# TimeGraph: a data management framework for Visual Analytics of large multivariate time-oriented networks

Albert Amor-Amorós*
Vienna University of Technology

Paolo Federico*
Vienna University of Technology

Silvia Miksch*
Vienna University of Technology

**ABSTRACT**

Large multivariate time-oriented networks have been gaining an increasing relevance in different domains. In order to support Visual Analytics processes on this kind of data, appropriate storage and retrieval methods are needed that take into account the scale, dimensionality, and in particular the complex nature of time. We introduce TimeGraph, a data management framework consisting of a data model and two levels of abstraction. TimeGraph captures both the topology of networks and the inherent structure of time into a property graph data structure, and transparently handles them by graph-based operations. TimeGraph aims to be an expressive, easy-to-use and extensible framework, enabling data reduction by selection and aggregation over both the temporal and the topological properties of data, to foster interactive visualization and analysis.

## 1 INTRODUCTION

Network data is used and analyzed in several domains, such as social sciences, life sciences and engineering [5]. Due to the availability of devices with continuously growing monitoring and storage capabilities, data has been getting more and more complex, mainly because of increased scale and dimensionality; moreover, it is commonly used to describe phenomena that evolve over time. Visual Analytics (VA) has been defined as the science of analytical reasoning facilitated by interactive visual interfaces [8]; it can also be understood as the integration of three essential components: data management, interactive analysis, and visualization [2]. Thus, a reliable and efficient data management is a key prerequisite for VA applications. Information visualization systems usually rely on in-memory databases to manage their data; however, a solid database system would result in more scalable and robust VA applications. Even if an integrated database is generally crucial to perform any kind of data analysis, finding an effective representation for specific data types is one of the current problems of database technology [4]. The quality of the visual analysis is affected by the quality of the underlying data representation: a large, complex, and dynamic dataset must be appropriately represented, in order to adequately support analytical tasks conveying the important content [8]. In particular, time is a data dimension with special characteristics, which require special VA methods [1].

To the best of our knowledge, there is no data management infrastructure that addresses all the challenges and the needs of VA of this kind of data, properly representing both the temporal and topological aspects at once. To fill this gap, we introduce TimeGraph, a data management framework supporting VA of large multivariate time-oriented networks. The main contributions of TimeGraph are:

- A data model for multivariate time-oriented networks, that captures the complexity of the temporal dimension together with the topological aspects into of a property graph structure.
- A modular design for a data management framework supporting storage and retrieval of large multivarite time-oriented networks.
- A prototypical implementation of the framework on top of a graph database, that aims to support VA researchers in building their own applications.

*e-mail: {amor,federico,miksch}@ifs.tuwien.ac.at

## 2 DATA MODEL

The TimeGraph data model includes **temporal primitives**, i.e. the basic elements to relate data to time, such as instants, intervals, and spans [3]. Instants and intervals are anchored primitives referring to the time domain, referencing one (e.g., 6 pm) or a sequence (e.g., 9-14 November, 2014) of entities of time respectively; conversely, spans represent unanchored durations (e.g., an hour, two days, three weeks). Additionally, the data model also incorporates **temporal granularities**, i.e. human abstractions (e.g., hours, days, weeks, months) useful to handle the complexity of time and refer to it in a simpler way. They can be understood as mappings of time values to larger or smaller units and grouping rules (e.g., hours are grouped into days, days are grouped into weeks and months). TimeFrames can handle different calendars, which are sets of rules regulating this groupings in general form a lattice structure [3].

## 3 TIMEGRAPH FRAMEWORK

The TimeGraph framework contains two levels of abstraction for managing the described data model (Figure 1).

The **TimeFrames** component is responsible of directly manipulating the graph represention of a temporal dataset described by the preceding data model, handling the temporal aspects of the data. Its main responsibility is the translation of the temporal retrieval requests into graph operations, and sending them to the underlying graph database engine for execution. A CalendarManager takes care of creating and maintaining the calendar structure.

The **GraphFrames** component handles the relational aspect of the network. Nodes and relations in the domain model are mapped by the GraphFrames layer to TemporalObjects exposed by the TimeFrames layer. Non-temporal attributes are modelled as properties of the TemporalObject nodes, while temporal attributes are managed at the underlying abstraction layer data structure.

The framework allows for modelling any number of temporal as well as non-temporal attributes on nodes and relations, and it can be used for representing any class of dynamic or temporal network from any application domain.

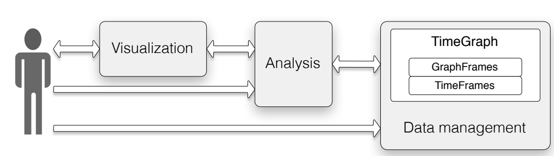


Figure 1: TimeGraph within the three layers of VA applications

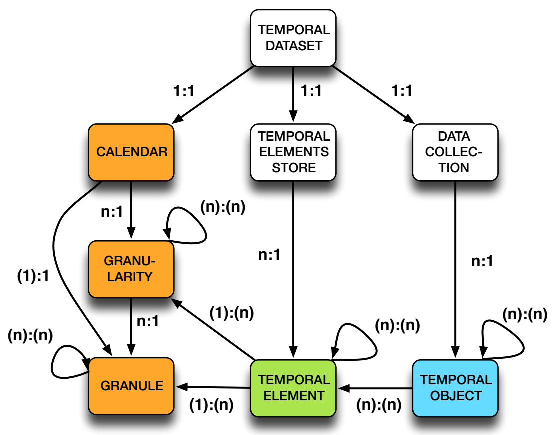


Figure 2: The **TimeGraph data model** for representing time-oriented information into a property graph structure: non-temporal aspects of data are modeled as properties of the TemporalObjects (cyan), temporal aspects are modeled as TemporalElements (green), while the graph substructure modelling the temporal reference system comprises Granule, Granularities, and Calendars (orange).

