Acknowledgements

* Publikationsrecht für Bilder
* Korrekturen Honorar

\[ \text{Vis Master}^2 \rightarrow \text{politischer} \]

→ Umbi/Siebka II: Acknowledgements
   → 2 Zeilen
• **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 are data where each temporal primitive is associated with at most one data value. Multivariate data are characterized by the fact that each temporal primitive holds multiple data values.

### 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 [9]. This applies accordingly to the development of visualization tools for time-oriented data. To specify the problem domain, so-called task models are widely used in the related field of human computer interaction [4].

A prominent example of such task models is the ConcurTaskTree (CTT) [9], describing a hierarchical composition of a goal with 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 manually established by a domain expert, and mostly for the purpose of driving automatic user interface generation [9]. First approaches have begun to use this notation for visualization purposes. For instance, [9] apply the CTT notation for structured tests and evaluation of interactive visualization techniques.

In the visualization domain, tasks are usually given only at a low level in form of informal verbal lists. An accepted low-level task description specifically addressing the temporal domain has been introduced by [4]. There, 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**: 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**: 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?”
4.1 Visualization Aspects

For example, the PlanningLines technique might be implemented on top of a time model in a point-based as well as an interval-based scope.

**Characteristics of the time axis** The time axis can be specified by the following criteria:

- **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 also “temporal distances” can 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 a temporal extent equal to zero. Thus, no information is given about the region between to 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 collection of temporal primitives, i.e., time proceeds from the past to the future. A cyclic time axis is composed of a finite set of recurring temporal primitives (e.g., the seasons of the year).

- **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 on time (as for instance required to structure eyewitness reports).

Besides these general aspects describing the time axis, different levels of granularity as well as three different time elements used to relate data to time have to be distinguished: instant vs. interval vs. span. An instant time primitive can be considered as a single moment in time without duration. An interval is a temporal primitive with an extent. Additionally, spans represent relative durations in time that are not absolutely fixed on the time axis.

**Characteristics of time oriented data** In addition to the time domain, the data have major impact on analytical and visual approaches. A great deal of different modeling approaches for data exist. As indicated, data that relate to time can be characterized by the following criteria:

- **Scale**: qualitative vs. quantitative: Quantitative data are based on a metric scale (discrete or continuous). Qualitative scales includes an unordered (nominal) or ordered (ordinal) set of data elements.

- **Frame of reference**: abstract vs. spatial: Abstract data have been collected in a non-spatial context, and thus abstract data are not per se connected to some spatial layout. Spatial data contain an inherent spatial layout, e.g. geographical positions.
### Table 3.1: Characteristics of Time.

<table>
<thead>
<tr>
<th>Design Aspect</th>
<th>Short Description</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>scale along which time values are given</td>
<td>ordinal, discrete, continuous</td>
</tr>
<tr>
<td>Scope</td>
<td>scope of the basic elements that resemble the structure of the time domain</td>
<td>point-based, interval-based</td>
</tr>
<tr>
<td>Arrangement</td>
<td>arrangement of the time domain</td>
<td>linear, cyclic</td>
</tr>
<tr>
<td>Viewpoint</td>
<td>viewpoints on time that are modeled</td>
<td>ordered (totally, partially), branching, multiple perspectives</td>
</tr>
</tbody>
</table>

### Table 3.2: Modeling Time-Oriented Data.

<table>
<thead>
<tr>
<th>Design Aspect</th>
<th>Short Description</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>scale of the data variables</td>
<td>quantitative, qualitative</td>
</tr>
<tr>
<td>Frame of Reference</td>
<td>frame of the data variables</td>
<td>abstract, spatial</td>
</tr>
<tr>
<td>Kind of Data</td>
<td>kind of data variables</td>
<td>event, states</td>
</tr>
<tr>
<td>Number of Variables</td>
<td>number of data variables considered</td>
<td>univariate, multivariate</td>
</tr>
<tr>
<td>Internal Time</td>
<td>temporal dimension inherent in the data model</td>
<td>temporal, non-temporal</td>
</tr>
<tr>
<td>External Time</td>
<td>temporal dimension extrinsic in the data model</td>
<td>static, dynamic</td>
</tr>
</tbody>
</table>
3.3 Discussion

In this section we structured and specified the characteristics of time and time-oriented data. We approached this in two ways: firstly, we characterized the 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 ways and their different aspects.

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

All these aspects need to be considered to visualize and analyze data variables over time. We mainly focused of 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].

