Simulation and Modeling of Semantically Enriched Time Series

B. Božić [a]

[a]*Austrian Institute of Technology, Donau-City-Str. 1, Vienna, Austria, 1220*
Email: bojan.bozic@ait.ac.at

Abstract: Time series are a part of our day to day life. A lot of computation and data processing problems can be solved by time series processing. Therefore, the question arises how to model and simulate time series data, which contains semantics and meta-data on the one hand, and how to derive decisions out of this data on the other hand.

A lot of data and data streams could be efficiently and reasonably represented as time series. This allows a simple and highly accurate reuse in real-time or time-shifted. It is useful and natural for a lot of different domains to store and manage data in this manner. The processing of time series is necessary to draw conclusions out of data and to implement user and application specific software solutions. This is why time series processing is an optimal solution for software in many different fields of application.

Therefore, we have developed a generic language to process time series data, which supports homogeneous and heterogeneous time series, complex data structures, working with time patterns, time intervals and single slots, and complex calculation with predefined and user defined functions. The main advantages are high expressiveness, user-friendly syntax, good extensibility, and meaningful data models.

Nowadays existing decision support systems and systems for time series processing still have some weaknesses. For example the fact that meta-information may still be missing or not integrated in the processing, that ontologies are not used, which means that contexts and connections could not be correctly recognized and respected, and particularly no option to bind domain-specific ontologies during run-time, which makes domain-specific processing hard to implement.

A semantic-enabled time series processor could use predefined or user-generated ontologies to enrich information with the appropriate meaning. This would allow automatic consideration of domain-specific calculations and decision support, a low fault probability (as complex expressions are easier to be formulated), verification of meaningfulness and reasonability, and much more additional features.

The models provided by this language could be integrated in interactive decision support systems for end-users. The advantage is that they are dynamic and there is no need of touching the DSS code or manually apply models to DSS. As the current field of application is Environmental Informatics, the dynamic is restricted to this field at the current state of implementation. The main advantage in this dynamic usage is the possibility to replace models as needed and to process multiple models at the same time (which saves computation time and resources).

The fields of application are nearly endless. Whether in industrial measurement and control applications to provide a reliable processing of measurement data, in risk management applications for domain-crossing risk calculations, in environment monitoring applications for alerting and legally compliant reporting in different domains, in eHealth applications for linking of diagnosis-relevant data, such as subjective sense of well-being, or in meteorology.

As this technology is part of the TaToo project[1], its primary field of application is TaToo and applications which use the TaToo Framework, as well as other projects, technologies and applications based on TaToo.

However, semantic time series processing would be a promising supplement for every decision support system, and has the potential to contribute improvements to this scientific area.

Keywords: Time series processing, semantic web, modeling, simulation.

[1] http://www.tatoo-fp7.eu/tatooweb/

1 INTRODUCTION

Modeling and simulation of time series data is a problem that is solved only in very specialized fields of application so far. E.g. the approaches for time series modeling in the financial sector and in sensor networks are completely different. Therefore, our goal is to define a concept of time series modeling and simulation, which can be used in many different fields and areas. Moreover, we extended the approach and introduced a concept that facilitates decision making, as well.

As a first step, processing of time series, which contain raw-data, provides a lot of capabilities to the user. Depending on the result of the processing, the user or a software system is able to decide how to react in most of the cases. Therefore, our first implementation was a model for processing of so-called pure time series.

Based on the results of pure time series processing, only very limited decisions can be made. During our research activities, we figured out that decisions may vary tremendously depending on the context in which the users see the resulting time series. Hence, we decided to introduce Semantic Web technologies and provide a model for processing of semantically enriched time series data.

As a result of this work, we developed models and simulation approaches to represent pure time series data and semantically enriched time series data, as well as a contribution that can improve decision support systems as a result. The main achievements are techniques for more efficient time series processing, decision support based on time series semantics and ideas for future developments.

