Chapter 3
Time & Time-Oriented Data

What, then, is time?
If no one asks me, I know what it is.
If I wish to explain it to him who asks, I do not know.

Saint Augustine (AD 354-430, The Confessions)

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

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

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

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

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

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

3.1 Modeling Time

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

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

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

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

3.1.1 Design Aspects

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

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

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

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

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

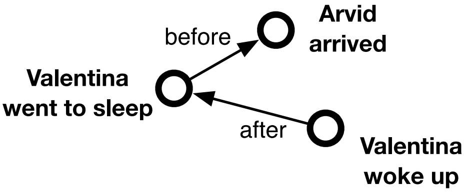

Fig. 3.1 Ordinal scale. Only relative order relations are present. At this level it is not possible to discern whether Valentina woke up before or after Arvid arrived.

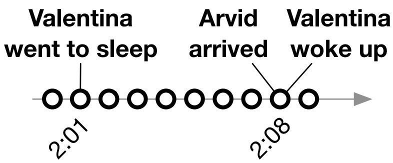

Fig. 3.2 Discrete scale. Smallest possible unit is minutes. Although Arvid arrived and Valentina woke up within the same minute, it is not possible to model the exact order of events.

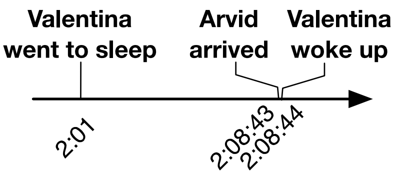

Fig. 3.3 Continuous scale. Between any two points in time, another point in time exists. Here, it is possible to model that Arvid arrived shortly before Valentina woke up.

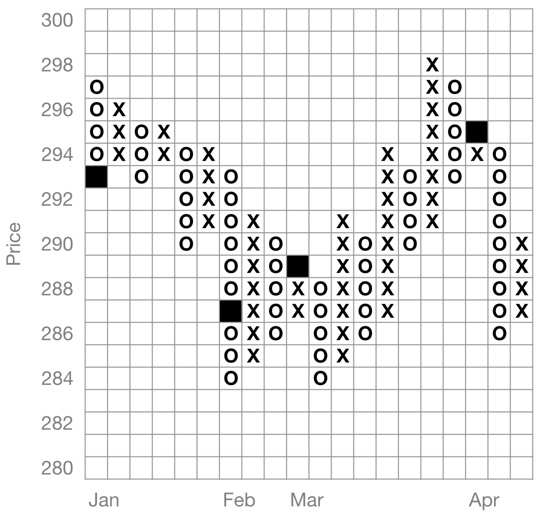

Fig. 3.4 Point and figure chart. Visualization technique tracking price and price direction changes. Uses an *ordinal time scale*. ○...positive price change of a certain amount, ×...negative price change of a certain amount, ■...begin/end of a trading period. *Source: Adapted from Harris (1999).*

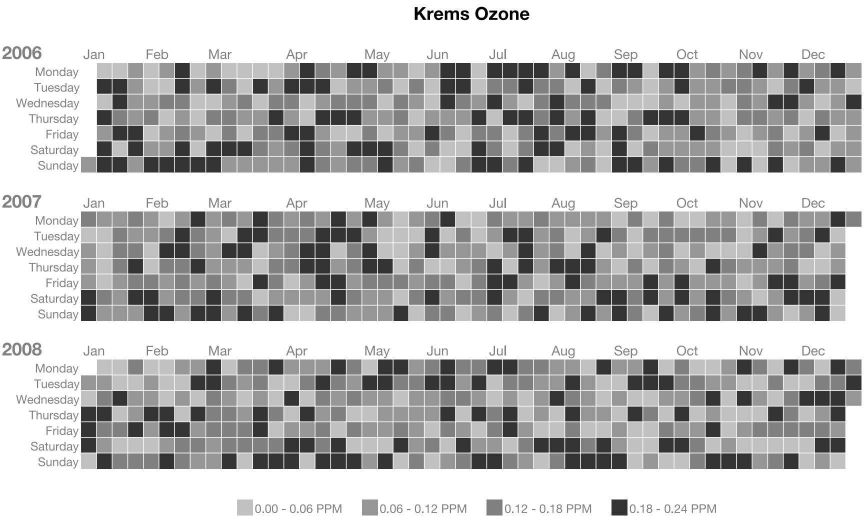

Fig. 3.5: Tile maps. Example shows average daily ozone measurements (scale: *discrete*, scope: *interval-based*) over the course of three years.
Source: Adapted from Mintz et al. (1997) with permission of David Mintz.

