Model-based time-distorted Contexts for efficient temporal Reasoning

Thomas Hartmann*, Francois Fouquet*, Gregory Nain*, Brice Morin‡, Jacques Klein* and Yves Le Traon*
*Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, first.last@uni.lu
‡SINTEF ICT Norway, Norway, first.last@sintef.no

I. Introduction

Intelligent systems continuously analyze their context to autonomously take actions. Building a proper knowledge representation of the context is key to take adequate actions. This requires context models, *e.g.* formalized as ontologies or meta-models. As these systems evolve in dynamic contexts, reasoning processes typically need to analyze and compare the current context with its history. A common approach consists in a temporal discretization, which regularly samples the context at specific timestamps (snapshots) to keep track of history. Fig. 1 shows a context sampled at three different timestamps. Reasoning processes would then need to mine a huge amount of data, extract a relevant view, and finally analyze it. This would require lot of computational power and be time-consuming, conflicting with the near real-time response time requirements of intelligent systems. To address these issues, we define *time-distorted* contexts as time-aware context models. Fig. 2 shows a context representation, where the context variables belong to different timestamps. Our approach considers temporal information as first-class property crosscutting any context element, and enables building time-distorted views of a context composed by elements from different times rather than a mere stack of snapshots. We claim that these time-distorted views can efficiently empower continuous reasoning processes and outperform traditional full sampling approaches by far.

II. Background

Over time different formalisms to represent the context of intelligent systems have been developed [1], [2], [3] for different purposes. Entity-relationship models [4], as a general modeling concept for describing entities and the relationships between them, are widely used for building context representations. Most of these approaches describe a context using a set of concepts (classes, types, elements), attributes (properties), and the relations between them. We refer to the representation of a context (set of elements) as a *context model* and to a single concept as *model element*. An emerging paradigm called *models@run.time* [5], [6], [7] proposes to use models both at design and runtime in order to support intelligent systems. Models support the design and implementation of the system, and are then embedded at runtime to support the reasoning processes of intelligent systems, as models offer a *simpler, safer and cheaper* [8] means to reason. Our implementation and the provided API are build on a models@run.time-based context representation approach and are integrated into an open source modeling framework, called Kevoree Modeling Framework [9] (KMF [1]). KMF is an EMF [10] alternative specifically designed to support the models@run.time paradigm.

III. Time-distorted Contexts

We consider temporal knowledge as part of a domain itself (*e.g.* electric load or wave propagation prediction, medical recommender systems, financial applications) and think that defining and navigating temporal data directly within domain contexts is far more efficient and convenient than regularly sampling a context and independently querying each model element with the appropriate time. Here, we define concepts to navigate into the time dimension of contexts and seamlessly combine elements from different points of time.

Temporal validity: Instead of relying on context snapshots, we define a context as a continuous structure, where each element can evolve independently. We define an implicit *validity* for model elements by associating a timestamp to each of them that defines a *version* $v_{m_e}(t)$ of a model element m_e at a time *t*. If a model element evolves, an additional version of the same element is created and associated to a new timestamp. Timestamps can be compared and thus form a chronological sequence. Although timestamps are discrete values, they logically define intervals in which a model element can be considered as *valid* (*i.e.* from the time it is captured until a new version is captured). New versions are only created if the model element changes. Because of this temporal validity, a relationship *r* from a model element m_{e1} to m_{e2} is no longer uniquely defined. Instead, the timestamps of model elements have to be taken into account for resolving relationships.

Navigating in time: Based on the assumption that intelligent systems need to consider not only the current context but also historical data, we provide means to enable an efficient navigation into time. We define three operations that can be called on each model element. The *shift* operator takes a timestamp as parameter, looks for the valid version of the element at the required timestamp, loads it from storage and returns the loaded version. *previous* and *next* are shortcuts to respectively retrieve the direct predecessor and successor of the current model element. These operations allow to shift model elements independently from each other through time, and make it possible to create context models combining model elements from different points of time.

The research leading to this publication is supported by the National Research Fund Luxembourg (grant 6816126) and Creos Luxembourg S.A. under the SnT-Creos partnership program.

