

# Temporal Reasoning for Supporting Temporal Queries in OWL 2.0

Sotiris Batsakis, Kostas Stravoskoufos, and Euripides G.M. Petrakis

Department of Electronic and Computer Engineering  
Technical University of Crete (TUC)  
Chania, Greece

e-mail:batsakis@softnet.tuc.gr,{kgstravo,petrakis}@intelligence.tuc.gr

**Abstract.** We propose an approach for reasoning and querying over temporal information in OWL ontologies. Representing both qualitative temporal (i.e., information whose temporal extents are unknown such as “before”, “after” for temporal relations) in addition to quantitative information (i.e., where temporal information is defined precisely e.g., using dates) is a distinctive feature of the proposed ontology. Qualitative representations are very common in natural language expressions such as in free text or speech and can be proven to be valuable in the Semantic Web. Reasoning rules applying over temporal relations, infer implied relations, detect inconsistencies and retain soundness, completeness and tractability over the supported sets of relations using path consistency. Temporal representations are defined on time instants rather than on intervals (as it is typical in the literature), resulting into simpler yet equivalent representations. A SPARQL-based temporal query language capable of exploiting the characteristics of the underlying representation is also implemented and discussed.

## 1 Introduction

Ontologies offer the means for representing high level concepts, their properties and their interrelationships. Dynamic ontologies will in addition enable representation of information evolving in time. Representation of dynamic features calls for mechanisms allowing for uniform representation of the notions of time (and of properties varying in time) within a single ontology. Methods for achieving this goal include (among others), temporal description logics [4], concrete domains [6], property labeling [5], versioning [2], named graphs [8], reification<sup>1</sup> and the 4D-fluents (perdurantist) approach [3].

Welty and Fikes [3] showed how quantitative temporal information (i.e., in the form of temporal intervals whose start and end points are defined) and the evolution of concepts in time can be represented effectively in OWL using the so called “4D-fluents approach”. In [1] this approach was extended in certain ways: (a) The 4D fluents mechanism was enhanced with qualitative (in addition to quantitative) temporal expressions allowing for the representation of temporal intervals with unknown starting and ending points by means of their relation (e.g., “before”, “after”) to other time intervals. SWRL and OWL 2.0 constructs (e.g., disjoint properties) are combined, offering a sound and complete reasoning procedure ensuring path consistency [13].

<sup>1</sup> <http://www.w3.org/TR/swbp-n-aryRelations/>

This approach still suffers from two disadvantages: (a) relying on an interval-based representation didn't allow for reasoning over time instants similarly to intervals, (b) Implementing path consistency in SWRL called for additional temporal relations thus complicating the representation. The proposed approach tackles both these problems. In addition, we propose an extension to SPARQL with temporal operators for effectively querying over temporal information in OWL ontologies. The query language is independent of the 4D fluent representation introduced into this work (i.e., may work equally well with any other representation). To the best of our knowledge, this is the first work dealing with both qualitative and quantitative temporal information in ontologies and at the same time handling all issues referred to above within the same representation.

Related work in the field of knowledge representation is discussed in Section 2. This includes issues related to representing and reasoning over information evolving in time. The proposed ontology is presented in Section 3 and the corresponding reasoning mechanism in Section 4. The temporal query language is presented in Section 5 followed by evaluation in Section 6 and conclusions and issues for future work in Section 7.

## 2 Background and Related Work

The OWL-Time temporal ontology<sup>2</sup> describes the temporal content of Web pages and the temporal properties of Web services. Apart from language constructs for the representation of time in ontologies, there is still a need for mechanisms for the representation of the evolution of concepts (e.g., events) in time.

*Temporal Description Logics (TDLs)* [4] extend standard description logics (DLs) that form the basis for semantic Web standards with additional constructs such as “always in the past”, “sometime in the future”. TDLs offer additional expressive capabilities over non temporal DLs but they require extending OWL syntax and semantics with the additional temporal constructs. Representing information concerning specific time instants requires support for concrete domains. *Concrete Domains* [6] introduce datatypes and operators based on an underlying domain (such as decimal numbers). The concrete domains approach requires introducing additional datatypes and operators to OWL, while our work relies on existing OWL constructs. This is a basic design decision in our work. TOWL [11] is an approach combining 4D fluents with concrete domains but didn't support qualitative relations, path consistency checking (as this work does) and is not compatible with existing OWL editing, querying and reasoning tools (e.g., Protege, Pellet, SPARQL).

