Temporal Representation and Reasoning in OWL 2

Sotiris Batsakis [a,*], Euripides G.M. Petrakis [b] , Ilias Tachmazidis [a] and Grigoris Antoniou [a]

[a] *School of Computing and Engineering,University of Huddersfield, Queensgate, Huddersfield, UK*
E-mail: {S.Batsakis,I.Tachmazidis,G.Antoniou}@hud.ac.uk
[b] *School of Electronic and Computer Engineering, Technical University of Crete (TUC) Chania, Crete, Greece*
E-mail: petrakis@intelligence.tuc.gr

Abstract. The representation of temporal information has been in the center of intensive research activities over the years in the areas of knowledge representation, databases and more recently, the Semantic Web. The proposed approach extends the existing framework of representing temporal information in ontologies by allowing for representation of concepts evolving in time (referred to as "dynamic" information) and of their properties in terms of qualitative descriptions in addition to quantitative ones (i.e., dates, time instants and intervals). For this purpose, we advocate the use of natural language expressions, such as "before" or "after", for temporal entities whose exact durations or starting and ending points in time are unknown. Reasoning over all types of temporal information (such as the above) is also an important research problem. The current work addresses all these issues as follows: The representation of dynamic concepts is achieved using the "4D-fluents" or, alternatively, the "N-ary relations" mechanism. Both mechanisms are thoroughly explored and are expanded for representing qualitative and quantitative temporal information in OWL. In turn, temporal information is expressed using either intervals or time instants. Qualitative temporal information representation in particular, is realized using sets of SWRL rules and OWL axioms leading to a sound, complete and tractable reasoning procedure based on path consistency applied on the existing relation sets. Building upon existing Semantic Web standards (OWL), tools and member submissions (SWRL), as well as integrating temporal reasoning support into the proposed representation, are important design features of our approach.

Keywords: Semantic Web, Temporal Representation, Temporal Reasoning

1. Introduction

The rapid growth of the World Wide Web (WWW) in recent years has generated the need for tools and mechanisms to automatically handle tasks typically handled by humans. For example, planning a trip requires selecting and purchasing tickets at specific dates at the best available price. Typically, these tasks are handled by searching the Web (e.g., using a search engine). Semantic Web is intended to provide a solution to these needs by developing Web services that accomplish these tasks automatically without requiring user intervention, besides task description. These services must be capable to understand the meaning of Web pages and reason over their content in a way similar to the way humans do. Semantic Web will realize this technology by introducing formal, machine readable semantics for representation of knowledge, combined with reasoning and querying support.

Formal definitions of concepts and their properties form ontologies, which are defined using the RDFS and OWL languages [23]. Ontologies contain definitions of concepts and their properties by means of binary relations. The syntactic restriction of OWL to bi-

---

*Corresponding author. E-mail: S.Batsakis@hud.ac.uk

nary relations complicates the representation of n-ary (e.g., ternary) relations. For example, an employment relation for a specific temporal interval that involves an employee, an employer and a temporal interval, is in fact a ternary relation. In general, properties of objects that change in time (dynamic properties) are not binary relations, since they involve a temporal interval in addition to the subject and the object. The availability of temporal information in Linked Data is analysed in [31]. Representing information that evolve in time in ontologies is the problem this work is dealing with.

We introduce an approach for handling temporal information in OWL while being consistent with existing Semantic Web standards (e.g., OWL [23]), W3C member submissions (e.g., SWRL [15]) and tools (e.g., Pellet [36] and HermiT [35] reasoners). The latter is a basic design decision in our work. Earlier work by Welty and Fikes [42] 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 in OWL using the so called "4D-fluents approach". In our work, this approach is extended as follows: The 4D-fluents and the N-ary mechanisms are 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", "overlaps") to other time intervals. To the best of our knowledge, this is the first work dealing with both qualitative and quantitative temporal information in ontologies, while supporting both time points and intervals.

In our approach, SWRL and OWL 2 constructs (e.g., disjoint properties) are combined, offering a sound and complete reasoning procedure over qualitative relations ensuring path consistency [41]. This is an issue which is not examined in the original work by Welty and Fikes or by other known approaches for temporal information representation (e.g., [4], [21], [25], [12], [18]). CNTRO ontology [38] contains SWRL rules for temporal reasoning, but is not combined with a sound and complete reasoning mechanism over Allen's interval relations (only reasoning over points and timestamps is supported in CNTRO) as this work does. SOWL ontology [6] supports only qualitative Allen relations using SWRL, but not time points, or intervals and points defined using dates as this work does. The proposed reasoner handles both quantitative and qualitative information using tractable sets of relations on which path consistency applies. Reasoning is implemented using SWRL rules and is capable of infer-

ring temporal relations and detecting inconsistent assertions. The reasoning mechanism is an integral part of the ontology and is handled by standard reasoners (such as Pellet). Reasoning over time instants, in addition to time intervals, is also a distinctive feature of our work. For this reason, the temporal representation is complemented by instant (or point) based representations as well, which was a limitation of previous work in [6].

Apart from 4D-fluents, a representation of both forms of temporal information (i.e., quantitative, qualitative) based on N-ary relations [25] is also proposed. Both approaches (4D-fluents and N-ary) are selected since they are OWL compliant and users have the choice to select the representation that they are more familiar with.

In summary, current work extends the representation presented in [6] in certain ways: (a) time points are supported in addition to intervals, (b) quantitative and combined quantitative-qualitative representations are supported in addition to qualitative representations, (c) several interval and point representations are proposed and evaluated, and (d) applications and related tools are presented.

Related work in the field of knowledge representation is discussed in Section 2. The proposed ontology model for temporal information is presented in Section 3. The corresponding reasoning mechanism is presented in Section 4, followed by evaluation in Section 5, related applications in Section 6 and conclusions and issues for future work in Section 7.

