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

Viorel Milea[1], Flavius Frasincar[1], Uzay Kaymak[1], and Tommaso di Noia[2]

[1] Erasmus University Rotterdam, Netherlands
Burgemeester Oudlaan 50, 3062 PA Rotterdam
{milea, frasincar, kaymak}@few.eur.nl
[2] Politecnico Di Bari, Italy
via E. Orabona, 4 - 70125 Bari
t.dinoia@poliba.it

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 Logic 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. We also present a proof of concept for the developed approach. This consists of 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.

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, currently the language offers only little support for representing temporal information in ontologies. 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 (DL) and enables the representation of time and time-related aspects such as change in ontologies.

The approach we take in this paper focuses on two main goals that need to be achieved when talking about time in ontologies: representing *time* itself (in the form of dates, times, etc.) and representing *temporal aspects* (changing individuals, temporal knowledge, etc.). The 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 combining the latter for obtaining new expressions [1]. Representing time in such a manner allows the use of the 13 Allen relations [2] 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 [1]. 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 (perdurants) in a unique manner at different points in time. The actual individual is then nothing more than the sum of its (temporal) parts. Following the approach taken in [3], we implement this representation by making use of time slices (the temporal parts of an individual) and fluents (properties that hold between timeslices).

The remainder of this paper is organized as follows. In section 2 we present a detailed description of the TOWL language. An extended example of the possible use(s) of the language is presented in section 3. Section 4 provides some concluding remarks and suggestions for further research.

2 The TOWL Language

The focus of this section is on introducing the concepts necessary for describing time and change in ontologies. The resulting ontology language, TOWL, is a symbiosis of the $\mathcal{SHOIN}(\mathcal{D})$ description logic and its extension. This extension consists of a concrete domain that represents time and the perdurantist approach towards modeling the changing aspects of entities through time. The first part of this section consists of a more general overview of the TOWL language, while in the second part the afore mentioned language is formally introduced by means of describing its syntax and semantics.

2.1 Introducing TOWL

The TOWL ontology language is intended to be an extension of the current Web Ontology Language (OWL), and thus an extension of the $\mathcal{SHOIN}(\mathcal{D})$ description logic. The thus obtained language allows the representation of knowledge beyond the constraints of a static world, enabling the representation of dynamic entities that change (some) traits through time by expressing them as perdurants. A number of powerful time relations/operators, as described by Allen [4], are available for the purpose of reasoning with time.

The Concrete Domain: Intervals and Allen's Relations
A concrete representation of time is available in TOWL through *time intervals*, modeled as a concrete domain. A time interval is defined as a pair of *time points* and is available in TOWL through the *towl:TimeInterval* class. The ends

of an interval, represented as time points, are available in TOWL through the *towl:TimePoint* class. Following the reasoning in [5], time points are modeled as reals, and a time interval I_t can thus formally be defined as a pair (t_1, t_2) with $t_1, t_2 \in \mathbb{R}$. A proper interval is then defined as a pair (t_1, t_2) where $t_1 < t_2$. Intervals are related by Allen's thirteen relations: equals, before, after, meets, met-by, overlaps, overlapped-by, during, contains, starts, started-by, finishes, finished-by. These relations are *exhaustive* and *mutually exclusive* [5], i.e. for each pair of intervals there exists at least one relation holding true between them and, respectively, for each pair of intervals there exists at most one relation that holds true amongst them. It should be noted that all of Allen's 13 relations can be expressed in terms of the endpoints of intervals and the set of predicates $\{<, =\}$, the only two predicates of the concrete domain, that apply to all reals.