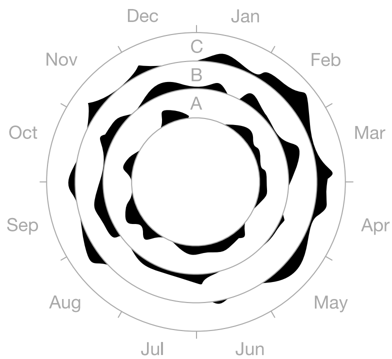

Fig. 3.6 Circular silhouette graph. Enables the representation of time along a *continuous scale* with a *cyclic arrangement*. The representation emphasizes the visual impression by filling the area below the plotted line in order to create a distinct silhouette. This eases comparison when placed side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

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

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

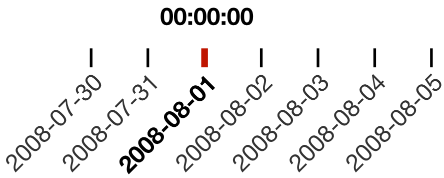

Fig. 3.7 Time value "August 1, 2008" in a point-based domain. No information is given in between two time points.

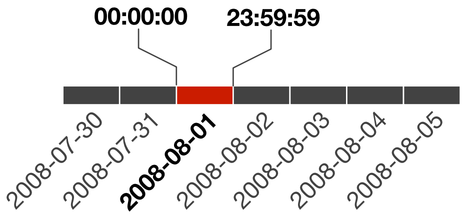

Fig. 3.8 Time value "August 1, 2008" in an interval-based domain. Each element covers a subsection of the time domain greater than zero.

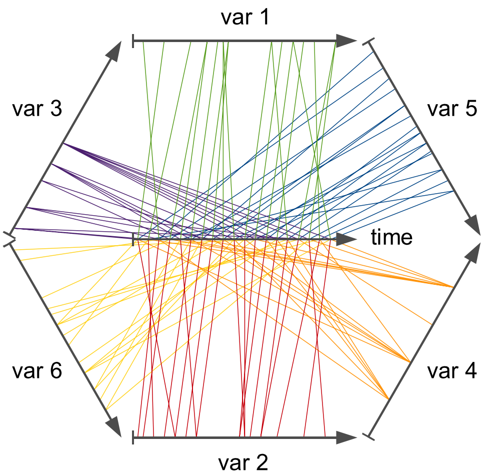

Fig. 3.9 TimeWheel. Parameter axes are arranged around the horizontal point-based time axis in a regular polygonal manner. For each time-step, lines descend from the time axis to the corresponding points on the parameter axes.

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

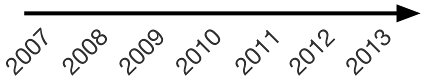

Fig. 3.10 Linear time. Time proceeds linearly from past to future.

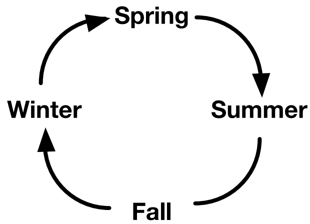

Fig. 3.11 Cyclic time. Set of recurring time values such as the seasons of the year.

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

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

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

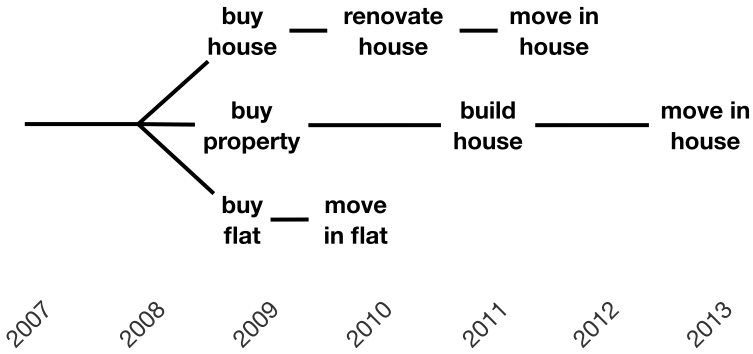

Fig. 3.12: Branching time. Alternative scenarios for moving into a new living space.

