Chapter 4
Visualization Aspects

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

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

Besides these dedicated techniques, time-oriented data can also be visualized using generic approaches. Since time is mostly seen as a quantitative dimension or at least can be mapped to a quantitative domain (natural or real numbers), general visualization frameworks like the Xmdv-Tool, Visage, or Polaris (see Ward, 1994; Kolojejchick et al., 1997; Stolte et al., 2002) as well as standard visualization techniques like parallel coordinates by Inselberg and Dimsdale (1990) or more or

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

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

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

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

4.1 Characterization of the Visualization Problem

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

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

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

4.1.1 What? – Time & Data

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

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

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

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

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

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

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

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

4.1.2 Why? – User Tasks

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

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

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

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

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

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

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

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

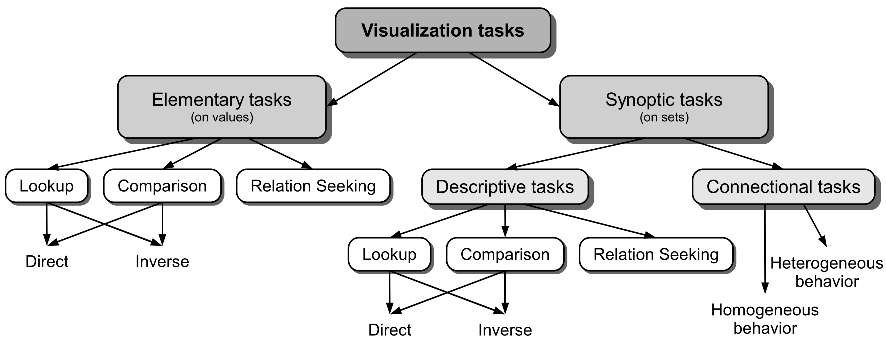


Fig. 4.1: Taxonomy of visualization tasks.

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

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

Elementary tasks:

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

Synoptic tasks:

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

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

4.1.3 How? – Visual Representation

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

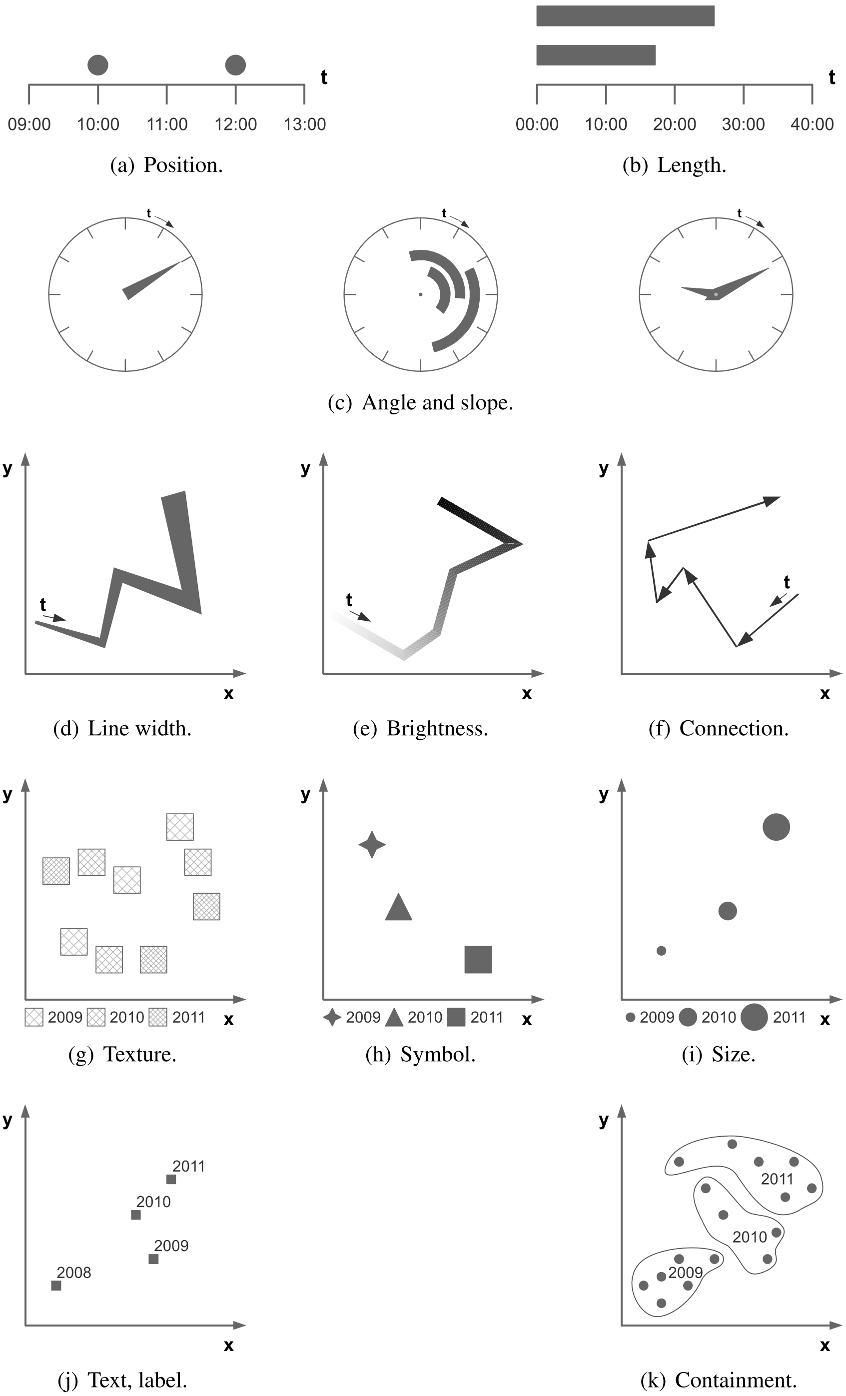
Mapping of time

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

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

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

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



(a) Position.
(b) Length.
(c) Angle and slope.
(d) Line width.
(e) Brightness.
(f) Connection.
(g) Texture.
(h) Symbol.
(i) Size.
(j) Text, label.
(k) Containment.

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

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

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

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

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

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

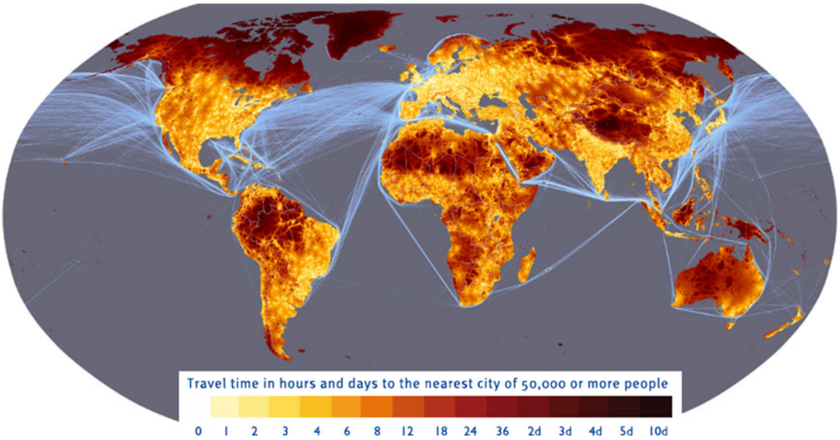


Fig. 4.4: Mapping time to color. Color encodes the time it takes to travel from a location on our planet to the nearest major city.
Source: Nelson (2008). Used with permission.

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

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

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

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

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

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

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

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

Dimensionality of the presentation space

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

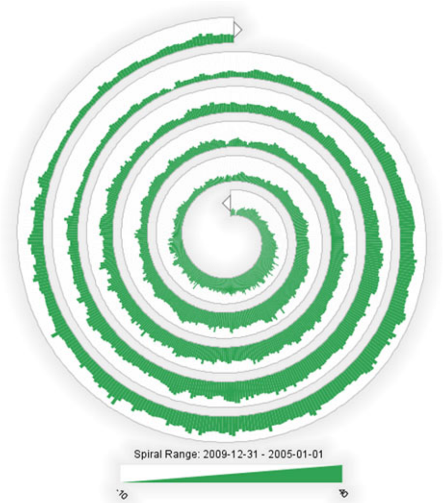


Fig. 4.6: Mapping to 2D. Spiral geometry is used to represent the time axis and data are encoded to the width of spiral segments.

