

## 2 Time Interval Data Analysis

This chapter is structured as follows: Section 2.1 introduces terms and temporal aspects relevant to be considered when analyzing time interval data. In section 2.2, the different features required by an information system are discussed. The introduced terms, temporal aspects, and presented features are results of several workshops with users from different domains (e.g., service providers like ground-handlers, airlines, call centers, and hospitals, as well as linguists and production workers) and aligned with an extended literature research. The chapter is completed with a summary in section 2.3.

### 2.1 Time

When referring to time within the context of information systems and analytics, it is necessary to utilize a temporal framework. A temporal framework defines how time is represented (i.e., temporal models, section 2.1.3), how time can be used (i.e., temporal operators, section 2.1.4), and which semantic is applied (i.e., temporal concepts, section 2.1.5). In addition, constraints and limitations are implicitly defined within a temporal framework, i.e., circumstances which cannot be formalized are assumed to be invalid. In order to motivate a temporal framework in the context of time interval data analysis, section 2.1.1 introduces the term time interval informally (a formal definition is given in section 4.3) and in section 2.1.2 the aggregation of time intervals is presented, which is the predominant operator in the field of data analysis (cf. section 2.2.1 and section 7.3.4). Lastly, special characteristics of time like leap years, daylight saving, or time zones are discussed in section 2.1.6.

#### 2.1.1 Time Intervals

A time interval can be specified by two endpoints (e.g.,  $t_{\text{start}}$  and  $t_{\text{end}}$ , with  $t_{\text{start}} \leq t_{\text{end}}$ ). Generally, the interval's endpoints can be included or excluded, denoting the former by rounded and the latter by square brackets. As an

example, the denotation [10:00, 12:12) is used to specify all time points between 10:00 (included) and 12:12 (excluded). In real life, time intervals are used to express the validity of, e.g., an observation, a state, or of a more complex situation, over a period of time:

- The red apple with a weight of 250.00g was falling from the tree between 09:45:12 and 09:45:57.
- The accused was out on bail from the first of January 2015 until the fifth.
- The machine only produced 16 items between 09:00 and 12:28, even though it could have produced 25.
- The translator typed the word ‘treasure’ and looked up the word ‘Schatzinsel’ within two minutes.

Looking at these sentences reveals some peculiarities to be considered when working with time intervals. For example, it may be impossible to tell if the endpoints are in- or excluded or if they are absolute (e.g., 01/01/2015) or relative (e.g., "within two minutes"). In addition, the granularity used to express an endpoint may differ (e.g., 09:00 uses a minute granularity, whereas the granularity of 09:45:12 is seconds). Furthermore, the examples indicate that the provided information used to describe can vary (e.g., "red apple" as categorization vs. "16 items" as fact). Figure 2.1 illustrates a first example of a time interval and different types of associated information.

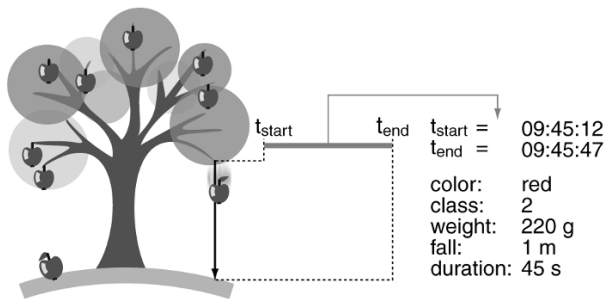


Figure 2.1 *Apple falling from tree*, example of a time interval and associated information observed, measured or calculated during the process of an apple falling from a tree.

The example shown in Figure 2.1 illustrates an observation which started at 09:45:12 and ended at 09:45:57 (i.e., a time interval of [09:45:12, 09:45:57]). During (or after) the observation the properties color, class, weight, fall, and duration were measured. Without providing a formal classification at this point (c.f. section 4.2), it is noticeable that properties may have to be handled differently from a semantical and analytical point of view. For example, the property color can be of interest when filtering, whereas the property class may be useful to determine a price, which can be important when aggregating. Other interesting properties are those which are not constants within the interval, e.g., the property fall is not constant. The presented value of 1.00 m is only valid for time points  $t \geq t_{\text{end}}$ . For time points  $t_{\text{start}} > t > t_{\text{end}}$ , the property's value can be calculated using the formula  $\text{fall} = \frac{1}{2} \cdot g \cdot (t - t_{\text{start}})^2$  and for  $t \leq t_{\text{start}}$  the value is 0.00 m.

Another example is shown in Figure 2.2. The example illustrates tasks (i.e., time intervals) performed by a machine. Such an example can typically be found in production environments.

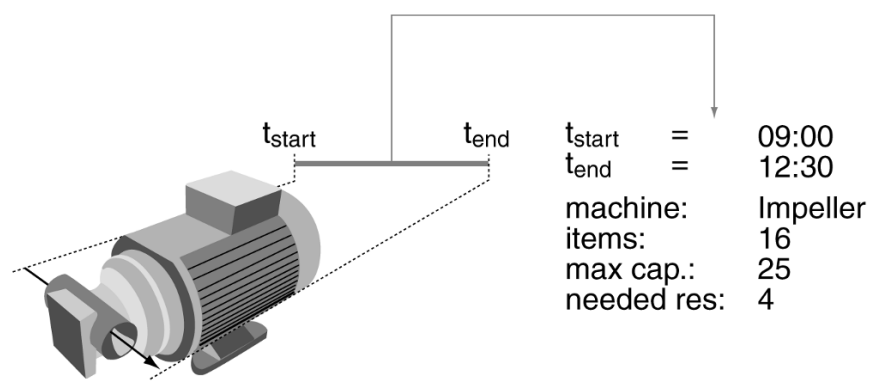


Figure 2.2 *Machine performance*, example of a time interval and associated information observed, measured, or calculated during the execution of a task by a machine.

The time interval of the *machine performance* example uses, compared to the previously discussed *apple falling from tree* example, a minute granularity for the time interval, i.e., [09:00, 12:30]. The example defines four

properties associated to the time interval: machine, items, maximal capacity, and needed resources. The items property is not constant (i.e., the value changes during the interval), whereby the maximum capacity property may be assumed to be constant (e.g., when filtering) or not (e.g., when used to calculate the utilization of the machine over time). In addition, the needed resources property is of special interest regarding aggregation. As introduced further in section 3.2.1 and discussed in more detail in section 7.3.4, this property can lead to summarizability problems if not aggregated correctly (Lenz, Shoshani 1997; Song et al. 2001; Mazón et al. 2008). The reason lies in the indivisibility of the value, i.e., the value is 4 for every time point of the interval but it is still 4 even if several time points of the interval are selected (i.e., summarizability is not given).

Within the next sections, the introduced examples are used to exemplify time interval data aggregation and are used to motivate the usage and exemplify the impact of temporal models, concepts and operators.

### *2.1.2 Time Interval Data Aggregation*

Data aggregation is the predominant operation in the field of data analysis (Zhang et al. 2008). Aggregating time interval data is more difficult than the aggregation of time point data. The reasons lie above all in the intricate semantic (cf. section 2.1.4), e.g., an interval expresses typically the validity of a fact or description over a period of time. When aggregating intervals within a specified time window several questions have to be answered, e.g., "Should the time window be partitioned" (e.g., using a time window of a year, it may be needed to aggregate data by day) or "What is the semantic meaning of the aggregation and does it fulfill the expectation" (e.g., is count a useful aggregation to determine the needed resources within a time window). In literature, different forms of temporal aggregations are introduced in the field of temporal databases and data analysis, i.e., Instant Temporal Aggregation (ITA), Moving-Window Temporal Aggregation (MWTa), Span

Temporal Aggregation (STA), General Temporal Aggregation (GTA) (Böhlen et al. 2008), as well as the Two-step Aggregation Technique (TAT) (Meisen et al. 2015b).

When aggregating time interval data, the set of intervals to be grouped is defined by the values of specified properties (e.g., the color of the apple in the *apple falling from tree* example (cf. Figure 2.1)) and, in addition, by a temporal grouping criterion (e.g., month, day or hour) used to partition the time axis. Depending on the form of temporal aggregation, the returned result of a query might contain so called constant intervals (ITA, MWTA, and GTA) or fixed partitions (STA, TAT, and GTA). A constant interval is an interval in which the aggregated value<sup>4</sup> is constant, i.e., consecutive time partitions are coalesced. Conversely, a fixed partition is defined by the specification of the aggregation (e.g., group by month) and the result contains a value for each partition (e.g., each month).