*Versioning* [2] suggests that the ontology has different versions as time evolves. When a change takes place, a new version is created. Versioning suffers from several disadvantages: (a) changes even on single attributes require that a new version of the ontology be created leading to information redundancy, (b) searching for events requires exhaustive searches in multiple versions of the ontology, (c) it is not clear how the relation between evolving classes is represented. *Named Graphs* [8] represent the temporal context of a property by inclusion of a triple representing the property in a named graph

---

<sup>2</sup> <http://www.w3.org/TR/owl-time/>

(i.e., a subgraph into the RDF graph of the ontology specified by a distinct name). The default (i.e., main) RDF graph contains definitions of interval start and end points for each named graph, so that a temporal property is represented by the start and end points corresponding to the temporal interval that the property holds. Named graphs are neither part of the OWL specification<sup>3</sup> (i.e., there are not OWL constructs translated into named graphs) nor they are supported by OWL reasoners.

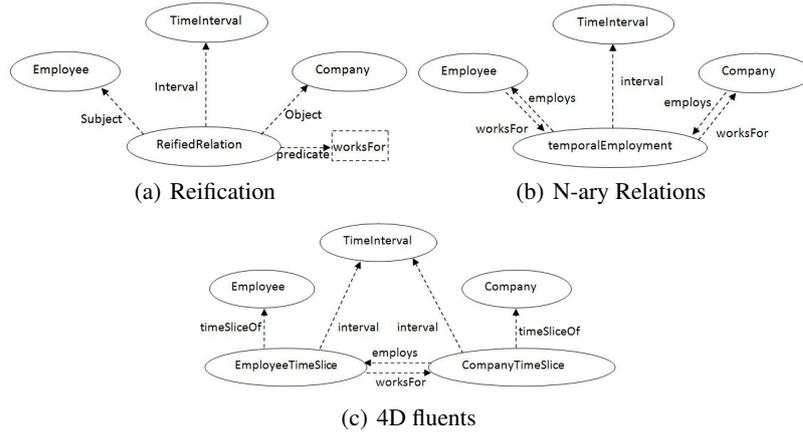


Fig. 1: Example of (a) Reification (b) N-ary Relations and (c) 4D-fluents

*Reification* is a general purpose technique for representing  $n$ -ary relations using a language such as OWL that permits only binary relations. Specifically, an  $n$ -ary relation is represented as a new object that has all the arguments of the  $n$ -ary relation as objects of properties. For example, if the relation  $R$  holds between objects  $A$  and  $B$  at time  $t$ , this is expressed as  $R(A,B,t)$ . In OWL this is expressed as a new object with  $R$ ,  $A$ ,  $B$  and  $t$  being objects of properties. Fig. 1(a) illustrates the relation  $WorksFor(Employee, Company, TimeInterval)$  representing the fact that an employee works for a company during a time interval. The extra class “ReifiedRelation” is created having all the attributes of the relation as objects of properties. Reification suffers mainly from two disadvantages: (a) a new object is created whenever a temporal relation has to be represented (this problem is common to all approaches based on OWL) and (b) offers limited OWL reasoning capabilities [3]. Because relation  $R$  is represented as the object of a property, OWL semantics over properties (e.g., inverse properties) are no longer applicable (i.e., the properties of a relation are no longer associated directly with the relation itself). CNTRO [14] describes a temporal medical ontology using reification.

The N-ary relations approach suggests representing an  $n$ -ary relation as two properties each related with a new object (rather than as the object of a property, as reification does). This approach requires only one additional object for every temporal relation,

<sup>3</sup> <http://www.w3.org/TR/owl2-syntax/>

maintains property semantics but (compared to the 4D-fluents approach discussed in this work) suffers from data redundancy in the case of inverse and symmetric properties (e.g., the inverse of a relation is added explicitly twice rather than once as in the 4D fluents approach). This is illustrated in Fig.1(b). Furthermore, domains and ranges of properties have to be adjusted taking into account the classes of intermediate objects representing the relation (for example the *worksfor* relation is no longer a relation having as object an individual of class *Company* and subject of class *Employee* as they are now related to the new object *TemporalEmployment*).