2. Background and Related work

Semantic Web standards and related work in the field of temporal knowledge representation are discussed in the following.

2.1. Description Logics and OWL

Description Logics (DLs) [5] are in most cases based on decidable fragments of First Order Logic (FOL) that form the basis for the Semantic Web standards for defining rich ontologies (notice that not all DLs are decidable). The basic components of a Description Logic formalism are the *concepts* or *classes*, their *properties* or *roles* and the *individuals* or *objects*. The expressivity of a description logic formalism is defined by the set of allowable constructs and expressions.

The expressive power of DLs is complemented by inference procedures dealing with *subsumption* (i.e., determining subclass-superclass relations), *consistency* (i.e., determining contradictions in concept definitions and individual assertions) and *instance* (i.e., determining the class(es) that an individual belongs to). Decidability of inference is a highly desirable characteristic so that, in practice, expressivity is often sacrificed (i.e., restricted) in order to guarantee decidability. The OWL language is based on DLs and it is the basic component of the Semantic Web initiative.

A description logic or language is fully characterized by the allowable constructs that are used for the definitions of concepts and properties, as expressions of basic (atomic) concepts and properties. The set of such definitions for an application domain forms the Terminological Box (TBox) of an ontology. Assertions involving concepts and properties of individuals form the Assertional Box (ABox) of the ontology. Reasoning is applied on both TBox definitions and ABox assertions. Recently, a shift towards the so called "tableaux" based reasoning is observed [5]. Popular reasoners such as FaCT++[1], Pellet[2], HermiT[3] and RACER[4] are examples of tableaux based reasoners.

RDF and RDFS represent properties or relations between entities by means of triples of the form *subject-predicate-object* (e.g., Google employs John). Specific individuals can belong to classes (e.g., John is-a Person, where *John* is an individual and *Person* is a class). Properties such as *employs* can relate individuals of specific classes. Classes that the subject and the object of a property are members, are abbreviated as *domain* and *range* respectively. Basic taxonomic relations between classes and properties can be specified as well. For example it can be stated that *Employee* is a *subclass* of *Person*, (i.e., every employee is also a person). OWL extends RDF/RDFS expressivity and OWL-DL is a decidable variant of OWL based on Description Logics.

The evolution of the OWL specification was based on the observation that additional constructs can be added in OWL-DL without compromising decidability, while increasing expressivity. Extending OWL-DL with additional constructs led to the adoption of OWL

2 as the current Semantic Web standard for defining rich ontologies [23].

SWRL[5] is a language for specifying rules applying on Semantic Web ontologies and Rule Interchange Format (RIF) [17] is the W3C recommendation for exchanging rules. *Horn Clauses* (i.e., a disjunction of classes with at most one positive literal), can be expressed using SWRL, since Horn clauses can be written as implications (i.e., $\neg A \vee \neg B ... \vee C$ can be written as $A \wedge B \wedge ... \Rightarrow C$). The efficiency of reasoning over Horn clauses using forward chaining algorithms is a reason for choosing this form of rules. The antecedent (body) of the rule is a conjunction of clauses. Notice that neither disjunction nor negation of clauses is supported in the body of rules. Also, the consequence (head) of a rule is one positive clause. Neither negation nor disjunction of clauses can appear as a consequence of a rule. To guarantee decidability, rules are restricted to *DL-safe rules* [22] that apply only on named individuals in the ABox.

2.2. Representation of Time

Time can be conceptualized as discrete or continuous, linear or cyclical, absolute or relative, qualitative or quantitative [9]. Also, time can be represented using time instances or intervals. Temporal concepts are represented by the OWL-Time ontology [13]. OWL-Time is an ontology of the concepts of time, but OWL-Time does not specify how these concepts can be used to represent evolving properties of objects (i.e., properties that change in time) and it does not specify how to reason over qualitative relations of temporal intervals and instants. This is also a problem this work is dealing with.

Choosing between a point or an interval-based representation is an important issue [41]. When using an interval-based representation, relations between intervals can be asserted directly, without representing their end-points, thus representation is more compact. The main disadvantage of an interval-based representation is that relations between points and between points and intervals can not be represented, while a point-based representation can support both points and intervals. Point-based representations assume linear ordering of time points with three possible relations the "$<$","$>$","$=$" often referred to as *before, after* and *equals* respectively. Based on these ordering relations,

intervals can also be defined as ordered pairs of points s, e with $s < e$, often referred to as *start* and *end* of an interval respectively. An interval temporal relation can be one of the 13 pairwise disjoint Allen's relations [1] of Figure 1.

In cases where the exact durations of temporal intervals are unknown (i.e., their starting or ending points are not specified), their temporal relations to other intervals (or points) can still be asserted qualitatively by means of temporal relations (e.g., "event A happens before B" even in cases where the exact durations of A or B or, of both A and B are unknown). Quantitative representations, on the other hand, are expressed using OWL datatypes (such as *xsd:date*) which can be used for comparing dates (e.g., such as the starting or ending points of intervals) and for yielding the Allen relation between time points or intervals.
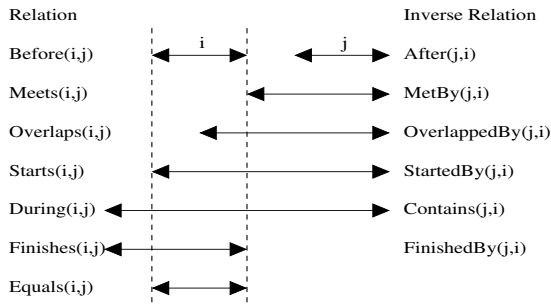

Fig. 1. Allen's Temporal Relations

2.3. Temporal Reasoning

Inferring implied relations and detecting inconsistencies are handled by a reasoning mechanism. In the case of a quantitative representation, such a mechanism is not required because temporal relations are extracted from the numerical representations in polynomial time (e.g., using datatype comparisons).