Figure 2.3 illustrates the ITA and MWTA forms, both returning constant intervals. In the figure, the intervals are grouped by the machine property, i.e., two groups are identified: furnace and machine. Furthermore, the time axis is on month granularity, and the example counts the amount of machines per month. As mentioned, ITA and MWTA both create constant intervals. Thus, in the case of ITA the results contains, e.g., the constant interval [3, 5] for the value 2. On the other hand, MWTA uses a defined time window  $[t - w, t - w']$  for each instance  $t$  of the defined temporal grouping and determines the set of intervals to be grouped. Thus, the example illustrated in Figure 2.3 calculates the aggregated values for the impeller group and the different time windows are, e.g.:  $\text{count}([1, 2]) = 1$ ,  $\text{count}([2, 3]) = 2$ ,  $\text{count}([3, 4]) = 2$ , ...,  $\text{count}([11, 12]) = 1$ , and  $\text{count}([12, 12]) = 0$ . The created constant values are shown in the table of the figure.

---

<sup>4</sup> Some implementations consider lineage information, i.e., the implementation validates if the resulting aggregated value is based on the same time intervals (cf. Böhlen et al. 2008).

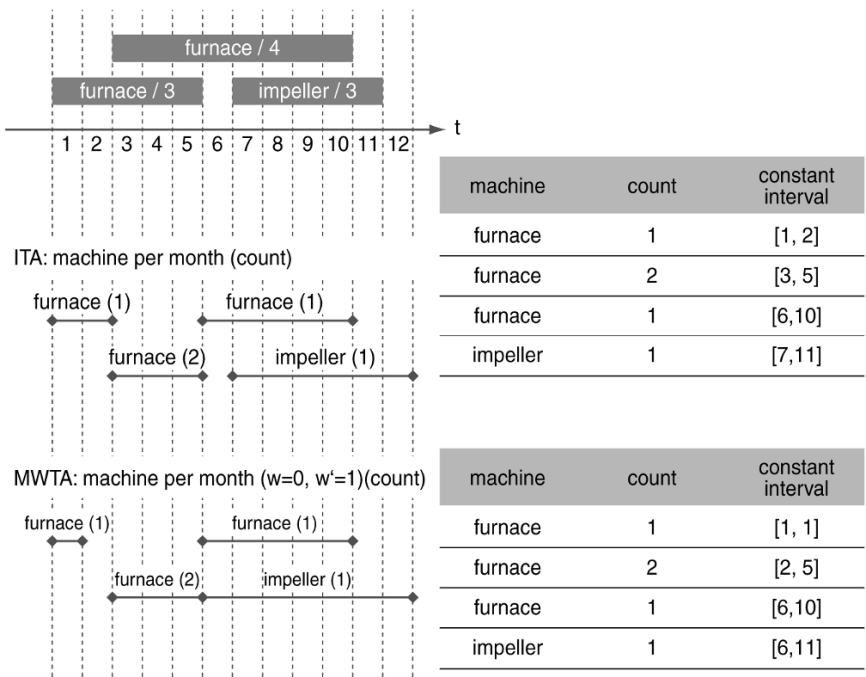


Figure 2.3 Example of ITA and MWTA (temporal aggregation forms creating constant intervals).

In general, ITA uses the defined temporal grouping criterion to determine the set of intervals for a specific group. On the other hand, MWTA uses a defined time window  $[t - w, t - w']$  for each instance  $t$  of the defined temporal grouping and determines the set of intervals to be grouped. Thus, using MWTA with  $w = 0$  and  $w' = 0$  leads to the same results as ITA provides. Empty groups are typically not included within the result (e.g., cf. Figure 2.3: (impeller; 0;  $[12, 12]$ ) and (furnace; 0;  $[12, 12]$ ) are not included; Snodgrass (1995), Böhlen et al. (2000)).

In contrast to ITA or MWTA, the application of STA or TAT leads to fixed partitions. Consequently, the result contains one aggregated value for each instance of the temporal grouping specified, if at least one time interval overlaps with the instance. It depends on the chosen implementation, if the

result contains empty groups or not. Meisen et al. (2015b) present a bit-map-based implementation for TAT which ensures that the result contains all empty groups. Regarding STA, empty groups are not included referring to Snodgrass (1995) and Böhlen et al. (2000). Figure 2.4 illustrates STA and TAT. As exemplified, STA determines the set of intervals for each instance within the specified temporal grouping criterion (i.e., instance [1, 6] overlaps with two intervals, whereas [7, 12] overlaps with three). The same result could be achieved using TAT with a count operator. Within the example shown in Figure 2.4, TAT applies the max-count operator. Thus, the aggregated value of count is determined for each instance of the lowest granularity of the underlying time axis (i.e., for each chronon, cf. section 2.1.3). Next, the results of each month are aggregated using the maximum operator (i.e., max). Therefore, the result for [7, 12] is 2 (i.e.,  $\max(\{2, 2, 2, 2, 2, 1\})$ ) instead of, compared to STA, 3.

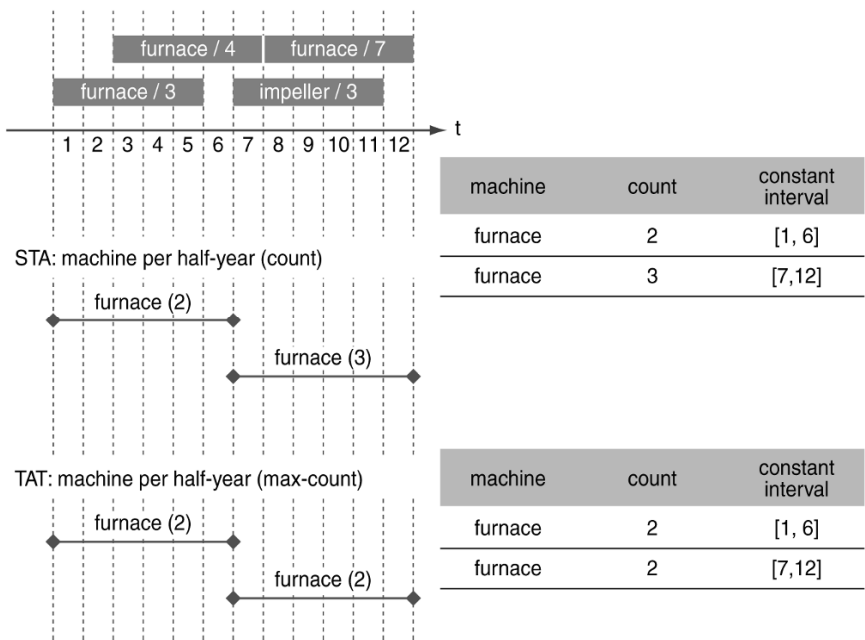


Figure 2.4 Example of STA and TAT (temporal aggregation forms creating constant intervals).

The earlier mentioned, but so far not further discussed, GTA is a generalized framework for temporal aggregation accommodating ITA, MWTA, and STA, as well as partly TAT. Generally, the framework allows specification of any kind of partition over the time axis. In addition, it is possible to define mapping functions in order to manipulate the instances of the partition. The framework covers TAT only partly because it only allows the definition of one aggregation function. Nevertheless, considering GTA, several challenges have not been solved. In addition, GTA is a theoretical definition which "offers a uniform way of expressing concisely the various forms of temporal aggregation" and "does not imply an efficient implementation" (Böhlen et al. 2008).

Temporal aggregations are discussed within this book several times: section 2.2.1 introduces features which are required regarding temporal aggregation, section 3.2.1 discusses the usage of temporal aggregators, as well as summarizability problems. Chapter 5 introduces a query language supporting the usage of temporal aggregations.

### *2.1.3 Temporal Models*

In literature about time, various temporal models have been proposed to represent physical time. Generally it can be stated that physical time can be modeled as discrete, dense, or continuous (Dyreson et al. 1994; Hudry 2004). In addition, literature introduces other aspects namely linear, branching, or circular temporal models, as well as bounded or unbounded temporal models (Frühwirth 1996). Within this section, the different aspects of a model will be introduced and discussed in matters of time interval data analysis. Also, the usage of a discrete, linear, bounded temporal model in the context of time interval data analyses is motivated. Figure 2.5 depicts the different temporal models which are introduced in detail in this section.



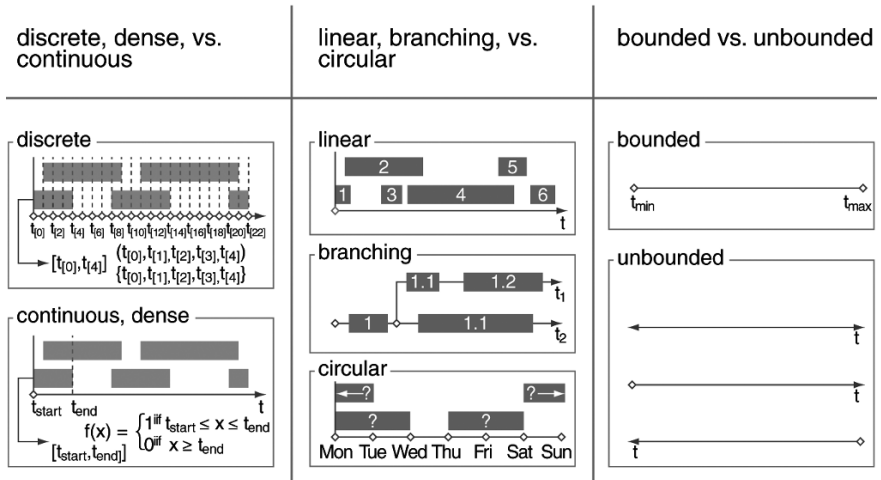


Figure 2.5 Overview of the different aspects of a temporal model.