TimeGraph has been implemented in the Java environment, leveraging the Blueprints API and the TinkerPop graph management stack. It has been tested with common graph databases that exploit the property graph model [7], such as Neo4j and Titan.

## 4 Temporal Queries on Multivariate Networks

TimeGraph provides methods for accessing large multivariate network data by both structure and attributes, in particular temporal attributes. When dealing with large networks, the most important data transformations for reducing scale and complexity and enable visualization, are selection and aggregation [5, chap. 3].

**Temporal selection** is understood as the operation of filtering data by its temporal predicates, i.e., selecting a subset of the original data imposing contraints on the value of a (set of) dimension(s). TimeGraph supports two specific selection operators: the timeslice operator and the history operator [3]. The *timeslice operator* enables the direct selection of data by temporal aspects, returning all the data whose temporal values are within the given (set of) range(s). It can be seen as an operation that cuts the dataset orthogonally to the time dimension. The *history operator* enables the direct selecion of data by topological aspects, returning all versions/instances of some network entites (nodes, links, components, the entire network) over time. In other words, this operator cuts the dataset along the time dimension, and returns the time-series (i.e., histories) associated to the selected (set of) element(s).

**Aggregation** reduces the dataset by grouping data records by commonalities shared by their raw or derived attributes. TimeGraph supports aggregation by temporal attributes, specializing the general granulation in order to group up temporal granules at different temporal granularities, by using an extensible set of aggregation functions.

**Temporal predicates** for selection and aggregation can be expressed in terms of granularities, primitives, and relations between them. Thus, the framework supports the extended Allen's intervals logic, enabling the pairwise comparison of instants, intervals, and spans. Moreover, since time constraints might be expressed at different granularities, the framework has the capabilities to transform granules, by passing from a granularity to another as well as by shifting a granule back and forth along the time axis of the specified amount. TimeGraph hides the complexities arising from managing the temporal aspects of data to the developers by exposing them an abstraction of the model in which no structural differences between temporal and non-temporal aspects appear. This means that the methods for manipulating temporal attributes and non-temporal ones share the same semantics, and differences in representation and retrival are not directly visible to the developer. In particular, temporal predicates can be defined analogously as other predicates as a triple (termA, comparison operator, termB), where the terms are temporal primitives and the comparison operator is one of the relations of the Allen's extended instant-interval logic. This temporal predicate is translated into the corresponding graph matching query, including trasformations between granularities (if the terms are expressed in different granularities) and any other calendric operation needed. Then the results are returned, or remaining non-temporal predicates are evaluated in case of composite queries.

The framework enables the definition of **temporal indexing** structures to optimize the formulation and the execution of specific queries along the temporal dimension like, for example, searching for all intervals intersecting a given instant (stabbing queries) or all intervals intersecting a given interval (range-interval queries).

## 5 Future Work

We plan to demostrate the capabilities of TimeGraph by VA application examples and to validate its design by assessing performances. Neverthless, we believe that the realease of TimeGraph as a free open-source software will provide interesting insights about its usability and effectiveness.

In order to make the framework even more scalable and optimize query execution times, an interesting research path would be the design of an interaction-aware middleware that tighltly couples TimeGraph with in-memory analysis and visualization components, by providing a caching mechanism A candidate component might be TimeBench [6], a software library providing data structures and algorithms for VA of time-oriented data. TimeBench is tightly integrated with the visualization and supports rapid prototyping; however, it stores data in relational tables held in memory, and therfore its scalability is limited. Since TimeBench and TimeGraph rely on a compatible conceptual model for time-oriented data, they could be integrated in a scalable VA system, in which TimeGraph provides the data management capabilities, while TimeBench provides an interactive visualization environment.

## Acknowledgements

This work was funded by the Austrian BMVIT (EXPAND project, #835937) and BMWFW (CVAST Centre of Expertise, #822746).

## References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer, 2011.

[2] J.-D. Fekete. Visual analytics infrastructures: From data management to exploration. *Computer*, 46(7):22–29, July 2013.

[3] C. Jensen, C. Dyreson, M. Böhlen, J. Clifford, R. Elmasri, and S. Gadia. The consensus glossary of temporal database concepts. In *Temporal Databases: Research and Practice*, volume 1399 of *LNCS*, pages 367–405. Springer Berlin Heidelberg, 1998.

[4] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering the information age - Solving problems with visual analytics*. Eurographics Association, 2010.

[5] A. Kerren, H. C. Purchase, and M. O. Ward, editors. *Multivariate network visualization*, volume 8380 of *LNCS*. Springer, 2014.

[6] A. Rind, T. Lammarsch, W. Aigner, B. Alsallakh, and S. Miksch. Timebench: A data model and software library for visual analytics of time-oriented data. *TVCG*, 19(12):2247–2256, Dec 2013.

[7] M. A. Rodriguez and P. Neubauer. Constructions from dots and lines. *Bulletin of the ASIST*, 36(6):35–41, 2010.

[8] J. Thomas and K. Cook. *Illuminating the path: the research and development agenda for visual analytics*. IEEE CSP, 2005.