In the case of qualitative relations, assertions of relations holding between temporal entities (e.g., intervals and points) restrict the possible assertions holding between other temporal entities in the knowledge base. Then, reasoning on qualitative temporal relations can be transformed into a *constraint satisfaction problem*, which is known to be an *NP-hard* problem in the general case [30].

Inferring implied relations is achieved by specifying the result of *compositions* of existing relations. Specifically, when a relation (or a set of possible relations) R_1 holds between entities A and B and a re-

lation (or a set of relations) R_2 holds between entities B and C then, the *composition* of relations R_1, R_2 (denoted as $R_1 \circ R_2$) is the set (which may contain only one relation) R_3 of relations holding between A and C. Typically, all possible compositions of pairs of relations are stored in *composition tables* [30].

Qualitative relations under the intended semantics may not apply simultaneously between a pair of individuals. For example, given time instants p_1 and p_2, p_1 cannot be simultaneously *before* and *after* p_2. Typically, in temporal representations (e.g., using Allen relations), all basic relations (i.e., simple relations and not disjunctions of relations) are pairwise disjoint. When disjunctions of basic relations hold true simultaneously, then their set intersection holds true as well. For example, if p_1 is *before or equals* p_2 and simultaneously p_1 is *after or equals* p_2 then p_1 *equals* p_2. In case the intersection of two relations is empty, these relations are disjoint. Checking for consistency means checking if asserted and implied relations are disjoint.

Reasoning over temporal relations is known to be an NP-hard problem and identifying tractable cases of this problem has been in the center of many research efforts over the last few years [30]. The notion of *k-consistency* is very important in this research. Given a set of n entities with relations asserted between them imposing certain restrictions, *k-consistency* means that every subset of the n entities containing at most k entities does not contain an inconsistency. Notice that checking for all subsets of n entities for consistency is exponential on the number n.

There are cases where, although *k-consistency* does not imply *n-consistency*, there are specific sets of relations R_t (which are subsets of the set of all possible disjunctions of basic relations R), with the following property: if asserted relations are restricted to this set, then *k-consistency* implies *n-consistency* and R_t is a *tractable set* of relations or a *tractable subset of R* [30]. Tractable subsets for point algebra have been identified in [41] and tractable sets of Allen interval algebra have been identified in [24].

2.4. The Semantic Web Approach

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. Representation of time in the Semantic Web can be achieved using *Temporal Description logics (TDLs)* [4], *Concrete do-*

mains [21], *Quadtuples* [20], *Reification* [25], *Temporal RDF* [12], *Versioning* [18], *named graphs* [39] and *4D-fluents* [42].

Temporal RDF is an RDF and not an OWL based approach. Named graphs are also RDF-based approaches, since named graphs are not part of the OWL specification. Quadtuples are not always supported (since they are not part of OWL specification), thus this solution depends on the underlying triple store.

Temporal Description Logics (TDLs) [4] extend standard description logics (DLs) that form the basis for Semantic Web standards for defining rich ontologies on top of RDF/S, with additional constructs such as "always in the past" and "sometime in the future". TDLs offer additional expressive capabilities over non temporal DLs and retain decidability (with an appropriate selection of allowable constructs), but they require extending OWL syntax and semantics with the additional temporal constructs (the same as property labelling [12]).

Concrete Domains [21] 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 [10] is an approach combining 4D-fluents with concrete domains but did not support qualitative relations, path consistency checking (as this work does) and is not compatible with existing OWL editing, querying and reasoning tools (e.g., Protégé, Pellet, SPARQL).

Versioning [18] suggests that the ontology has different versions (one per instance of time). 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 occurred at time instances or during time intervals requires exhaustive searches in multiple versions of the ontology, and (c) it is not clear how the relation between evolving classes is represented.

Reification (Figure 2) 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. Using an improved form of reification (which is used at the CNTRO ontology [38]), the N-ary relations approach [25] suggests representing an

n-ary relation as three properties (subject, object and time) each related with a new object (rather than as the object of a property, thus retaining properties of the n-ary relation such as inverse properties or symmetry). Notice that the N-ary relationship approach expresses relations with arity greater than two as a new class rather than as a property [31]. This approach requires only one additional object for every temporal relation, maintains property semantics but (compared to the 4D-fluents approach below) suffers from data redundancy in the case of inverse and symmetric properties (e.g., the inverse of a relation is added explicitly twice instead of once as in 4D-fluents) or two separate properties (i.e., the second representing the inverse) must be used. This is illustrated in Figure 3. In the case of transitive properties additional triples are introduced as well.
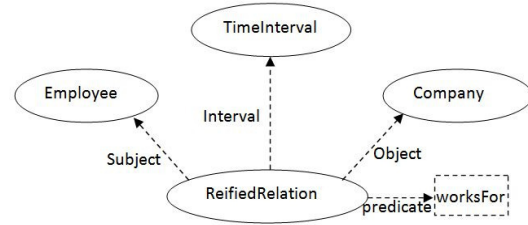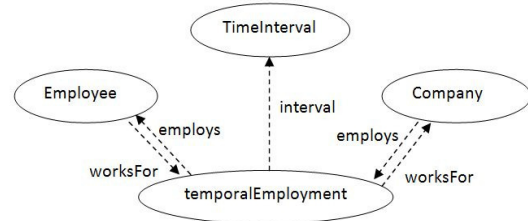

Fig. 2. Example of Reification
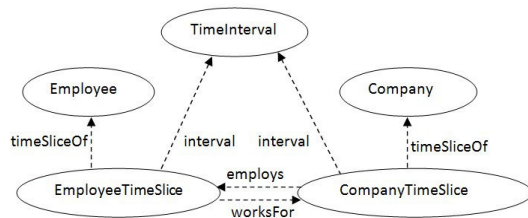

Fig. 3. Example of N-ary Relations

Fig. 4. Example of 4D fluents

The *4D-fluents* (perdurantist) approach [42] shows how temporal information and the evolution of tem-

poral concepts can be represented in OWL. Concepts 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 Figure 4. Changes occur on the properties of the temporal part of the ontology keeping the entities of the static part unchanged. The 4D-fluents approach still suffers from proliferation of objects since it introduces two additional objects for each temporal relation (instead of one in the case of N-ary relations). The N-ary relations approach referred to above is considered to be an alternative to the 4D-fluents approach also considered in this work. Avoiding proliferation of objects can be achieved if quintuples are used instead of triples as proposed in [20]. A detailed comparison of design patterns for representing temporal information for the Semantic Web is presented in [11].