### *Discrete, Dense, and Continuous Temporal Models*

A discrete time implies that a point in time can be represented by an integer (i.e., time is isomorphic to the natural numbers). If a dense or continuous temporal model is used, it infers that another time point between any two ‘unequal’ time points exists (i.e., time is isomorphic to the rational or real numbers)<sup>5</sup>. To understand the impact of the decision of which temporal model to use, it is necessary to understand the main differences between the different models considering the context of analyzing time interval data. Because of the isomorphic behavior of dense and continuous temporal models and the fields of application concerning dense temporal models (i.e., mainly model checking), the following discussion will discuss the usage of a discrete or continuous temporal model, whereby dense temporal models are – regarding the argumentation – ‘covered’ by the latter.

<sup>5</sup> As stated by Hudry (2004), a dense temporal model is isomorphic to rational numbers, whereby a continuous temporal model is isomorphic to the real numbers. In the context of analytics this differentiation is not important and is therefore not further mentioned.

To illustrate the differences between the temporal models, the *apple falling from tree* example (cf. Figure 2.1) is used. Applying a discrete temporal model to the example would let the apple ‘fall in steps’, i.e., at each discrete time point the apple would have a different falling distance, i.e., the fall property’s value would be different. The model would not clarify the apple’s position ‘in between’ two directly successive time points because in a discrete temporal model something between two directly following time points does not exist. Thus, within a discrete temporal model the falling distance of the apple would be specified for each discrete time point of the interval (e.g., at  $t_{\text{end}}$  the apple’s falling distance is 1.00 m). Furthermore, it would be possible to calculate an indivisible delta, which would be specified by the absolute value of the difference of the falling distance of two directly successive time points. Using a continuous temporal model, the falling distance would be specified for every moment  $t$  (using  $\frac{1}{2} \cdot g \cdot (t - t_{\text{start}})^2$ ). A delta between two time points can still be calculated but within such a model the delta is not indivisible. Figure 2.6 illustrates the falling distance in a discrete and continuous temporal model and shows the indivisible delta calculated for the discrete case (triangles).

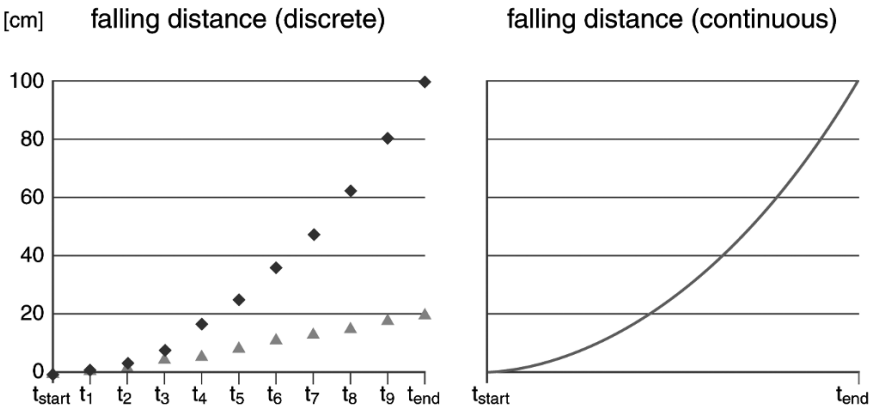


Figure 2.6 The *fall* property using a discrete (left) and continuous (right) temporal model. Within the discrete chart, the diamonds mark the value of the property and the triangles illustrate the indivisible delta between the previous and the current time point.

Regarding the *apple falling from tree* example, it may be intuitive to say that the information available when using the continuous temporal model is more precise. Nevertheless, looking at the *machine performance* example and the items property, this intuition may be different. Figure 2.7 shows the results recorded from an employee who checked the amount of created items every 15 minutes using both the discrete and the continuous temporal model.

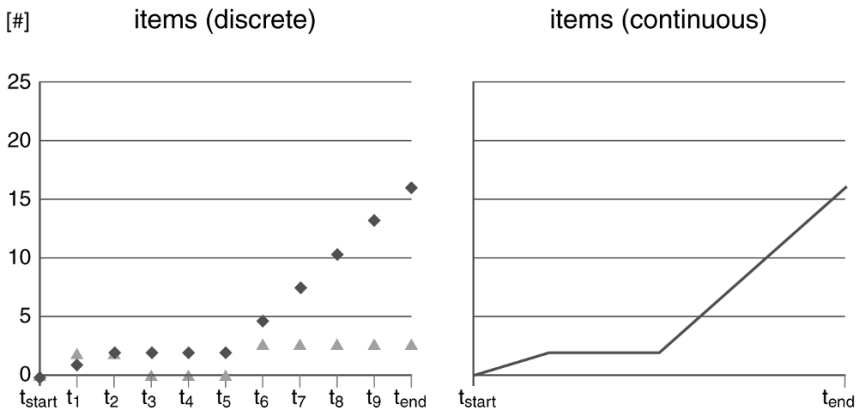


Figure 2.7: The item property using a discrete (left) and continuous (right) temporal model. Within the discrete chart, the diamonds mark the value of the item property and the triangles illustrate the indivisible delta between the previous and the current time point.

In this example, the information provided by the continuous model is too precise. Depending on the used function (e.g., if interpolation is used) it may even be invalid<sup>6</sup>. From an analytical point of view, one may argue that: ‘as long as the granularity of a discrete time-axis is selected correctly, the discrete temporal model is at least as good as the continuous one’. In addition, it has to be considered that data is typically collected by sensors (using a discrete sampling rate). Thus, the measured data is discrete and

<sup>6</sup> Figure 2.7 allows for the conclusion that the value of  $t = \frac{1}{2} \cdot (t_2 - t_1)$  is 0.5. Such an invalid value can be avoided by using a piecewise-defined continuous function. Nevertheless, from a domain-specific point of view, the correctness of the value is still not guaranteed because the employee did not check the amount at every time point.

the use of a continuous model is unnecessary. It should also be mentioned that a continuous property (e.g., a value based on a mathematical function) can be easily transformed into a discrete property using discretization techniques (Liu et al. 2002). Another aspect that should be considered when reaching a decision regarding a temporal model is the context. State of the art indicates that analyses dealing with temporal data are mostly based on discrete temporal models (cf. section 3.2).

As a result of these conclusions, the temporal model used within this book is discrete. Thus, the time axis consists of a finite number of chronons (i.e., "a nondecomposable [indivisible, remark of author] time interval of some fixed, minimal duration" (Dyreson et al. 1994, p. 55)).

### *Linear, Branching, and Circular Temporal Models*

Another aspect of temporal models addresses the future. Within a linear temporal model only one future is assumed, whereby a branching temporal model allows the existence of at least one but also multiple futures (paths). Moreover, a circular temporal model defines the future to be recurring. In the majority of cases regarding temporal data analysis, a linear temporal model is used. This is plausible because of the temporal concepts and operators mostly used within the field. If a branching or circular temporal model is utilized, simple concepts like before, or after may be difficult to be applied<sup>7</sup>. Thus, within this book a linear temporal model is assumed.

It should be mentioned that most data based on a circular temporal model can be pre-processed to fit a linear temporal model. If, e.g., data is retrieved from a simulation which is based on a circular temporal model it is necessary to 'roll out' the circular time, i.e., map time intervals of the circular time to time intervals of the linear time as indicated in Figure 2.8. The figure depicts a circular temporal model of a week and data generated in five iterations. The applied mapping links each circular week (i.e., each week of each iteration) to a week of the linear time.

---

<sup>7</sup> For discussions within other research areas the interested reader is referred to Alur, Henzinger (1992), Frühwirth (1996), Hudry (2004), and Ossimitz, Mrotzek (2008).

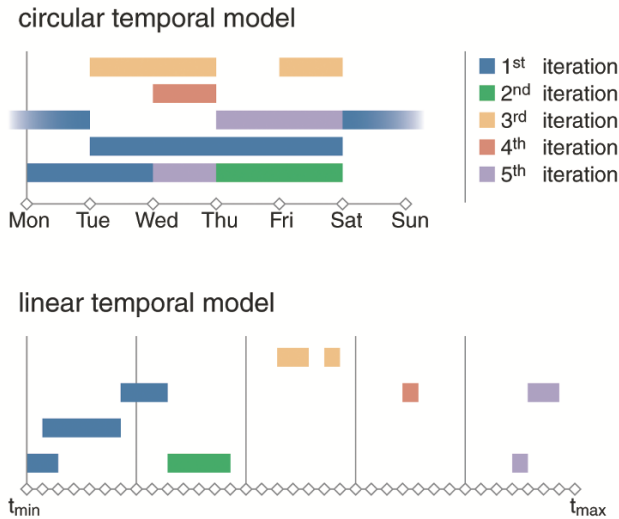


Figure 2.8: Example of a mapping between data of a circular temporal model to a linear temporal model.