Perdurants in TOWL Ontologies

The concrete domain approach for representing time provides a good foundation towards representing change in ontologies. For this purpose the following TOWL concepts are introduced following the reasoning in [3]: *towl:TimeSlice, towl:tsTimeSliceOf, towl:fluentObjectProperty, towl:fluentDatatypeProperty* and *towl:tsTime*. The temporal parts of a perdurant are described as timeslices, and each of these timeslices is an individual of type *towl:TimeSlice*. They can be regarded as snapshots (slides) of an individual at a particular moment (interval) in time. The period of time for which each individual timeslice holds true is described as a pair (t_1, t_2) from the concrete domain and is associated to the timeslice through the *towl:tsTime* property. In case the temporal information does not regard an interval, but a single time point, then this time point can also be associated to the timeslice through the *towl:tsTime* property. The individual that is described by each particular timeslice over a time interval I_t, i.e. the perdurant, is indicated by means of the *towl:tsTimeSliceOf* property. Finally, timeslices are connected through (subproperties of) the *towl:fluentObjectProperty* relation while the association between timeslices and literals is indicated by the *towl:fluentDatatypeProperty.*

Advantages of this Approach

The advantages of the approach presented here are twofold. First, unlike previous approaches [3, 6–8], the current approach provides the means to represent both time in its quantitative nature, as well as temporal entities. Each of the previous approaches mentioned here focusses on only one of these aspects, such as [7] 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 [8] 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.).

2.2 TOWL: Syntax and Semantics

In this subsection we formally introduce the syntax and semantics of the TOWL language. This presentation only includes the additional syntax and semantics of TOWL. Figure 1 gives an overview hereof. A further specification of the TOWL concepts is given in Figure 2, where the extensional semantics of the newly defined language is presented. Finally, the OWL schema of TOWL is presented in Figure 3, in OWL abstract syntax.

$C, D \longrightarrow TS \mid$	(towl:TimeSlice)
$TE \mid$	(towl:TimeEntity)
$I_t \mid$	(towl:TimeInterval)
$P_t \mid$	(towl:TimePoint)
$\ell \mid$	(rdfs:Literal)
$TS \sqsubseteq \forall FOP.TS \mid$	(towl:fluentObjectProperty)
$TS \sqsubseteq \forall FDP.\ell \mid$	(towl:fluentDatatypeProperty)
$TS \sqsubseteq \forall TSO_{ts}.C \mid$	(towl:tsTimeSliceOf)
$TS \sqsubseteq \forall T_{ts}.(I_t \sqcup P_t) \mid$	(towl:tsTime)
$I_t \sqsubseteq \forall S_{ti}.P_t \mid$	(towl:tiStart)
$I_t \sqsubseteq \forall E_{ti}.P_t \mid$	(towl:tiEnd)
$P_t \sqsubseteq \forall D_{tp}.\ell \mid$	(towl:tpDate)

Fig. 1. TOWL syntax rules

functional properties :	$TSO_{ts}, T_{ts}, S_{ti}, E_{ti}, D_{tp}$.
$(TS)^{\mathcal{I}}$	$= \{a \in \Delta^{\mathcal{I}} \mid TSO_{ts}(a) \in \Delta^{\mathcal{I}}\}$
$(I_t)^{\mathcal{I}}$	$= \{a \in \Delta^{\mathcal{I}} \mid S_{ti}(a) = x_1,\ E_{ti}(a) = x_2$ and $D_{tp}(x_1) < D_{tp}(x_2)\}$
$(P_t)^{\mathcal{I}}$	$= \{a \in \Delta^{\mathcal{I}} \mid D_{tp}(a) \in \ell\}$
$(\forall T_{ts}.I_t)^{\mathcal{I}}$	$= \{a \in (TS)^{\mathcal{I}} \mid \forall b \in \Delta^{\mathcal{I}} : (a,b) \in (T_{ts})^{\mathcal{I}} \rightarrow b \in (I_t)^{\mathcal{I}}\}$
$(\forall T_{ts}.P_t)^{\mathcal{I}}$	$= \{a \in (TS)^{\mathcal{I}} \mid \forall b \in \Delta^{\mathcal{I}} : (a,b) \in (T_{ts})^{\mathcal{I}} \rightarrow b \in (P_t)^{\mathcal{I}}\}$
$(\forall FOP.TS)^{\mathcal{I}}$	$= \{a \in (TS)^{\mathcal{I}} \mid \forall b \in ((TS)^{\mathcal{I}} \setminus \{a\}), \exists t_1, t_2 \in (I_t)^{\mathcal{I}} :$ $(a,b) \in (FP)^{\mathcal{I}}, a \in TI.t_1, b \in TI.t_2 \rightarrow t_1 = t_2\}$
$(\forall FDP.\ell)^{\mathcal{I}}$	$= \{a \in (TS)^{\mathcal{I}} \mid \forall (a,b) \in (FDP)^{\mathcal{I}} \rightarrow b \in \ell\}$
$(\forall TSO_{ts}.C)^{\mathcal{I}}$	$= \{a \in (TS)^{\mathcal{I}} \mid \forall b \in \Delta^{\mathcal{I}} : (a,b) \in (TSO_{ts})^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$

