On Temporal Cardinality in the Context of the TOWL Language

Viorel Milea[1], Michael Mrissa[2], Kees van der Sluijs[3], and Uzay Kaymak[1]

[1] Faculty of Economics, Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
`{milea, kaymak}@few.eur.nl`
[2] University of Namur
Rue de Bruxelles 61, B-5000 Namur, Belgium
`michael.mrissa@fundp.ac.be`
[3] Eindhoven University of Technology
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
`k.a.m.sluijs@tue.nl`

Abstract. The TOWL language is a temporal ontology language built on top of OWL-DL that enables descriptions involving time and temporal aspects such as change and state transitions. Extending OWL-DL into a temporal context does not only relate to providing the adequate expressiveness for such a goal, but also ensuring that static concepts preserve their meaning in a temporal environment. One such concept relates to cardinality. In this paper, we discuss temporal cardinality in the context of the TOWL language, and provide a possible approach towards representing temporal cardinality in this context.

1 Introduction

The role of Web Information Systems (WIS) on the Web is constantly increasing in importance. The Web 2.0 transformed Web pages from static pieces of text into dynamic applications with desktop-like functionality on a Web scale. It has thus become possible to exploit Web features, such as being able to integrate functionality of different applications into a new Web application. The Semantic Web adds a new dimension, as now not only functional blocks can be exchanged, but also knowledge, allowing information to be processed in entirely new ways. An important feature of the Semantic Web consists of its explicit semantics: metadata about the information is explicitly modeled in terms of classes and relationships between them. This provides rich descriptions that can be reasoned upon.

By using the state-of-the-art Semantic Web language OWL [1] we can, for example, model that an instance of the class *Person* is married with another instance of the class *Person*. If we want to be a bit more explicit we might consider modeling that a person is only married with one other person, i.e., that a marriage is symmetrical (x is married to y, implies y is married to x) and not all persons are married. In OWL this can be represented as follows.

```
ObjectProperty(:marriedTo Symmetric
    domain(:Person)
    range(:Person))
Class(:Person partial
    restriction(:marriedTo maxCardinality(1))
    restriction(:marriedTo minCardinality(0)))
```

However, this is a static model. If we want to be able to account for remarriage, e.g. due to divorce or to the death of one of the partners, we might want to slightly change our restrictions and state that a person is only married to one other person *at any point in time*. Expressing such a restriction requires the existence of a construct for the representation of temporal cardinality. However, such a construct is not available in OWL, nor in its temporal extension, TOWL.

Different application domains motivate the need for a temporal ontology language. In this paper, we focus on two specific application domains: first, the trading domain (StockBroker), which is a highly dynamic application domain, and second, the cultural heritage domain (CHI). In StockBroker we utilize the Semantic Web to meet the increased technological demands emerging in the world of trading. The information that one seeks to represent in such a context relates mostly to news. With the increased popularity of the Web as a broadcasting medium, the latter has also become the main source of information and signals for financial traders. Different large players among multimedia news agencies already provide professional products that come to meet the increased need for tools supporting automated trading. Reuters, for example, provides a range of such products, such as the NewsScope Archive[4] – an annotated archive of news messages aimed at "customers seeking to develop news-based programmatic trading strategies with a comprehensive, machine-readable archive of Reuters global news." NewsScope Real-Time[5] is similar to the 'archive' version of this tool, with the main difference that the annotated feeds are provided in real-time, thus enabling automatic reactions to market-moving events. The StockBroker application is aimed at employing market knowledge from several Web sources for assisting in making better trading decisions.

CHI is a real-life application for Regional Historic Centre Eindhoven (RHCe), an institute that governs cultural heritage related to the region around the city of Eindhoven. The purpose of the CHI application is to open up a large dataset of multimedia documents with the help of additional metadata (in [2] we describe how to acquire this metadata by relating tags to concepts in an ontology). Time is a major dimension in this application, as most objects refer to a time in the past and since time is an essential aspect of the disclosure of the collections. A major issue with respect to time is to describe how we model change and evolution. For example, we represent locations of objects in our datamodel, say the location of the "city hall". If we have a picture that shows the city hall, we

---

[4] http://about.reuters.com/productinfo/newsscopearchive/

[5] http://about.reuters.com/productinfo/newsscoperealtime/

annotate this object with the “city hall” concept. The “city hall” concept has an address and the city hall coordinates so that we can show its location on a map. However, the location of “city hall” changes over time, e.g. because the old building is replaced or municipalities merge.