Section 2 of this paper introduces our approaches of pure time series processing and shows how low-level decisions can be derived out of resulting data. Our concept of time series and the usage of our time series processing language is explained.

In Section 3 we present the concept of semantically enriched time series. The main goal of the section is to show the advantages of time series processing with Semantic Web technologies and to compare the approach with a general time series processing approach.

Section 4 shows the models used to implement the previously presented concepts. It explains ways of modeling time series data and how time series can be simulated with and without semantics. Moreover the possible improvements to decision support systems, as well as to manual and automatic decision making, are investigated.

Related work is presented in Section 5. Papers, which motivated our work and helped us to develop our approaches are listed and explained.

Finally, Section 6 shows the conclusion to the work and future trends of simulation and modeling of semantically enriched time series, as well as fields of application for our achievements.

2 TIME SERIES PROCESSING

Our understanding of time series is that time series consist of slots. Each slot has a certain time stamp and a value. Hence, the simplest time series is a table with two lines, where the first line contains the time stamps and the second line the values. The intervals between time stamps do not have to be equally spaced. Additionally, slots, as well as whole time series, can have certain properties. For time series this means that they are able to carry additional information, which is valid for slots and their values, and are specific for the whole time series. In the case of slots, this means that they are able to transport more than one value for a certain time stamp.

A simple time series, plotted in R^2, is shown in Figure 1. It consists of environmental measurements of a certain hazardous material for the last 20 years. The red line defines the threshold, which, when exceeded, leads to the fact that the air quality is considered dangerous.

To process time series data, we use certain expressions. E.g. equation (1) shows the processing of a time series with awareness of threshold exceeding. The input is a time series with measurements of ozone concentration in Vienna. The expression defines that the property warning should be set to ”ACTIVE” if

[2] http://www.r-project.org

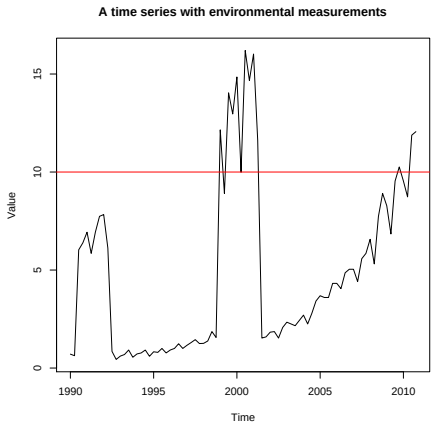

Figure 1: An environmental time series with threshold plotted in R.

the value of 10 ppm is exceeded, and to "INACTIVE" if the value is within limits.

$$@Ozone\_Vienna < warning = "ACTIVE" \; if \; Ozone\_Vienna[n] > 0.1 \, ppm \\ otherwise \; warning = "INACTIVE" > \tag{1}$$

Another example is shown in equation (2), where the acidity of water in the Danube is measured. The input time series contains values of the measurement. The expression calculates mean values of pollution for one week. Therefore, the time series data periods from the current week of each slot, whereas every week one slot is taken, until one week in the past. The result is a time series of mean values with a constant interval (one value per week).

$$@Acidity\_Danube < Acidity\_Danube[t - 1 \, week \, .. \, t].mean > every \, 1 \, week \tag{2}$$

Finally equation (3) shows the interconnection of two expressions with the pipe operator. This means that the two expressions are calculated one after another, and the output of the first is used as input of the second.

$$@Time\_Series < A[n] * 2 > \; || \; @Time\_Series < A[n] \; / \; 2 > \tag{3}$$

As we can see from the previous expressions, time series processing can be very powerful and help us with processing measurements and deriving decisions from the results. The processing can take place in real-time by streaming a time series or with a time series whose data is already complete. In our approach it does not matter in which domain the processing itself takes place or from which domain the data originates.

While it is clear that time series processing solves a lot of problems, there is still an open question left. The current situation is that there exist different users from different domains. All of them may be interested in a specific time series. But the problem arises as soon as the users claim to be interested in only some special information from the time series. This is the point where pure time series processing stops to be sufficient.

