occurrences and are layed out along a shared horizontal time axis. Individual event categories are color-coded and stacked vertically. LifeLines2 introduces a powerful set of the three operators align, rank, and filter for exploring the data: First, align arranges all records along a specific event type an temporal order. E.g., aligning a group of patients horizontally along their first heart attack. Additionally, the time line switches from an absolute time representation to relative time originating from the specified event (e.g., 1 week before, or 2 weeks after the first heart attack). Second, rank is ordering records according to the number of occurrences of a specified event type. Third, filter allows for searching of particular sequences of events including both, presence of events and absence of events (e.g., patients having had a heart attack but no stroke following it). Moreover, an aggregation of events is represented as histogram that shows the number of occurrences.

References


Similan [Wongsuphasawat and Shneiderman, 2009] uses the same visual representation layout as LifeLines2 but uses a different approach to data exploration. Instead of filtering according to an event specification or a given event sequence records are ranked according to their similarity (query-by-example). This can be used for example to search for groups of patients who share similar temporal patterns. For determining the similarity of event sequences a similarity measure (M&M measure) has been developed. Moreover, weights of factors that determine the similarity measure can be adjusted interactively by the user.

References

Fig. 5.44: PeopleGarden [Xiong and Donath, 1999] – Visualization of users’ interaction histories in discussion groups. It visualizes users of an online environment based on time of posting, amount of response, and whether a post starts a new conversation.

*PeopleGarden* [Xiong and Donath, 1999] uses the so-called data portrait as novel graphical representation of users’ interaction histories in discussion groups. It visualizes users of an online interaction environment based on time of posting, amount of response, and whether a post starts a new conversation. For intuitive understanding, the metaphors of a garden and a flower are used whereas the garden metaphor is used for the whole environment and the flower metaphor is used for the representation of individuals within the environment. The petals of a flower represent individual postings of a user (see Fig. 5.44, upper left). The time of posting is mapped to ordering and saturation of a flower’s 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 (see Fig. 5.44, lower left). Furthermore, the height of the flower gives information about how long a specific user is member of the discussion group. Using these visual representations, it is easy to spot for example dominant voices, long time participants, or how active a group is very quickly via intuitive metaphors (see Fig. 5.44, right).

References


Fig. 5.45: 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).

Data that describe items being 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. visualize temporal sequences of transactions by means of so-called Timeline Trees [Burch et al., 2008].

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 for the current level of detail multiple sequences of boxes, where color and box sizes are used to encode data values (e.g., product price) of an item (or group) at a particular time point. Thumbnails for each leaf of the information hierarchy show an overview of transaction masked by the corresponding leaf node. Enhanced with several interaction facilities, Timeline Trees help users in understanding trends in the data and in finding relation between different levels of abstractions (e.g., different product groups, or product groups and specific products).

References

Kiviat Tube

Fig. 5.48: A three-dimensional kiviat tube representing 7 time-dependent variables. Peaks and valleys indicate ups and downs in evolution of the data over time. "Wings" assist in associating features of the kiviat tube to particular variables in the data.

The kiviat tube visualizes multiple time-dependent variables [Tomiński et al., 2005]. The construction of a kiviat tube is as simple as placing multiple time diagrams circularly in 3D space, where all diagrams share a common time axis. Each diagram can be imagined as a time plot for a particular variable. However, instead of drawing multiple plots, a three-dimensional surface is constructed. This way, multiple otherwise separated time series are combined to form a single 3D body that represents the data set as a whole (see Figure 5.48). The spatial characteristics of a kiviat tube can be recognized easily, which allows users to identify peaks or valleys in the data. Additional semitransparent “wings” assist in relating identified patterns to particular variables. Common interaction methods such as zooming and rotation around arbitrary axes is complemented by specifically tailored interactions: rotation around the time axis enables users to quickly access variables on all sides of the kiviat tube, and interactive axes allow them to browse back and forth in time to visit different intervals of a possible large time range.

Temporal Star

Fig. 5.49: Temporal Star [Nothomme-Frature, 2002] – 3D representation of Circular Column Graphs that are arranged in a row to represent the each time step. Right: A transparent veil can be displayed to enhance the perception of the data set's evolution.

The Temporal Star technique [Nothomme-Frature, 2002] is an extended multi-axes plot for multivariate data structures using the 3rd dimension to represent time. For each point in time, a Circular Column Graph is drawn that represents each variable’s value as bar length in a circular arrangement. The Circular Column Graphs are aligned in a row to represent the development of the dataset over time (see Fig. 5.49). 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 data set’s evolution as a whole (see Fig. 5.49, right).

References


References

Fig. 5: Schematic diagram of the experimental apparatus. Fig. 6: Results of the experimental measurements. Fig. 7: Comparison of the theoretical and experimental results.
Feature-based flow visualization

Fig. 5.52: 3D ellipsoids are used to represent salient features of time-dependent flow data. Additional stream lines augment the visualization, in order to communicate the temporal character of the data. Reinders et al. employ interactively viewable animation.