[1] http://kevoree.org/kmf

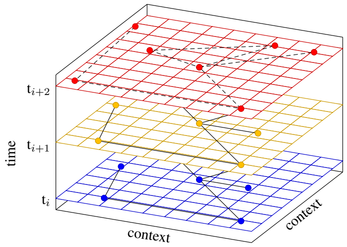

Fig. 1. Linear sampled context

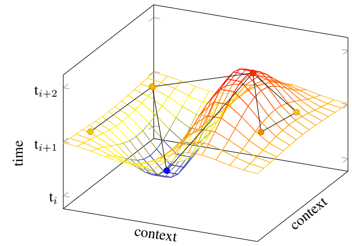

Fig. 2. Time-distorted context

Time-relative navigation: Navigating temporal data is complex, since a relationship r from an element m_{e1} to an element m_{e2} is not uniquely defined. Indeed, r can link different versions of m_{e2} depending on the timestamps t_1 and t_2 of m_{e1} and m_{e2}, and on the set of versions of m_{e1} and m_{e2}. This means that the version of m_{e2} linked to m_{e1} by r depends on the timestamp t of m_{e1}. Navigating in the model, while considering this time-relative navigation manually is complicated and error-prone. We therefore provide a navigation mechanism, hidden in the navigation methods of the context model, that automatically resolves the relationships transparently for the user. Hereby, a context time can be defined (the curve in fig. 2) and each model element is then resolved accordingly to this definition while traversing the model. For example, the context time can be defined as the current time of a model element minus one day. When navigating from model element m_{e1} at timestamp t_i to element m_{e2}, the version of m_{e2}, which is valid at timestamp $t_i - 1$ *day* is resolved. In case at timestamp t_i object m_{e2} does not exist the *prior existing version of* m_{e2} is returned. Considering model elements in the context of a specific time interval creates a navigable time dimension for model elements. This time relative data resolution is one of the novel concepts of this contribution. Indeed unlike in previous approaches (*e.g.* relationships in MOF [11] or predicates in RDF [12]), the navigation function is not constant but yields different results depending on the navigation context (*i.e.* the current observation date). This distortion in terms of navigable relations finally enables what we call a time-distorted context.

IV. Implementation and Evaluation

This approach has been integrated in an open source modeling framework: KMF. The integration relies on two properties: i) each model element must be uniquely identifiable, and ii) it must be possible to get a serialized representation of all attributes and relationships of each model element, with no relativity to a time. KMF offers a *path* mechanism to support the i) property, and *traces* to address the ii) requirement. Thanks to this integration, we ran an experimentation on a smart grid example to evaluate our approach by comparison with a full sampling strategy. To this end, we implement a reasoning engine, which aim is to predict if the electric load in a certain region will likely exceed or surpass a critical value. Our validation focuses on two key indicators: (1) performance of the reasoning process and (2) insertion time. In order to cover several use cases, our experimentation involves 4 different set-ups varying a) the size of the area used in the prediction (number of meters) and b) the size of the history. Our experimentation shows that the insertion time, compared to full sampling, is improved by a factor of 17 and that the time required by the algorithms to finish have been improved by factors from 598 to 1361.

V. Conclusion

Considering time as a crosscutting concern of data modeling has been discussed since more than two decades. However, recent data modeling approaches mostly still rely on a discrete time representation, which can hardly consider model elements coming from different points of time. We presented an approach, which considers time as a first-class property crosscutting any model element, allowing to organize context representations as time-distorted views dedicated for reasoning processes, rather than a mere stack of snapshots. By introducing a temporal validity for model elements we allow them to evolve independently and at different paces, making the full sampling of a context model unnecessary. Instead of introducing a dedicated querying language we provided operations to move model elements independently through time. Finally, we added a time-relative navigation, which makes an efficient navigation between model elements, coming from different timestamps, possible. Our approach has been implemented and integrated into KMF.

References

[1] M. Perttunen, J. Riekki, and O. Lassila, "Context representation and reasoning in pervasive computing: a review," *Int. Journal of Multimedia and Ubiquitous Engineering*, pp. 1–28, 2009.

[2] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, 2007.

[3] T. Strang and C. L. Popien, "A context modeling survey," in *UbiComp 1st Int. Workshop on Advanced Context Modelling, Reasoning and Management*, 2004, pp. 31–41.

[4] P. P. shan Chen, "The entity-relationship model: Toward a unified view of data," *ACM Trans. Database Syst.*, vol. 1, pp. 9–36, 1976.

[5] G. Blair, N. Bencomo, and R. France, "Models@ run.time," *Computer*, vol. 42, no. 10, pp. 22–27, 2009.

[6] B. Morin, O. Barais, J. Jezequel, F. Fleurey, and A. Solberg, "Models@ run.time to support dynamic adaptation," *Computer*, vol. 42, 2009.

[7] S. Kent, "Model driven engineering," in *IFM*, 2002.

[8] J. Rothenberg, L. E. Widman, K. A. Loparo, and N. R. Nielsen, "The nature of modeling," in *Artificial Intelligence, Simulation and Modeling*, 1989, pp. 75–92.

[9] F. Fouquet, G. Nain, B. Morin, E. Daubert, O. Barais, N. Plouzeau, and J. Jézéquel, "An eclipse modelling framework alternative to meet the models@runtime requirements." in *MoDELS*, 2012.

[10] F. Budinsky, D. Steinberg, and R. Ellersick, *Eclipse Modeling Framework : A Developer's Guide*, 2003.

[11] OMG, *OMG Meta Object Facility (MOF) Core Specification, Version 2.4.1*, Object Management Group Std., Rev. 2.4.1, 2011.

[12] O. Lassila and R. R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification," W3C, W3C Recommendation, 1999.