Fig. 2. TOWL semantics

When compared with the original fluents approach [3] that stands at the basis of the approach we present here, a number of additional features have been incorporated in the language, thus adding to its flexibility and expressiveness. One of these features is allowing the association of time slices not only with intervals, but also with time points, a representation essential for systems as the one described in this paper. Additionally, we make a distinction between two different types of fluent properties: datatype fluent properties, that point to objects of type rdfs:Literal, and object fluent properties that point to objects, i.e. actual timeslices of individuals present in the ontology. The case of the datatype fluent property is special in that it does not require a timeslice of a specific type of literal, but it may point to the actual value itself.

3 An Extended Example

In this section we present an example of how the TOWL language can provide for added value. For this purpose we sketch a system that uses last minute news for the generation of buy/hold/sell signals based on market consensus. The system consists of five parts, reflecting the essential components of the system: 1) the financial TOWL ontology - the ontology used to store all knowledge relevant to the system, 2) information extraction, the component that extracts the relevant knowledge from news messages, 3) ontology update - the actual updating of the ontology with the new information, 4) query evaluation - an important part of the process of answering queries regarding the current state of the world as described in the financial TOWL ontology and, finally, 5) the actual application that generates buy/hold/sell signals on an on-demand basis derived from the domain knowledge modeled in the ontology.

3.1 The Financial TOWL Ontology

For the purpose of this example a simple TOWL financial ontology has been developed. The schema of this ontology consists of a *Company* class, the class of all companies that are of relevance in the ontology, a *CompanyAdvice* class that denotes the advices issued by experts regarding companies, and a class *AdviceType* defined by means of its only three instances *buy*, *hold* and *sell* - the actual recommendation(s) of the expert for some company. Additionally, a number of properties have been defined that further specify the meaning of classes. The property *hasName* indicates the actual name of the individuals of type *Company*, the property *adviceType* relates all individuals of type *CompanyAdvice* to individuals of type *AdviceType*, while the property *priceTarget12* indicates, for all individuals of type *CompanyAdvice*, the expected price over 12 months, as formulated in a particular advice. Finally, two fluent properties 'connect' timeslices of individuals of type *Company* to corresponding timeslices of individuals of type *CompanyAdvice*. The *adviceIssuedBy* property indicates the company that has issued a particular advice and the *adviceIssuedFor* property indicates the company for which the particular advice has been issued. A formal representation of this ontology is given in Figure 4, in OWL abstract syntax.

```
Ontology(TOWL
  Class(TimeSlice)
  Class(TemporalEntity)
  Class(TimeInterval partial TemporalEntity)
  Class(TimePoint partial TemporalEntity
   restriction(complementOf(TimeInterval)))
  DisjointClasses(TimeSlice TemporalEntity)

  ObjectProperty(fluentObjectProperty Symmetric)
   domain(TimeSlice)
   range(TimeSlice))

  DatatypeProperty(fluentDatatypeProperty Symmetric)
   domain(TimeSlice)
   range(rdfs:Datatype))

  ObjectProperty(tsTimeSliceOf Functional
   domain(TimeSlice)
   range(complementOf(unionOf(TimeSlice TemporalEntity rdfs:Literal))))

  ObjectProperty(tsTime Functional
   domain(TimeSlice)
   range(TemporalEntity))

  DatatypeProperty(tiStart Functional
   domain(TimeInterval)
   range(TimePoint))

  DatatypeProperty(tiEnd Functional
   domain(TimeInterval)
   range(TimePoint))

  DatatypeProperty(tpDate Functional
   domain(TimePoint)
   range(xsd:dateTime)))
```