Flow visualization is in itself a demanding task. In order to cope with even more challenging time-varying flow data, Reinders et al. proposed feature-based visualization of flow data (Reinders et al., 2001). Via suitable methods, features are extracted from the data. These features are further abstracted to three dimensional ellipsoids. The visualization can then focus on these abstractions, rather than dealing with the raw data. Reinders et al. argue that an iconic representation of the ellipsoids is useful to guide users to the salient features of time-dependent flow data. The visualization is further augmented with stream lines to maintain a connection to the underlying flow data. The temporal character of the data is visualized by animating the iconic representation. Interactive browsing in time is supported via VCR-like controls.

References
Gravi++ [Hinum et al., 2005] was designed to find predictors during the treatment planning of anorectic girls. It represents patients and data gathered from questionnaires during treatment. Patients are represented by icons that are positioned according to a spring-based model relative to the surrounding question icons. This leads to the formation of clusters of persons who gave similar answers (see Fig. 5.56). To visualize the changing values over time, Gravi++ uses animation. The position of each person’s icon changes over time allowing to trace, compare and analyze the changing values. Alternatively, the change over time can also 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 (see screenshot in Fig. 5.56). To visualize the exact values of each question, rings around the question’s icon can be drawn where the ring size corresponds to the attraction to the question.

References


Flocking Boids

Fig. 5.57: Stock market data 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 as the majority of prices; Right: Implicit surfaces surrounding boids help in recognizing the spatial structure of the flock.

Stock market data change dynamically during the day as prices are constantly updated. Varda Moore proposes to visualize such data by means of information flocking boids (Moore, 2004). 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 in Figure 5.57. This 3D visual representation is enhanced by enclosing boids within implicit surfaces, which helps users recognize the spatial structure of the flock. The flocking boids visualization can be useful to detect 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

null
**References**

5.3 Summary

This chapter reviewed 64 existing techniques for visualizing time and time-oriented data. From the review one can see that few general concepts reoccur in several instantiations, as for instance the utilization of the third display dimension to encode time or the mapping to spiral shapes. However, the majority of publications is specific to a particular what and why, and as a consequence represent tailored solutions in terms of how the data is visualized. On the one hand, specific solutions are highly adapted and fine-tuned to be successful in supporting a small 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 daily work is to make sense of constantly changing data—a general framework would be favorable. How such a framework could look like and what its components would be will be discussed in the next chapter.

References

Chapter 6: Conceptual Framework for Visual Analytics

Fig. 6.1: Conceptual Framework for Visual Analytics

The framework for visual analytics involves the following components:

1. Data Collection
2. Data Preprocessing
3. Data Visualization
4. Analysis
5. Reporting
6. Feedback

This framework is designed to help users analyze and understand complex data sets through visual representations. It includes a step-by-step process for data collection, preprocessing, visualization, analysis, reporting, and feedback, ensuring that the analytic process is comprehensive and effective.

In the figure, the flowchart illustrates the interaction between these components, highlighting the iterative nature of visual analytics. Each step is crucial for ensuring that the final analysis is robust and informative.
6.2 The Analytic Kernel – Automatic Reasoning

An interesting question is how to gain the necessary specifications to fill the descriptor database. Although several of the needed information can be automatically extracted, and there actually exists methods to do so, in most cases it is required to manually specify these values. Therefore, the framework should contain a component supporting both: automatic preprocessing to extract metadata, and interactive specification of initial settings as well to complete the descriptions of available methods, the interests of a user and the data to be analyzed. We speak about initial settings, because these settings can be dynamically changed during the visual analysis process. For example, the user has finished an analysis task and his new interests or newly selected data subsets lead to adjusted metadata or user profiles.

The heart of the framework is given by the two core components, the visual kernel and the analytical kernel. The visual kernel contains a variety of visual methods for time and time-oriented data, as described in Chapter 5. The analytical kernel includes methods for the automatic analytical reasoning process. We will explain a couple of these methods with regard to time-oriented data in Subsection 6.2.

The last, but very important, component of the framework is the Graphical User Interface providing the means for an interactive exploration of the time-oriented data set. This interaction functionality will be discussed in more detail in Subsection 6.3.

6.2.1 Knowledge Discovery and Data Mining

The aim of automatic reasoning is to extract useful information and patterns from large heterogeneous data sets. The methods used range from statistics and exploratory data analysis to Data Mining and Knowledge Discovery in Databases (KDD). Exploratory data analysis was introduced by John W. Tukey [Tukey, 1977], where he illustrates the possibility to interact with data and their analysis results.

Data Mining is commonly defined as the application of algorithms to extract useful structures from large volumes of data [Fayyad et al., 2001; Han and Kamber, 2005]. It is a multidisciplinary field integrating work from areas including statistics, machine learning, information retrieval, database technology, and neural networks.