3 SEMANTIC TIME SERIES

To improve the capabilities of decision support systems in time series processing, we introduce the semantic time series approach. Our idea is to enrich time series with meta-information, integrate ontologies into processing, and hence make connections between data and meta-data. This process makes it possible

to dynamically change the process of time series processing from general to domain-specific by adding an ontology, or to switch the domain by replacing an ontology, during run-time.

The core component for this task is a semantic-enabled time series processor, which uses domain-specific ontologies as input and enriches data of time series with its meaning. As a consequence, the time series processor provides domain-specific calculations and decision support to external systems and software.

As a first step, visualization and filtering functionality is needed to extend time series processing in a semantic way. Visualization and filtering can be seen as a kind of tagging processor. Visualization can be understood as the visualization of tags (e.g. a tag cloud) of resources selected by a user, but also grouping tags, highlighting certain tags, etc. The filtering functionality filters tags by specific tag values, such as user name, tagged resource, time, location, etc.

Figure 2 shows a tag cloud generated by Wordle[3]. It corresponds to our visualization functionality and shows a tag cloud generated out of terms. The size of a word demonstrates the frequency of the appearance of the word

Figure 2: A tag cloud of this document generated with Wordle.

The same way as this document can be represented as a tag cloud, our visualization extension to the time series processing language is able to generate tag clouds out of time series of tags. Therefore, the input, which is not a document, but a time series, is processed by counting the frequency of tags and generating a tag cloud to visualize the importance and weight of tags.

Regarding the filtering functionality, our solution is an extension to the time series processing approach with ontologies. The following ontology in Listing 1 shows such an example.

```
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
@prefix sts:  <http://www.semantic-time-series.org/stsp.owl#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

stsp:isInterestedIn           rdf:type             owl:ObjectProperty .
stsp:InterestedGroup          rdf:type             owl:Class .
stsp:TimeSeries               rdf:type             owl:Class .
stsp:Bank                     rdf:type             stsp:InterestedGroup ;
                              stsp:isInterestedIn  stsp:FinancialTimeSeries .
stsp:EnvironmentalTimeSeries  rdf:type             stsp:TimeSeries .
stsp:FinancialTimeSeries      rdf:type             stsp:TimeSeries .
stsp:Government               rdf:type             stsp:InterestedGroup ;
                              stsp:isInterestedIn  stsp:EnvironmentalTimeSeries ,
                                                   stsp:FinancialTimeSeries .
stsp:GreenPeace               rdf:type             stsp:InterestedGroup ;
                              stsp:isInterestedIn  stsp:EnvironmentalTimeSeries .
stsp:InsuranceCompany         rdf:type             stsp:InterestedGroup ;
                              stsp:isInterestedIn  stsp:EnvironmentalTimeSeries ,
                                                   stsp:FinancialTimeSeries .
stsp:University               rdf:type             stsp:InterestedGroup ;
```

[3]http://www.wordle.net

```
                    stsp:isInterestedIn  stsp:EnvironmentalTimeSeries ,
                                         stsp:FinancialTimeSeries .
```

Listing 1: Turtle sample code.

The purpose of this ontology is to add information about different interested groups to the time series and to enable the semantic time series processor to provide the right output for each domain. At first, the ontology in Listing 1 defines some prefixes[4] to improve the legibility of the code. In the second step the ontology defines *isInterestedIn* as an object property, and *InterestedGroup* and *TimeSeries* as classes. This is the static part to define a general relation and general classes. The next step defines the instances (similar to object-oriented programming) and their relations to each other. E.g. the instance *Government* is of type *InterestedGroup* and is linked to the instances *EnvironmentalTimeSeries* and *FinancialTimeSeries* of type *TimeSeries* via the relation *isInterestedIn*.