### *Bounded and Unbounded Temporal Models*

The discussion about bounded or unbounded temporal models are, in the context of data analysis, more or less philosophical. A bounded temporal model is a model which has a defined start (i.e., a smallest time point) and a defined end (i.e., a greatest time point). Within an unbounded temporal model, infinitive time points are allowed, i.e., the interval  $[01.01.2015\ 09:00, \infty]$  is infinitive considering its end. If data from an unbounded temporal model should be analyzed it implies that there is no beginning or ending of time, i.e., there is always a time point earlier or later. Analyzing data within such a model would mean that unlimited data is available (i.e., defined by a discrete or continuous function); if not, limited data can be analyzed by the bounded temporal model by using the minimal and maximal time point of the limited data as boundaries. Nevertheless, unlimited data which is, e.g., defined by a recursively defined discrete function, could be analyzed within a time window which defines the boundaries used for the bounded temporal model (as illustrated in Figure 2.9).

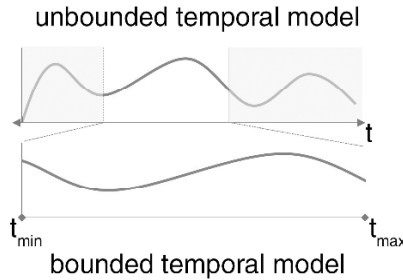


Figure 2.9: Selection of a time window from an unbounded temporal model to be presented and analyzable in a bounded temporal model.

Taking into consideration the above-mentioned findings, a bounded temporal model is used within this book.

### 2.1.4 Temporal Operators

A temporal operator for time intervals expresses the relation between typically, but not exclusively, two intervals. Within the last decades, several temporal operators were defined (cf. Moerchen (2009) for an extensive overview). In the majority of cases, the temporal operators of Allen (1983) are used. The primary reason for this is that the list of 13 defined operators is complete regarding possible combinations. Figure 2.10 depicts the defined operators.

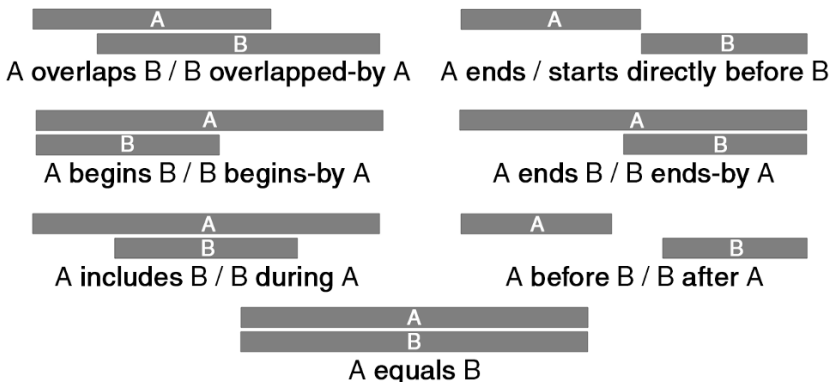


Figure 2.10: Overview of Allen's (1983) temporal operators.

Nevertheless, Moerchen (2009) states that Allen is not robust considering small changes and ambiguous regarding one's intuition. The first point can be ignored if exact boundaries are requested. However, the latter point mentioned refers to the problem that the size of overlaps or gaps is not taken into account using Allen's relations. Figure 2.11 illustrates the concerns mentioned by Moerchen. The relation between the intervals A and B are considered to be equal to C and D (both overlap). The same problem can be observed by looking at the relation between the intervals E and F and the relation between G and H which are both considered to be equal.

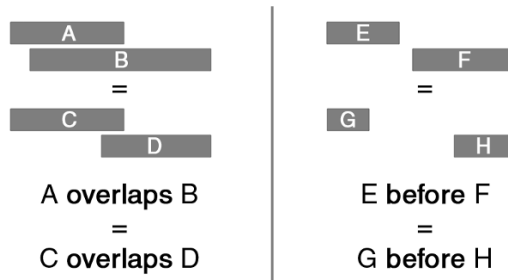


Figure 2.11: Illustration of the ambiguousness of Allen's (1983) temporal operators.

As already mentioned, several other temporal operators were published over the last decades. These other approaches mainly focus on overcoming the problems of Allen's definition regarding robustness and ambiguousness. Some try to achieve that by adding additional relations (e.g., Roddick, Mooney (2005) which define a total of 49 relations of which nine are different types of overlaps), others split intervals to generate partial relations (cf. Moerchen (2006a); Moerchen, Fradkin (2010); Peter, Höppner (2010)). Despite the doubts mentioned by Moerchen, this book uses the temporal operators of Allen, if not stated differently. If needed, additional precautions are introduced to overcome the mentioned problems (e.g., the distance-measure used to find similar time interval datasets introduced in section 6 utilizes the coverage ratio or spacing).

### 2.1.5 Temporal Concepts

Temporal concepts are used to define semantic categories for arrangements of temporal operators (Moerchen 2009). Several temporal concepts like past, present, or future, as well as order (i.e., before or after), duration, concurrency, coincidence, or synchronicity are commonly known and often used in natural language (cf. Moerchen (2006b), Kranjec, Chatterjee (2010)). Regarding the context of time interval data analysis and especially in the field of knowledge discovery (i.e., data mining) or even more specific in the field of temporal pattern mining, temporal concepts are often used to explain or classify patterns found within a time interval dataset. For example, the frequent occurrence of five periodically arranged time intervals may indicate an interesting observation. Nevertheless, searching for interesting and infrequent patterns may also be of interest, regarding coincidences or abnormal situations. A detailed discussion regarding temporal pattern mining as a part of time interval data analysis is provided in section 2.2.1 and 3.2.2. However, within this book commonly known temporal concepts, as exemplarily depicted in Figure 2.12, are used to express temporal arrangements of temporal operators.

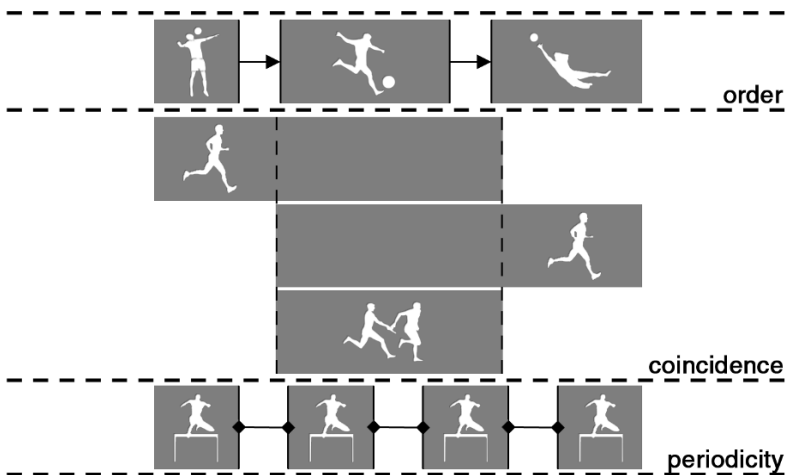


Figure 2.12: Examples of commonly used temporal concepts.



### 2.1.6 *Special Characteristics of Time*

In this section, several characteristics of time are introduced which have to be handled with special care with regards to time interval data analysis. Depending on the context of the analysis, some characteristics may be irrelevant. Thus, it is advisable to validate the impact of the characteristics within each analytical context. The introduced characteristics are: time zones, special days (like weekends, holidays, or vacation periods), leap seconds, leap years, absolute and relative time, as well as the general complexity of the time dimension.

#### *Time Zones and the Coordinated Universal Time (UTC)*

The world is divided in several time zones, each defined by the specification of an offset from the coordinated universal time (UTC). When analyzing temporal data the time zone information is of great importance to ensure the validity of the analytical results (cf. Kimball, Ross 2002, p. 240; Carmel 1999; Espinosa et al. 2007). Figure 2.13 illustrates an example which exemplifies the importance. The figure shows time interval data recorded within three time zones (i.e., UTC+1, UTC-8, and UTC-5). The example implies that data collected in the time zones UTC+1 and UTC-8 represent tasks performed at different airports. The interval shown within the UTC-5 time zone indicates an event having significant impact (e.g., 09/11, a stock market crash, or the moon landing). Analyzing the pictured scenario without taking the time zones into consideration is possible and valid, e.g., if the dataset of an airport is analyzed separately from the other. To compare the work-performance between the two airports (e.g., in the morning) it is necessary to analyze the time interval dataset using local times, ignoring any time zone information. If, on the other hand, the goal of the analysis aims to determine the impact of the event occurred within the UTC-5 time zone, it is necessary to perform the analysis using a normalized time (e.g., UTC).