The *4D-fluent* (perdurantist) approach [3] shows how temporal information and the evolution of temporal concepts can be represented in OWL. Concepts evolving in time are represented as 4-dimensional objects with the 4th dimension being the time (*timeslices*). Time instances and time intervals are represented as instances of a *TimeInterval* class, which in turn is related with concepts varying in time as shown in Fig. 1(c). Changes occur on the properties of the temporal part of the ontology keeping the entities of the static part unchanged. The 4D-fluent approach still suffers from proliferation of objects since it introduces two additional objects for each temporal relation (instead of one in the case of reification and N-ary relations). The N-ary relations approach referred to above is considered to be an alternative to the 4D fluents approach considered into this work.

### 3 Temporal ontology

Following the approach by Welty and Fikes [3], to add the time dimension to an ontology, classes *TimeSlice* and *TimeInterval* with properties *TimeSliceOf* and *TimeInterval* are introduced. Class *TimeSlice* is the domain class for entities representing temporal parts (i.e., “time slices”) and class *TimeInterval* is the domain class of time intervals. A time interval holds the temporal information of a time slice. Property *TimeSliceOf* connects an instance of class *TimeSlice* with an entity, and property *interval* connects an instance of class *TimeSlice* with an instance of class *TimeInterval*. Properties having a temporal dimension are called *fluent properties* and connect instances of class *TimeSlice* (as in Fig.1(c)).

In our previous work [1] the 4D-fluents representation was enhanced with qualitative temporal relations (i.e., relations holding between time intervals whose starting and ending points are not specified) by introducing temporal relationships as object relations between time intervals. A temporal relation can be one of the 13 pairwise disjoint Allen’s relations [7] of Fig. 2. Notice that, temporal instants still cannot be expressed; subsequently, relations between time instants or between instants and intervals cannot be expressed explicitly.

In this work, an instant-based (or point-based) approach is adopted. Also, definitions for temporal entities (e.g., instants and intervals) are provided by incorporating OWL-Time into the same ontology. Each interval (which is an individual of the *ProperInterval* class) is related with two instants (individuals of the *Instant* class) that specify its starting and ending points using the *hasBeginning* and *hasEnd* object properties respectively. In turn, each *Instant* can be related with a specific date represented using the concrete *dateTime* datatype. One of the *before*, *after* or *equals*

relations may hold between any two temporal instants with the obvious interpretation. In fact, only relation *before* is needed since relation *after* is defined as the inverse of *before* and relation *equals* can be represented using the *sameAs* OWL keyword applied on temporal instants. In this work, for readability we use all three relations. Notice also that, property *before* may be also qualitative when holding between time instants or intervals whose values or end points are not specified. This way, we can assert and infer facts beyond the ones allowed when only instants or intervals with known values (e.g., dates) or end-points are allowed.

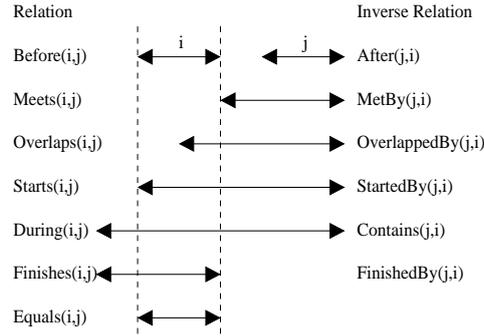


Fig. 2: Allen's Temporal Relations.

Relations between intervals are expressed as relations between their starting and ending points which in turn are expressed as a function of the three possible relations between points (time instants) namely *equals*, *before* and *after* denoted by “=”, “<” and “>” respectively, forming the so called “point algebra” [13]. Let  $i_1 = [s_1, e_1]$  and  $i_2 = [s_2, e_2]$  be two intervals with starting and ending points  $s_1, s_2$  and  $e_1, e_2$  respectively; then, the 13 Allen relations of Fig. 2 are rewritten as follows:

$$\begin{aligned}
 i_1 \text{ before } i_2 &\equiv e_1 < s_2 \\
 i_1 \text{ equals } i_2 &\equiv s_1 = s_2 \wedge e_1 = e_2 \\
 i_1 \text{ overlaps } i_2 &\equiv s_1 < s_2 \wedge e_1 < e_2 \wedge e_1 > s_2 \\
 i_1 \text{ meets } i_2 &\equiv e_1 = s_2 \\
 i_1 \text{ during } i_2 &\equiv s_1 > s_2 \wedge e_1 < e_2 \\
 i_1 \text{ starts } i_2 &\equiv s_1 = s_2 \wedge e_1 < e_2 \\
 i_1 \text{ finishes } i_2 &\equiv s_1 > s_2 \wedge e_1 = e_2
 \end{aligned}$$

The relations *after*, *overlappedby*, *metby*, *contains*, *startedby* and *finishedby* are the inverse of *before*, *overlaps*, *meets*, *during*, *starts* and *finishes* and are defined accordingly (by interchanging  $s_1, s_2$  and  $e_1, e_2$  in their respective definitions). These tem-

poral relations and the corresponding reasoning mechanism are integrated both in the 4D-fluents and the n-ary based ontologies.