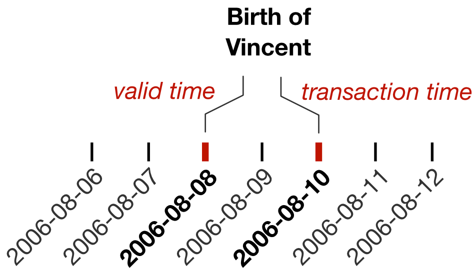

Fig. 3.13 Multiple perspectives. Vincent was born on August 8, 2006 (valid time) and this fact was stored in the register of residents two days later on August 10, 2006 (transaction time).

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

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

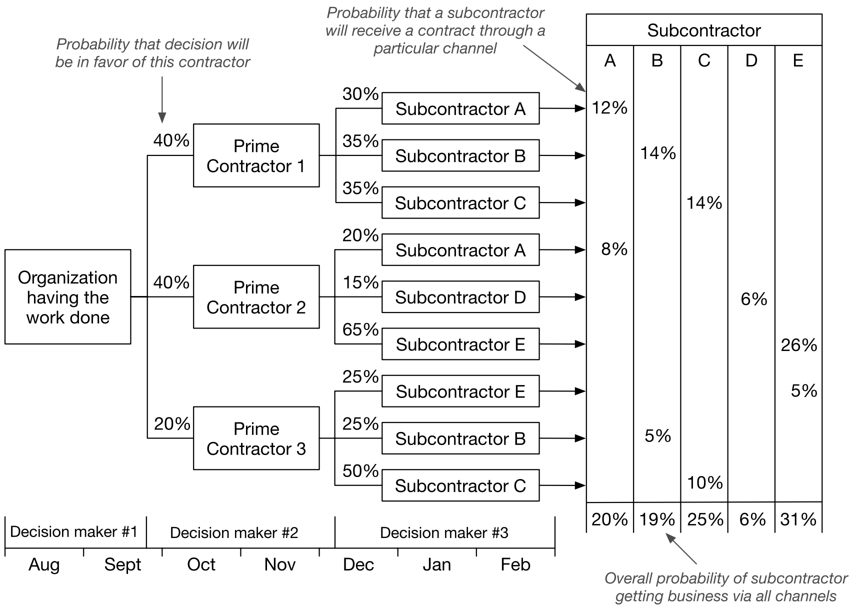

Fig. 3.14: Decision chart. Example of a visualization technique capable of representing branching time. Future decisions and potential alternative outcomes along with their probabilities can be depicted over time.
Source: Adapted from Harris (1999).

3.1.2 Granularities & Time Primitives

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

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

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

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

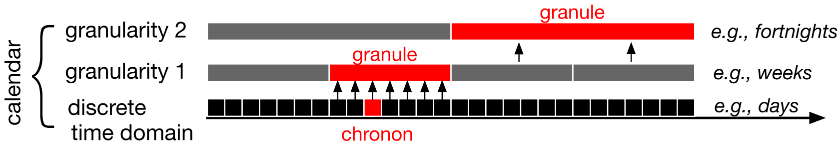

Fig. 3.15: Example of a discrete time domain with multiple granularities. The smallest possible unit (chronon) is one *day*. Based on this, the granularity *weeks* contains granules that are defined as being a continuous set of seven days. Moreover, the granularity *fortnights* consists of granules that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

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

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

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

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

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

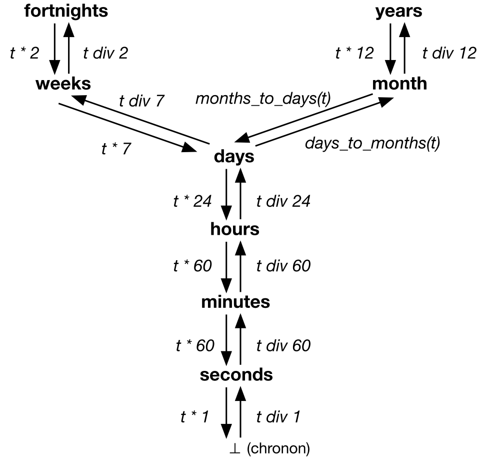

Fig. 3.16 Annotated granularity lattice of the Gregorian calendar that contains regular and irregular mappings (leap seconds are not considered in the granularity lattice).