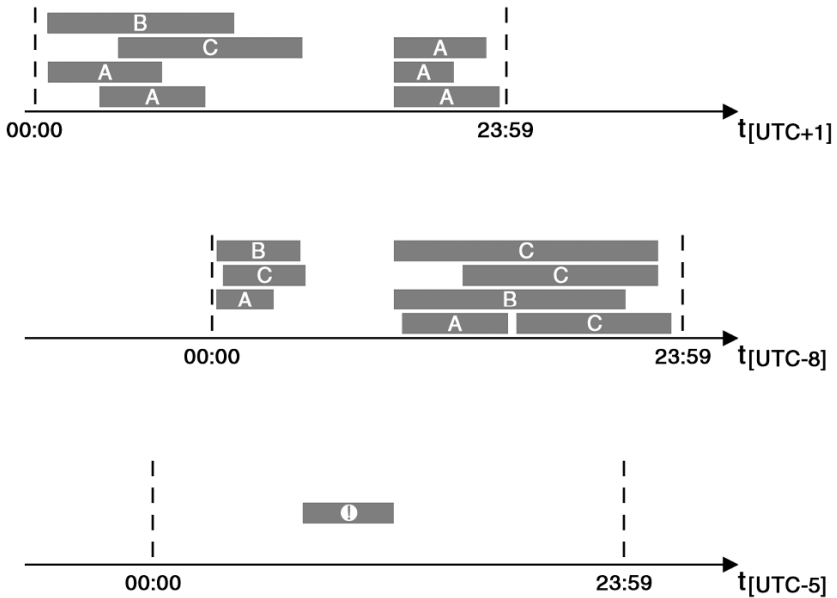


Figure 2.13: Example of the impact of different time zones within the scope of temporal analytics.

In order to meet the requirements, it is necessary for an information system and the underlying data model to understand the difference between normalized and local time, as well as the concept of time zones. The impact of time zones is addressed in section 4.1 (regarding the modeling of the time axis), 4.4 (with regard to different dimensional modeling), and 7.2.1 (concerning the implementation).

### *Daylight Saving Time (Summer Time)*

Changing the time during summer to increase the duration of daylight into the evening is a common practice in several countries. Nowadays, there are ongoing discussions if this practice is still meaningful and a minority of countries decided to abandon daylight saving time (DST). Nevertheless, from an analytical point of view DST is a difficulty which has to be considered and managed (cf. Celko 2006, pp. 26–27). The main issues while

dealing with temporal data and DST occur during two days a year (i.e., one when the time must be adjusted back one hour, the other one when it is forwarded). These days have 23 or 25 hours which makes it difficult to compare these days to any others. The problem can be exemplified when assuming a company utilizing an app to measure the employees' performed tasks during a day. Analyzing the average amount of performed tasks within an hour may lead to false results and therefore to erroneous decisions. Figure 2.14 illustrates the problem regarding DST and statistical values. Calculating the amount of time intervals between 03:00:00 and 04:00:00 results in 1 for the default (DEF), 2 for the forward (DST↑) and 0 for the backward case (DST↓).

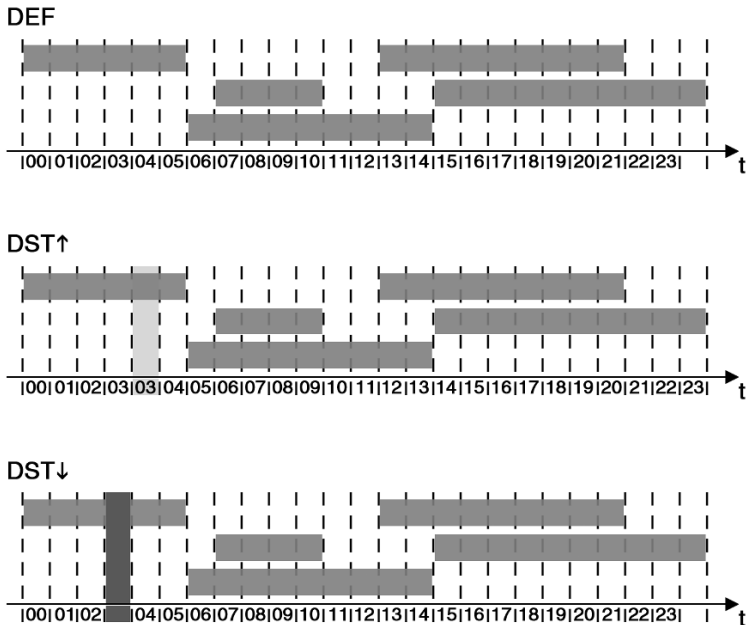


Figure 2.14: Illustration exemplifying the error of calculating statistical values, e.g., the amount of intervals per hour.

In general, several other statistical measures (depending on the context) may be affected by DST, e.g., in the context of work time management:

the daily performance, workload, or throughput. In addition, similarity measures (e.g., searching for similar days), which do not consider DST, may provide incorrect matches. A further discussion on how to analyze days with DST is presented in section 6 and 7.3.4.

### *Weekends, Holidays, Vacation Periods and Special Days*

Depending on the context of the analysis, weekends, holidays, vacation periods, and context specific special days may be of importance to understand specific observations, patterns, or anomalies. As already mentioned in the case of time zones, an event like a holiday or the beginning or ending of a vacation period can have a significant impact. For example, a travel agency's amount of customers, and therefore the amount and duration of consultations, may increase. Analyzing the workload without considering vacation periods may lead to invalid conclusions. Patterns searched across days, may differ meaningfully regarding holidays, weekends, and work days.

Supporting different types of days<sup>8</sup>, is an important feature when analyzing time interval data (cf. Kimball, Ross 2002, pp. 38–41). The need or importance of this additional information in the context of time interval data analysis may depend on the location the data is recorded at (e.g., a municipal holiday) and/or the goal of the analysis (e.g., 9/11 may be an important date considering cause studies, cf. Figure 2.13) . Some ideas on how to handle this additional information are discussed in chapter 9.

### *Leap Seconds*

Leap seconds are applied to the UTC to keep it close to the mean solar time. If not applied, the UTC would drift away (Whibberley et al. 2011). Thus, a leap second is inserted whenever the International Earth Rotation

---

<sup>8</sup> A not further discussed part of analyzes is the detection of special days within a specific domain by, e.g., using cluster or classification analysis. For further information, the reader may consider Grabbe et al. (2014), which applies clustering technique to find related days based on weather information, and Christie (2003), which uses classification techniques to identify outlying performances so called major event days.

and Reference Systems Service (IERS) decides to apply one. In the majority of cases, leap seconds are not relevant for analysis. However, Google states in their blog-post "Time, technology and leaping seconds" that "having accurate time is critical to everything we do at Google". Furthermore, Pascoe states that "keeping replicas of data up to date, correctly reporting the order of searches and clicks, and determining which data-affecting operation came last are all examples of why accurate time is crucial to our products and to our ability to keep your data safe" (Pascoe 2011). To achieve that, Google introduced the concept of leap smear. The idea behind a leap smear is to spread the additional (or shortened) second over a specific time window (e.g., the last minute before midnight), instead of waiting or shorting the last minute. It was mainly introduced, so that developers and engineers can rely on the system time without considering leap seconds at all. Within common operation systems and programming languages, leap seconds are not supported, i.e., the clock or internal counter does not display nor handle leap seconds. Instead, the second is added by counting the last second of the minute the leap second is scheduled for twice.

Summarized it can be stated that leap seconds may influence the results of temporal analytics. This may be the case if the selected granularity of time is in the range of seconds or less and the operation system handles leap seconds by counting the last second twice. If the concept of leap smear is applied or other specialized time protocols (e.g., Precision Time Protocol) are used, leap seconds should not lead to any problems. Nevertheless, statistical calculation may be off by up to one second. Within this book, the handling of leap seconds in association with the introduced information system is discussed in section 4.1.

### *Leap Years*

The Gregorian calendar differentiates between common years and leap years. The former has 365 days, whereas the latter has 366 days (adding

the 29<sup>th</sup> of February, namely the leap day). Depending on the level of aggregation used when analyzing temporal data, the existence of a leap day within a year may or may not invalidate the results. Thus, statistical measures aggregated on a year-level (e.g., sum or count) are not comparable between a leap year and a common year. A solution, to overcome this problem is the usage of relative values (e.g., mean or median) or a comparison on a valid level (e.g., by comparing sorted sets ignoring the additional day). In this book, the handling of leap years is discussed in section 6 and 7.3.4.

### *Absolute vs. Relative Time*

Time dependent data can be collected in an absolute or a relative manner. In general, an absolute time interval consists of two time points each specified by date, time, and time zone<sup>9</sup>. Contrary to this, a relative time interval consists of two time points each typically specified by an integer or a floating point number. Thus, relative time interval data is mostly found in scenarios in which the absolute time is irrelevant, e.g., when comparing time interval data collected from several process runs, each starting at a normalized moment in time, e.g., 0. Most researches within the field of data mining assume relative time interval data for their pattern mining algorithm. Nevertheless, in the context of on-line analytical processing (OLAP) and mining (OLAM), which both considers the existence of dimensions, absolute time interval datasets are mostly used. Thus, an information system has to be capable of handling relative and absolute time interval data (cf. section 4.1).

