International Workshop on Visual Analytics (2012)
K. Matkovic and G. Santucci (Editors)

Developing an Extended Task Framework for Exploratory Data Analysis Along the Structure of Time

T. Lammarsch, A. Rind, W. Aigner, and S. Miksch

Institute of Software Technology and Interactive Systems (ISIS), Vienna University of Technology, Austria

Abstract

Exploratory data analysis of time-oriented data is an important goal that Visual Analytics has to tackle. When users from real-world domains are asked about time-oriented tasks, they often refer to the unique structure of time (e.g., calendars, primitives, etc.). Several task frameworks have been developed, but none of them combines a complete, systematic approach with explicit attention to the structure of time. To fill this gap, we aim for complementing an established task framework with a rule set that explicitly models the structure of time for tasks. This rule set allows to consistently formulate tasks for evaluating time-oriented data analysis methods.

Categories and Subject Descriptors (according to ACM CCS): Information Systems [H.1.1]: Models and Principles—Systems and Information Theory; Computing Methodologies [I.m]: Miscellaneous—

1. Introduction

Human judgement plays a fundamental role in Visual Analytics (VA) and is primarily mediated through interactive visual interfaces [TC05]. Therefore, it is necessary to take into account the users and be aware of their goals and mental models. For exploratory data analysis (EDA) of time-oriented data, they usually consider the structure of time, for example the aspect of calendric systems (see Section 3). Smuc et al. [SML*09] present detailed examples resulting from an insight study:

> "Starting in the morning, it rises to a peak around 10 or 11 a.m. It then calms down by noon, but there is a second peak around 4 or 5 p.m., after which it decreases again."
> "The first Monday is high, the second is lower, but it rises again on the third and fourth."

The authors organize these insights using a bottom-up, and also a top-down approach, but both are spread around specific examples, even if they try to generalize from there. Thus, they cannot make a statement about the completeness of the insights or explain for which kinds of insights a tool is suitable [SML*09]. Existing task frameworks, like the one by Andrienko and Andrienko [AA06] approach this problem by starting at the most general and abstract level, where it is possible to define a complete set of tasks. For example, they phrase tasks like "look for the characteristics at a given reference" and provide a formal rule set that describes these. They do provide details in the form of illustrative example cases, and only those are formulated according to the aspects of the structure of time. However, these example cases do not cover the design space completely, and the rules account for the unique characteristics of time only implicitly. Thus, there is a gap between the complete formal a-priori definition of tasks, for example performed in the Andrienko and Andrienko [AA06] task framework (AATF), and tasks lists that stem from free exploration, for example shown by Smuc et al. [SML*09] or arbitrary consideration by task developers. To evaluate an application in a top-down approach, or to evaluate the completeness of insights in a bottom-up approach, a task taxonomy for the dataset used is necessary. The structure of time imposes a number of aspects on such taxonomies that are always the same. The actual tasks contain a subset of them. We phrase these aspects for fitting them into the AATF, which is used because it is formally complete but also extendable (see Section 2). We have to adapt the aspects so that they fit into the framework's formalism. The result is a rule set that explains how to phrase tasks in a way that pays heed to the specific characteristics of time-oriented data. Hence, a main contribution of our work is a task framework that guides the development of test cases.

2. Related Work

Many task frameworks exist in the visualization and HCI communities. Most of them are concerned with low-level

tasks [AS05,TC05]. Shneiderman [Shn96] presents a task by data type taxonomy, listing seven tasks. Amar et al. [AES05] determines a taxonomy of ten analytical tasks from 196 concrete task. Also the user intents, which Yi et al. [YKSJ07] abstracts from the interaction techniques described in academic literature and commercial systems, can be regarded a low-level task framework. These frameworks are general and do not cater to the unique structure of time. Even though Shneiderman tackles time-oriented data, his considerations are limited to intervals and their relation, which our approach covers in Section 4.2. Tasks related to time have received special attention in the context of geographic information systems (GIS). Peuquet [Peu94] proposes a triad framework for GIS comprised of three perspectives space, time, and objects. This allows her to discern three possible task types, asking for one perspective while the other two are given. MacEachren [Mac95] presents a more detailed list of tasks relating to time in maps: Existence of an entity, temporal location, time interval, temporal texture, rate of change, sequence, and synchronization. Most of these frameworks are simple lists of tasks, where each task is described by typical questions and typical answers. While some frameworks such as [Peu94] are on a very high level of abstraction, for others like [Mac95] and [AES05] it is hard to show completeness. To overcome these problems, Andrienko and Andrienko formulate a task framework (AATF) [AA06] which allows fine-grained description of exploration tasks and which is complete in respect to their chosen data model and level of abstraction. Their data model separates between referential and characteristic components and explains the data set as a functions that associates each *reference* with a *characteristic*. In addition, they work with *relations* between references or characteristics. In a time series, for example, the time points are references, the values are characteristics, and a 20% increase of value is a relation. Tasks are categorized as *lookup, comparison,* or *relation seeking,* depending on the the target and the constraints of the task. Furthermore, they distinguish between *elementary tasks* and *synoptic tasks*. The former are concerned with the the characteristics or references of separate data elements, whereas the latter examine behaviors or patterns of the data set or subsets of the data. The AATF and its underlying data model only consider the structure of time implicitly, which means that these aspects are considered in principle, but are only phrased in terms of examples and not explicitly on the formal level. Yet, their formal definitions and structured approach allows us to tackle structure of time as an extension of this framework. Therefore, research in that area also has to be considered.