3. Temporal Representation

We propose an ontology for representing and reasoning over dynamic information in OWL. Building upon well established standards (OWL 2) and tools the proposed ontology enables representation of static as well as of dynamic information based on the 4D-fluents [42] (or, equivalently, on the N-ary [25]) approach. Representing both qualitative temporal information (i.e., information whose temporal are unknown such as "before" for temporal relations) in addition to quantitative information (i.e., where temporal information is defined precisely) is a distinctive feature of this work. Both, the 4D-fluents and the N-ary relations approaches are expanded to accommodate this information. The corresponding reasoner implements path consistency [30], and is capable of inferring new relations and checking their consistency, while retaining soundness, completeness, and tractability over the supported sets of relations.

3.1. Temporal Representation using 4D-Fluents

Following the approach by Welty and Fikes [42], to add time dimension to an ontology, classes *TimeSlice* and *TimeInterval* with properties *tsTimeSliceOf* (tsTimeSliceof is an equivalent property of timeSliceOf property defined in [42]) and *tsTimeInterval* 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 *tsTimeSliceOf* (or the equivalent property *timeSliceOf*[6] connects an instance of class *TimeSlice* with an entity. Property *tsTimeInterval* (or the equivalent property *interval*, the equivalent properties are used in order to allow users use both names) connects an instance of class *TimeSlice* with an instance of class *TimeInterval*. Properties having a time dimension are called fluent properties and connect instances of class *TimeSlice*.

Fig. 5. Dynamic Enterprise Ontology

Figure 5 illustrates a temporal ontology with classes *Company* (with datatype property *companyName*), *Product* (with datatype properties *Price* and *productName*), and *Location* which represents the location of the company. In this example, *CompanyName* is a static property (its value does not change in time), while properties *produces*, *productName*, *locatedAt* and *Price* are dynamic (fluent) properties. Fluent properties are the properties whose values may change in time. Because they are fluent properties, their domain (and range) is of class *TimeSlice*. *CompanyTimeSlice*, *LocationTimeslice* and *ProductTimeSlice* are instances of class *TimeSlice*.

In this work, the 4D-fluents and N-ary representations are enhanced with qualitative temporal relations

---

[6]the equivalent properties are used in order to allow users use both names

holding between time intervals whose starting and ending points are not specified. This is implemented by introducing temporal relationships as object relations between time intervals. This can be one of the 13 pairwise disjoint Allen's relations [1] of Figure 1. Definitions for temporal entities (e.g., intervals) are provided by incorporating OWL-Time into the same ontology.

Qualitative relations are used for increasing the expressive power of the representation. Specifically, a fact can be asserted even when exact dates are not known by means of the qualitative temporal relations with other intervals. Typically, the 4D-fluents model (similarly to other approaches such as Temporal RDF [12]) assumes closed temporal intervals for the representation of temporal information, while semi-closed and open intervals cannot be represented effectively in a formal way. This is handled by Allen relations: for example if interval t_1 is known and t_2 is unknown but we know that t_2 starts when t_1 ends, then we can assert that t_2 is *met by* t_1. Likewise, if t_3 is an interval with unknown endpoints and t_3 is *before* t_1 then, using compositions of Allen relations [1], we infer that t_3 is *before* t_2 although both intervals' endpoints are unknown and their relation is not represented explicitly in the ontology. Semi-closed intervals can be handled in a similar way. For example, if t_1 starts at time point 1, still holds at time point 2, but it's endpoint is unknown, we assert that t_1 has *started by* interval t_2:[1,2].

Our approach demonstrates enhanced expressivity compared to previous approaches [39,10] that require specific dates for all asserted facts by combining 4D-fluents with Allen's temporal relations; their formal semantics and composition rules are defined in [1]. For example, if A happened before B, and C happens during B, then we can assert these facts using qualitative interval relations and infer that A happened before C, even if all dates are unknown. This is not possible using quantitative only approaches such as [39,10]. 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 proposed as well. As in the case of temporal intervals, OWL-Time provides with definitions for instants: each interval (which is an individual of the $ProperInterval$ class of OWL-Time) is related with two temporal instants (individuals of the $Instant$ class) that specify its starting and ending points using the $hasBegining$ and $hasEnd$ object properties re-

spectively. In turn, each $Instant$ can be related with a specific date using the concrete $dateTime$ datatype.

One of the $before$, $after$ or $equals$ relations may hold between any two temporal instants with the corresponding point located before, after or at the same location with the reference point on the axis of time respectively. 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 property applied on temporal instants. In this work, for readability, we use all three relations. Notice that property $before$ may be qualitative when holding between time instants or intervals whose values or end points are not specified. Instants can also be defined quantitatively using the $dateTime$ datatype.

