TemporalEMF: A Temporal Metamodeling Framework - Extended Abstract

Abel Gómez[1], **Jordi Cabot**[1,2], **Manuel Wimmer**[3]

1. *Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Spain*
   agomezlla@uoc.edu
2. *ICREA, Spain*
   jordi.cabot@icrea.cat
3. *CDL-MINT, TU Wien, Austria*
   wimmer@big.tuwien.ac.at

KEYWORDS: *Temporal Models, Model-Driven Engineering*

Modeling tools and frameworks have improved drastically in the last decade due to the maturation of metamodeling concepts and techniques (Brambilla *et al.*, 2017). A concern which did not yet receive enough attention is the temporal aspect of metamodels and corresponding models when it comes to model valid time and transaction time dimensions instead of just arbitrary user-defined times (Gregersen, Jensen, 1999).

Indeed, existing modeling tools provide direct access to the most current version of a model, but very limited support to inspect the model state at specific past time periods (Bill *et al.*, 2018; Benelallam *et al.*, 2017). This typically requires looking for a model version stored in some kind of model repository roughly corresponding to that time period and using it to manually retrieve the required data. This approximate answer is not enough in scenarios that require a more precise and immediate response to temporal queries like complex collaborative co-engineering processes or runtime models (Mazak, Wimmer, 2016).

To deal with these new scenarios, temporal language support must be introduced as well as an infrastructure to efficiently manage the representation of both historical and current model information. Furthermore, query means are required to validate the evolution of a model, to find interesting modeling states, as well as execution states. Using existing technology to tackle these requirements is not satisfactory as we later discuss.

To tackle these limitations, we reuse well-known concepts from temporal languages to propose a temporal metamodeling framework, called *TemporalEMF*, that adds native temporal support. In *TemporalEMF*, models are automatically treated as temporal, and temporal query support allows to retrieve model elements at any point in time. Our framework is realized on top of EMF (Steinberg *et al.*, 2009).

Models history is transparently stored in a NoSQL database, thus supporting large evolving models. We evaluated the resulting framework using an Industry 4.0 case study of a production system simulator (Mazak *et al.*, 2017). The results showed good scalability for storing and accessing temporal models without requiring changes to the syntax and semantics of the simulator.

Specifically, the contribution of this work is three-fold: (*i*) we present a **lightweight extension** of current metamodeling standards to build a temporal metamodeling language. This metamodel is also represented as a profile for augmenting existing metamodels with information about temporal aspects; (*ii*) we introduce an **infrastructure to manage temporal models** by combining EMF and HBase (The Apache Software Foundation, 2018), an implementation of Google's BigTable storage (Chang *et al.*, 2006); and (*iii*) we outline a **temporal query language** to retrieve historical information from models as an extension of the well-known Object Constraint Language.

Please note that contributions do not change the general way how models are used: if only the latest state is of interest, the model is transparently accessed and manipulated in the standard way as offered by the EMF. Thus, all existing tools are still applicable, and the temporal extension is considered to be an add-on.

As further work, we plan to extend *TemporalEMF* in several directions. At the modeling level, we will predefine some useful temporal patterns to facilitate the definition of temporal queries and operations. At the technology level, we will explore the integration of our infrastructure in other NoSQL backends and Web-based modeling environments to expand our potential user base. Finally, we aim to exploit the generated temporal information for a number of learning and predictive tasks to improve the user experience with modeling tools. For instance, we could classify users based on their typical modeling profile and dynamically adapt the tool based on that behaviour.

References

Benelallam A., Hartmann T., Mouline L., Fouquet F., Bourcier J., Barais O. *et al.* (2017). Raising time awareness in model-driven engineering: Vision paper. In *Proc. of models*, p. 181-188.

Bill R., Mazak A., Wimmer M., Vogel-Heuser B. (2018). On the need for temporal model repositories. In *Proc. of staf workshops*, pp. 136–145.

Brambilla M., Cabot J., Wimmer M. (2017). *Model-driven software engineering in practice, 2nd edition*. Morgan & Claypool Publishers.

Chang F., Dean J., Ghemawat S., Hsieh W. C., Wallach D. A., Burrows M. *et al.* (2006). Bigtable: A distributed storage system for structured data. In *Proc. of osdi*, pp. 15–15.

Gregersen H., Jensen C. S. (1999). Temporal entity-relationship models - A survey. *IEEE Trans. Knowl. Data Eng.*, Vol. 11, No. 3, pp. 464–497.

Mazak A., Wimmer M. (2016). Towards liquid models: An evolutionary modeling approach. In *Proc. of cbi*, pp. 104–112.

Mazak A., Wimmer M., Patsuk-Boesch P. (2017). Reverse engineering of production processes based on Markov chains. In *Proc. of case*, p. 680-686.

Steinberg D., Budinsky F., Paternostro M., Merks E. (2009). *EMF: Eclipse Modeling Framework 2.0* (2nd ed.). Addison-Wesley Professional. (ISBN: 0321331885)

The Apache Software Foundation. (2018). *Apache HBase.* (http://hbase.apache.org/)