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

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

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

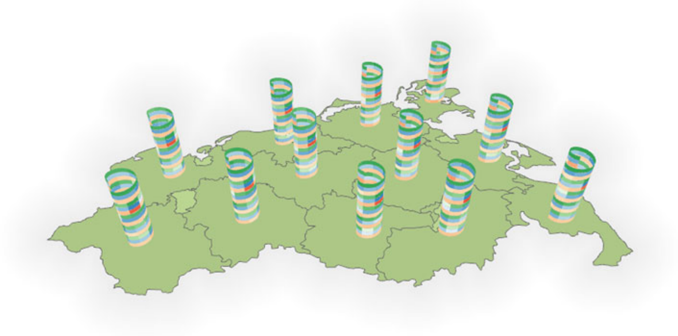


Fig. 4.7: Mapping to 3D. Three-dimensional helices represent time axes for individual regions of a map and associated data are encoded by color.

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

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

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

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

4.2 Visualization Design Examples

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

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

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

4.2.1 Data Level

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

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

Time characteristics: linear vs. cyclic representation of time

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

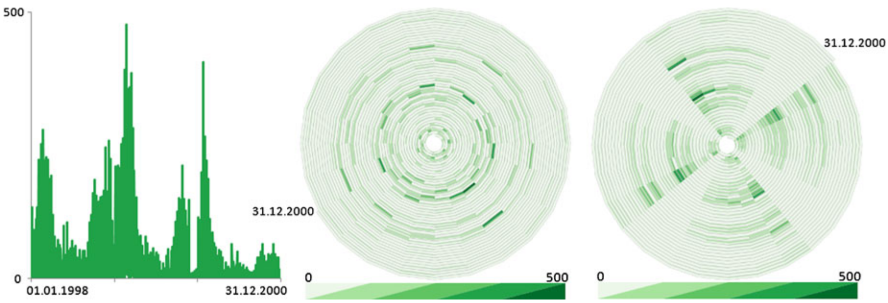


Fig. 4.8: Different insights can be gained from visual representations depending on whether the linear or cyclic character of the data is emphasized.

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

Data characteristics: abstract data vs. spatial data

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

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

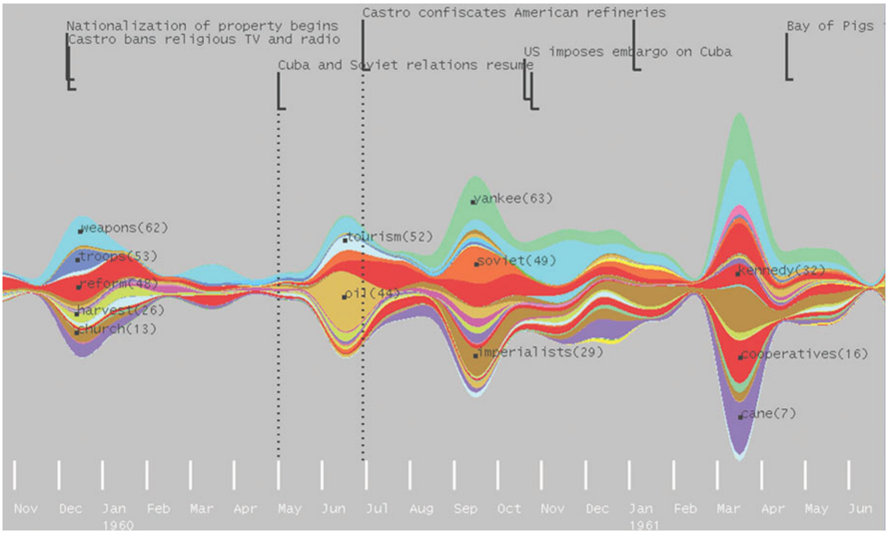


Fig. 4.9: The ThemeRiver technique is fully focused on communicating the temporal evolution of abstract time-oriented data.
Source: Havre et al. (2002). © 2002 IEEE. Used with permission.