More specifically, Data Mining is mainly concerned with the tasks of clustering, classification, association-rule discovery, generalization, and prediction. KDD emphasizes the discovery process and that knowledge is the end product of such a data discovery process [Platzerky-Shapiro, 1991]. KDD has evolved, and continues to evolve, from the intersection of research fields such as machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, data visualization, and high-performance computing. The unifying goal is extracting high-level knowledge from low-level data in the context of large data sets [Fayyad et al., 1996].

To adhere to the special role of time in Data Mining, the research branch Temporal Data Mining has emerged. For the scope of this book, we will follow the grouping of temporal Data Mining tasks as presented in [Laxman and Sastry, 2006]: prediction, classification, clustering, search & retrieval, and pattern discovery. A similar categorization is used in the recently published book about Temporal Data Mining [Minu, 2010].

Prediction An important task in analyzing time-oriented data is the prediction of likely future behavior by inferring from data collected in the past and present. One essential aspect is to build a predictive model for the data. Examples for this are autoregressive models, nonstationary and stationary models, or rule-based models.

Classification The goal of classification is to automatically determine which class or category a data set, sequence, or subsequence belongs to. Examples for this are speech recognition, gesture recognition, or genome sequence classification.

Clustering Clustering is concerned with grouping data sets, sequences, or subsequences into clusters based on their similarity. For example, in the analysis of financial data, clustering could be used to group stocks that exhibit similar behavior over time. In contrast to classification, where the classes are known a priori, clusters are not defined up front.

Search & retrieval This task encompasses searching for a priori specified queries in possibly large volumes of data and is often also 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
patterns. Higher-order temporal abstraction methods are needed to derive unified qualitative values and patterns. Therefore, we have investigated methods of complex temporal abstraction.

**VIE-VENT [Miksch et al., 1996]** addresses context-sensitive and expectation-guided temporal abstraction methods in a medical application domain, namely in an intensive care unit setting. The developed methods incorporate background knowledge about data points, data intervals, and expected qualitative trend patterns to arrive at unified qualitative descriptions. They are based on context-aware schemas for data point transformation [see Fig. 6.2] and curve fitting to express the dynamics of and the reaction to different degrees data abnormalities. Smoothing and adjustment mechanisms are used to keep qualitative descriptions stable in case of shifting contexts or data oscillating near thresholds. For example, during intermittent positive pressure ventilation (IPPV), the transformation of the quantitative value $P_rCO_2 = 55$ mmHg results in a qualitative $P_rCO_2$ value of "substantially above target range". During intermittent mandatory ventilation (IMV) however, $56mmHg$ represent the "target value". Qualitative $P_rCO_2$ values and schemata of curve fitting are subsequently used to decide if the value progression happens too fast, at normal rate, or too slow (see Fig. 6.2).

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

---

1 $P_rCO_2$ = transcutaneous partial pressure of carbon dioxide

---

Fig. 6.2: VIE-VENT [Miksch et al., 1996] – The user interface of VIE-VENT. The left-hand side region shows the blood gas measurements, their corresponding qualitative temporal abstraction on the top and the actual and recommended ventilator settings below. The right-hand side region gives plots of the most important variables over the last four hours (e.g., transcutaneously assessed blood gas measurements and ventilator settings).

---

6.3 The Analysis Kernel – Automatic Reasoning

one way to deal with cases of oscillating data where abstractions and hence interpretations are frequently changing.

Another solution is presented in the The Spread [Miksch et al., 1999]. It implements a time-oriented data abstraction method to derive steady qualitative descriptions from oscillating high-frequency data. We distinguish the following steps of processing and abstracting the data:

1. **Eliminating data errors.** Sometimes up to 40% of the input data are obviously erroneous, i.e., exceed the limits of plausible values.

2. **Clarifying the curve.** Transform the still noisy data into a steady curve with some additional information about the distribution of the data along that curve.

3. **Qualifying the curve.** Abstract quantitative values to qualitative values like "normal" or "high" and join data points with equal qualitative values to time intervals.

The Spread provides parameters to adjust the abstraction process (e.g., length of time window, permitted gaps, or points of changing the qualitative value). As an example, consider a physician who is observing continuously assessed measurements and wants to find time intervals of different qualitative regions like "$P_rCO_2$ is high for 5 minutes". When looking at the raw data, which typically oscillate, the physician will certainly have difficulties in finding reasonably long time spans with stable values. The Spread is able to support the physician in making qualitative assessments of the time intervals she is interested in (see Fig. 6.3).