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" [41]. 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 Figure 1 are rewritten as follows (according to Allen [1], for any given interval with starting point s and ending point e, the following holds: $s < e$) :

$$i_1 \; before \; i_2 \equiv e_1 < s_2$$
$$i_1 \; equals \; i_2 \equiv s_1 = s_2 \wedge e_1 = e_2$$
$$i_1 \; overlaps \; i_2 \equiv s_1 < s_2 \wedge e_1 < e_2 \wedge e_1 > s_2$$
$$i_1 \; meets \; i_2 \equiv e_1 = s_2$$
$$i_1 \; during \; i_2 \equiv s_1 > s_2 \wedge e_1 < e_2$$
$$i_1 \; starts \; i_2 \equiv s_1 = s_2 \wedge e_1 < e_2$$
$$i_1 \; finishes \; i_2 \equiv s_1 > s_2 \wedge e_1 = e_2$$

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). Notice that in the case of Allen relations, additional relations (representing disjunctions of basic relations) are introduced in order to implement path consistency, totalling a set of 29 supported relations (although, such relations are not required by a point algebra). Example of such relations is the disjunction of relations *during, overlaps* and *starts*. The full set of supported relations is presented in Section 4.2. These temporal relations and the corresponding reasoning mechanism are integrated within the ontology.

The aforementioned reasoning mechanism is also used for enforcing restrictions in the 4D-fluents mech-

anism. Specifically, in the original work by Welty and Fikes [42], the following restriction is imposed on timeslices: whenever two timeslices are related by means of a fluent property, their corresponding temporal intervals must be equal. However, no mechanism for enforcing this restriction is provided. In this work, the following SWRL rule in conjunction with the reasoning mechanism of Section 4 (which is used for checking if equality of intervals causes an inconsistency) imposes the required restriction:

$$fluent(x,y) \wedge tsTimeInterval(y,z)$$
$$\wedge\, tsTimeInterval(x,w) \qquad\qquad (1)$$
$$\rightarrow equals(w,z)$$

3.2. Representation using N-ary Relations

The N-ary version of the ontology introduces one additional object for representing a temporal property. This object is an individual of class $Event$ and this name convention is also adopted by other approaches such as the LODE ontology [34]. In our work, the temporal property remains a property (and not object of property as in Figure 2) relating the additional object with both the subject and object (e.g., an $Employee$ and a $Company$) involved in a temporal relation. This is illustrated in Figure 3. The representation of qualitative relations between temporal intervals or instants (and the corresponding reasoning mechanisms) remains identical to the 4D-fluents based version of the model.

If a fluent property is transitive, specific rules must be defined since the equality of the related intervals must also hold when a temporal (fluent) property is transitive. For example if $worksFor$ is transitive, in case of dynamic properties this holds only if the corresponding intervals are equal (e.g., if *A worksFor B during interval I1* and *B worksFor C during interval I2* and the property *workFor* is transitive then *A worskFor C if intervals I1 and I2 are equal*). This can be achieved using an SWRL rule such as in the case of 4D-fluents.

3.3. Representation of Points and Intervals

This work deals with qualitative relations between points in addition to interval Allen relations. Qualitative relations of two points are represented using an object property specifying their relative position on the axis of time. Specifically between two points three re-

lations can hold, these relations are "$<$", "$>$", "$=$" also referred to as *before, after* and *equals* respectively. If date/time is available then a corresponding datatype property can be used. Qualitative and quantitative representations can be combined (see Figure 6). An in-

Fig. 6. Point Representation

terval temporal relation can be one of the 13 pairwise disjoint Allen relations [1] of Figure 1. In cases where the exact durations of temporal intervals are unknown (i.e., their starting or ending points are not specified), their temporal relations to other intervals can still be asserted qualitatively by means of temporal relations (e.g., "interval i_1 is before interval i_2" even in cases where the exact starting and ending time of either i_1, i_2, or both are unknown).

Fig. 7. Direct Interval Representation

Intervals can be represented using two directly attached datatype properties, corresponding to starting

and ending time of each interval (see Figure 7). This straightforward approach can be applied only when start and end time of intervals are known. Interval relations can be inferred using comparisons of starting/ending dates using SWRL rules.
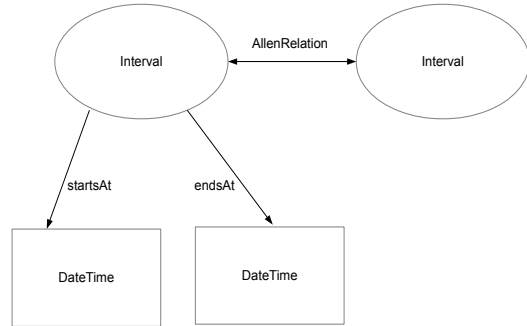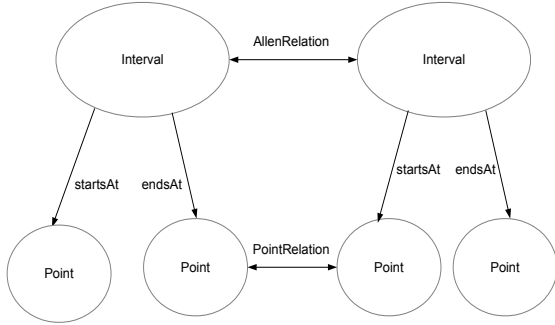

Fig. 8. Point-Interval Representation

Another more flexible and more complex approach is presented in Figure 8. In this case, intervals are related with starting and ending points using object properties, and not directly with dates. These points can be associated with dates, as in Figure 6, and/or with other points using object point relations (such as *after*). Point relations can be inferred using comparisons of dates and/or reasoning rules over asserted point relations. When point relations are inferred, then Allen relations between intervals can be inferred using SWRL rules implementing the definitions of Figure 1.

Fig. 9. Allen-Based Interval Representation

Finally, reasoning over qualitative defined Allen relations can be applied directly without using dates or points as in Figure 9.

4. Temporal Reasoning

Temporal reasoning in this work is realized by introducing a set of SWRL[7] rules for asserting inferred temporal Allen relations. Reasoners that support DL-safe rules (i.e., rules that apply only on named individuals in the knowledge base) such as Pellet [36] can be used for inference and consistency checking over temporal relations.

Specifically, reasoning is applied either on temporal intervals directly [6] or by applying point-based reasoning [8] operating on representations of intervals involving their starting and ending points. Both approaches have been implemented and are discussed in the following.