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

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

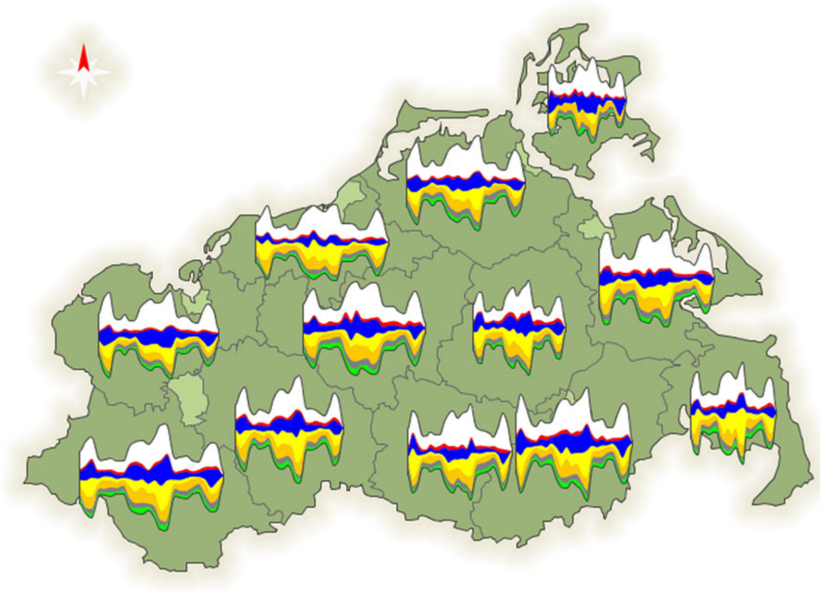


Fig. 4.10: Embedding ThemeRiver thumbnails on a map allows for communicating both temporal and spatial dependencies of spatial time-oriented data.

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

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

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

4.2.2 Task Level

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

Color coding

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

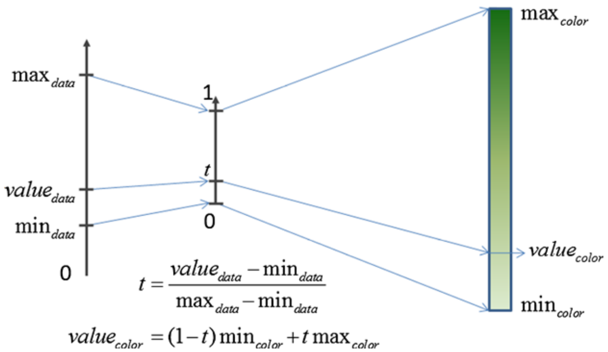


Fig. 4.11: Simple strategy for mapping data to color.

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

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

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

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

Task-dependent color coding

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

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

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

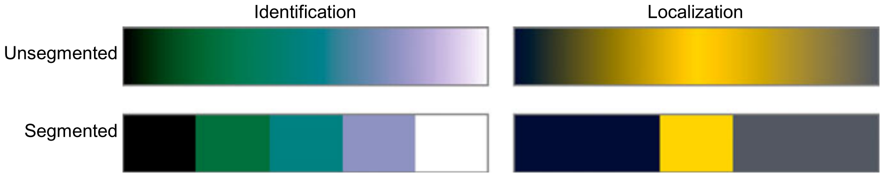


Fig. 4.12: Examples of unsegmented and segmented color scales for identification and localization of data values in a visual representation.

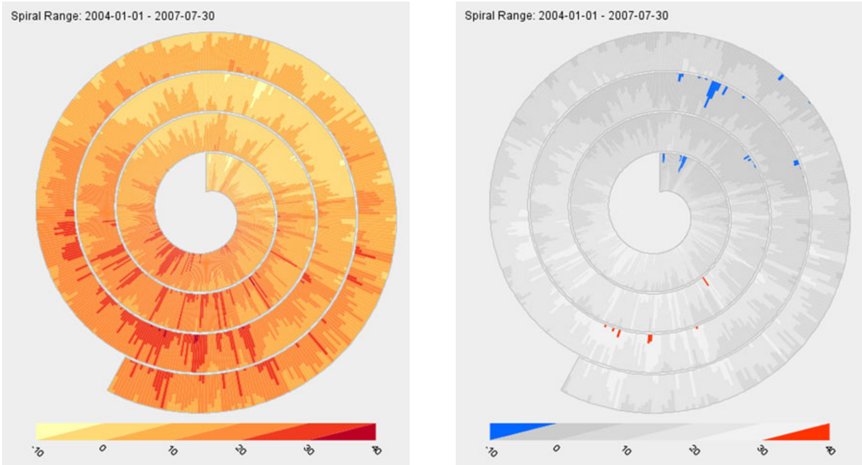
Color-coding individual data values requires unsegmented color scales. Unsegmented color scales associate unique colors with all individual data values, that is, every color of the color scale represents exactly one data value. In contrast to that, segmented color scales should be used to encode sets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

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

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

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

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

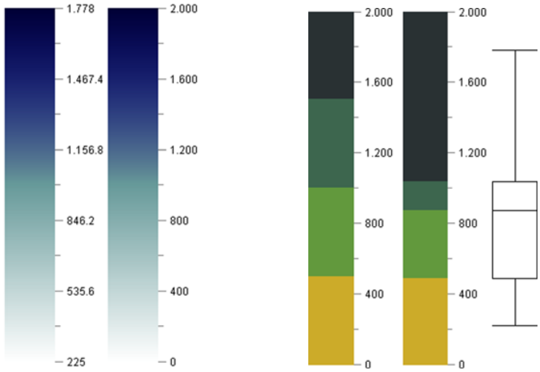


(a) Color scale for identification of values. (b) Color scale for localization of extrema.

