An OWL-Based Approach for Representing Time in Web Information Systems

Viorel Milea[a] Flavius Frasincar[a] Uzay Kaymak[a] Tommaso di Noia[b]

[a] *Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062 PA Rotterdam*
[b] *Politecnico Di Bari, via E. Orabona, 4 - 70125 Bari, Italy*

The full version of this paper appeared in:
Proceedings of the 4th Workshop on Information Systems Modeling (WISM '07), pp.791-802, 2007.

Abstract

In this paper we present an approach towards representing dynamic domains by means of concrete domains and perdurants. This approach is based on Description Logics and enables the representation of time and time-related aspects such as change in ontologies. The approach taken in this paper focuses on two main goals that need to be achieved when talking about time in ontologies: representing *time* itself and representing *temporal aspects*. We employ an explicit representation of time and rely on the internal method for the purpose of reflecting the changing aspects of individuals over time.

1 Introduction

One of the challenges posed by the Semantic Web is dealing with temporal aspects in a variety of domains, such as knowledge reasoning. While the Web Ontology Language (OWL) is the preferred alternative for representing domain knowledge, the language currently offers only little support for representing temporal information in ontologies. The representation of this temporal information is needed when one considers modeling dynamic domains, i.e., domains that change their properties in time. In this paper we present an approach for representing dynamic domains by means of concrete domains and perdurants. This approach is based on Description Logics and enables the representation of time and time-related aspects such as change in ontologies.

We distinguish between two aspects of time that are relevant for the current purpose: *concrete time* and *temporal entities*. The concrete representation of time is rather straight forward, and relates to making the latter available in the ontology in the form of dates, hours, minutes, etc. This type of representation allows for more (semantically) useful time representations such as instants and intervals. We talk here about an *explicit* representation of time, as this representation allows the usage of temporal operators and the combination of the latter for obtaining new expressions [2]. Representing time in such a manner allows the use of the 13 Allen interval relations [1] in combination with temporal intervals. The symbiosis between the time intervals and Allen's relations represents the concrete domain employed for the current purpose.

Representing time is an essential feature of a language that seeks to represent dynamic domains. However, this representation must be supported by means of consistently expressing temporal aspects, such as change, in ontologies. Two approaches are possible for this purpose, namely the internal and the external method [2]. In this paper we employ the *internal method* for the purpose of reflecting the changing aspects of individuals over time. This method relates to representing entities in a unique manner at different points in time. The actual individual, i.e., the perdurant, is then nothing more than the sum of its (temporal) parts. In this way one can capture, in each temporal component, the individual's properties that hold at that particular time. Extending the approach taken in [5], we implement this representation by making use of time slices (the temporal parts of an individual) and fluents (properties that hold between timeslices).

2 The Approach

The concrete domain that we considered is the set of real numbers with the binary predicates $<$, $\leq$, $>$, $\geq$, $=$, $\neq$. This concrete domain is used to represent time points in a quantitative manner and the temporal relationships between time points. In this way one can define the interval class as $\mathit{Interval} \equiv \exists\, (\mathit{start}, \mathit{end}).\leq$, i.e., the *start* and *end* concrete features need to fulfill the "$\leq$" constraint [3].

The real numbers are based on an encoding scheme of the XML Schema *xsd:dateTime* that we used for the quantitative representation of time. The 13 Allen relations (*equals*, *before*, *after*, *meets*, *met-by*, *overlaps*, *overlapped-by*, *during*, *contains*, *starts*, *started-by*, *finishes*, *finished-by*) [1] can be simulated by using straightforward translation rules. For example two intervals are *after* each other if the *start* time point of the first interval is after the *end* time point of the second interval. In this manner one is able not only to quantify time, but also to specify temporal constraints.

The concrete domain approach for representing time provides a good foundation towards representing change in ontologies. For this purpose we introduce in TOWL the concept of perdurants, materialized in explicit constructs for timeslices and fluents. The temporal parts of a perdurant are described as timeslices that can be regarded as snapshots of an individual at a particular moment (interval) in time. Timeslices are connected through (subproperties of) the fluent property, i.e., the fluent is indicating what is changing at a particular moment (interval) in time.

As a proof of concept of our approach we built a system that extracts the relevant information regarding company shares from analyst recommendations and uses this aggregate information to generate buy/hold/sell signals based on predefined rules. The dynamicity of this domain enables us to demonstrate the usefulness of our approach for representing temporal information.

3 Advantages of this Approach

The advantages of the approach presented here are twofold. First, unlike previous approaches, the current approach provides the means to represent both time in its quantitative nature, as well as temporal entities. Previous work in this area focuses on only one of these aspects, such as in [3], where the main focus is on representing time in its quantitative meaning by employing concrete domains. Other approaches that try to tackle the same problem, such as [4] where time is made available through an ontology of time, offer little to no support for automated temporal reasoning, thus bringing the discussion to the second advantage of the representation we chose for the purpose of representing time. Since all concepts are modeled at language level, this provides the basis for designing appropriate algorithms that will enable temporal reasoning with regard to both meanings of time as underlined here: quantitative time (order, duration, etc.) as well as temporal entities (change, temporally bounded existence, evolution, etc.).

References

[1] J.F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11): 832-843, 1983.

[2] A. Artale and E. Franconi. A Survey of Temporal Extensions of Description Logics. *Annals of Mathematics and Artificial Intelligence*, 30(1-4):171-210, 2001.

[3] F. Baader and P. Hanschke. A Scheme for Integrating Concrete Domains into Concept Languages. *12th International Joint Conference on Artificial Intelligence*, pp. 452-457, 1991.

[4] J.R. Hobbs and F. Pan. An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing*, 3(1):66–85, 2004.

[5] C. Welty, R. Fikes and S. Makarios. A Reusable Ontology for Fluents in OWL. *Fourth International Conference on Formal Ontology in Information Systems*, pp. 226-236, 2006.