Temporal abstraction methods as provided in VIE-VENT and The Spread are generic methods that can be used for different purposes. In the Midgard project [Bade et al., 2004] these methods have been extended by several visualization techniques to enhance the understanding of qualitative and quantitative characteristics of a given time-oriented dataset. The challenges were not only to support the user in exploring the

---

Fig. 6.3: The Spread [Miksch et al., 1999] – The thin line shows the raw data. The red area depicts the Spread, the blue rectangles represent the derived temporal intervals of steady qualitative values. Increased oscillation leads to increased width of the spread, but not to a change of the qualitative value. The lower part of the figure shows the used parameter settings.
First we will discuss the integration of temporal data abstraction into the visualization process according to [Dos Santos and Brodlie, 2004]:

- **Data Analysis and Temporal Data Abstraction** One important step in the visualization process is the data analysis step. Temporal data abstraction allows to get an overview about univariate and multivariate huge volume of data variables by reducing the value range from quantitative values to qualitative values or patterns. Therefore, in VIE-VENT the user - the data observer - gets the raw data in combination with the point-based qualitative values or in The Spread and Midgaard approach the user observes the raw data together with the interval-based qualitative patterns. Second, the data are more erroneous, have outliers, or some data are missing. This can be tackled with knowledge-based techniques and smoothing and adjustment mechanisms, like presented in VIE-VENT and The Spread. Third, domain knowledge can be used to guide the abstraction process, as VIE-VENT's context-aware schemas for data point transformations [see Fig. 7] and curve fitting schemas to express the dynamics of the parameters and their expected development according to some activities. The fourth important aspect are the interaction with the data and the data level, as provided by the Midgaard approach, also called semantic zoom functionality. According to these different methods of temporal data abstraction, the user gets a compact and informative overview of the multivariate data and the user can investigate in the sometimes hidden dependencies of the variables.

- **Filtering and Temporal Data Abstraction** Temporal data abstraction does not provide filtering in the classical sense, because it supports reduction of data value ranges of time-oriented data.

- **Mapping and Temporal Data Abstraction** Mapping involves both the selection of an appropriate visualization technique and the mapping of data values onto expressive and effective visual attributes. Color coding schemata, positions, and markers are used to express the qualitative values and patterns of the basic and complex temporal data abstraction methods. For example, in VIE-VENT arrows, their size, and their direction (see Fig. 6.2) depict the different kind of trends of the observed variables. Usually gray values are used for erroneous or missing data and outliers. The combination of raw data with the different color-coding of qualitative values and patterns allow for easy interpretation of the qualitative regions.

- **Rendering and Temporal Data Abstraction** Midgaard extensively use different visual representations to provide a smoothly integrated semantic zoom as well as interact with and abstract the data over time.

Next we will examine how time is tackled in the above temporal abstract methods. VIE-VENT uses point- and interval-based time. The spread and Midgaard are mainly based on interval-based time. VIE-VENT utilizes different kinds of trends (e.g., very short-term trends (sample of data points based on the last minute), short-term trends (sample of data points based on the last 10 minutes), or long-term trend (sample of data points based on the last 3 hours) to abstract time interval-based qualitative values and patterns as well as to detect errors in data and outliers. Interpolations are applied to cope with missing data. The Spread uses a slanted window of variable length to calculate trends and derives stable time interval-based qualitative values and patterns to be able to cope with erroneous and missing data and outliers. Midgaard supports to abstract and interact with data points over time, the strength of Midgaard is the semantic zoom, which can be applied to multivariate variables on the data value level as well as on the time axis (compressing data over time using the ideas of the Information Mural [7] and connected box plots. Therefore, the temporal abstraction methods applied heavily depend on the users' tasks and which kinds of data and background knowledge is available and asks for a tight coupling of the analytical analysis and visual methods as it is aimed by the Visual Analytics Framework.

After discussing temporal data abstraction and dimension reduction with PCA, we now want to take a closer look on data aggregation. Clustering methods provide a basis for this purpose. Clustering relates to partitioning a dataset into subsets exhibiting a certain similarity. The clustering process also provides an abstraction of the data. Concentrating on the clusters, rather than on individual data values allows for an analysis of datasets with a much larger number of tuples. Appropriate distance or similarity measures lay the ground for clustering. Distance and similarity measures are profoundly application dependent. This has lead to a large number of different measures and clustering algorithms [7]. Selecting appropriate algorithms is typically difficult. Careful adjustment of parameters and regular validation of the results are also essential tasks in the process of clustering. Different to PCA, the variable "time" is typically included in the clustering process to reveal clustering with respect to temporal aspects. The resulting clustering may also lead to a temporal data abstraction.

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

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

low dimensional subspace of the original space for visualization purposes, if he will concentrate on the exploration of main trends.

• Mapping and PCA

Mapping is the most crucial visualization step. It involves both the selection of an appropriate visualization technique and the mapping of data values onto expressive and effective visual attributes, such as position, marker sizes, and marker color. Showing both original variables and PCs as well, or showing original data values and scores as well, supports a user and task-oriented design of visualization techniques. The user can adjust the mapping process on the basis of this information to either communicate trends, or to emphasize outliers.

• Rendering and PCA

Here a promising approach is to utilize the PCA results for tuning visual output. Figures ?? and ?? depict examples for focusing on information of the demographic data set by applying a table based view (see ??). Data objects corresponding to high trend values, or that represent outliers respectively, are rendered with more detail (by larger row space). PCA results also can be applied to automatically arrange original variables, e.g. by reordering them by their scores of the first PC.