3. The Structure of Time

According to Aigner et al. [AMST11], time-oriented data can be categorized according to

Scale Time scale can be ordinal, discrete, and continuous.

Scope Temporal data can be given in the form of instants ("point-based") or intervals ("interval-based").

Arrangement Time can be linear or cyclic. Cyclic time can be modeled as periodic grouping of granularities.

Viewpoints Temporal data is often given ordered. Variants are branching time, and multiple perspectives.

Granularities Time can be divided according to structures that, for example, derive from calendric systems. A full and formal definition is given by Bettini et al. [BJW00]. They base their work on a view on the discrete time domain that is composed of atomic units called chronons. A granularity is defined as a mapping from integers that represents chronons of the discrete time domain to subsets. They also define it as the union of a number of granules, making a granule the set of a certain amount of integers from the discrete time domain. Furthermore, they define grouping operations that allow for finer granularities to be grouped into coarser granularities. E.g., if the chronons are days, they can be grouped to months or to years.

Time Primitives Instants are a model for single points in time, intervals for ranges between instants. Spans are durations (of intervals) without a fixed position. Time primitives can be used to model scope, but it is possible to consider several point-based data elements an interval. Allen [All83] provides a set of possible relations between intervals which is a time-related expansion of order theory. The relations are further extended by Aigner et al. [AMST11].

Determinacy Time-oriented data can contain uncertainties. Aigner et al. [AMST11, AMTB05] show that indeterminate instants and intervals can be modeled by using a combination of standard intervals and spans.

4. Tasks for Time-oriented Data

We intend to apply the AATF, but the aspects of time's structure require special considerations. In the following section, we add this part to the task framework. The AATF usually considers time as reference. For most EDA cases involving time-oriented data, this approach seems sensible. We will show an important exception in Section 4.4.

4.1. Scale

As the task framework itself is rather abstract, it does not have requirements regarding scale. When introducing time as a reference, we still have to consider it. In practical application, time can only be measured discretely. So on the one hand, when two characteristics seem to happen at the same time, humans can decide based on domain knowledge that this is not possible, but they cannot deduce it from the data if the level of discretization is too coarse. Our relations, on the other hand, work for both kinds of data. The difference between discrete time and ordinal time becomes apparent when dealing with relations between references. For ordinal time, the relations between two references $r_1, r_2 \in R_O$ with R_O being the ordinal time domain, are:
$r_1 = r_2$: "r_1 and r_2 happen at the same time"
$r_1 < r_2$: "r_1 happens before r_2"

$r_1 > r_2$: "r_2 happens before r_1" Logical combinations of those relations are also possible.

For discrete time, all relations between two references $r_1, r_2 \in R_D$ with R_D being the discrete time domain can be brought to the form $r_1 - r_2 = d$, with $d \in \mathbb{Z}$, "there are d chronons between r_1 and r_2."