Fig. 3. OWL Schema of TOWL

3.2 Information Extraction

The information extraction phase is responsible for providing the system with the necessary input in the form of processed knowledge from news messages. For this example we focus on a particular type of news - analyst recommendations - in the form of buy/hold/sell signals, sometimes accompanied by a price target. In this example we use the following three news messages.

```
Ontology(finTOWL
  Class(Company)
  Class(CompanyAdvice)
  EnumeratedClass(AdviceType buy hold sell)
  DisjointClasses(Company CompanyAdvice AdviceType)

  DatatypeProperty(hasName
   domain(Company)
   range(xsd:String))

 ObjectProperty(adviceType Functional
   domain(CompanyAdvice)
   range(AdviceType))

 ObjectProperty(adviceIssuedBy super(fluentObjectProperty) Functional
   domain(restriction(tsTimeSliceOf(allValuesFrom Advice)))
   range(restriction(tsTimeSliceOf(allValuesFrom Company))))

  ObjectProperty(adviceIssuedFor super(fluentObjectProperty) Functional
   domain(restriction(tsTimeSliceOf(allValuesFrom Advice)))
   range(restriction(tsTimeSliceOf(allValuesFrom Company))))

  DatatypeProperty(priceTarget12 Functional
   domain(CompanyAdvice)
   range(xsd:double)))
```

Fig. 4. The Financial TOWL Ontology

News1

(MarketAdvices.com) New York(7-17-2006) - Mark Hebeka of Standard & Poors reiterates his buy recommendation for the American bank and insurance company Citigroup. The 12-months target price for Citigroup is reiterated at 55 USD.

News2

(MarketAdvices.com) New York (1-19-2007) - The analysts of Goldman Sachs reiterate their hold recommendation for the American bank and insurance company Citigroup (ISIN: US1729671016 / Symbol: C). The 12-months target price for Citigroup is 59 USD.

News3

(MarketAdvices.com) New York (1-29-2007) - Analyst Frank Braden of Standard & Poors reiterates his buy recommendation for the American bank and insurance company Citigroup (ISIN:US1729671016 / Symbol: C). A price target was not provided.

For all of the three news messages, a feature selection has been performed, and the selected features have been highlighted in the examples above. An analysis of the *News1* example provides the following: *17/7/2006*, the date when the advice was issued and thus the date starting at which the advice holds true, *Standard and Poor's*, the company that has issued the advice, *Citigroup*, the company for which the advice was issued, *buy*, the advice type, and *55 USD*, the value of the *12-months target price* for one Citigroup share according to the expectation of the analysts at Standard and Poor's. This process is repeated for each of the news messages.

3.3 Knowledge Base Update

Having performed the extraction phase, the resulting knowledge relevant to the domain is modeled explicitly in the knowledge base (KB). The way in which this can be achieved is presented below, in OWL abstract syntax, for each of the news messages previously processed. One assumption is that static knowledge regarding the three companies involved is already present in the ontology, and modeled as presented in Figure 5.

```
Ontology(finTOWL
  Individual(iCitigroup
   type(Company)
   value(name "Citigroup"^^xsd:String))

  Individual(iStandardPoors
   type(Company)
   value(name "Standard and Poor's"^^xsd:String))

  Individual(iGoldmanSachs
   type(Company)
   value(name "Goldman Sachs"^^xsd:String))
```