The objective of visualization and filtering is to visualize tags to the user and perform filtering operations requested by the user. The time series processor with visualization and filtering functionality takes meta-information (e.g. an RDF document) in XML format and user input to perform certain operations.

4 MODELS AND SIMULATIONS

Our language provides models which are ready to use by third party software components. The current version is being tested in the environmental field of application, but the approach is a generic one. By default, we use the auto-regressive model and the moving average model. This is because we assume that a data source provides linear data.

The auto-regressive model is used to predict time series data. It is linear and random, which means that we use time series with already measured data to predict future values of a certain time series.

$$X_t = \delta + \sum_{i=0}^{p} \phi_i X_{t-i} + A_t \tag{4}$$

Equation (4) shows the formula for the auto-regressive model. X_t represents the time series, A_t is the white noise, and δ is a constant value. The value p is called the order of the auto-regressive (AR) model and represents the length of the time series. Finally, ϕ is the parameter of the model.

The moving average model is another common model for modeling univariate time series. It is a linear regression of the current value of the time series against the white noise or random shocks of one or more prior values of the time series.

$$X_t = \mu + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \theta_3 A_{t-3} - ... - \theta_p A_{t-p} \tag{5}$$

Equation (5) shows the formula for the moving average model. X_t is again the time series, μ is the mean value of the time series, A is the white noise, and θ is a parameter of the model. The order of the model is again represented by p.

These two models are already implemented in our time series processing software. Since our time series processing components are implemented in the Python programming language[5], the user is able to implement own models as python functions and provide them to our semantic time series processing language. This allows a dynamic change of the model used for processing and simulation of time series data.

5 DECISION SUPPORT SYSTEMS

Regarding decision support systems (DSS), we target especially data-driven DSSs. As data-driven DSSs deal with time series data and their manipulation. The novelty here is the enrichment of data for data-driven DSSs with Semantic Web technologies. As we have seen in the previous chapters, our idea is to

[4]Prefixes are a kind of namespaces.
[5]http://www.python.org

provide addional information for better decisions regarding not only data but also it's context. Therefore, the presented technologies have the aim to build a basis for future developments and to improve the efficiency and functionality of future data-driven DSSs.

6 RELATED WORK

In time series processing, modeling, and forecasting, some very interesting developments and achievements have been published. Although, the publications happened around the year 2000, they present interesting approaches, which influenced our developments.

The time series modeling approach developed by Lehtokangas et al. [1996] has a neural network structure with three layers and uses a general autoregressive model, Figwer [1997] presents an approach to the simulation of wide-sense stationary random time-series defined by its power spectral density, and the study of Zhang et al. [2001] presents an experimental evaluation of neural networks for non-linear time-series forecasting.

In the application area of decision support systems, we found out that time series processing, as well as semantic technologies are able to improve the efficiency of problem solving. Therefore we analysed related work on decision support systems as well and implemented some concepts and approaches from this field.

As Fazlollahi et al. [1997] state, the effectiveness of decision support systems (DSS) is enhanced through dynamic adaptation of support to the needs of the decision maker, to the problem, and to the decision context. Chuang and Yadav [1998] developed an integrated conceptual model of an adaptive decision support system (ADSS).

Semantic technologies and web service technologies play the most important part in our work. The following related work and publications helped us to build up a semantic solution to our problem and shown us approaches used in the field of Semantic Web.

Gibbins et al. [2004] describe that the Web Services world consists of loosely-coupled distributed systems, which adapt to changes by the use of service descriptions. Almendros-Jiménez [2008] investigates an extension of XQuery for querying from RDF documents. Agarwal et al. [2004] explain that the way web services are currently being developed, places them beside rather than within the existing World Wide Web. Automatic metadata generation may provide a solution to the problem of inconsistent, unreliable metadata describing resources on the Web, as explained by Jenkins et al. [1999]. High quality domain ontologies are essential for successful employment of Semantic Web services, as Sabou et al. [2005] describe. The work of Lukasiewicz and Straccia [2008] shows that ontologies play a crucial role in the development of the Semantic Web. Golbreich et al. [2006] present a method developed for migrating the Foundational Model of Anatomy (FMA) from its representation with frames to its logical representation. The results of Shekarpour and Katebi [2010]'s paper are a review and analysis of well known methods of trust modeling and evaluation. Viinikka et al. [2009] found out that the main use of intrusion detection systems (IDS) is to detect attacks against information systems and networks. Noy and Rubin [2007]'s Foundational Model of Anatomy (FMA) represents the result of manual and disciplined modeling.