4.2. Time Primitives

Modeling time primitives also allows for including the different variants of scope as well as determinacy. When tasks with time as reference are formulated considering time primitives, the possible relations between them have to be used in relations between references. Each reference can be an instant in time, or an interval in time. Aigner et al. [AMST11, p. 59] show variants without considering scale, we formulate them first for ordinal scale, then for discrete scale: An interval is a range in time that starts at an instant and finishes at an instant. Let $r_1, r_2, s_1, s_2, e_1, e_2 \in R_O$ be instant references in the ordinal time domain and $\overline{r_1}, \overline{r_2}$ be interval references where $\overline{r_1}$ starts at the instant s_1 and finishes at the instant e_1 while $\overline{r_2}$ is similarly given by s_2, e_2. For instants, the cases are the same as shown in Section 4.1. Following the notation of Allen [All83], new cases (that partially overlap) are:
$r_1 < s_1$: "r_1 happens before $\overline{r_1}$"
$r_1 = s_1$: "r_1 starts $\overline{r_1}$"
$r_1 = e_1$: "r_1 finishes $\overline{r_1}$"
$s_1 < r_1 < e_1$: "r_1 happens during $\overline{r_1}$"
$e_1 < s_2$: "$\overline{r_1}$ happens before $\overline{r_2}$"
$s_1 < s_2 \wedge s_2 \leq e_1 \wedge e_1 < e_2$: "$\overline{r_1}$ overlaps $\overline{r_2}$"
$s_1 = s_2 \wedge e_1 < e_2$: "$\overline{r_1}$ starts $\overline{r_2}$"
$s_2 < s_1 \wedge e_1 < e_2$: "$\overline{r_1}$ happens during $\overline{r_2}$"
$s_1 > s_2 \wedge e_1 = e_2$: "$\overline{r_1}$ finishes $\overline{r_2}$"
The relation of two intervals meeting each other cannot be formulated for ordinal data.

When considering a discrete scale, we can again use the difference $d \in \mathbb{Z}$ as the number of chronons between instants $r_1, r_2, s_1, s_2, e_1, e_2 \in R_D$:
$s_1 - r_1 = d$: "r_1 happens d chronons before $\overline{r_1}$"
$s_1 = r_1$: "r_1 starts $\overline{r_1}$"
$e_1 = r_1$: "r_1 finishes $\overline{r_1}$"
$r_1 - s_1 = d \wedge d > 0 \wedge e_1 - r_1 > 0$: "$r_1$ happens during $\overline{r_1}$, d chronons after the start"
$s_2 - e_1 = d \wedge d > 0$: "$\overline{r_1}$ happens d chronons before $\overline{r_2}$"
$s_2 - e_1 = 1$: "$\overline{r_1}$ meets $\overline{r_2}$"
$s_2 - s_1 = d_1 \wedge d_1 > 0 \wedge e_2 - e_1 = d_2 \wedge d_2 > 0 \wedge s_2 \leq e_1$: "$\overline{r_1}$ overlaps $\overline{r_2}$, starting d_1 chronons earlier and ending d_2 chronons earlier"
$s_1 = s_2 \wedge e_2 - e_1 = d \wedge d > 0$: "$\overline{r_1}$ starts $\overline{r_2}$, ending d chronons earlier"
$s_1 - s_2 = d_1 \wedge d_1 > 0 \wedge e_2 - e_1 = d_2 \wedge d_2 > 0$: "$\overline{r_1}$ happens during $\overline{r_2}$, starting d_1 chronons later and ending d_2 chronons earlier"
$s_1 - s_2 = d \wedge d > 0 \wedge e_1 = e_2$: "$\overline{r_1}$ finishes $\overline{r_2}$, starting d chronons earlier"

Finally, it is possible that spans are references. However, they can only be related to other spans, and then they can be treated like integers.

4.3. Viewpoints

An ordered dataset is the normal case and branching time is usually considered in conjunction with predicting values. For EDA, this is out of scope, but it is an important case when advancing to further tasks on a broader scope. Multiple perspectives can be modeled in the AATF by defining each one as a data function. All the tasks that consider more than one function can access these perspectives. For example, the task to compare two different attributes corresponding to the same reference $?y_1, y_2, \lambda : f_1(r) = y_1; f_2(r) = y_2; y_1 \lambda y_2$ [AA06, p. 66], can be phrased "compare the degree of customer satisfaction as reported by group 1 with the degree as reported by group 2". Multiple perspectives can also be used to model dynamic systems, like the interplay between valid time and transaction time in temporal databases.

4.4. Granularities

Granularities are formed by grouping time, so in many cases it makes sense to consider granule references, like it is usually done with time. As the AATF is based on a symmetric data model, this does not limit the possibilities. We will integrate granularities in a way that is most convenient according to one of two different task groups: (1) Performing the tasks as defined in the AATF, but basing the reference domain on granularities instead of flat and linear time. (2) Finding the granularities that are relevant in the first place. Applications of the calendar aspect of time have so far only been considered on a basis where the important granularities in a dataset are already known. Finding those granularities is a challenging task on its own right that we also describe.