Possible problems we want to detect and bring to the attention of the RHCe domain experts are, for example, “This picture shows building X and has been annotated with date Y, but we also have in our knowledge base that building X was built on date Z and Z>Y, so this might be an inconsistency”.

The most expressive fragment of the Web Ontology Language (OWL) that maintains desirable computational properties for reasoning is OWL-DL [1]. However, this language only provides limited support for the representation of time. The TOWL language [3], built as an extension of OWL-DL, comes to address this shortcoming by providing support for the representation of time and time-related aspects such as change and state transitions. In this paper, we discuss how the static OWL-DL concept of cardinality changes in a temporal context. More precisely, we discuss the concept of temporal cardinality in the context of the TOWL language, and provide a possible representation of the semantics of temporal cardinality when timeslices are employed for the representation of change, as provided by TOWL.

This paper is structured as follows. In Section 2. we provide an overview of related work. An overview of the TOWL language is given in Section 3. A discussion on the issue of temporal cardinality in the context of the TOWL language is given in Section 4. We discuss our results and give some insights for future work in Section 5.

2 Related Work

In this section, we overview existing works that tackle time-related issues for semantic representation. The representation of temporal semantics has for long been subject of investigation from the research community. It relates to several domains such as artificial intelligence, temporal databases or schema evolution [4]. The rise of the Semantic Web has lead to new attempts to integrate the temporal dimension into semantic languages. Some approaches choose to extend RDF triples (to quadruples) with time based information (e.g. [5]). The setbacks of extending RDF with time however are inflexibility and incompatibility: inflexibility in the sense that for every property a time interval has to be repeated while this might not be necessary by just using additional modeling primitives, incompatibility in the sense that no Semantic Web frameworks support quadruples which is why we would prefer to have a solution that chooses a modeling approach.

Many approaches in literature are built on description logic-based languages such as OWL [1] that have been widely adopted by the community, by extending these languages with temporal semantics in diverse ways [6, 7, 8, 9]. These approaches rely on the notions of instant and time interval to describe temporal events.

In [6], an ontology of time is proposed for the semantic Web. This ontology is now accepted as a W3C working draft. As an extension to this work, the notion of temporal aggregates is introduced in [7]. Temporal aggregates are useful for describing recurrent events such as "every Tuesday". The authors rely on set theory in order to describe temporal aggregates as ordered sets of time intervals.

In [8], the notion of complex temporal events is also introduced as a set of instants or time intervals, but the authors describe the temporal aspect of data with an object-oriented conceptual data model called MADS that is integrated into OWL. Time is represented with the help of MADS-OWL classes that link MADS constructs to OWL descriptions. The practical approach presented by the authors relates to spatio-temporal schema integration. The validation of schemas from a temporal perspective is performed by introducing additional properties and constraints over MADS-OWL classes, thus specifying and restricting the way information can be described. For instance, two classes with the same spatial properties are labeled with a "s_equal" property that expresses spatial equivalence.

In [9] an approach is presented for the description of fluents in OWL. In this context, fluents are nothing more than properties that hold between timeslices (temporal parts of an individual), and indicate what is changing. This approach builds on a perdurantist (or 4D) view where temporal objects are spatially represented over three dimensions, with time as the fourth dimension. This view is opposed to the previously described endurantist (or 3D) view that makes the distinction between objects and events. In a 3D view, objects remain always present and the temporal aspect is represented by different events occurring at certain times.

In the 4D view, temporal objects are described as *spacetime worms* made of slices, where each slice represents the state of the object at a time t. Spacetime worms have properties (fluents) that hold within certain time intervals (i.e. they span over several timeslices). This 4D view simplifies temporal representation as the time dimension is equally important to the other dimensions of the objects. Also, it has the benefit not to affect the use of several OWL constructs such as the inverse or transitivity operators. However, when it comes to cardinality representation, the fluent-based approach appears to be very verbose and suffers from several limitations. For instance, cardinality restrictions over the number of timeslices at a certain time cannot be represented. We tackle this problem and propose appropriate OWL constructs for cardinality description in the remainder of this paper.