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

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

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



(a) Value range expansion. (b) Control point adjustment.

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

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

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

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

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

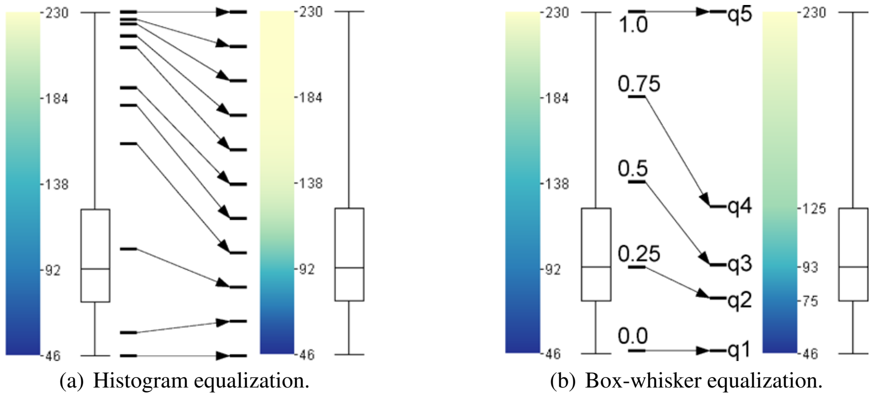


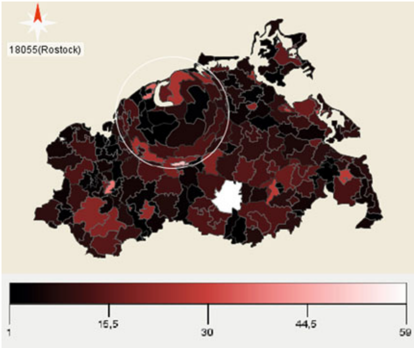
Fig. 4.15: Equalization schemas for adapting a color scale to the data distribution, which is depicted as box-whisker plots.

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

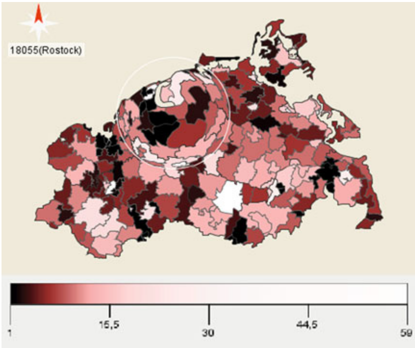
- Box-whisker equalization works similarly. Here, colors are sampled at points determined by quartiles. Quartiles partition the original data into four parts, each of which contains one-fourth of the data. The second quartile is defined as the median of the entire set of data (one half of the data lies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

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

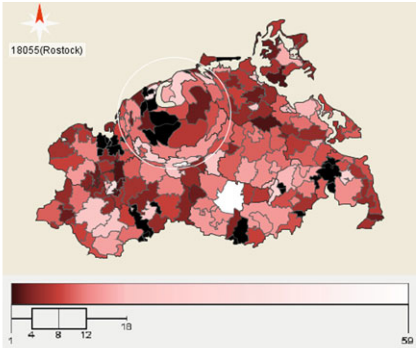
Color-coding for the comparison task The comparison of two or more time-dependent variables requires a global color scale that comprises the value ranges of all variables participating in the comparison. Particularly problematic are comparisons where the individual value ranges are quite different. For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



(a) Color coding without equalization.



(b) Histogram equalization.