Now we will briefly discuss the aspect of retaining the temporal context by example. We consider the use case of visually analyzing a meteorological dataset from the Potsdam Institute of Climate impact research that focuses on summer weather conditions and contains daily observations of temperature for a period of more than 35,500 days (100 years) [Nucke et al., 2004]. The data set includes five variables: summer warmth (sum of max temperatures for days with $T_{max} \geq 25^\circ$C), summer days (number of days with $T_{max} \geq 25^\circ$C), hot days (number of days with $T_{max} \geq 30^\circ$C), summer mean temperature (mean of daily average temperature $T_{mean}$), and mean of monthly temperature (mean of max day temperatures $T_{max}$). All five are quantitative variables and strongly correlated. The dataset can be visualized with a ThemeRiver (see Figure ??). In this graph, constrictions in the river stand for low data values, which indicate particularly cold summers. Broad flow snapshots characterize particularly hot summers. On first impression, a general overview about temporal developments is depicted well. However, PCA and an additional simple bar chart representation can help to derive further information from the data to find major temporal trends in terms of climate. For this purpose, the variable “time” is excluded from PCA. The Bar chart plot in Figure ?? shows the combined visualization of PC1 and PC2 for the dataset from Figure ?? i.e. Figure ?? depicts the first PC only (PC1), to which all variables contribute because of their correlations. Upward bars represent warmer conditions, whereas downward bars stand for colder summers. Frequencies of data values are mapped onto color to further distinguish typical (blue) and outlier (orange) years (the colors are not related to temperature values). Major trends are clearly visible: The first third of the time line is dominated by average warm summers mixed with the coldest summers; hot summers occur followed by cold summers in the end of this period; in general, outlier summers cumulate at the end of the time line. The PC visualization in Figure ?? depicts corresponding trends very well. Nonetheless, one should recall that this climate dataset represents a special case where all variables are strongly correlated. That correlation is the reason why PC1 separates warm and cold summers so well. When analyzing arbitrary temporal datasets, further PCs may be necessary to describe all trends. In such cases, not only more responsibility of the user is required, but also flexible mechanisms and controls are needed to determine variables that should be considered for PCA and to select PCs that should be visualized. This calls for a tight integration of analytical analysis and visualization methods as it is intended by the Visual Analytics Framework.

6.2.3 Challenges and Opportunities

In this part we will list the most important challenges that KDD and Data Mining have to deal with in the connection of a Visual Analytics framework tackling time-oriented data.

• Data Quality: How to handle missing and erroneous data? How to handle uncertain data?
• Data Integration: How can the various heterogeneous data sources be integrated?
• Data Types and Temporal Aspects: How to handle various data types with their temporal aspects and relations?
• Semantics:
• Data Provenance and Result Explanation: Where does the data come from? How to interpret the results?
• Data Streaming: How to handle continuous flow of data?
• Integrating visual and analytical methods: How to allow more user interaction using Data Mining methods? How to support communication of the Data Mining results? How can visually-controlled Data Mining work? How can data-mining-controlled visualization work?

According to these challenges one very important part are interactions, which will be covered in the next section.

6.3 Interaction

Many application fields have recognized visual techniques as a potential tool to order, manage, and understand time-oriented data. With increased need for visual exploration tools for time-oriented data in real world applications, a high degree of interactivity and advanced interaction techniques are becoming more and more important. Interaction helps users in understanding the visual mapping, in realizing the effect of visualization parameters, and in getting confident about the data. Interaction provokes curiosity — users want to put hands on their data — which is particularly useful when exploring unknown data. The importance of interaction is
The process of expectation formation and confirmation can be held in the concept of "confirmation bias." This concept suggests that people have a natural tendency to search for and interpret information in a way that confirms their preconceptions. This can lead to a self-reinforcing cycle of information seeking, where individuals seek out information that supports their existing beliefs and ignore information that contradicts them. This can be seen in various contexts, such as in the spread of conspiracy theories, where people tend to believe in theories that align with their existing worldview and discount information that contradicts it.

As a result, confirmation bias can have significant implications for decision-making, especially in situations where there is a high stakes or when people are particularly committed to their beliefs. It is important to be aware of this bias and to try to overcome it by seeking out information from multiple sources, considering alternative viewpoints, and being open to changing one's mind based on new evidence.

Diagram: The Gulf of Evaluation

The diagram above illustrates the concept of the "Gulf of Evaluation." This refers to the distance between the initial formation of an expectation and the point at which the confirmation or disconfirmation is actually measured. The gulf can be wide, leading to a situation where individuals may be more influenced by the confirmation or disconfirmation of their initial expectations than by the actual evidence.

The diagram shows a cycle with several steps: formation, confirmation, and evaluation. The formation step represents the initial expectation, which can be influenced by various factors such as personal biases or外界 information. The confirmation step involves seeking out information that supports or contradicts this expectation. Finally, the evaluation step is where the actual evidence is compared against the initial expectations to determine whether or not they are confirmed or disconfirmed.