3 The TOWL Language

The TOWL language [3] is an extension of OWL-DL that enables the representation of time and temporal aspects such as change and state transitions. It comes to meet some shortcomings of previous approaches, such as [6, 9] that only address this issue to a limited extent and do not seek to enable automated reasoning in a temporal context. The language is designed by means of layers built on top of OWL-DL, each adding to the expressive power of the language. The TOWL layer cake is presented in Figure 1. In what follows, we present an overview of the different layers and their representational characteristics.

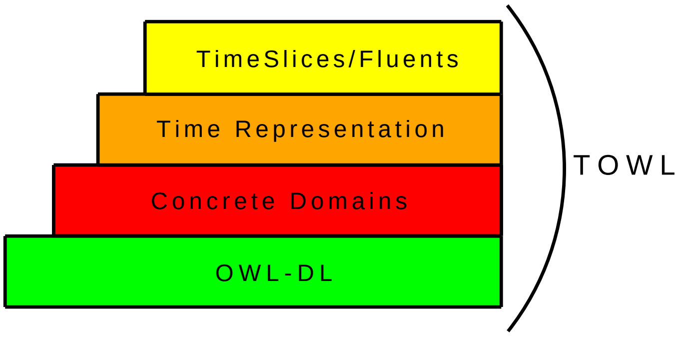

Fig. 1. The TOWL layer cake.

The *concrete domains* layer enables functional role chains in the language, of the DL form shown below.

$$f_1 \circ f_2 \circ ... \circ f_n \circ g$$

Such chains consist of compositions of functional roles f_i with a concrete feature g that points to the concrete domain. Due to their nature, such chains are denoted as concrete feature chains in TOWL, and can be represented in TOWL abstract syntax as:

$$\texttt{ConcreteFeatureChain}(f_1 \;\; f_2 \;\; ... \;\; f_n \;\; g)$$

where $f_1, ..., f_n$ are abstract features and g is a concrete feature pointing to some value in the concrete domain.

Based on such chains and concrete domain predicates, TOWL enables existential and universal quantification, as shown in Figure 2, where u_i is a concrete feature chain and p_d denotes a concrete domain predicate.

DL Notation	TOWL Abstract Syntax
$\exists u_1, u_2.p_d$	`dataSomeValuesFrom`$(u_1 \; u_2 \; p_d)$
$\forall u_1, u_2.p_d$	`dataAllValuesFrom`$(u_1 \; u_2 \; p_d)$

Fig. 2. Existential and universal quantification in TOWL.

The *time representation* layer includes basic representations of time in the form of temporal intervals, as well as Allen's 13 interval relations that may hold between pairs of intervals. The core of this layer consists of a particular type of concrete domain in the form of a constraint system [10].

The *timeslices/fluents* layer, building upon the approach in [9], enables the representation of change and state transitions. This is achieved by extending the OWL-DL syntax and semantics to include timeslices and fluents. Additionally, the two other TOWL layers presented in this section contribute to the representational power of this layer by enabling more specific semantics at a concrete level. It thus becomes possible to express the fact that fluents may only connect timeslices that hold over the same interval [3].

Timeslices, as employed for the purpose of the TOWL language, are aimed at representing some concept (static individual) over a certain period of time (interval). Two main properties describe timeslices, namely the *timeSliceOf* property, that connects the timeslice to the static concept it represents, and the *time* property, indicating the time interval across which the timeslice holds. Additionally, fluent properties are employed to describe what is changing, i.e., what only holds true for a bounded period of time.

Fluents are the properties that hold for timeslices, and thus indicate what is changing. The semantics of fluents allows timeslices to be related to other timeslices, or to datatypes as attribute values. This differentiation in the range of fluents comes to address the proliferation of objects, inherent to the current approach. Hence, timeslices are created each time something is changing. However, creating timeslices for concrete values is deemed meaningless in the current context. Thus, for this case only, a timeslice may be connected to something else than another timeslice, namely a datatype. It should also be noted that the semantics of fluents enforce that the connected timeslices must invariably hold over the same time interval.