4.1. Reasoning over Interval-Based Representations

Reasoning is realized by introducing a set of SWRL rules operating on temporal intervals. The temporal reasoning rules are based on the composition of pairs of the basic Allen's relations of Figure 1 as defined in [1]. Specifically, if relation R_1 holds between $interval_1$ and $interval_2$ and relation R_2 holds between $interval_2$ and $interval_3$, then the composition table defined in [1] denotes the possible relation(s) holding between $interval_1$ and $interval_3$. Not all compositions yield a unique relation as a result. For example, the composition of relations $During$ and $Meets$ yields the relation $Before$ as a result, while the composition of relations $Overlaps$ and $During$ yields three possible relations namely *During, Overlaps and Starts*. In our proposed approach, reasoning is realized by sets of rules corresponding to compositions of relations R_1, R_2 [8]. Rules yielding a unique relation R_3 as a result can be represented using SWRL as follows:

$$R_1(x,y) \land R_2(y,z) \rightarrow R_3(x,z) \qquad (2)$$

An example of temporal inference rule is the following:

$$During(x,y) \land Meets(y,z) \rightarrow Before(x,z) \quad (3)$$

Rules yielding a set of possible relations cannot be represented directly in SWRL since, disjunctions of

---

[7]http://www.w3.org/Submission/SWRL/
[8]We have made this representation available on the Web at http://www.intelligence.tuc.gr/prototypes.php

atomic formulas are not permitted as a rule head. Instead, disjunctions of relations are represented using new relations whose compositions must also be defined and asserted into the knowledge base. For example, the composition of relations $Overlaps$ and $During$ yields the disjunction of three possible relations *During, Overlaps* and *Starts* (*DOS*) as a result:

$$Overlaps(x,y) \wedge During(y,z) \rightarrow$$
$$During(x,z) \vee Starts(x,z) \vee Overlaps(x,z) \quad (4)$$

If the relation DOS represents the disjunction of relations *During, Overlaps* and *Starts*, then the composition of $Overlaps$ and $During$ can be represented using SWRL as follows:

$$Overlaps(x,y) \wedge During(y,z) \rightarrow DOS(x,z) \quad (5)$$

The set of possible disjunctions over all basic Allen's relations contains 2^{13} relations and complete reasoning over all temporal Allen relations has exponential time complexity. However, tractable subsets of this set that are closed under composition (i.e., compositions of relation pairs from this subset yield also a relation in this subset) are also known to exist [24,41]. In this work, we use the subset presented in Section 4.2. In addition, inverse axioms (relations *After, Metby, Overlappedby, Startedby, Contains* and *Finishedby* are the inverse of *Before, Meets, Overlaps, Starts, During* and *Finishes* respectively) and rules defining the relation holding between two intervals with known starting and ending points (e.g., if the ending point of $interval_1$ is before the starting point of $interval_2$, then $interval_1$ is *before* $interval_2$) are also asserted into the knowledge base.

The starting and ending points of intervals are represented using concrete datatypes such as *xsd:date* that support ordering relations. Axioms involving disjunctions of basic relations are denoted using the corresponding axioms for these basic relations. Specifically, compositions of disjunctions of basic relations are defined as the disjunction of the compositions of these basic relations. For example, the composition of relation DOS (representing the disjunction of *During, Overlaps* and *Starts*) and the relation $During$ yields the relation DOS as follows (notice that in this example DOS is decomposed to basic relations for illustrative purposes, the corresponding SWRL reasoning rule

consists of the head and body and not the intermediate decomposition):

$$DOS \circ During \rightarrow$$
$$(During \vee Overlaps \vee Starts) \circ During \rightarrow$$
$$(During \circ During) \vee (Overlaps \circ During)$$
$$\vee (Starts \circ During) \rightarrow \quad (6)$$
$$(During) \vee (During \vee Overlaps \vee Starts)$$
$$\vee (During) \rightarrow During \vee Starts \vee Overlaps$$
$$\rightarrow DOS$$

The symbol $\circ$ denotes composition of relations. Compositions of basic (non-disjunctive) relations are defined at [1]. Similarly, the inverse of a disjunction of basic relations is the disjunction of the inverses of these basic relations illustrated in Figure 1. For example, the inverse of the disjunction of relations $Before$ and $Meets$ is the disjunction of their inverse relations, $After$ and $MetBy$ respectively.

By applying compositions of relations, the implied relations may be inconsistent (i.e., yield the empty relation $\perp$ as a result). Consistency checking is achieved by applying path consistency [30,24,41]. Path consistency is implemented by consecutive application of the formula:

$$\forall x,y,k\, R_i(x,y) \cap (R_j(x,k) \circ R_k(k,y)) \rightarrow R_s(x,y)$$
$$(7)$$

representing intersection of compositions of relations with existing relations R_i, R_j, R_k, (symbol $\cap$ denotes intersection). The formula is applied until a fixed point is reached (i.e., application of rules does not yield new inferences) or until the empty set is reached, implying that the ontology is inconsistent. Implementing the formula requires definition of rules for both composition and intersection.