Even defining all these aspects is cumbersome and time consuming, but ease visualization and analysis tasks, which are explained in the next sections.
3.2 Modeling Time-Oriented Data

versely, *external time* is considered to be extrinsic to the data model. The external time is necessary to describe how a dataset behaves in (external) time. Depending on the number of temporal elements in the internal respectively the external time, the following distinction of time related datasets can be made:

**Static Non-temporal Data (see Fig. 3.30(a))** If both, internal and external time, are each comprised of only one temporal element, the data are completely independent from time. A fact sheet containing data about the products offered by a company is an example of static non-temporal data. This kind of data is not addressed in this book.

**Static Temporal Data (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 which explicitly consider time a special data dimension address static temporal data (e.g. the TimeWheel [Tominski 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 which explicitly focus on dynamic non-temporal data. These techniques are mostly applied in monitoring scenarios, for instance to visualize process data [Matković et al., 2002]. However, since internal time and external time can usually be mapped from one to the other, some of the known visualization techniques for static temporal data can be applied for dynamic non-temporal data as well.

**Dynamic Temporal Data (see Fig. 3.30(d))** If both, internal and external time, are comprised 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 which contain measures depending on time (e.g. daily number of cases of influenza or daily average temperature), and which are updated every 24 hours with 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, statements on visualization techniques for dynamic temporal data are beyond the scope of this book.
Chapter 5
Survey of Visualization Techniques

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

The survey lists the techniques on a per-page basis. This allows for easy access when a quick reference to a particular technique is sought by the reader. Each page briefly describes the background, explains the main idea and concepts, and indicates application of a particular technique. The description is accompanied with a reference to the original publication or a list of references in the case that multiple publication propose or make use of the same approach. As this is a visualization book, a figure demonstrates the technique in use or the conceptual construction of the visual representation. Additionally, we provide a side-bar that categorizes the technique. To keep the categorization at a manageable level, we do not use the full-scale classification introduced in the previous chapters, but instead focus on three key criteria: data, time, and vis(ualization). For each key criterion, we introduce further sub-criteria and corresponding characteristics. The side-bar information follows this pattern:

- data
  - variables – univariate vs. multivariate
5 Survey of Visualization Techniques

- frame of reference – abstract vs. spatial
  
bullet time
  - arrangement – linear vs. cyclic
  - time primitive – instant vs. interval

bullet vis
  - mapping – static vs. dynamic
  - dimensionality – 2D vs. 3D

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

As univariate data are usually less complex, the survey will start with techniques for this class of time-oriented data in Section 5.1. Later in Section 5.2, techniques for multivariate data will be listed.
5.1 Visualization Techniques for Univariate Data

konstent references in the appendix

name
Multi-Scale Temporal Behavior

Fig. 5.9: Multi-Scale Temporal Behavior [Shimabukuro et al., 2004] - Comprises different levels of granularity and aggregation to explore patterns at different temporal levels. Example shows precipitation data in Brasil.

The Multi-Scale Temporal Behavior technique comprises different levels of granularity and aggregation to explore patterns at different temporal levels. Basis for the visualization is a matrix that is divided vertically into three regions for the three scale levels daily, monthly, and yearly (see Fig. 5.9). Each column of the matrix represents a year worth of data. Particularly, the topmost region contains cells for each month whereas each cell comprises of colored pixels for each day. In the mid region, the monthly cells are colored uniformly (one color for each cell in contrast to different colors for each pixel of a cell). Here, the color represents the aggregation of daily values to a single monthly value. In the bottom region the same principle is applied by aggregating the monthly values to a single yearly value represented by one cell. A significant and non-trivial problem in dealing with “real world” data sets are missing values. This issue is tackled by the authors in terms of preprocessing as well as visually.

References

DateLens

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

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

Time Annotation Glyph

Fig. 5.22: Time Annotation Glyph [Miksch and Kosara, 1999] — Uses the metaphor of bars that lie on "pillars". Four vertical lines on the base specify earliest and latest starting and ending times. Supported by these pillars lies a bar that is as long as the maximum duration. On top of the maximum duration bar, a bar that represents the minimum duration lies upon two diamonds for latest start and earliest end. Left: Single glyph and represented parameters (ESS = Earliest Starting Shift, LSS = Latest Starting Shift, EFS = Earliest Finishing Shift, LFS = Latest Finishing Shift, MinDu = Minimum Duration, MaxDu = Maximum Duration). Right: Usage in a tool to represent the temporal and hierarchical aspects of a medical treatment plan as well as the execution order of individual parts.