Finally, it should be remarked that the issue of cardinality has a strong relation with the identity of timeslices. For this purpose, two timeslices (TS_1 and TS_2) are defined as identical ($\mathtt{eq_{TS}}$), if the following holds true:

$$\begin{aligned}(TS_1, TS_2).\mathtt{eq_{TS}} \equiv\ & (TS_1.\mathtt{time}, TS_2.\mathtt{time}).\mathtt{equal} \wedge \\ & \wedge (TS_1.\mathtt{timeSliceOf}, TS_2.\mathtt{timeSliceOf}).\mathtt{sameAs}\end{aligned}$$

4 Temporal Cardinality

The concept of temporal cardinality becomes relevant in the context of timeslices, as employed for the representation of change in the TOWL language. In what follows, we present an illustrative example of how the concept of cardinality changes semantics in a temporal context, and present a possible formalization of this concept.

The timeslices approach introduced in TOWL allows for the representation of change through the creation of timeslices. These temporal parts of individuals are related by fluents - the properties that connect timeslices and thus indicate what is changing and how. However, some of the OWL-DL constructs lose their meaning in this context of change, mostly due to the static semantics that pinpoint their definition. One such concept is the *cardinality* construct, present in OWL-DL in three closely related forms [11]:

- *minCardinality*: if stated to have the value a on a property P, with respect to a class C, then any instance of C will be related through P to **at least** a individuals (of which the type may further be restricted by the *range* of P);
- *maxCardinality*: if stated to have the value a on a property P, with respect to a class C, then any instance of C will be related through P to **at most** a individuals (of which the type may further be restricted by the *range* of P);
- *cardinality*: if stated to have the value a on a property P, with respect to a class C, then any instance of C will be related through P to **exactly** a individuals (of which the type may further be restricted by the *range* of P). In other words, both the *minCardinality* of a and the *maxCardinality* of a are simultaneously satisfied.

Moving to a temporal context, an extension of the static concept of cardinality may be envisioned in the sense that, at any point in time, only a restricted number of timeslices may describe a concept. In other words, temporal cardinality is meant to restrict the number of timeslices that may overlap, at any point in time for the same individual. These restrictions should be stated on fluents, with respect to static individuals whose timeslices are described by those fluents.

One example of a context where such restrictions are meaningful comes from the financial domain, and is also discussed in [9]. The example is built around the *Company*, *Person*, and *hasCEO* entities. In the current context we seek to represent the fact that, at any point in time, a company must have exactly 1 Chief Executive Officer (CEO), in the form of a person. This restriction targets two situations:

- *fluent cardinality*: the (static) cardinality of the *hasCEO* fluent should be equal to one, following the description above. In other words, the *hasCEO* fluent must be associated to exactly one timeslice of a static individual of type *Person* each time it is defined for a timeslice of an individual of type *Company*. This issue can easily be addressed by employing the OWL-DL *cardinality* construct, as done in [9];
- *overlapping timeslices*: the (temporal) cardinality of the *hasCEO* fluent should be equal to 1. In other words, at any point in time, the *hasCEO* relation must be described by one timeslice of a static individual of type *Person* [6].

These two situations are graphically illustrated in Figure 3, for the CEO example discussed above. Here *Company1-TS1* and *Company1-TS2* are timeslices of the static *Company1* individual, and *Person1-TS1* and *Person2-TS1* are timeslices of the *Person1* and *Person2* static individuals, respectively.

---

[6] It should be noted that the TOWL semantics enforce equal intervals for timeslices connected by a fluent.

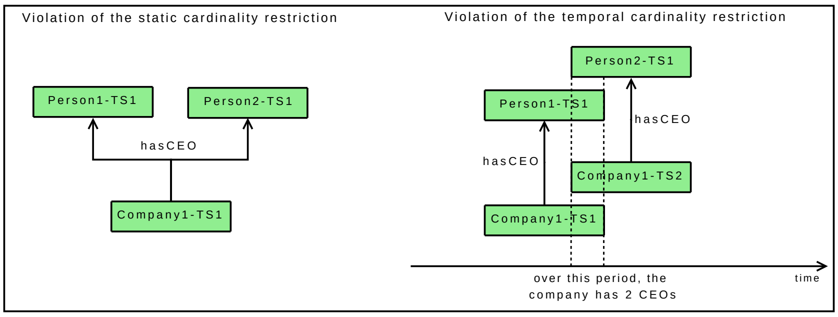

Fig. 3. Static vs. Temporal Cardinality.