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

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

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

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

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

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

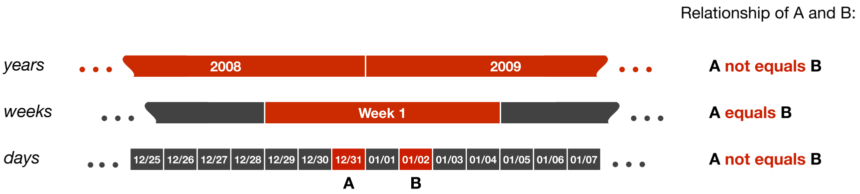

Fig. 3.17: Granularities influence equality relationships. The times of A and B are not equal on the granularity of days, but are equal on the granularity of weeks, and then again are not equal on the coarser granularity of years.

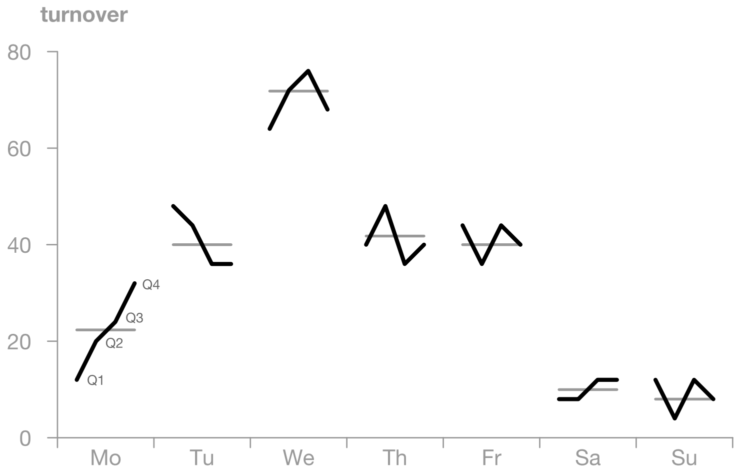

Fig. 3.18: Cycle plot. Visualization technique that utilizes two time granularities to represent cycles and trends. The example shows trends of measurements of weekdays over quarters. For example, on Mondays, the values show an increasing trend over the year while on Tuesdays the trend is decreasing.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

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

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

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

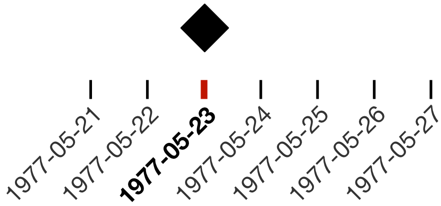

Fig. 3.19 Instant in a point-based time model. A point in time that has no duration.

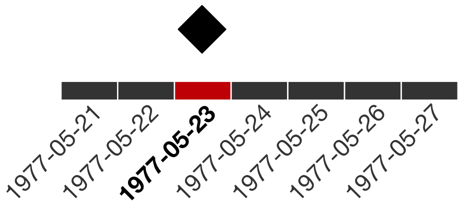

Fig. 3.20 Instant in an interval-based time model. A point in time that has a duration which depends on its granularity.

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

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

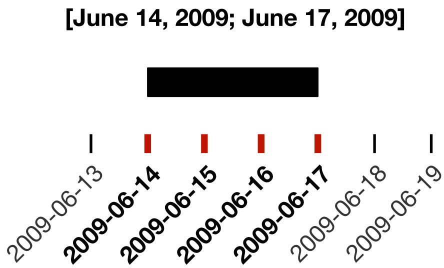

Fig. 3.21 Interval [June 14, 2009; June 17, 2009] in a point-based time model.