## 4 Temporal Reasoning

Reasoning is realized by introducing a set of SWRL<sup>4</sup> rules operating on temporal relations. Reasoners that support DL-safe rules such as Pellet<sup>5</sup> can be used for inference and consistency checking over temporal relations. Alternatively, OWL axioms on temporal properties can be used instead of SWRL. As we shall see later, this approach cannot guarantee decidability and is therefore not compatible with W3C specifications. The three temporal relations between points are *before*, *after* and *equals*, denoted by symbols “<”, “>”, “=” respectively. Table 1 illustrates the set of reasoning rules based on the composition of existing relation pairs.

Relations	<	=	>
<	<	<, =, >	>
=	<	=	>
>	<, =, >	>	>

Table 1: Composition Table for point-based temporal relations.

The composition table represents the result of the composition of two temporal relations. For example, if relation  $R_1$  holds between  $instant_1$  and  $instant_2$ , and relation  $R_2$  holds between  $instant_2$  and  $instant_3$ , then the entry of the Table 1 corresponding to line  $R_1$  and column  $R_2$  denotes the possible relation(s) holding between  $instant_1$  and  $instant_3$ . Also, the three temporal relations are declared as pairwise disjoint, since they can’t simultaneously hold between two instants. Not all compositions yield a unique relation as a result. For example, the composition of relations *before* and *after* yields all possible relations as a result. Because such compositions doesn’t yield new information these rules are discarded. Rules corresponding to compositions of relations  $R_1, R_2$  yielding a unique relation  $R_3$  as a result are retained (7 out of the 9 entries of Table 1 are retained) and are expressed in SWRL using rules of the form:

$$R_1(x, y) \wedge R_2(y, z) \rightarrow R_3(x, z)$$

The following is an example of such a temporal inference rule:

$$before(x, y) \wedge equals(y, z) \rightarrow before(x, z)$$

A series of compositions of relations may imply relations which are inconsistent with existing ones (for example the rule referred to above will yield a contradiction if *after*( $x, z$ ) has been asserted into the ontology for specific values of  $x, y, z$ ). Consistency

<sup>4</sup> <http://www.w3.org/Submission/SWRL/>

<sup>5</sup> <http://clarkparsia.com/pellet/>

checking is achieved by ensuring path consistency [13]. Path consistency is implemented by consecutively applying the following formula:

$$\forall x, y, k R_s(x, y) \leftarrow R_i(x, y) \cap (R_j(x, k) \circ R_k(k, y))$$

representing intersection of compositions of relations with existing relations (symbol  $\cap$  denotes intersection, symbol  $\circ$  denotes composition and  $R_i, R_j, R_k, R_s$  denote temporal relations). The formula is applied until a fixed point is reached (i.e., the consecutive application of the rules above doesn't yield new inferences) or until the empty set is reached, implying that the ontology is inconsistent. Thus, in addition to rules implementing compositions of temporal relations, a set of rules defining the result of intersecting relations holding between two instances must also be defined in order to implement path consistency. These rules are of the form:

$$R_1(x, y) \wedge R_2(x, y) \rightarrow R_3(x, y)$$

where  $R_3$  can be the empty relation. For example, the intersection of the relation representing the disjunction of *before*, *after* and *equals* (abbreviated as *ALL*), and the relation *before* yields the relation *before* as result:

$$ALL(x, y) \wedge before(x, y) \rightarrow before(x, y)$$

The intersection of relations *before* and *after* yields the empty relation, and an inconsistency is detected:

$$before(x, y) \wedge after(x, y) \rightarrow \perp$$

As shown in Table 1, compositions of relations can yield one of the following four relations: *before*, *after*, *equals* and the disjunction of these three relations. Intersecting the disjunction of all three relations with any of these leaves existing relations unchanged. Intersecting any one of the tree basic (non disjunctive) relations with itself also leaves relations unaffected. Only compositions of pairs of different basic relations affect the ontology by yielding the empty relation as a result, thus detecting an inconsistency. By declaring the three basic relations (*before*, *after*, *equals*) as pairwise disjoint, all intersections that can affect the ontology are defined. Path consistency is implemented by defining compositions of relations using SWRL rules and declaring the three basic relations as disjoint. Notice that, path consistency is sound and complete when applied on the three basic relations [13].