Fig. 5. Static individuals of finTOWL

News1

For the purpose of representing the information contained in the first news message, a number of timeslices have to be created of the individuals StandardPoors, Citigroup and CompanyAdvice. This results in three timeslices, one for each of the aforementioned individuals: *iSandP_TS1*, *iCitigroup_TS1* and finally *iCitiAdvice1_TS1* for the CitiAdvice1 individual. The beginning of the period in which the advice holds true is modeled as an individual of type TimePoint that contains the relevant xsd:dateTime object: *iTP1*. Finally, the timeslice of the advising company, iStandardPoors_TS1 is associated with the timeslice of the issued advice, iCitiAdvice_TS1, through the adviceIssuedBy property. Similarly, the timeslice of the company that received the advice, iCiti_TS1 is associated with the timeslice of the received advice - iCitiAdvice_TS1, through the adviceIssuedFor property. It should be noted that at the moment this knowledge became available, no additional information is available on the duration of this advice, hence only the starting moment of this advice has been modeled as an individual of type TimePoint. This representation is summarized in Figure 6, where a model of the *News1* news message is given in OWL abstract syntax. As soon as a new advice is issued by Standard and Poor's for the company Citigroup, the duration of the new advice will be known and will equal the time between the already known starting point (iTP1) and the time point at which the new advice has been issued. Thus, the tsTime property will not have an argument of type TimePoint, but of type TimeInterval as soon as this information becomes available. This is the case after the issuing of a new advice by Standard and Poor's for Citigroup, as in the *News3* news message. The concrete changes in the KB, that ideally are automatically performed, are illustrated in Figure 7.

3.4 Query Evaluation

The most basic operations that the TOWL language enables become evident through the queries that may be posed upon the system. These queries form an essential part of the rules used to determine the final output (the buy/hold/sell signals). In this section we present some examples of queries and how the results of these queries can be inferred by the system.

A number of four query examples are presented below, where the most relevant part referring to time or some aspect of time is in bold:

IEX_1 Was any advice for Citigroup issued **after** Goldman Sachs issued an advice for Citigroup on January 19th, 2007?

IEX_2 Was any advice for Citigroup issued **while** the Goldman Sachs advice for Citigroup issued on January 19th, 2007 was holding?

IEX_3 When were **the last two** buy advices issued for Citigroup?

IEX_4 Was there any positive (buy) advice for Citigroup **in** January 2007?

The first query example, IEX_1, relates to comparing individuals of type TimePoint, whether they individuals are present individually or as part of a TimeInterval object, i.e. as argument of the tiStart or tiEnd of an individual of type TimeInterval. First, the specific advice issued by Goldman Sachs on January 19th 2007 should be identified as the individual iCitiAdvice2_TS1 of type TimeSlice. Next, the moment in time when this advice was issued must be retrieved; this moment in time is the argument of the tpDate property of iCitiAdvice2_TS1 in case this property is defined, or the argument of the tiStart property in case this property is defined. Next, all advices that have been issued after the date of January 19th, 2007 must be retrieved. These advices are individuals of type TimeSlice for which the property tsTimeSliceOf has an argument of type CompanyAdvice. The property adviceIssuedFor of the previously selected individuals must point to a timeslice of Citigroup. Additionally, one of the properties tpDate (in the case of a TimePoint) or tiStart (in the case of a TimeInterval) must be defined, and the argument of this property should be strictly larger than January 19th, 2007. If the set of individuals satisfying all these constraints is not empty, then the answer to this query is positive.

A similar procedure can be applied for answering the remaining query examples (2 through 4) by trying to find a set of individuals that satisfies the constraints specified in the query. A special case is the query example *IEX_3*, where the answer to this query is not of Boolean type, but consists of a set of date objects. In order to answer this query, the set of buy advices for Citigroup must be selected and ordered according to the date these advices were issued. Then, the arguments of the tpDate properties or of the tiStart properties should be returned.

3.5 Application

At application level, the basic query examples described in the previous section can be combined for the purpose of generating buy/hold/sell signals based on predefined rules. A simple example of such a rule is given below.

APR_1 "If a buy advice is issued for a company while another buy advice holds, then buy".

