A Multi-Domain Framework for Community Building Based on Data Tagging

Bojan Božić

Austrian Institute of Technology,
Donau-City-Straße 1,
1220 Vienna, Austria

Abstract. In this paper, we present a doctoral thesis which introduces a new approach of time series enrichment with semantics. The paper shows the problem of assigning time series data to the right party of interest and why this problem could not be solved so far. We demonstrate a new way of processing semantic time series and the consequential ability of addressing users. The combination of time series processing and Semantic Web technologies leads us to a new powerful method of data processing and data generation, which offers completely new opportunities to the expert user.

1 Introduction

Nowadays time series processing is not only a very complex research field, but also a very specialized one. There are a lot of parties interested in time series data and all of them have certain "tailor-made" solutions for their specific problems. Therefore, we have developed the Time Series Semantic Language (TSSL). TSSL evolves from a conservative, general-purpose time series processing language, to a processing language for semantically enriched time series.

Our idea is to use semantically enriched time series to improve data processing in the Semantic Web, i.e. to be able to annotate data flows as sensor data with additional information, e.g. tagging postings of scientist with a specific research topic, which can be seen as time series, or the other way around, to use time series data as input for the creation of ontologies.

The first early prototype of the language has been developed at the Austrian Institute of Technology as scripts for processing environmental time series data. We extended the language implementation from a very specific usage to a general and dynamic language for many fields of application. To empower this, we added semantic functionality and implemented first prototypes.

To demonstrate our work and ideas, this paper describes the use of this dedicated language, which enables time series generation and processing enriched with semantics.

2 Language Specification

The language on which this research is based, is originally a classical time series processing language. This means that it is a generic language for processing time series data.

The language supports homogeneous (with fixed time grids) and inhomogeneous time series processing. Time series can have very complex data structures. It is also possible to work with time patterns, time intervals, and single slots. The complex types of aggregation can be performed with predefined, but also with user-defined functions.

Expression	**Meaning**
`< [n].sin * 2 + 3 >`	Calculation is applied to all slots.
`A, B < A + 2 * B >`	Combination of two time series (aggregation).
`< [n] > every 2 hours`	Projection to a fixed time grid.
`< (t .. t-2).mean > every 1 hour`	Sliding mean value.
`< [n]->hot if [n].temperature > 100 otherwise [n]->cold >`	Filtering, classification.

Table 1. General expressions and their meanings.

Some common expressions are shown in table 1. The first expression calculates the sine of the value from each slot, multiplies it by 2 and adds the value of 3. Expression 2 specifies two time series, where each slot of time series A is added to the doubled value of each slot from time series B. The usage of a time grid is shown in expression 3, where only the slot default value is taken every 2 hours and copied to the output time series. Expression 4 calculates a mean value for each slot and the previous two slots, but only every 1 hour. Finally, the last expression takes the property "hot" if the temperature is higher then 100, and "cold" otherwise.

TSSL has been implemented in the Python programming language, to guarantee the ease of extensibility and interoperability with other programming languages. Therefore it is usable as a standalone library on major platforms[1].

3 Semantic Time Series Processing

The main innovative contribution of our work is semantic time series processing. It tries to fix the weaknesses of current time series processing systems, such as:

[1] currently Java, .Net, and native

- meta-information is often non-existent or not bound to the processing of data,
- the linkage of ontologies is missing and therefore connections of information cannot be respected automatically,
- no possibility to add domain-specific ontologies at runtime, hence domain-specific processing is hard to implement.

The semantically enriched time series processing language, introduced in this paper is able to use predefined or user-provided ontologies to assign meaning to information. It supports automatic consideration of domain-specific calculations and functions, such as mean value calculation, thresholds, etc. for certain domains. This means that it depends on the domain how certain processing steps are affected. The advantage is the lower fault probability, because complex expressions are easier to phrase. Another issue is the verification of reasonability, e.g. there needs to be a difference between the water temperature, room temperature, and outdoor temperature. For all mentioned functionality extensions there is no need to change the syntax of the language.

This principle becomes even more obvious, if we take a look at exemplary expressions for both processors. The following expressions are examples of filtering a meteorologic time series. The name of the time series is *MeteoTS*. The first expression defines that a warning should be returned if the precipitation is greater than 1000 $\mathrm{l/m^2}$ or the temperature is greater than 40 ^{o}C or the wind speed is greater than 56 knots, etc. If no semantics is supported, one needs to specify every single condition and every single kind of warning in the expression.

```
MeteoTS < warning if precipitation > 1000 l/m^2
or temperature > 40^oC or wind > 56 knots ... >
```

The second expression has the same meaning as the first one. The only difference is that it is written for a time series processor that supports semantic time series processing. The name of the time series is again *MeteoTS*. Again a warning is returned if the value exceeds the allowed limit. The difference is that the value and the limit are not specified exactly, they rather depend on the used ontology. This means that we have different values and different limits depending on the targeted domain. The same expression can thus be used for a number of different processings and many different target groups.

```
MeteoTS < warning if value > allowed >
```

The example above shows only one possible use case for semantic extensions. It does not mean that the only improvement of semantics in time series processing is the flexible formulation of thresholds. There are many other use cases, like consideration of special information in different domains (e.g. data of a meteorologic time series may be interesting for many different domains like government, event management, air traffic, agriculture, tourism, etc., but every domain is interested in a different view of the data). Thus, semantics can help us to provide the right information to the right interest group.

4 Related Work

Semantic Web technologies have undergone a huge development in the last couple of years. New tools, technologies and projects are being introduced almost on a daily basis and first steps have been undertaken to combine the concepts of Semantic Web and Web 2.0 [1].

In ontology-based knowledge management, the SEKT[2] project produced very interesting results. Current issues in social ontologies [3], and a discussion on the relation between sociability and semantics [2] are important and could be of high interest.

The state-of-the-art in Semantic Web and Web Mining is developing very fast, and these two research areas are more and more combined, as results of Web Mining are improved by exploiting semantic structures in the Web, and Web Mining techniques are used for building the Semantic Web [4].

5 Conclusion

Our time series processing language for semantically enriched time series is an attempt to assign the right time series to the right person. The language itself is first of all a time series processing language, which covers classical time series processing functionality like arithmetic calculations, time patterns, slot selection, aggregation, mean value calculation, and much more.

Semantic time series processing is one of the features that distinguishes our language from others. It enables the consideration of meta-information, the integration of ontologies and the possibility to add domain-specific ontologies at run-time.

The time series processing language is already used in different domains like environment, traffic, etc., and several prototypes for the Semantic Web like components for accessing RDF stores, visualization and filtering, user context management.

References

1. A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandečić. The two cultures: Mashing up web 2.0 and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, (6):70–75, November 2007.
2. P. Mika. Social networks and the semantic web: the next challange. *IEEE Intell. Syst.*, 1(20):82–85, February 2005.
3. P. Mika and A. Gangemi. Descriptions of social relations. In *Proceedings of the First Workshop on Friend of a Friend.* Social Networking and the (Semantic) Web, 2004.
4. G. Stumme, A. Hotho, and B. Berendt. Semantic web mining state of the art and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*, (4):124–143, February 2006.

[2] http://www.sekt-project.com