By understanding the concept of the Gulf of Evaluation, individuals can become more aware of the potential for confirmation bias and take steps to mitigate its effects. This can involve actively seeking out diverse perspectives, engaging in critical thinking, and remaining open to revising one's beliefs in light of new evidence.
6.3 Interaction

The bottom line is that we have two opposite 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 (i.e., unresponsive) while computing is the worst scenario. On the other hand, we do need asynchrony! —for both generating the feedback (i.e., computation) and presenting the feedback (i.e., animation). The difficulty is to integrate synchrony and asynchrony. This difficulty is unique to interactive visualization systems; without interaction, this challenge would not exist.

6.3.2 Interaction Intents

[Yi et al., 2007] identified several user intents for interaction. They introduced a list of categories that describe on a high level why users would like to interact. Applied to the case of interacting with time-oriented data, the following list makes sense:

Select — Mark something as interesting. Once users spot something interesting in the visual representation, they want to mark it as such. This is usually realized as a selection operation that tags the time primitives of interest in the data and visually highlights them in the visual representation.

Explore — Show me something else. During the process of visual analysis, users have to visit different parts of the time line. This is often realized by time sliders that allow users to navigate to specific points in time or to select an entire interval to be shown.

Encode/Reconfigure — Show me a different representation/arrangement. Users want to look at the data from different perspectives, be it to conduct comparison tasks or to confirm a hypothesis generated from one representation against an alternative view. An example is switching from a linear representation of the time line to a cyclic one and vice versa.

Abstract/Elaborate — Show me more or less detail. Depending on the current task, a user may need to view certain time points in more or less detail. The natural hierarchical structure of time is often used to drive such an interactive information drill-down.

Filter — Show me something conditionally. When users have a hypothesis about the data, it is useful to provide mechanisms that assist in restricting the visualization to data items that obey to the constraints of the hypothesis. Filtering mechanisms, in particular dynamic filtering, are the tool of choice to support this user intent.

Connect — Show me related items. Starting from an initial view, users are often interested in parts of the data that are similar to the currently visualized one. Sorting the rows of a table is an example of bringing similar rows together. Highlighting elements that belong to the same cluster is another example that focusses on showing related items.

Undo/Redo — Let me go where I have been already. Undo and redo operations are needed to support the explorative nature of analytic processes. Users have to try different parameter settings and a history mechanism should support them in doing this effortlessly. While commonly supported in standard office applications, history mechanisms only slowly find their way into visual analysis tools.

Change configuration — Let me adjust the interface. In addition to adapting the visualization to data and task at hand, do users want to adapt the user interface to their particular needs.

Many of the approaches described in our survey in Chapter 5 offer support for one or the other user intent. However, a visual analytics system for time-oriented data should support all user intents in order to take full advantage of the synergy of the human’s perceptual power and the machine’s computational capabilities.

6.3.3 Fundamental Interaction Concepts

In this section, we will briefly introduce some of the key concepts that are needed or that complement the communication between the user and the visualization system.

Graphical User Interface. The user interface is largely involved in the communication between human and machine. It usually consists of multiple visualization views and one or more panels that hold several user interface controls (e.g., buttons, sliders, etc.) that are bound to visualization parameters to be adjusted to the data and task at hand. Figure 6.7 shows an example. Certainly, there are visualization parameters that are adjusted more often than others during interactive visual exploration. To ease the adjustment of important parameters, it makes sense to provide instant access and process interaction requests immediately. It has been shown that duplicating important functionality from an all-encompassing control panel to an exposed position is a useful way to drive adaptable user interfaces [Gajos et al., 2006]. For example, tools allow for accessing interaction that is most frequently used, whereas rarely applied tools have to be selected from an otherwise hidden menu structure.

Graphical user interfaces can implement two different strategies for altering visualization parameters. For discrete interaction users perform a whole bunch of adjustments, but these are transmitted to the system in a single transaction once the user wants to apply them to get visual feedback (e.g., indicated by pressing the "Apply" button). Research in human-computer-interaction has long been emphasizing the significance of continuous interaction as a requirement of interactive systems to support native human behavior [Faccioli and Masinik, 2001]. Once the user adjusts a control (e.g., dragging a slider), the input is immediately committed to generate instant visual feedback for an interaction. This is particularly useful, because examining multiple "what if" scenarios is a key aspect of exploratory analysis of time-oriented data. A scenario could, for example, refer to setting a model parameter to
6.3 Interaction

display filtering results: filtered objects can be dimmed or they can be made invisible. Dimming objects 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.

Elementary filters are commonly applied to describe conditions that must be satisfied for an object to pass through. Threshold or range sliders are an effective mechanism to filter by any particular numerical attribute. In addition to sliders, textual filters that extract objects with specified labels are useful. Such basic filters allow users to specify manifold filter conditions by simply moving a slider or typing letter strings. For complex data sets with a multitude of attributes, relying solely on elementary filters is not sufficient. The next natural step is to combine elementary filters to provide some form of multidimensional data reduction. Composite filters can be created by logically combining elementary filters. A logical AND combination generates a filter that can be passed only if an object obeys all conditions. An object can pass a logical OR filter if it satisfies any of the involved filter conditions.