A possible way of firing this rule reduces to checking, each time a new buy advice is issued for some company X, whether another buy advice holds true for X. If this is the case, then generate a buy signal. Of course, this rule does not say anything about the situation in which there are already two buy advices still holding true for company X when a third buy advice is issued, but in this case we just assume that the already generated (buy) signal is not changed, or perhaps it is reiterated.

The trading signal generation system based on TOWL can also be used to generate buy/hold/sell signals based on more complex rules, such as in rule *APR_2*. Here we assume that advices are of the form -1, 0, 1 for sell, hold and buy, respectively, and the final result returned by $SYST_{ADV}$ can be rounded to the nearest whole number. The m, n and p variables represent prespecified weights specific to each company that issued an advice.

APR_2 "Companies X, Y, Z issued advices A, B, C for company W. For each point in time the advice issued by the system, is a weighted average of the three

```
Individual(iStandardPoors_TS1
 type(TimeSlice)
 value(tsTimeSliceOf StandardPoors)
 value(tsTime iTP1))

Individual(iCitigroup_TS1
 type(TimeSlice)
 value(tsTimeSliceOf iCitigroup)
 value(tsTime iTP1))

Individual(iTP1
 type(TimePoint)
value(tpDate "17/7/2006"^^xsd:date))

Individual(iCitiAdvice1
 type(CompanyAdvice)
 value(adviceType buy)
 value(priceTarget12 "55"^^xsd:double))

Individual(iCitiAdvice1_TS1
 type(TimeSlice)
 value(tsTimeSliceOf iCitiAdvice1)
 value(tsTime iTP1)
 value(adviceIssuedBy iStandardPoors_TS1)
 value(adviceIssuedFor iCitigroup_TS1))
```

Fig. 6. A TOWL representation of *News1*

```
Individual(iTI1
 type(TimeInterval)
 value(tiStart iTP1)
 value(tiEnd iTP3))

Individual(iStandardPoors_TS1
 value(tsTime iTI1))

Individual(iCitigroup_TS1
 value(tsTime iTI1))

Individidual(iCitiAdvice1_TS1
 value(tsTime iTI1))
```

Fig. 7. Ontology update after *News3* has been issued

individual advices: $SYST_{ADV} = (mA + nB + pC)/(m + n + p)$ if all three companies have an advice for W at time T".

4 Conclusions and Further Research

This paper presents a new ontology language that allows the expression of time and change in ontologies: the TOWL language. Two aspects of time are deemed essential: the actual/concrete time and the concept of change. The TOWL language offers the possibility of representing both these aspects in ontologies and offers a consistent way of expressing the changing aspect of the entities in some world by means of perdurants. Although the concept of concrete domains or fluents is not new, the symbiosis between the two is unique in representing time and change in KR languages. Moreover, the original approach described in [3] has been extended towards added expressivity and increased flexibility, while the perdurants syntax has a basic underlying semantics. There are however a number of issues requiring attention in further research, such as cardinality restrictions on fluents with regard to overlapping timeslices.

Acknowledgement

The authors are supported by the EU funded IST STRP Project FP6 - 26896: Time-determined ontology-based information system for realtime stock market analysis. More information is available on the official website[3] of the project.

References

[1] Artale, A., Franconi, E.: A survey of temporal extensions of description logics. Annals of Mathematics and Artificial Intelligence **30** (2000) 171–210

[2] Allen, J.F., Ferguson, G.: Actions and events in interval temporal logic. Journal of Logic Computation **4** (1994) 531–579

[3] Welty, C., Fikes, R., Makarios, S.: A reusable ontology for fluents in owl. In: Proceedings of FOIS, IOS Press (2006) 226–236

[4] Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26** (1983) 832–843

[5] Lutz, C.: The Complexity of Description Logics with Concrete Domains. PhD thesis, RWTH Aachen (2002)

[6] Artale, A., Franconi, E.: Introducing temporal description logics. In: TIME '99: Proceedings of the Sixth International Workshop on Temporal Representation and Reasoning, Washington, DC, USA, IEEE Computer Society (1999) 2

[7] Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. Technical Report RR-91-10, Deutsches Forschungszentrum für Künstliche Intelligenz (1991)

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

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