4.4.1. Granularities in the Reference Domain

Without granularities, the references for time-oriented data are timestamps. To use granules as a measure, we need to count them. Bettini et al. [BJW00] define a way to assign labels to granules. We use a simplified form that is more compatible to the AATF: Let $g(t) = l; t \in R_D; l \in \mathbb{Z}$ be the label function that maps a chronon in the discrete time domain to an integer label. This label refers to a granule of a granularity—for example, 1 can have a text equivalent of January. A dataset using granularities therefore has a data functions $\dot{f}(l) = c$, mapping a granule label reference to a characteristic c. A conventional data function $f(x)$ can be mapped to $\dot{f}(l)$. However, as granularities are formed by grouping, the characteristics also need to be grouped. Possibilities include replacing one characteristic by a set of them, or aggregating them, for example by mean, median, or sum. Furthermore, a data function can be formed $\dot{f}(l_1; l_2; \ldots) = c$, having different values for granule combinations, like January in 1970, and so on. All tasks working on chronons can

also be performed working on granules of one granularity. Furthermore, it is possible to use two different functions using different granularities, but stemming from the same original function, when tasks with two functions are performed. For example, $?y_1, y_2, \lambda : \dot{f}_1(l_1; l_2) = y_1; \dot{f}_2(l_2) = y_2; y_1 \lambda y_2$, can mean "compare the value in January 1970 with the average of 1970". Behavior comparison tasks get an important meaning in conjunction with granularities. Often, a pattern is characterized by telling that a range in time belonging to one granularity is similar to another granularity. For example, "bridging days are similar to holidays".

4.4.2. Finding Granularities

When searching the granularities that are important for a dataset, many comparisons with different label functions are needed. This is easier when considering the label functions equivalent to data functions. So in that case, the chronons are the references and the labels are the characteristics. A simple task can look like this: $?y, l, x : f(x) = y; g(x) = l; y \Lambda l$ and an example would be "Which Januarys have high average values?". The same task could then be performed with other granularities, till something significant shows up, rendering one granularity interesting. More suitable for finding granularities seem to be connectional tasks (see AATF [AA06, p. 124]): $\rho(f(x), g(x) | x \in R)$ can be considered the mutual behavior of the data and a granule label, which can be directly translated to the question "does this granularity have an influence?". The scatterplots used as an example in the AATF [AA06, p. 126], can only show the influence of one granularity at a time, but visualizations, like GROOVE [LAB*09], based on the recursive pattern technique by Keim et al. [KKA95], can show the mutual behavior of one data characteristic and four or more different granularities.

4.5. Application and Rule Set

If we consider one of the insights from Section 1, like "The first Monday is high, the second is lower, but it rises again on the third and fourth." [SML*09], the task would be "describe the behavior of the characteristic value over the Mondays" which can be formulated $?p : \beta(\dot{f}(l; 1) | l \in \mathbb{Z}) \approx p$, where l is a variable week label and the second parameter gives the day being always Monday. AATF also allows to spread time across more dimensions in the form $?p : \beta(f(x_1; x_2) | x_1, x_2 \in R) \approx p$, but there are no rules how to distribute time.

Another example: Data of a stock index and individual buy and sell orders about the stock are to be analyzed. $?R_1, R_2, p_1, p_2 : R_1 \Psi R_2; \beta(f(x) | x \in R_1) \approx p_1; \beta(f(x) | x \in R_2) \approx p_2; p_1 \Lambda p_2$ could lead to the question "are there any times with many sales while the stock price is dropping?", but this is only one of many. For the data dimension, many and few transactions, falling and rising stocks are well-known terms, but what about time? Our Section 4.2 gives a full list of relations: starts, finishes, happens during, happens before, meets, overlaps, starts, happens during, finishes. People developing a task set might need to decide which of them to include, and whether they need only ordinal or discrete relations. But they have a list to check, and might, for example, find the important case of "are there any times with many sell orders meeting an interval when the stock price is dropping?"—a possible cue for insider trading.

To discern if all tasks have been found (or to state which tasks have to be searched), task developers have to phrase all relevant tasks by going through the AATF and for each task, going through the aspects mentioned in this paper:

Scale/Time Primitives When the task involves relations on time, go through all temporal primitives aspects according to the scale of the dataset.