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 logical combinations of filters.

Multiple views The multiple coordinated views concept addresses the separation of visualization aspects into multiple views, and hence is a natural match to the multitude of aspects occurring when visualizing time and time-oriented data [Baldasano et al., 2000, Aigner et al., 2007]. The key idea is to use multiple views, each of which is dedicated to a particular aspect of visualization task. Coordination among these views is achieved to show a consistent overall image of the visualized data. Coordination of views can be modeled by means of the model-view-controller concept [Kraus and Pape, 1988]. When a user performs an interaction in a view, the request is propagated to a controller that in turn notifies all other views of the pending operation. If no view vetoes the interaction, the controller performs the interaction on the data model. Once the necessary work is done, the controller informs all views about the particular change, the view having initiated the interaction being informed first. Each view notified this way updates itself to reflect the change visually.

When it comes to arranging multiple views on the display, there are two extremal positions one could take. One extreme is to use a fixed arrangement that has been designed by an expert and has proved to be efficient. The other extreme is to provide users with the full flexibility of windowing systems, allowing them to move and resize views arbitrarily. Both extremes have their pros and cons and they are actually applied. On the other hand, an interesting alternative is to make use of view docking. The main reason for applying docking is that it maintains flexibility, but imposes certain order in terms of what arrangements are possible. For instance, preferably, views should not overlap partially; a view should either be visible or not. Therefore, the available screen space is partitioned into regions, each containing 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). Partially visible views can thus be avoided.

6.3.4 Integrating Interactive and Automatic Methods

The methods presented in the previous sections are useful tools to facilitate visualization and analysis of time-oriented data. We already indicated that this is true only if the methods are parameterized according to the users' needs and tasks. This brings us to the third major point of our discussion—the user. User interaction is a way to manually parameterize the described visualization and analysis tools. Many tools provide an interactive graphical user interface to adjust the parameters of analytical methods (e.g., via sliders or check boxes). Visualization views can usually be adjusted via common view navigation (zoom, pan, rotation) [T], dynamic queries [T], and brushing [T].

However, it is not always easy for users to find parameter values that suit the analysis task at hand. Particularly analytical methods often have parameters that are not self-explanatory, and hence, are not easy to set. Moreover, the increasing complexity of visualization makes it more difficult for users to parameterize the visualization properly. What is needed is some form of support that helps in steering the visual analysis. A promising concept that addresses the automatic parameterization of visual representations is event-based visualization [7]. The thought behind this concept is to gain benefit from incorporating visualizability and event methodology. Commonly, events are considered happenings of interest that trigger some automatic actions. This concept is prevalent in various application fields, including active databases, software engineering, and software visualization.

In our understanding, events occur if user-defined conditions, which are expressed with respect to entities of a dataset, become true. The basic idea of event-based visualization is to let users specify their interests as event types (i.e., encapsulations of conditions), to determine if and where these interests match in the data (i.e., detect event instances), and to consider detected event instances when generating the visual representation. This basic procedure requires three main steps: 1) event specification, 2) event detection, and 3) event representation. We will give detailed descriptions of each of these steps in the next paragraphs. Fig. 6.9 illustrates how event-related components can be attached to the visualization pipeline (seg [72]), which internally comprises data analysis, filtering, mapping, and rendering. Data analysis and filtering can be realized by the methods presented in Secion 77. How time-oriented data can be mapped (and rendered) to graphical representation was shown in Section 77.

Describing user interests The event specification is the step where users describe their interests. To be able to find actual matches of user interests in the data, the event specification must be based on formal descriptions. For this purpose, event
methods operate on a differential dataset, rather than on the whole data. However, incremental methods also impose restrictions on possible event types.

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

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

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

We distinguish two basic possibilities for representing events. On the one hand, it makes sense to visualize event instances, rather than the whole dataset. In this way, the focus is set exclusively on the interests of the user. Since the number of events is usually smaller than the number of data items, even large datasets can be analyzed (certainly, the same holds true for principal components and clusters as presented in Section 7.7). This way of representing events is referred to as **explicit event representation**. On the other hand, adjusting the parameters of visual representations according to occurred event instances is a promising alternative. By pursuing what we call **implicit event representation**, we can automatically set visualization parameters according to interests detected in the data. If we assume that user interests are related to user tasks and vice versa, implicit event representation can help to achieve better targeted visual representations. The big challenge is to meet the above stated requirements merely by adapting visualization parameters. Apparently, availability of adequate visualization parameters is a prerequisite for implicit event representation.