An additional set of rules defining the result of intersection of relations holding between two intervals is thus introduced. These rules are of the form:

$$R_1(x,y) \wedge R_2(x,y) \rightarrow R_3(x,y), \quad (8)$$

where R_3 can be the empty relation. For example, the intersection of relation DOS (represents the disjunction of *During, Overlaps and Starts*) with relation

During, yields relation *During* as a result:

$$DOS(x,y) \wedge During(x,y) \rightarrow During(x,y) \quad (9)$$

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

$$Starts(x,y) \wedge During(x,y) \rightarrow \bot \quad (10)$$

The maximal tractable subset of Allen relations containing all basic relations when applying path consistency comprises of 868 relations [24]. Tractable subsets of Allen relations containing 83 or 188 relations [41] can be used instead, offering reduced expressivity but increased efficiency over the maximal subset of [24]. A tractable set of relations is a set of basic relations or disjunctions of basic relations with the following property: when asserted properties into the knowledge base are restricted to this set then a polynomial time algorithm such as path consistency can be used to infer all implied relations and detect all inconsistencies (i.e., the algorithms is sound and complete). This is not the case of arbitrary disjunctions of Allen relations, in this case exponential algorithms must be applied [1]. Furthermore, since the proposed temporal reasoning mechanism affects only relations of temporal intervals, it can be also applied to other temporal representation methods (besides 4D-fluents) such as N-ary relations. Reasoning operating on temporal instants rather on intervals is also feasible [41]. Specifically, qualitative relations involving instants form a tractable set if relation $\neq$ (i.e., a temporal instant is before *or* after another instant) is excluded. Reasoning involving relations between interval and instants is achieved by translating relations between intervals to relations between their endpoints [1].

Path consistency requires composition of properties, intersection of properties and role complement. Notice that disjointness of properties can be represented in terms of complement of properties (i.e., two properties are disjoint when one of them is *subproperty* of the *complement* of the second property). However, the combination of property composition, intersection and complement has been proven to be undecidable [32]. Instead of property complement, the disjointness of two properties can be represented as an *at most 0* cardinality constraint over their intersection. However, the intersection and the composition of two properties is a composite (i.e., not simple) property and applying cardinality constraints over composite properties has

been proven to be undecidable [16]. Therefore, reasoning using SWRL, as proposed in this work, is the only solution complying with current OWL specifications while retaining decidability.

Implementing path consistency over Allen relations requires minimizing the required additional relations and rules for implementing the mechanism. Existing work (e.g., [29]) emphasizes on determining maximal tractable subsets of relations, while practical implementations calls for minimizing of such relation sets (i.e., finding the minimal tractable set that contains the required relations). For example, implementing path consistency over the maximal tractable set of Allen relations [29], containing 868 relations is impractical, since defining all intersections and compositions of pairs of relations by means of SWRL rules requires millions of such rules.

In this work we propose the closure method of Table 1 for computing the minimal relation sets containing a tractable set of basic relations: starting with a set of relations, intersections and compositions of relations are applied iteratively until no new relations are produced. Since compositions and intersections are constant-time operations (i.e., a bounded number of table lookup operations is required at the corresponding composition tables) the running time of closure method is linear to the total number of relations of the identified tractable set. Applying the closure method over the set of basic Allen relations yields a tractable set containing 29 relations, illustrated in Section 4.2.

Table 1
Closure method

Input:Set S of tractable relations
Table C of compositions
WHILE S size changes
BEGIN
Compute C:Set of compositions of relations in S
S=S ∪ C
Compute I:Set of intersections of relations in S
S= S ∪ I
END
RETURN S

Notice that implementing path consistency using rules of the form of Equation 7 over n relations requires $O(n^3)$ rules (i.e., rules for every possible selection of three relations must be defined), while implementing path consistency using rules according to Equation 2 and Equation 8 (as implemented in this work) requires $O(n^2)$ rules since rules for every pair of

relations must be defined. Further improvements and reductions can be achieved by observing that the disjunction of all basic Allen relations when composed with other relations yields the same relation, while intersections yield the other relation. Specifically, given that All represents the disjunction of all basic relations and, R_x is a relation in the supported set then the following hold for every R_x:

$$All(x,y) \wedge R_x(x,y) \rightarrow R_x(x,y)$$
$$All(x,y) \wedge R_x(y,z) \rightarrow All(x,z) \quad (11)$$
$$R_x(x,y) \wedge All(y,z) \rightarrow All(x,z)$$

Since relation All always holds between two individuals, because it is the disjunction of all possible relations, all rules involving this relation, both compositions and intersections, do not add new relations into the ontology and they can be safely removed. Also, all rules yielding the relation All as a result of the composition of two supported relations R_{x1}, R_{x2}:

$$R_{x1}(x,y) \wedge R_{x2}(y,z) \rightarrow All(x,z) \quad (12)$$

can be removed as well. Thus, since intersections yield existing relations and the fact that the disjunction over all basic relations must hold between two intervals, all rules involving the disjunction of all basic relations and consequently all rules yielding this relation can be safely removed from the knowledge base. After applying this optimization the required number of axioms for implementing path consistency over the minimal tractable set of Allen relations is reduced to 983.

4.2. Supported Set of Tractable Allen Relations

The following is the set of tractable Allen relations used for implementing the reasoning mechanism of Section 4.1. Relations *Before, After, Meets, Metby, Overlaps, Overlappedby, During, Contains, Starts, Startedby, Finishes, Finishedby* and *Equals* are represented using symbols B, A, M, Mi, O, Oi, D, Di, S, Si, F, Fi and Eq respectively. These are basic Allen relations or disjunctions of basic relations represented as a set of relations into brackets:
{B},{A},{A, D, Di, O, Oi, Mi, S, Si, F, Fi, Eq}, {A, D, Oi, Mi, F}, {A, Di, Oi, Mi, Si}, {A, Oi, Mi}, {B, D, Di, O, Oi, M, S, Si, F, Fi, Eq}, {B, D, O, M, S}, {B, Di, O, M, Fi}, {B, O, M}, {D}, {D, Di, O, Oi, S, Si, F, Fi, Eq}, {D, Oi, F}, {D, O, S}, {Di}, {Di, Oi, Si},

{Di, O, Fi}, {Eq}, {F}, {F, Fi, Eq}, {Fi}, {M}, {Mi}, {O}, {Oi}, {S}, {S, Si, Eq}, {Si}.

4.3. Reasoning over Point-Based Representations

In the following, we propose a reasoner relying on the instants-based representation suggested in Section 3. The possible relations between temporal instants are $before$, $after$ and $equals$, denoted as "<",">","=" respectively. Table 2 illustrates the set of reasoning rules defined on the composition of existing relation pairs.