7 CONCLUSIONS

This paper proposes an approach to combine Semantic Web technologies with time series processing models. This approach makes a lot of new applications for time series processing (e.g. in social web, semantic web portals, visualization of time series, etc.) possible. The main advantage is the ability to process and prepare the same time series data for different domains. Based on the results of processing and simulation, systems are able to make better decisions and improve their own results.

We developed several prototypes to extend our time series processing language. They prove the feasibility and applicability of our approaches. The time series processing language itself is a classical one and provides standard means to deal with time series data. The semantic time series concept is a truly new idea. Our prototype currently supports visualization of semantic time series data in tag clouds and filtering based on meta-information of time series. The models we use are easily and dynamically extensible by users.

In our future work we will extend semantic functionality of the language and implement a framework to pool various components together and to provide a ready-to-use solution for building of semantic time series applications. The framework will be able to find resources by semantic discovery, to support manual and automatic tagging, to process these tagged semantic time series, to build groups of interest and communities for the processing output, and to present and visualize tagged resources and communities to an end-user.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr. 247893.

REFERENCES

Agarwal, S., S. Handschuh, and S. Staab (2004). Annotation, composition and invocation of semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web 2*, 31–48.

Almendros-Jiménez, J. M. (2008). An rdf query language based on logic programming. *Electronic Notes in Theoretical Computer Science 200*, 67–85.

Chuang, T.-T. and S. B. Yadav (1998). The development of an adaptive decision support system. *Decision Support Systems 24*, 73–87.

Fazlollahi, B., M. A. Parikh, and S. Verma (1997). Adaptive decision support systems. *Decision Support Systems 20*, 297–315.

Figwer, J. (1997). A new method of random time-series simulation. *Simulation Practice and Theory 5*, 217–234.

Gibbins, N., S. Harris, and N. Shadbolt (2004). Agent-based semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web 1*, 141–154.

Golbreich, C., S. Zhang, and O. Bodenreider (2006). The foundational model of anatomy in owl: Experience and perspectives. *Web Semantics: Science, Services and Agents on the World Wide Web 4*, 181–195.

Jenkins, C., M. Jackson, P. Burden, and J. Wallis (1999). Automatic rdf metadata generation for resource discovery. *Computer Networks 31*, 1305–1320.

Lehtokangas, M., J. Saarinen, K. Kaski, and P. Huuhtanen (1996). A network of autoregressive processing units for time series modeling. *Applied Mathematics and Computation 75*, 151–165.

Lukasiewicz, T. and U. Straccia (2008). Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web 6*, 291–308.

Noy, N. F. and D. L. Rubin (2007). Translating the foundational model of anatomy into owl. *Web Semantics: Science, Services and Agents on the World Wide Web 6*, 133–136.

Sabou, M., C. Wroe, C. Goble, and H. Stuckenschmidt (2005). Learning domain ontologies for semantic web service descriptions. *Web Semantics: Science, Services and Agents on the World Wide Web 3*, 340–365.

Shekarpour, S. and S. D. Katebi (2010). Modeling and evaluation of trust with an extension in semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web 8*, 26–36.

Viinikka, J., H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen (2009). Processing intrusion detection alert aggregates with time series modeling. *Information Fusion 10*, 312–324.

Zhang, G. P., B. E. Patuwo, and M. Y. Hu (2001). A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers and Operations Research 28*, 381–396.