Extending the static concept of cardinality to a temporal setting, we introduce the following constructs:

- *temporalMinCardinality*: the equivalent of the *minCardinality* OWL-DL construct in a temporal setting;
- *temporalMaxCardinality*: the equivalent of the *maxCardinality* OWL-DL construct in a temporal setting;
- *temporalCardinality*: the equivalent of the *cardinality* OWL-DL construct in a temporal setting.

A more formal description for each of the introduced constructs is provided in definitions 1 through 3.

Definition 1 (temporalMinCardinality)
Given a fluent property f, a class C, an individual i of type C and a value a such that $a \in \mathbb{N}$, we represent by $temporalMinCardinality(f, a)$ the restriction on f with respect to timeslices of i for which f is defined that, at any point in time, any timeslice of i is described by **at least** a timeslices through f.

Definition 2 (temporalMaxCardinality)
Given a fluent property f, a class C, an individual i of type C and a value a such that $a \in \mathbb{N}$, we represent by $temporalMaxCardinality(f, a)$ the restriction on f with respect to timeslices of i for which f is defined that, at any point in time, any timeslice of i is described by **at most** a timeslices through f.

Definition 3 (temporalCardinality)
Given a fluent property f, a class C, an individual i of type C and a value a such that $a \in \mathbb{N}$, we represent by $temporalCardinality(f, a)$ the restriction on f with respect to timeslices of i for which f is defined that, at any point in time, $temporalMinCardinality(f, a)$ and $temporalMaxCardinality(f, a)$ simultaneously hold.

We next focus on giving a more formal semantic representation of the three types of temporal cardinality we introduced. In achieving this, we first define a function g that, given a fluent f, a static individual i and a point in time t, returns the number of timeslices of different individuals j holding at t, for which f is explicitly defined and linked from a timeslice of i that also holds at t. The result of this function is a natural number, obtained by counting the unique individuals returned by the $g_{(f,i,t)}$ function.

$$\begin{aligned} g_{(f,i,t)} = |\{j \in C^{\mathcal{I}} \mid \exists x, y, s, e \texttt{ s.t. } x, y \in TS^{\mathcal{I}} \wedge (x,i) \in \texttt{timeSliceOf}^{\mathcal{I}} \wedge \\ \wedge\ (y,j) \in \texttt{timeSliceOf}^{\mathcal{I}} \wedge (x,y) \in f^{\mathcal{I}} \wedge s = \texttt{start}(\texttt{time}(y)) \wedge \\ \wedge\ e = \texttt{end}(\texttt{time}(y)) \wedge s \leq t \leq e\}| \end{aligned}$$

Moving to Description Logics, the semantics of the three constructs relating to temporal cardinality can be represented as follows, where $\geq_{\mathcal{T}}$, $\leq_{\mathcal{T}}$ and $=_{\mathcal{T}}$ denote *temporalMinCardinality*, *temporalMaxCardinality* and *temporalCardinality*, respectively, a, f and t preserve their meaning as previously, and C denotes a concept.

$$\begin{aligned} (\geq_{\mathcal{T}}\ a\ f)^{\mathcal{I}} &= \{x \in TS^{\mathcal{I}} \mid \forall t\ \exists i,\ i \in C^{\mathcal{I}} \wedge (x,i) \in \texttt{timeSliceOf}^{\mathcal{I}} \wedge g_{(f,i,t)} \geq a\} \\ (\geq_{\mathcal{T}}\ a\ f)^{\mathcal{I}} &= \{x \in TS^{\mathcal{I}} \mid \forall t\ \exists i,\ i \in C^{\mathcal{I}} \wedge (x,i) \in \texttt{timeSliceOf}^{\mathcal{I}} \wedge g_{(f,i,t)} \leq a\} \\ (=_{\mathcal{T}}\ a\ f)^{\mathcal{I}} &= (\geq_{\mathcal{T}}\ a\ f)^{\mathcal{I}} \cap (\leq_{\mathcal{T}}\ a\ f)^{\mathcal{I}} \end{aligned}$$

Based on this definitions, a syntactic and semantic extension for TOWL can be proposed, that enable the representation of temporal cardinality when change and state transitions are represented by employing TOWL timeslices and fluents, as in the example presented in Figure 3.