Viewpoints When the dataset has multiple viewpoints, phrase all tasks involving two functions accordingly, calling the different viewpoints.

Granularities Phrase the tasks for finding the appropriate granularities. Only a finite list of granularities can be checked, but this list can be expanded by an automated search for cycles in the dataset. Perform the tasks to actually find the granularities. Phrase the tasks involving relations on time using the granularities found.

A complete list would most likely exceed the number of tasks that can be performed in a study, but task developers can use it to make sure nothing important is missed.

5. Conclusion and Future Work

First, we have listed restrictions of state-of-the-art task frameworks. Second, we have described the structure of time, which is an important influence on time-oriented data, and shown that it is not considered sufficiently by existing task frameworks. To help setting up data-centric tasks in order for top-down analysis or to evaluate the results from a bottom-up analysis, we have then provided a rule set for integrating the structure of time into a complete and formal task framework. This rule set allows to consistently formulate tasks for evaluating time-oriented data analysis methods.

So far, our rule set does not contain concrete tasks. These tasks can be formulated for time-oriented data in general, but in practice, it will be more important to formulate them directly for a dataset that will be used to test various systems. So the main part of future work will be the application of this work. The tasks as defined in the AATF [AA06] are used as a basis for EDA. Further task groups that VA intends to solve are forecasting and developing options [TC05, KMS*08]. Task frameworks involving these groups also have to consider the structure of time.

Acknowledgments This work was supported by the FWF Austrian Science Fund. Project number: P22883. We also wish to thank Heidrun Schuman and Christian Tominski from the University of Rostock for their valuable input.

References

[AA06] ANDRIENKO N., ANDRIENKO G.: *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer, Berlin, 2006. 1, 2, 3, 4

[AES05] AMAR R., EAGAN J., STASKO J. T.: Low-Level components of analytic activity in information visualization. In *Proc. IEEE Symp. Information Visualization (INFOVIS 2005)* (2005), pp. 111–117. 2

[All83] ALLEN J.: Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 11 (1983), 832–843. 2, 3

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*. Springer, London, UK, 2011. 2, 3

[AMTB05] AIGNER W., MIKSCH S., THURNHER B., BIFFL S.: PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proc. 9th Int. Conf. Information Visualisation (IV 2005)* (2005), Banissi E., et al., (Eds.), IEEE, pp. 457–463. 2

[AS05] AMAR R. A., STASKO J. T.: Knowledge precepts for design and evaluation of information visualizations. *IEEE Trans. Visualization and Computer Graphics 11*, 4 (2005), 432–442. 2

[BJW00] BETTINI C., JAJODIA S., WANG S.: *Time Granularities in Databases, Data Mining and Temporal Reasoning*. Springer, Berlin, 2000. 2, 3

[KKA95] KEIM D., KRIEGEL H.-P., ANKERST M.: Recursive Pattern: A Technique for Visualizing very Large Amounts of Data. In *Proc. IEEE Visualization (Vis95)* (1995), pp. 279–286. 4

[KMS*08] KEIM D., MANSMANN F., SCHNEIDEWIND J., THOMAS J., ZIEGLER H.: Visual Analytics: Scope and Challenges. In *Visual Data Mining*, Simoff S. J., Böhlen M. H., Mazeika A., (Eds.), LNCS 4404. Springer, Berlin, 2008, pp. 76–90. 4

[LAB*09] LAMMARSCH T., AIGNER W., BERTONE A., GÄRTNER J., MAYR E., MIKSCH S., SMUC M.: Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-based Visualizations. In *Proc. Int. Conf. Information Visualization (IV09)* (2009), IEEE, pp. 44–49. 4

[Mac95] MACEACHREN A. M.: *How Maps Work*. Guilford Press, New York, 1995. 2

[Peu94] PEUQUET D. J.: 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 (1994), 441–461. 2

[Shn96] SHNEIDERMAN B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. IEEE Symp. Visual Languages* (1996), pp. 336–343. 2

[SML*09] SMUC M., MAYR E., LAMMARSCH T., AIGNER W., MIKSCH S., GÄRTNER J.: To Score or Not to Score? Tripling Insights for Participatory Design. *IEEE Computer Graphics and Applications 29*, 3 (2009), 29–38. 1, 4

[TC05] THOMAS J., COOK K.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005. 1, 2, 4

[YKSJ07] YI J. S., KANG Y. A., STASKO J. T., JACKO J. A.: Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Visualization and Computer Graphics 13*, 6 (2007), 1224–1231. 2