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

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

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

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

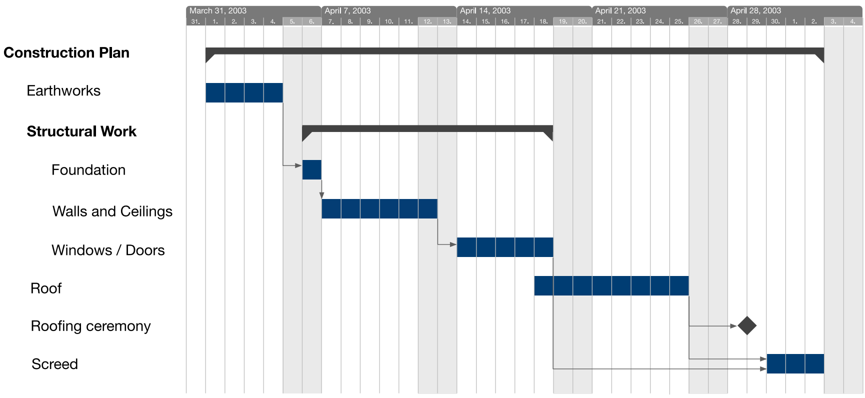

Fig. 3.23: Gantt chart. Example of a visualization technique capable of representing intervals. The tasks of a project plan are displayed as a list in the left part of the diagram. For each task, a horizontal bar (timeline) displays the extent of the task in time.

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

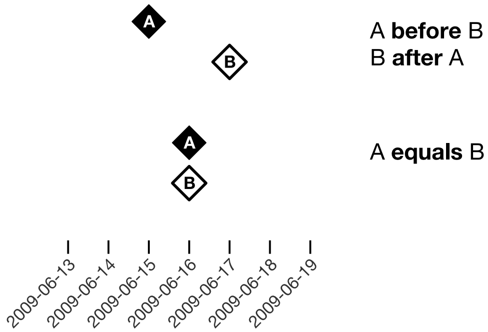

Fig. 3.24 Instant relations. Instants can be related in three different ways.

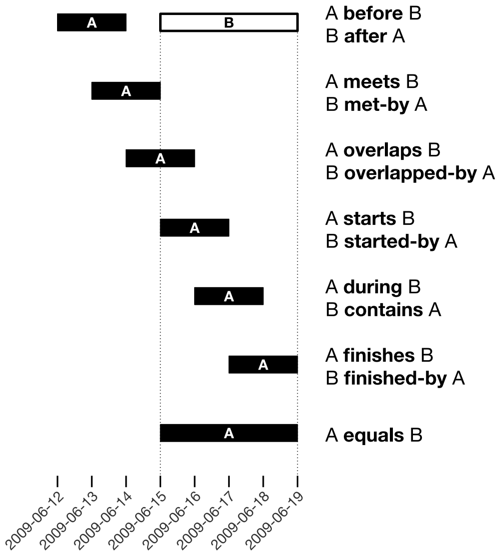

Fig. 3.25 Interval relations. Instants can be related in thirteen different ways.

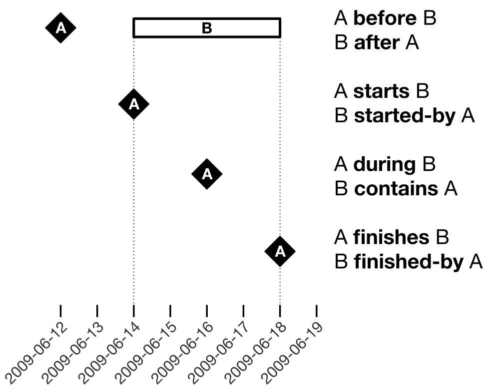

Fig. 3.26 Instant+interval relations. Instants and intervals can be related in eight different ways.