5 Conclusions & Future Work

This paper provides a discussion of temporal cardinality in the context of the TOWL language. One of the challenges of representing such a concept is related to how the language enables the representation of change, i.e., by means of timeslices. This representation alone poses difficulties due to the fact that, unlike in the case of static cardinality, one should consider the case of overlapping timeslices in addition to just the property (or in this case fluent) alone. However, the representational power provided by the TOWL layers proves to be sufficient for enabling such a construct. Feature chains, as enabled by the *concrete domains* TOWL layer, and Allen's relations, as enabled by the *time representation* TOWL layer, allow the introduction of a function that enables counting the number of overlapping timeslices at some point in time, given a fluent and a static concept. Based hereon, we have shown that the static concept of OWL-DL cardinality can be extended with a temporal dimension, and more particularly in the context of the TOWL language.

Currently, the specification of the TOWL language is being finalized. Part of the future work will be aimed at building implementations for querying and reasoning support for TOWL. Furthermore, we aim at incorporating TOWL in a number of practical Semantic Web applications, such as StockBroker and CHI, as presented in the introductory part of this paper.

Acknowledgement

The first author is supported by the EU funded IST STREP Project FP6 - 26896: Time-determined ontology-based information system for realtime stock market analysis. More information is available on the official website[7] of the project.

References

[1] Patel-Schneider, P., Hayes, P., Horrocks, I.: Web ontology language (OWL) abstract syntax and semantics. http://www.w3.org/TR/owl-semantics/ (2004)

[2] van der Sluijs, K., Houben, G.J.: Relating user tags to ontological information. In: Proceedings of the 5th International Workshop on Ubiquitous User Modeling (UbiqUM'2008), Workshop at the 2008 International Conference on Intelligent User Interfaces (IUI2008). (2008)

[3] Milea, V., Frasincar, F., Kaymak, U.: Knowledge engineering in a temporal semantic web context. In: The Eighth International Conference on Web Engineering (ICWE 2008), IEEE Computer Society Press (2008)

[4] Roddick, J.F., Patrick, J.D.: Temporal semantics in information systems - a survey. Information Systems **17**(3) (1992) 249–267

[5] Pugliese, A., Udrea, O., Subrahmanian, V.S.: Scaling RDF with time. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 605–614

[6] Hobbs, J., Pan, F.: An ontology of time for the semantic web. ACM Transactions on Asian Language Information Processing **3**(1) (2004) 66–85

[7] Pan, F., Hobbs, J.R.: Temporal aggregates in OWL-Time. In Russell, I., Markov, Z., eds.: FLAIRS Conference, AAAI Press (2005) 560–565

[8] Sotnykova, A., Cullot, N., Vangenot, C.: Spatio-temporal schema integration with validation: A practical approach. In Meersman, R., Tari, Z., Herrero, P., Méndez, G., Cavedon, L., Martin, D., Hinze, A., Buchanan, G., Pérez, M.S., Robles, V., Humble, J., Albani, A., Dietz, J.L.G., Panetto, H., Scannapieco, M., Halpin, T.A., Spyns, P., Zaha, J.M., Zimányi, E., Stefanakis, E., Dillon, T.S., Feng, L., Jarrar, M., Lehmann, J., de Moor, A., Duval, E., Aroyo, L., eds.: OTM Workshops. Volume 3762 of Lecture Notes in Computer Science., Springer (2005) 1027–1036

[9] Welty, C., Fikes, R.: A reusable ontology for fluents in OWL. In: Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006), IOS Press (2006) 226–336

[10] Lutz, C., Milicic, M.: A tableau algorithm for description logics with concrete domains and general tboxes. Journal of Automated Reasoning **38**(1–3) (2007) 227–259

[11] Smith, M., Welty, C., McGuinness, D.: OWL web ontology language guide. http://www.w3.org/TR/owl-guide/ (2004)

---

[7] http://www.towl.org