### *Complexity of Time Dimension*

The time dimension is an important and probably the most frequently used dimension within multidimensional models (cf. Kimball, Ross 2002, pp. 38–41). Considering OLAP and temporal data, aggregating data along the time

---

<sup>9</sup> The time zone information is often omitted because the system's local time zone is expected to be implicitly used.

dimension is one of the pre-dominant operations (Agarwal et al. 1996; Chaudhuri, Dayal 1997; Zhang et al. 2001), e.g., analyze the different months, detect anomalies, and understand the reasons for the anomalies by looking at the days of the months. In the field of temporal pattern mining, the different levels of the time dimension are often used to specify time dependent filters or ranges, e.g., detect frequent patterns occurring on Mondays. Using the time dimension in the context of analytics reveals several problems.

One of the problems to deal with is the fact that a calendar week does not neatly fit into a month nor a year. Thus, a time hierarchy like day → calendar week → month → year risks summarizability and comparison problems (Hutchison et al. 2006; Mansmann, Scholl 2006; Mazón et al. 2008). Solving, or at least revealing this problem to the querying user, is an important aspect to ensure correct usage of provided results. In section 3.2.1, several solutions on a conceptual or logical level are presented. In section 4.4, the modeling of the time dimension considering an information system for time interval data analysis is introduced and the handling of the mentioned problem is further discussed.

Another problem when dealing with the time dimension is the already mentioned variety of additional information attached to a member. A day may be, e.g., a global or municipal holiday, a memorial day, or a special event like tax day or 9/11 (cf. *Weekends, Holidays, Vacation Periods and Special Days*). Considering the time dimension, such additional information may be used to define special hierarchies (e.g., days may be rolled-up to a level containing members like none, municipal, national, and international holiday). Special time hierarchies are discussed in section 4.4.

## **2.2 Features of Time Interval Data Analysis Information System**

As noted in the introduction of this chapter, several workshops with analysts from different domains were organized addressing the issues occurring when analyzing time interval data. The first workshop "Business

Intelligence: How do you use your temporal data?" was held with 64 international companies (mainly aviation industry, logistics providers, and ground-handling service providers) during the "Inform Users Conference 2012". Additional workshops were organized during the following years aiming to reveal further insights, understand specific problems (e.g., occurring using proprietary software products), or to specify requirements (e.g., regarding the query language or special visualizations). The number of participants varied according to the purpose of the workshop and was distributed among a number of different sectors, i.e., aviation, logistic, ground-handling, call-center, hospitals, temporary employment, and linguistic. Altogether, more than 20 workshops, organized as expert discussions (i.e., between three or six experts from one or different companies), as business users workshop (i.e., up to 10 managers and experts were invited to discuss expected results), or as part of a users' conference (i.e., more than 20 experts), were held between 2012 and 2015.

The following sections present features derived from the results of the workshops and complemented by an extended literature research. The different features are categorized in analytical features (section 2.2.1), features defined along a time interval data analysis process (section 2.2.2), and features associated to the user interface (UI) of an information system for time interval data analysis (section 2.2.3). These features can also be understood as functional requirements. Nevertheless, non-functional requirements (e.g., regarding the performance or robustness) are not discussed in detail. Instead, relevant non-functional requirements are discussed and motivated implicitly within the different sections and used to motivate specific implementation strategies (i.e., authorization and authentication in section 5.1, indexing in section 7.3.2, and caching in section 7.3.3).

### *2.2.1 Analytical Capabilities*

In the field of data analysis, a distinction is made between different analytical techniques. In general, techniques are categorized in descriptive



("What has happened"), predictive ("What could happen"), and prescriptive ("What should happen") analytics (IBM Corporation 2013). During the workshops one of the goals was to determine which techniques must be supported and how the support may be realizable by specifying desired features. The results indicate that, regarding the analysis of time interval data, a demand for all three categories exists. Nevertheless, none of the categories is currently satisfactorily covered by any available information system and the importance differs between the three categories.

### *Descriptive Analytics*

The results of the workshops indicate that the need for descriptive analytics is very high. Experts stated that "understanding the current situation and past observations", as well as "being able to determine causes for anomalies" are important first tasks. The feature requests assigned to the type of descriptive analytics are listed in Table 2.1.

**Table 2.1:** Overview of the features requested in the category descriptive analytics.

| Identifier | Description   | Priority |
|------------|---|----------|
| DA-01      | As an analyst, I want to aggregate the time interval data along the time-axis, using different aggregation methods (must: SUM, COUNT, MAX, MIN, MEAN; should: MEDIAN; can: MODE). The aggregation must be correct considering summarizability.  | critical |
| DA-02      | As an analyst, I want to be able to use temporal aggregation methods along the time-axis (must: COUNT STARTED, COUNT FINISHED).   | high     |
| DA-03      | As an analyst, I want to be able to retrieve the raw time interval data within a specified time window (i.e., by using a query language). In addition, it should be possible to specify the temporal operator specifying the relation between the interval to be retrieved and the time window (e.g., retrieve all intervals equal to the specified time window). | high     |

|       |  |          |
|-------|--|----------|
| DA-04 | As an analyst, I want to roll-up and drill-down the time dimension. The levels of the different time hierarchies should support the definition of buckets for lower granularities (i.e., minutes and seconds).       | critical |
| DA-05 | As an analyst, I want to specify dimensions for the different properties associated to the time interval. Furthermore, I want to use these dimensions to generalize or specialize the result.                        | critical |
| DA-06 | As an analyst, I want to analyze data from different time zones. More specifically, I want to be able to analyze data from different time zones using local time zones, as well as a generalized time zone like UTC. | medium   |
| DA-07 | As an analyst, I want to be able to compare, e.g., hours, days, or weeks. In addition, I should be capable of searching for similar situations by selecting a template, e.g., hour, day, or week.                    | medium   |
| DA-08 | As an analyst, I want the system to provide a query language to retrieve analytical results (i.e., time series, mining results)  | critical |

Figure 2.15 exemplifies selected features, i.e., DA-01 (aggregate), DA-03 (select records), DA-04 (roll-up & drill-down time dimension), and DA-05 (roll-up to department & drill-down to work-area). The raw intervals (top left, DA-03) are aggregated applying count aggregation on the lowest granularity (top middle, DA-01). The roll-up and drill-down operations are applied (illustrations on the lower part of the figure, DA-04). The realization of these features is addressed in the context of modeling the time axis (cf. section 4.1) and dimensional modeling (cf. section 4.4). In addition, solutions for overcoming the summarizability problems occurring while realizing these features<sup>10</sup>, are presented in section 7.3.4.

<sup>10</sup> The problems occur when using available proprietary software (cf. Mazón et al. (2008)) or algorithms presented in the field of temporal databases (cf. section 2.1.2). Lately, several proprietary tools like icCube, Microsoft Analysis Services, or IBM Cognos presented features to support many-to-many relationship (cf. Russo, Ferrari (2011)). Nevertheless, as

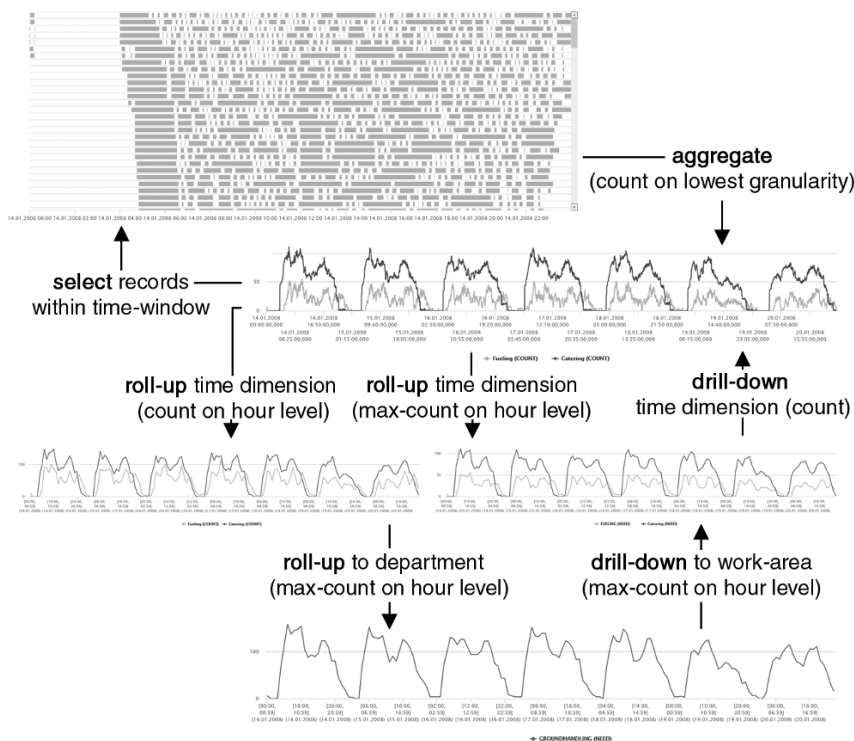


Figure 2.15: Overview of selected features defined in the category descriptive analytics in the context of time interval data analysis (cf. Table 2.1).