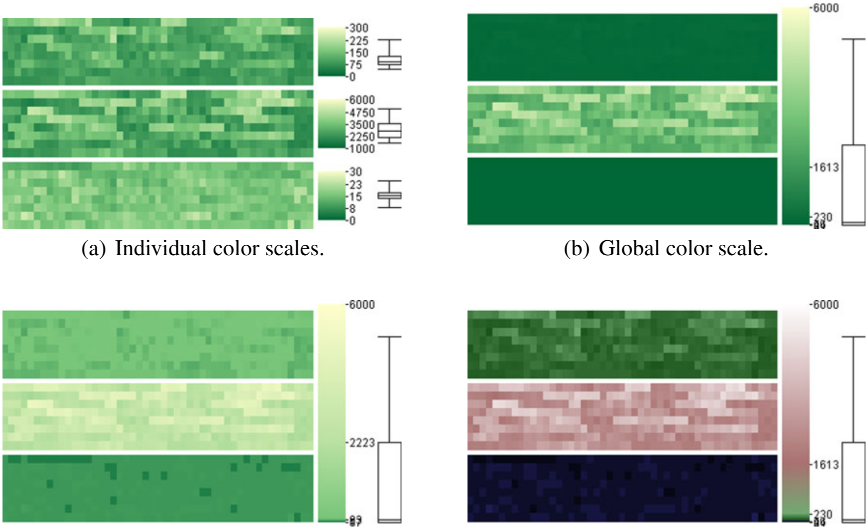


(c) Box-whisker equalization.

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

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

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



(a) Individual color scales.

(b) Global color scale.

(c) Box-whisker equalized color scale.

(d) Global comparison color scale.

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

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

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

4.2.3 Presentation Level

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

2D presentation of time-oriented data

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

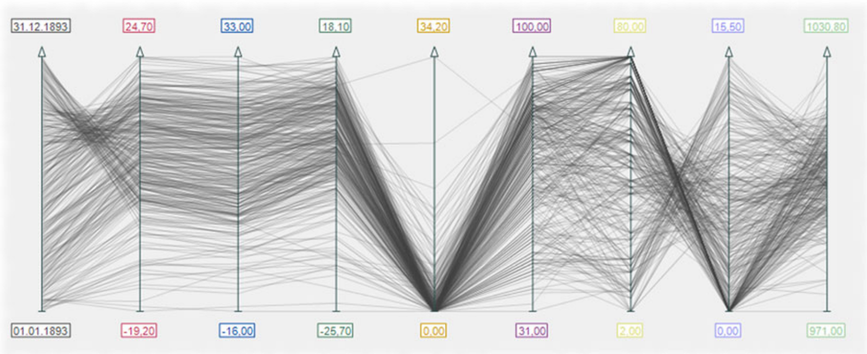


Fig. 4.18: In parallel coordinates, the time axis (here the leftmost axis) is just one of many axes and it is not treated in any particular way to emphasize the importance of time.

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

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

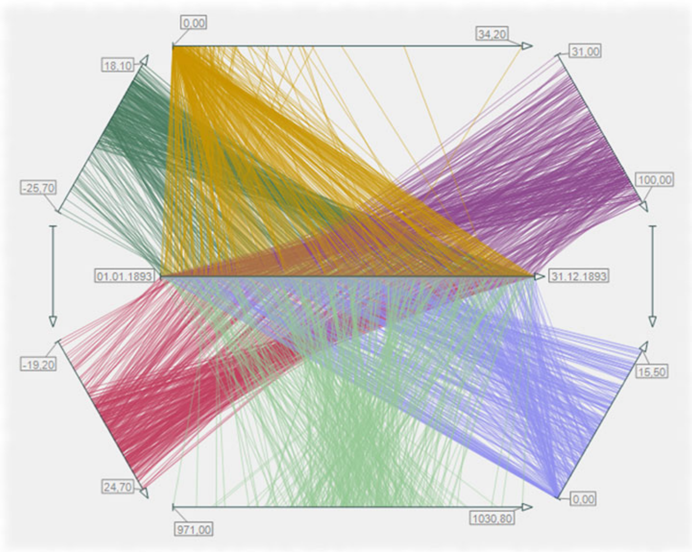


Fig. 4.19: The TimeWheel shows the reference time axis in a prominent central position and arranges data axes representing time-dependent variables around the time axis. Data are visualized by drawing lines between points at the time axis and values at data axes.

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

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

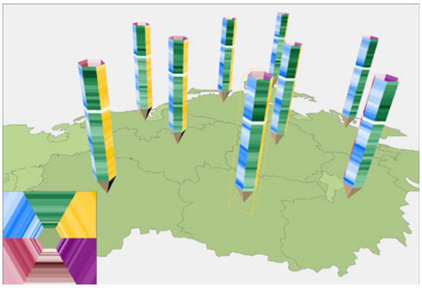
3D presentation of time-oriented data

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

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

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

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



(a) Pencil icons for linear time.

(b) Helix icons for cyclic time.

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

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

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

4.3 Summary

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

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

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

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

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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