To illustrate the potential of event-based visualization, we will discuss an example. We assume a user who has to search time-dependent human health data for unusually high numbers of cases of influenza. The task at hand is to detect where and in what situations these conditions have occurred. A possible way to accomplish this task is to use the TimeWheel technique [Tomiński et al., 2004]. However, without event integration the user will be provided with a TimeWheel that uses a standard parameterization (see Fig. 6.10(a)). The standard view shows influenza on the upper left axis (light green), time is represented on the central axis. Alpha-blending has been applied by default to reduce visual clutter. From the TimeWheel in Fig. 6.10(a) one can only guess from the labels of the axis showing influenza that there are higher numbers of cases; the alpha-blending made the particular lines almost invisible (see question mark). Several interaction steps are necessary to re-parameterize the TimeWheel to accomplish the task at hand.

In contrast to that, in an event-based visualization environment, the user can specify the interest "Final days with a high number of cases of influenza" as an event type (\( \sum (I) \geq 500 \)) to be considered for the current analysis task. The event type can be stored and may be reused in further visualization sessions or by other users. If a new dataset is opened or if new tuples are added dynamically to a time-oriented dataset, the event detection is run to determine whether or not the data conform to the condition expressed in the event type. If this is the case, event instances are created for those data portions that fulfill the condition. To reflect the interest of the data analysis, i.e., to provide an individually adjusted TimeWheel, the parameters of the visual representation have to be altered. Parameter changes can be implemented either as instantaneous actions or gradual processes (e.g., smooth animation). In our particular example, we use an action that switches color and transparency of line segments representing event instances. Days with high numbers of influenza cases are excluded from alpha-blending and are drawn in white color. Additionally, the TimeWheel is rotated (as a whole) such that the axis representing influenza is moved gradually to an exposed position. The application of a gradual process is important in this case to support users in maintaining their mental map of the visual representation. The result of applying parameter changes as response to event instances is depicted in Fig. 6.10(b). This figure illustrates that event-based visualization cues the visual analysis of time-oriented data significantly, since the visual representation is adapted to the current visualization task. In the example, the identification of days with higher numbers of influenza infections is easy.

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

Until now, event-based visualization is not suited to automatically mine potential events in time-oriented data, i.e., to support undirected search, where users have no hypotheses about the data. With a tighter integration of visual and analytical methods, it should be possible to alleviate this concern. A second challenge for future work is to find general guidelines on how to realize parameter changes that indeed highlight event instances. Because the parameter space of visualization methods is usually very large and contains many interdependencies, we have to apply sophis-
The diagram shows two different views of a structure. The two views are labeled as "6.4.1 Planning views" and "6.4.2 Vierow." The diagrams illustrate the layout and perspective of the structure from two different angles.
The time-oriented data are characterized as follows:

**Scale:** quantitative
- the variables are given by numerical values (float).

**Frame of Reference:** abstract
- the modeled and visualized data have no spatial frame of reference associated.

**Kind of Data:** state
- the state of the observed variable is measured at discrete steps in time.

**Relation to Time:** direct
- measurements are directly related to values of the time domain (each measurement belongs to a single value of the time domain).

**Number of Variables:** multivariate
- multiple stocks and multiple attributes per stock (e.g., opening and closing prices) can be dealt with in the case of supporting the application scenario of investigating stock price data.

**Internal Time:** temporal
- the data contains time values explicitly.

---

**External Time:** static
- data is given at a certain point of time and reflect the history of past values.

The analytical kernel of TimeSearcher 2 in particular supports the temporal data mining task “search & retrieval”. Basically, two options for data querying are offered by TimeSearcher 2. First, the “TimeBox query” allows for the specification of a rectangular query region that defines both, a time period and value range of interest. All time-series that comply with this query are shown whereas all others are filtered out. Moreover, multiple timeboxes can be combined to refine the query further. Second, using a “SearchBox query” the user can highlight an interesting part (pattern) of a selected time-series which is then searched for in all other time-series (see Figure 6.11).

The visual kernel of TimeSearcher 2 supports the following:

**Mapping:** static
- TimeSearcher 2 uses a mapping of time to space and represents data via time-series plots where horizontal, linear time axes are used.

**Dimensionality:** 2D
- the visual representation uses a two-dimensional representation space.

Finally, TimeSearcher 2 allows for an interactive specification of query parameters. That means, parameters for both provided query methods (TimeBox and SearchBox) are set via direct manipulation by the user. This also includes the tolerance settings for searching exactness. Figure 6.11 shows an exemplary screenshot to demonstrate the GUI of TimeSearcher 2.

---

**6.4.4 LandVis**

**6.4.5 MOSAN**

**6.5 Discussion**

In this chapter we have described both the architecture and the components of a conceptual framework supporting the interactive visual analysis of time-oriented data. We presented concrete visual analysis tools that focus on handling the data's dependency on time and explained that interaction plays a key role in the analysis process.

The conceptual framework covers the main functionality that has to be provided by Visual Analytics tools for time-oriented data, and in this way serves as a conceptual basis for assisting application designers in developing concrete systems. Several
Chapter 7
Concussion

7.1 Summary
7.2 Open Issues
7.3 Challenges
References