At this point, the features DA-02, DA-06, DA-07, and DA-08 are not presented in the figure. A detailed introduction for these features is given in the relevant section which introduces a concrete solution, several examples, as well as modeling, definition, and implementation aspects, i.e., section 7.3.4 (DA-02), section 4.4 (DA-06), chapter 6 (DA-07), and section 5.3.3 (DA-08).

discussed in section 3.2.1, these solutions cannot be applied satisfactorily in the context of time interval data.

### *Predictive Analytics*

In the case of prescriptive analytics the workshops have shown that the need is not rated as high as for descriptive analytics. One of the reasons stated by experts is the assumption that without appropriate descriptive analysis tools, features regarding predictive or prescriptive analysis are difficult to formulate. Another reason, indicated by experts, may be the availability of appropriate, proprietary software. For example, in the case of workforce management, several software products are available, e.g., useful to create rule-based rosters or simulate defined scenarios. The issues arising when using these tools are the definition of the rule-sets or the scenario's parameters. To formulate such a rule-set or determine the parameters, a better understanding of current and past situations is required which support the necessity of descriptive analytics. Nevertheless, some aspects of predictive analytics were classified as meaningful and are summarized in Table 2.2.

**Table 2.2:** Overview of the features requested in the category predictive analytics.

| Identifier | Description  | Priority |
|------------|--|----------|
| PD-01      | As a manager/supervisor, I want to be able to observe specified measures and be alerted if a defined threshold may be reached in the near future.  | medium   |
| PD-02      | As an analyst, I want to be able to find patterns or rules within a time interval dataset. Thus, it is necessary to specify the scope of the mining (e.g., just Mondays or holidays). In addition, it is of interest to validate if a pattern found within Mondays can also be found within other sets, e.g., Tuesdays, weekdays, or days of July. | low      |

*Prescriptive Analytics*

The aim of prescriptive analytics is to optimize upcoming situations by knowing what should ideally happen and rate different outcomes. The arguments mentioned in the case of predictive analytics apply, as well, in the case of prescriptive analytics. There are several tools used by data scientist enabling prescriptive analytics. However, the access to time interval data is quite difficult. Thus, an information system, as introduced in this book, is needed to provide an easy access and help for analyzing data in a descriptive way, prior to any prescriptive analysis. Regarding the results of the workshops, the requests expressed in the field of predictive analytics overlap with the once of prescriptive analytics. Table 2.3 shows a concise summary for the mostly openly formulated feature requests.

**Table 2.3:** Overview of the features requested in the category prescriptive analytics.

| Identifier | Description   | Priority |
|------------|---|----------|
| PR-01      | As a manager, I want the system to be able to predict upcoming situations (e.g., staff shortages) and provide solutions to the responsible dispatcher.                                      | low      |
| PR-02      | As an analyst, I want the system to be usable with other tools useful for prescriptive analytics (e.g., R <sup>11</sup> , Apache Spark <sup>12</sup> , or Watson Analytics <sup>13</sup> ). | low      |

*2.2.2 Time Interval Data Analysis Process*

Another purpose of the workshops was the determination of a generalized process for time interval data analysis, applicable to an information system.

<sup>11</sup> <http://www.oracle.com/technetwork/database/database-technologies/r>

<sup>12</sup> <https://spark.apache.org>

<sup>13</sup> <http://www.ibm.com/analytics/watson-analytics>

In general, the process of data analysis<sup>14</sup>, also known as data science process, is defined by several iterative phases (Schutt, O'Neil 2014, pp. 41–44). Figure 2.16 depicts the data science process.

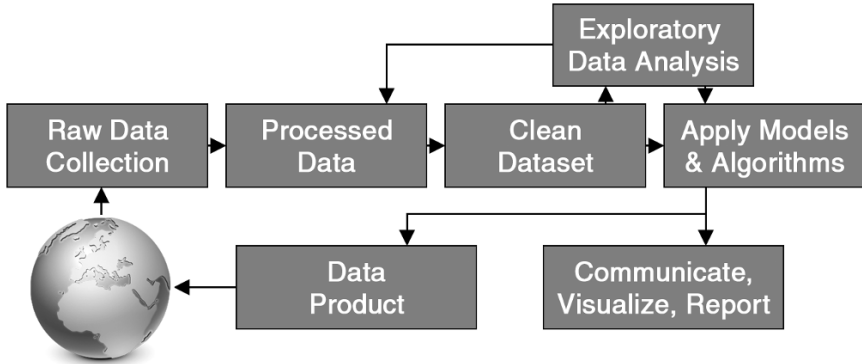


Figure 2.16: The data science process following Schutt, O'Neil (2014).

The process starts with the "Raw Data Collection" step, which is followed by the "Processed Data" step. Typically, data integration techniques are used by an analyst to process data in a way to create organized data ready for analysis. Nevertheless, the organized data may contain missing information, invalid entries, or duplicates. Thus, a clean dataset is derived during the second step by applying, e.g., data enrichment, outlier detection, or plausibility check techniques. In order to obtain a clean dataset or understand the data, it may be necessary to use exploratory data analysis (EDA) techniques, used to reveal further insights and clarify the validity. Having a clean dataset and understanding it, enables the analyst to detect, e.g., relationships, patterns, or causalities ("Apply Models & Algorithms"). Models may be generated and applied during this step to simplify the analysis. During the last steps, i.e., "Data Product" and "Communicate, Visualize, Report" the results created (e.g., a model, a rule, or a cause) and insights gained are used by a data product (i.e., an application) to create

<sup>14</sup> The process is comparable to the knowledge discovery in databases (KDD) process (Fayyad et al. 1996) or the more general visual analytics process (Keim 2010, pp. 10–11).

(automated) results (e.g., recommendations) or are presented to a decision maker.

The data science process aims to encapsulate the tasks performed by an analyst when analyzing any kind of data. Thus, it is applicable to time interval data analytics. Nevertheless, from an information system point of view the process is too generic and wide. Discussions during the different workshops have shown that from an analyst point of view several steps should be redefined or narrowed. In addition, it was pointed out that an information system may have to perform tasks automatically on each single time interval data record pushed into the system (c.f. feature request PD-01). Figure 2.17 illustrates the time interval data analysis process based on the results of the workshops. The figure differentiates between steps which should be supported by an information system (colored boxes) and steps performed by other systems, an analyst or a user (white boxes). Supporting describes the ability of the information system to perform the step automatically (e.g., based on configuration or modeling). In contrast to the data science process, the depicted time interval data process described the steps from an information system or data point of view instead of the perspective of an analyst. The analyst uses the information system to query, interact, or understand the time interval dataset and additionally configure and model the system (which is a cross-sectional task, and therefore not illustrated).

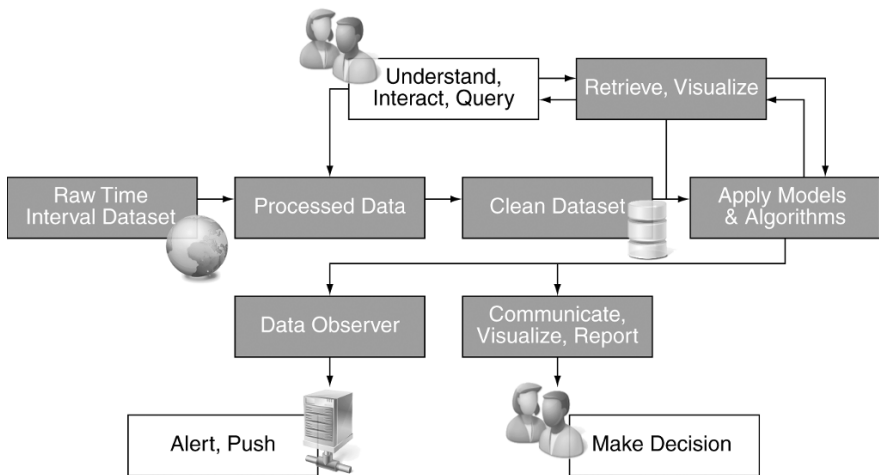


Figure 2.17: The result of the workshops regarding the time interval data analysis process.

The process starts with the collection of time interval data from an available and configured source. The collection might be a recurring (i.e., load the data whenever new data is available) or a one-off task (i.e., load data once into the system to analyze the set). The information system processes the incoming data using defined data integration techniques (step: "Processed Data"). Within the next step, the processed data is cleaned and a valid dataset is received (step: "Clean Dataset"). At this point, the analyst is capable to interact with the system, e.g. by firing queries or using a provided UI, useful to perform hypothesis testing, validation, or monitoring (step: "Retrieve, Visualize"). In addition, the analyst might retrieve and visualize results created by defined exploratory data analysis tasks, data mining algorithm, or machine learning concepts (step: "Apply Algorithms & Models"). Depending on the configuration of the information system, the defined algorithms and models are applied automatically used to determine if an alert has to be generated (step: "Data Observer") or report results to a decision maker (step: "Communicate, Visualize, Report").