Alternatively, we can define the composition of *before* with itself as a transitivity axiom rather than by an SWRL rule. In this case, there would be no need for SWRL rules applying only on named individuals into the ontology ABox. The resulting representation will apply on the TBox as well. However, there is an obstacle in this approach forcing the use of SWRL rules: the relation *before* must be declared as transitive in order to infer implied relations and disjoint with *after*, it's inverse relation, (also *before* is asymmetric and irreflexive) in order to detect inconsistencies. But OWL specifications<sup>6</sup> disallow the combination of transitivity and disjointness (or asymmetry) axioms on a property because they can lead to undecidability [9]. This restriction is necessary in order to guarantee decidability in reasoning with OWL 2 DL.

<sup>6</sup> <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/#The.Restrictions.on.the.Axiom.Closure>

## 5 Temporal Query Language

We also design and implement a SPARQL-like temporal query language based on the idea of introducing a set of temporal operators (i.e., the AT and Allen operators) on top of SPARQL. Query execution relies on the intermediate translation of the temporal operators into an equivalent SPARQL query. This translation depends on the underlying ontology representation (e.g., it is expressed as a set operations over fluent properties in case the 4D-fluents representation is used). In this work, we have experimented with both representations described in Section 2 namely, N-ary relations and the 4D-fluent approach. The motivation of extending SPARQL with new operators (rather than using SPARQL with its existing operators for querying temporal information) is that this approach offers additional flexibility in expressing temporal queries concisely, while ensuring independence of the temporal expressions from the peculiarities of the underlying ontological representation (i.e., query syntax is the same regardless of the temporal representation adopted). The query language inherits SPARQL syntax and semantics (e.g., queries over non-temporal information is expressed in SPARQL) with the addition of the temporal operators.

We introduce clauses AT, SOMETIME\_AT and ALWAYS\_AT for comparing a fluent property (i.e., the time interval during which the property holds true) with a time period (time interval) or time point. Such queries return fluents holding true at the specified time interval or point. Queries involving static properties (properties not changing in time) are issued as normal SPARQL queries applied on the static part of the ontology. Operator SOMETIME\_AT returns fluents holding for intervals that share common time points with the interval in question. Operator ALWAYS\_AT returns fluents holding for intervals which contain all points of the interval in question. For example the following query retrieves the name of the company “x” where employee “y” was always working for, during the specified interval:

```
select ?x ?y where {
  ?x ex1:hasEmployee ?y ALWAYS_AT
  "2007 - 02 - 01T00 : 00 : 00Z", "2007 - 02 - 05T00 : 00 : 00" }
```

The “AT” operator returns fluents holding at intervals that contain the time point in question. Notice that, integrating interval and instance representations allows for inferring relations between points and intervals in addition to relations between intervals. For example the following query retrieves the name of the company “x” that employee “y” was working for, at the specified date:

```
select ?x ?y where {
  ?x ex1:hasEmployee ?y AT "2007 - 02 - 05T00 : 00 : 00" }
```

The following Allen operators are also supported: BEFORE, AFTER, MEETS, METBY, OVERLAPS, OVERLAPPEDBY, DURING, CONTAINS, STARTS, STARTEDBY, ENDS, ENDEDBY and EQUALS, representing the relations holding between two time intervals specified (operators BEFORE and AFTER support temporal points as well). In this work, relations can be quantitative (i.e., involving specific temporal instants or intervals) or qualitative (i.e., the exact values of temporal instants or intervals

are unknown or not specified). For example the following query retrieves the name of the company “x” that employee “y” was working for, before *Employee2* worked for *Company1*:

```
select ?x ?y where {
  ?x ex1:hasEmployee ?y before
  ex1:Company1 ex1:hasEmployee ex1:Employee2}
```

## 6 Evaluation

Reasoning is achieved by employing DL-safe rules expressed in SWRL that apply on named individuals in the ontology A-box, thus retaining decidability while offering a sound and complete inference procedure for asserted temporal intervals or instants. Furthermore, reasoning has polynomial time complexity since only the basic Allen or point-based relations are supported [12, 13].