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

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

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

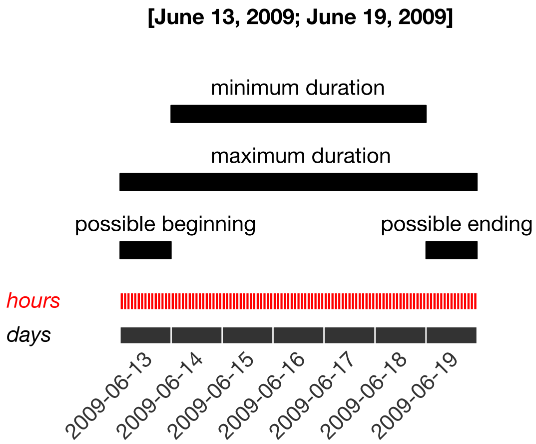

Fig. 3.27: Indeterminacy. Implicit indeterminacy when representing the interval [June 14, 2009; June 17, 2009] that is given at a granularity of *days* on a finer granularity of *hours*.

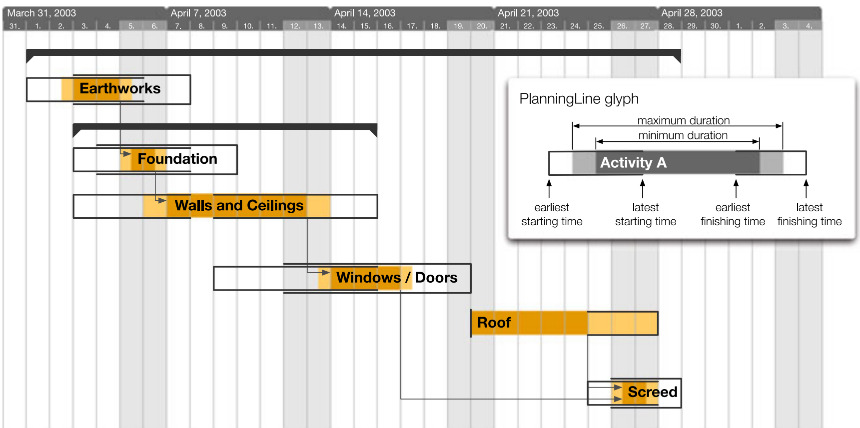

Fig. 3.28: PlanningLines allow the depiction of temporal indeterminacies via a glyph consisting of two encapsulated bars representing minimum and maximum duration, that are bounded by two caps that represent the start and end intervals.
Source: Adapted from Aigner et al. (2005).

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

3.2 Characterizing Data

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

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

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

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

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

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

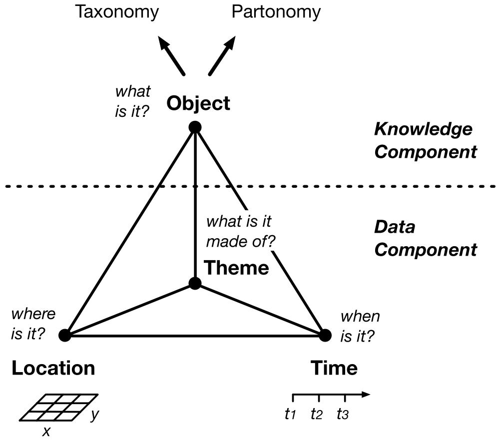

Fig. 3.29 Pyramid framework. Data are conceptualized along the three perspectives of location, time, and theme. Derived interpretations form objects on the cognitively higher level of knowledge. *Source: Adapted from Mennis et al. (2000).*

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

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

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

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

3.3 Relating Data & Time

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

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

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

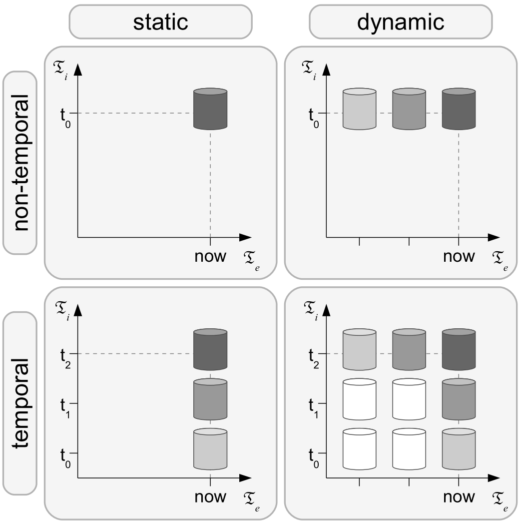

Fig. 3.30 Temporal characteristics of time related data. A dataset is related to the two temporal domains internal time $\mathfrak{T}_i$ and external time $\mathfrak{T}_e$. *Source: Adapted from Steiner (1998).*