In the following, the requested features for the steps: "Raw Time Interval Dataset" (Data Linkage & Collection), "Processed Data and Clean Dataset" (Data Integration & Cleansing), and "Apply Algorithms & Models" (Application of Models & Algorithms) are introduced and discussed. Features demanded in the context of visualization and interaction (i.e., steps "Retrieve, Visualize" and "Communicate, Visualize, Report") are presented in section 2.2.3. Requirements considering the "Data Observer" step are considered in section 2.2.1 (cf. Predictive and Prescriptive Analytics).

### *Data Linkage & Collection*

An information system for time interval data analysis has to provide interfaces enabling the loading of data into the system. During the first development phases and workshops several different ways on loading data into the system were discussed. Furthermore, scalability and data integrity were important topics when discussing the topic of data collection. Table 2.4 shows the subsumed features requested.

**Table 2.4:** List of requested features for the information system considering data collection.

| Identifier | Description   | Priority |
|------------|---|----------|
| DC-01      | As a system provider, I want the system to support different data sources, e.g. databases (i.e., relational DBMS), files (i.e., CSV or XML), and streams (i.e., JSON). If not supported, a simple application programming interface (API) must be available to enable me to add unsupported data sources. | High     |
| DC-02      | As an analyst, I want the provision of a Java Database Connectivity (JDBC) driver and a query language which allows the insertion and deletion of data. In addition, bulk loading operations should be supported.   | Critical |
| DC-03      | As a system provider, I want to be able to specify pre-aggregates to be calculated by the system, to increase query performance.  | High     |

Although the features requested are mostly self-explanatory, it should be mentioned that the realization of these feature is presented and discussed further in section 7.2.1 (DC-01), section 5.3.1 (DC-02), and section 7.3.4 (DC-03).

### *Data Integration & Cleansing*

Whenever data is loaded into the information system, it is important that the data is integrated and cleaned, so that invalid entries are detected, missing data is enriched, and the internally needed data structure is applied. The discussions considering data integration and cleansing was diversely, especially the question: "Which data integration techniques must be available by the system and at which point dedicated data integration tools should be applied as pre-processor". Table 2.5 shows the results of the discussions and additional feature requests defined within the workshops.

**Table 2.5:** List of requested features for the information system considering data integration & cleansing.

| Identifier | Description  | Priority |
|------------|--|----------|
| DI-01      | As an analyst, I want the system to be capable to handle complex data structures, in particular many-to-many relationships (cf. Kimball, Ross (2002), Mazón et al. (2008)).  | Critical |
| DI-02      | As an analyst, I want to be able to validate the descriptive values (properties) associated to the time interval. Validation must ensure that the value is not empty (i.e., mark a property as required), that the value is allowed to be used (i.e., by providing a white-list), or how a new value is handled (i.e., add it, use null, or fail). | High     |
| DI-03      | As an analyst, I want to be able to define how undefined intervals (i.e., intervals which have no start, end, or neither defined) are handled. Typically, I should be able to pick one of the following strategies: use time axis boundaries, use the  | High     |

|       |  |        |
|-------|--|--------|
|       | (other) specified value (i.e., create a time point), or fail.  |        |
| DI-04 | As an analyst, I want to be able to write scripts applied to the raw data prior to any processing or cleansing. Thus, I am able to manipulate the incoming data without pre-processing it using integration tools. | Medium |

The feature requests DI-02 and DI-03 are defined to cover important and often, in the context of time interval data analysis, applied strategies. The specified strategies are used to ensure data quality (by plausibility checks) or to offer the possibility to enrich missing values. DI-04 is requested as a last resort, i.e., the information system should offer a scripting interface useful to implement integration or cleansing techniques. This interface enables an analyst to apply techniques prior to using additional data integration tools. In addition, the interface might even be used to trigger a more complex integration process defined with a proprietary integration tool (cf. Meisen et al. (2012)).

The requirement formulated with feature request DI-01 addresses the already mentioned summarizability problem, which occurs when using many-to-many relationships and is introduced in detail in section 3.2.1. Regarding the used model introduced in chapter 4, the feature request DI-02 is partly covered by so called mapping functions (cf. section 4.1 and 4.2). In addition, the final implementation provides additional strategies to fulfill the request (cf. section 7.2.1).

### *Application of Models & Algorithms*

The requested capabilities of the information system considering descriptive, predictive and prescriptive analytics are listed in section 2.2.1. In addition, this section specifies architectural requirements to be met by the system to support these analytical capabilities. The features requested are listed in Table 2.6 and the implementation is introduced in section 7.1.

**Table 2.6:** The features required to support the application of models and analytical algorithms.

| Identifier | Description  | Priority |
|------------|--|----------|
| MA-01      | As an analyst, I want to be able to apply models or algorithms to the data stream, i.e., I want to determine problems, generate alerts, report anomalies, or classify the current data.            | Medium   |
| MA-02      | As an analyst, I want to be able to schedule analysis (e.g., daily) using the currently available data. Depending on the result of the analysis I want to trigger an action (e.g., send an email). | Medium   |

### 2.2.3 User Interface, Visualization, and User Interactions

An important criterion regarding the user acceptance of a system is its interface. The UI may be graphical (e.g., showing a graph) or a query language. In general, the user needs capabilities to interact with the system, so that a request can be specified or an alert be understood. Table 2.7 shows the features relevant for the information system. Features dealing with specific visualization<sup>15</sup> are not listed, because the development of specific visualizations are not in the scope of this book. Nevertheless, the interested reader is referred to section 3.2.3., which introduces current state of the art visualizations regarding time interval data and time series. Ideas considering the usage of visual analytics techniques in the context of time interval data analysis are discussed in section 7.4.

**Table 2.7:** Overview of the features requested for the UI, visualization, and user interaction.

| Identifier | Description  | Priority |
|------------|--|----------|
| VIS-01     | As an analyst, I want to be able to retrieve data from the information system using a JDBC driver to visualize the results, e.g., using a third party business intelligence tool, a visualization, | High     |

<sup>15</sup> E.g., a specific request for a line chart was to show the involved time intervals in a tool tip when hovering the value.

|        |   |          |
|--------|---|----------|
|        | or another analytical framework. Thus, I implicitly request a query language useful to retrieve data as needed.   |          |
| VIS-02 | As an analyst, I want to be able to subscribe to the system's alerts and analytical results. The system must publish the requested information to any subscribed instance.  | Medium   |
| VIS-03 | As a system provider, I want to have a UI for user management (i.e., delete or add users, define roles, grant or revoke a permission).                                      | Critical |
| VIS-04 | As an analyst, I want to have a minimal graphical user interface (GUI) useful to request and visualize results (e.g., a time series, resulting datasets, or a Gantt-chart). | High     |
| VIS-05 | As a web-developer, I want the system to provide web-friendly services, i.e., requesting and receiving data through a JSON interface.                                       | High     |

## 2.3 Summary

Within this chapter, several important terms within the context of time interval data analysis were introduced. In addition, features related to an information system supporting analytical tasks were presented. These features are motivated along temporal aspects and characteristics of time (e.g., temporal models, leap years, or time zones), as well as subsumed results from several workshops and an extended literature research. Furthermore, some subordinate features mentioned during the workshops, like specific requirements regarding specific statements of the query language, are not listed. Nevertheless, these feature requests are stated within the different upcoming chapters, if relevant.

This chapter also provides the answer to the first RQ: "Which features must be supported by an information system to enable time interval data analysis". An information system has to support the time characteristics, as well as provide the specified features in a performant way. An evaluation

regarding the fulfillment of the features is presented in section 8.1. In addition, these features provide the basis for the other research questions. A model for time interval data analysis (as mentioned in RQ2) is needed as formal framework for such an information system. The need for a query language (as addressed by RQ3) is explicitly or implicitly mentioned in several features (e.g., DA-01, DA-02, DA-03, DA-08, PR-02, DC-02, or VIS-01). The performance of an analytical information system is, even if not explicitly mentioned, of importance and the core issue of RQ4. The similarity among difference sets of time interval data is requested by feature DA-07 and topic of RQ5. The architecture and configuration of an information system are aspects to consider when realizing such a system. In addition, the needed interfaces (e.g., JDBC, JSON, or visualization) of time interval data and results of analyses are addressed by, e.g., DC-01, DC-02, VIS-01, VIS-04, and VIS-05. The RQ6 subsumes the mentioned aspects regarding the architecture, configuration, and interfaces.

Analyzing Time Interval Data

Introducing an Information System for Time Interval  
Data Analysis

Meisen, P.

2016, XXXI, 232 p. 65 illus., 8 illus. in color., Hardcover

ISBN: 978-3-658-15727-2