A visualization technique representing the same temporal parameters as SOPOs is the Time Annotation Glyph [Miksch and Kosara, 1999]. But in contrast to the geometric SOPO technique, the Time Annotation Glyph is timeline- and glyph-based (see Fig. 5.22). Specifically, it uses the simple metaphor of bars that lie on "pillars". Four vertical lines on the base specify earliest and latest starting and ending times. Supported by these pillars lies a bar that is as long as the maximum duration. On top of the maximum duration bar, a bar that represents the minimum duration lies upon two diamonds for latest start and earliest end. Furthermore, undefined parts are displayed in gray and different granularities are indicated by using zigzag lines. Because of this metaphor, a few simple time parameter constraints can be understood intuitively. For example that the minimum duration cannot be shorter than the interval between latest start and earliest end — if it was, the minimum duration bar would fall down between its supports. All parameters might be defined relative to a reference point that is also represented graphically. Together with SOPO View, the Time Annotation Glyph is applied to represent the time annotations of medical treatment plans within the AsbruView application [Miksch and Kosara, 1999].

References

SpiraClock

Fig. 5.26: In the SpiraClock the current time is represented using common clock hands. The spiral on the clock face is used to represent events that will happen in the future, where each cycle of the spiral (inward) is one more hour in the future.

The SpiraClock [Dragicevic and Hoot, 2002] is a technique that visualizes temporal entities by using a clock metaphor. The common setup of the visual representation consisting of a face and two hands makes it easy for users to read the current time. Moreover, the clock face shows a spiral at which temporal intervals can be visualized. Such intervals can be, for instance, a bus schedule or a personal agenda. The outermost spiral cycle shows events of the current hour. Intervals on the ensuing cycles mark events to happen in the hours to come. In this sense SpiraClock is unique, because it combines the representation of the current time with a preview of the near future. SpiraClock allows users to drag the clock handles to visit different points in time, and to drag the spiral to expand or shorten the view into the future. Recently, the SpiraClock has been coupled with the Google Calendar in a tool called HeliCal.

References

Maximum Analysis

Fig. 5.30: Maximum Analysis [Tominski et al., 2003] – The path of the maximum of a variable for each time step can be traced in space and time.

The Maximum Analysis technique [Tominski et al., 2003, Carlstein et al., 1978] is used to represent maximum values of a selected parameter spatially and temporally. It can be seen as a variation of the Space-Time Cube and is based on the "beads metaphor". For each location of interest, a cord of beads is depicted perpendicular on top of the map. For each time step, a bead is added to the cord. If the parameter value at a particular location is the maximum among all locations, a red bead is added, otherwise a white one. This way, the path of the maximum value can be traced in space and time. Furthermore, the maxima might be connected by lines in order to ease analysis. The technique has been used for example to represent and analyze the spread of diseases over space and time [Tominski et al., 2003].

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 at assist users in spotting highly active activities.

References

PostHistory

![Calendar and Contacts Panels](image)

**Fig. 5.46:** PostHistory [Viegas et al., 2004] – The calendar panel on the left shows the intensity of e-mail activity on a daily basis. The contacts panel on the right is used for displaying the names of people who sent messages to the user.

*PostHistory*’s [Viegas et al., 2004] goal is to uncover different patterns of e-mail activity visually – social networks, e-mail exchange rhythms, and the role of time in these patterns. These social patterns are derived from analyzing e-mail header information. Hence, not the content of messages, but tracked traffic is the basis for analysis of people’s e-mail conversations over time. *PostHistory* is user-centric and focuses on a single user’s direct interactions with other people through e-mail. Basically, the user interface visualizes a full year of e-mail activity and is divided into two main panels – a calendar panel on the left and a contacts panel on the right (see Fig. 5.46). The calendar panel shows the intensity of e-mail activity on a daily basis whereas a square represents a single day and each row of squares represents a week. The size of a square is determined by the quantity of e-mail received on that day and its color represents the average quality of messages ("directedness") – the brighter the color, the more directed the messages on that day. The contacts panel is used for displaying the names of people who sent messages to the user.

**References**

Gravi++

Fig. 5.56: Gravi++ [Hinum et al., 2005] — Patient icons in the middle of the display are positioned according to a spring-based model relative to the surrounding question icons. The representation might be stepped through time manually or animated via the time control panel on the lower left. Furthermore, traces might be displayed that convey information about the evolution of values over time as shown in detail on the lower right.

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

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 \textit{what and why}, and as a consequence represent tailored solutions in terms of \textit{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.