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

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

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

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

3.4 Summary

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

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

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

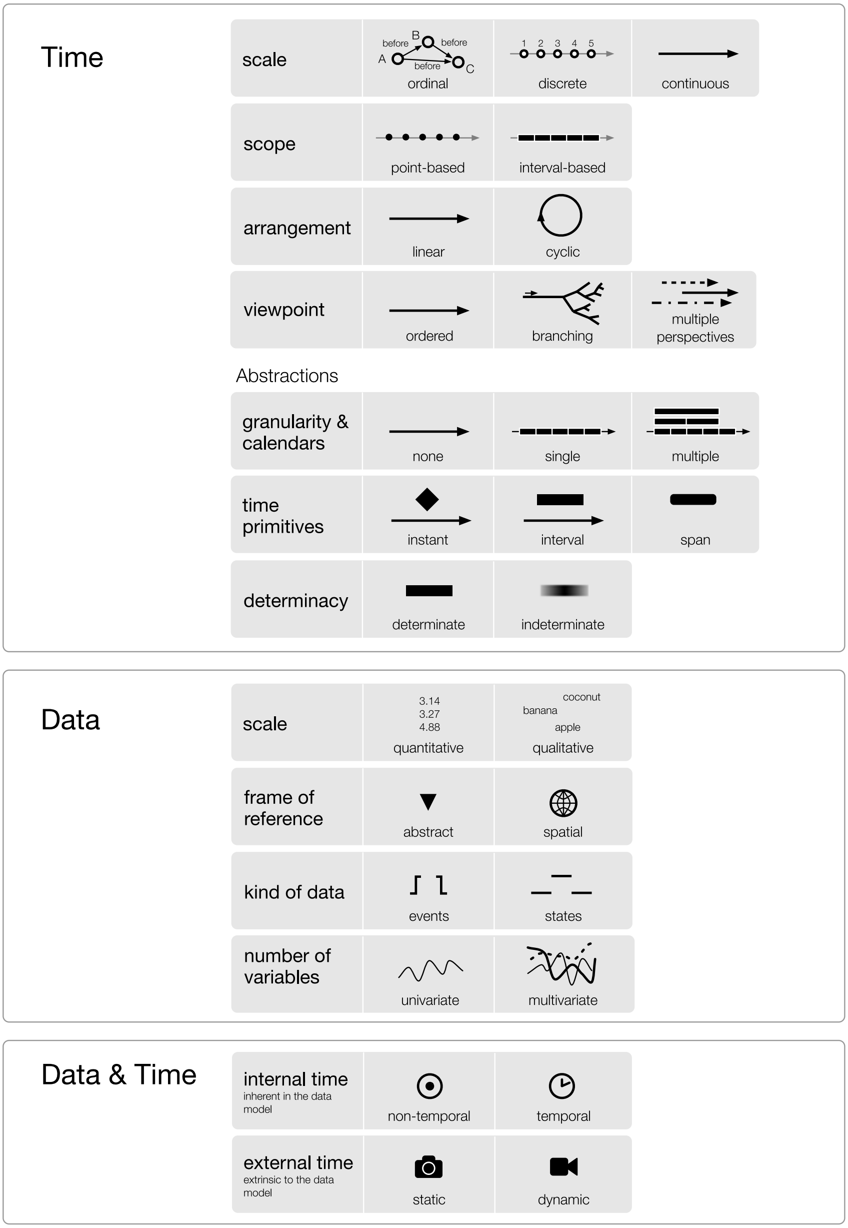

Fig. 3.31: Design aspects of time-oriented data.

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

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

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

References

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

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

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

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

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

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

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

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

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

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

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

Hochheiser, H. and Shneiderman, B. (2004). Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization*, 3(1):1–18.

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

Lee, J. Y., Elmasri, R., and Won, J. (1998). An Integrated Temporal Data Model Incorporating Time Series Concept. *Data and Knowledge Engineering*, 24(3):257–276.

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

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

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

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

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

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

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

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

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

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

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

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

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

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