Because any time interval can be related with every other interval with one basic Allen relation (basic Allen relations are mutually exclusive) between  $n$  intervals, at most  $(n - 1)^2$  relations can be asserted and this also holds in case of instants. Furthermore, path consistency has  $O(n^5)$  worst time complexity (with  $n$  being the number of intervals or instants) and is sound and complete for the supported sets of relations. In the most general case where disjunctive relations are supported in addition to the basic ones, any time interval (or time instant) can be related with every other interval (or instant) by at most  $k$  relations, where  $k$  is the size of the set of supported relations. Therefore, for  $n$  intervals or instants, using  $O(k^2)$  rules, at most  $O(kn^2)$  relations can be asserted into the knowledge base. In the case of temporal instants [13], qualitative relations on time instants form a tractable set (i.e., a set of relations applying path consistency on this is a sound and complete method) if the relation  $\neq$  (i.e., a temporal instant is before *or* after another instant) is excluded. Reasoning can be extended with disjunctive relations such as  $\geq$  denoting that an instant is *after or equals* to another.

In this work, a point-based representation is adopted for handling both time instants and intervals. Relations between intervals are expressed as a function of relations between their end-points. A representation relying on intervals is also possible. However, since the number of basic relations is 13 (Fig. 2) and because all possible disjunctions appearing in the supported tractable set must also be supported, the representation may become particularly involved. Notice also that, time instants, the same as semi-closed temporal intervals (i.e., intervals with only one quantitative defined end-point) cannot be represented efficiently in an interval-based representation which doesn’t handle points. For example, if interval  $A$  *contains* interval  $B$  and point  $C$  is *into* interval  $B$  we can infer using the point based representation that  $C$  is *into* interval  $A$ .

## 7 Conclusions and Future Work

The contributions of this work are twofold: First, we propose an approach for handling temporal knowledge in ontologies using OWL and SWRL. It handles both time instants and temporal intervals (and also semi-closed intervals) equally well using a sound

and complete inference procedure based on path consistency. Second, we introduce a SPARQL-based temporal query language for expressing temporal queries. Extending our approach for spatial and spatio-temporal information and addressing scalability issues are directions of future work.

## References

1. S. Batsakis, and E.G.M. Petrakis. "SOWL: Spatio-temporal Representation, Reasoning and Querying over the Semantic Web" 6th International Conference on Semantic Systems, Graz, Austria, 1–3 September 2010.
2. M. Klein and D. Fensel. "Ontology Versioning for the Semantic Web". In *International Semantic Web Working Symposium (SWWS'01)*, pages 75–92, California, USA, July–August 2001.
3. C. Welty and R. Fikes. "A Reusable Ontology for Fluents in OWL". *Frontiers in Artificial Intelligence and Applications*, 150:226–236, 2006.
4. A. Artale, and E. Franconi. "A Survey of Temporal Extensions of Description Logics". *Annals of Mathematics and Artificial Intelligence*, 30(1-4), 2001.
5. C. Gutierrez, C. Hurtado, and A. Vaisman. "Introducing Time into RDF". In *IEEE Trans. on Knowledge and Data Engineering*, 19(2), pp. 207-218, 2007.
6. C. Lutz. "Description logics with concrete domains - A survey." In *Advances in Modal Logics Volume 4*. Kings College, 2003.
7. J. F. Allen. "Maintaining Knowledge About Temporal Intervals". *Communications of the ACM*. 26:832-843, 1983.
8. J. Tappolet, and A. Bernstein "Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL." In: *Proceedings of the European Semantic Web Conference, LNCS 5554*, 308-322, 2009.
9. I. Horrocks, O. Kutz and U. Sattler "The Even More Irresistible SROIQ" In: *Proc. KR 2006*, Lake District, UK, 2006.
10. B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider and U. Sattler. "OWL 2: The Next Step for OWL." In: *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, pp. 309-322, 2008.
11. V. Milea, F. Frasincar, and U. Kaymak "Knowledge Engineering in a Temporal Semantic Web Context" In *The Eighth International Conference on Web Engineering (ICWE 2008)*, 2008.
12. B. Nebel, and H.J. Burckert "Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra" *Journal of the ACM (JACM)*, Vol.42(1), pages:43–66, 1995.
13. P. van Beek, and R. Cohen "Exact and approximate reasoning about temporal relations." *Computational intelligence*, Vol 6(3), pp. 132–147, 1990.
14. C. Tao, W.Q. Wei, H.R. Solbrig, G. Savova, and C.G. Chute "CNTRO: A Semantic Web Ontology for Temporal Relation Inferencing in Clinical Narratives" In *AMIA Annual Symp. Proc.* 2010, pp. 787-791, 2010.