Table 2
Composition Table for point-based temporal relations.

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 Table 2 corresponding to row 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 cannot 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 do not yield new information these rules are discarded. Rules corresponding to compositions of relations R_1 *and* R_2 yielding a unique relation R_3 as a result are retained (7 out of the 9 entries of Table 2 are retained) and are expressed in SWRL using rules of the form (Equation 2):

$$R_1(x,y) \wedge R_2(y,z) \rightarrow R_3(x,z) \quad (13)$$

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

$$before(x,y) \wedge equals(y,z) \rightarrow before(x,z) \quad (14)$$

Therefore, 7 out of the 9 entries in Table 2 can be expressed using SWRL rules, while the two remaining entries do not convey new information. A series of compositions of relations may imply relations which are inconsistent with existing ones. Consistency check-

ing is achieved by imposing path consistency [41]. Path consistency is implemented by iteratively applying formula of Equation 7. 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 of Equation 8:

$$R_1(x,y) \wedge R_2(x,y) \rightarrow R_3(x,y) \qquad (15)$$

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:

$$ALL(x,y) \wedge before(x,y) \rightarrow before(x,y) \quad (16)$$

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

$$before(x,y) \wedge after(x,y) \rightarrow \bot \qquad (17)$$

As shown in Table 2, compositions of relations may 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 existing 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 by declaring the three basic relations as disjoint. Notice that path consistency is sound and complete when applied on the three basic relations [41].

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, this is not compatible with OWL 2 specification: relation $before$ must be declared as transitive in order to infer implied relations and disjoint with $after$, its inverse relation, (also $before$ is asymmetric and irreflexive) in order to

detect inconsistencies. However, OWL specifications[9] disallow the combination of transitivity and disjointness (or asymmetry) axioms on a property since they can lead to undecidability [14]. This restriction is necessary in order to guarantee decidability of the basic reasoning problems for OWL 2 DL. Thus, reasoning and consistency checking requires the use of SWRL rules.

In cases where temporal information is provided as dates, the qualitative relations are specified using SWRL rules that apply on the quantitative representation. An example of such a rule is the following:

$$Instant(x) \wedge Instant(z) \wedge inXSDDateTime(x,y)$$
$$\wedge inXSDDateTime(z,w) \wedge lessThan(y,w)$$
$$\rightarrow before(x,z)$$

$$(18)$$

Replacing the $lessThan$ operator in the rule with $greaterThan$ and $equal$ yields the corresponding rules for relations $after$ and $equals$ respectively. These qualitative relations can be combined with asserted and inferred qualitative relations using path consistency.

All interval relations can be represented by means of point relations between their end-points. Rules implementing transformation of Allen relations to endpoint relations and rules yielding Allen relations from endpoint relations have been implemented as well. For example, the rule yielding the *During* Allen relation from endpoint relations is the following:

$$ProperInterval(a) \wedge ProperInterval(x) \wedge before(b,y)$$
$$\wedge before(z,c) \wedge hasBeginning(a,b)$$
$$\wedge hasBeginning(x,y) \wedge hasEnd(a,c) \wedge hasEnd(x,z)$$
$$\rightarrow intervalDuring(x,a)$$

$$(19)$$

Rules similar to the above, yielding all basic Allen relations are implemented. Notice that the inverse transformation cannot be expressed by a single SWRL rule: one Allen relation corresponds to four end-point relations and conjunctions at the rule head are not sup-

---

[9]http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/#The_Restrictions_on_the_Axiom_Closure

ported in SWRL. Conjunctions can be expressed as rules with identical antecedent part and different head. For example, the following rules represent the transformation of relation $IntervalOverlaps$:

$$hasBeginning(a, b) \wedge hasBeginning(x, y)$$
$$\wedge \, hasEnd(a, c) \wedge hasEnd(x, z) \qquad (20)$$
$$\wedge \, intervalOverlaps(x, a) \rightarrow before(z, c)$$

$$hasBeginning(a, b) \wedge hasBeginning(x, y)$$
$$\wedge \, hasEnd(a, c) \wedge hasEnd(x, z) \qquad (21)$$
$$\wedge \, intervalOverlaps(x, a) \rightarrow before(b, z)$$

$$hasBeginning(a, b) \wedge hasBeginning(x, y)$$
$$\wedge \, hasEnd(a, c) \wedge hasEnd(x, z) \qquad (22)$$
$$\wedge \, intervalOverlaps(x, a) \rightarrow before(y, b)$$

$$hasBeginning(a, b) \wedge hasBeginning(x, y)$$
$$\wedge \, hasEnd(a, c) \wedge hasEnd(x, z) \qquad (23)$$
$$\wedge \, intervalOverlaps(x, a) \rightarrow before(y, c)$$

Notice that if data consistency can be assured, then reasoning can be significantly speeded-up. In cases where all relations are specified quantitatively (i.e., by numerical values) reasoning with path consistency can be dropped. For example, for intervals with known end-points, all possible relations between them can be computed in quadratic time from their end-point dates. The computed set of relations is guaranteed to be consistent and reasoning is not needed.

If consistency checking is not needed (in case instance assertions do not contain conflicts, implied or direct), then temporal properties need not be declared disjoint. For example, if sequences of events are recorded using sensors, then there is a valid arrangement of the events on the axis of time (i.e., the sequence of their recording), thus their temporal relations are consistent by definition. In this case, reasoning can be achieved using OWL role inclusion axioms instead of SWRL rules that apply on the ontology TBox as well. Such axioms are of the form:

$$before \circ equals \sqsubseteq before \qquad (24)$$

All relation compositions can be defined similarly. Intersections of relations are not required in case of basic point algebra relations and if the consistency checking requirement is dropped, only OWL axioms are sufficient for implementing the reasoning mechanism.

In total, based on the reasoning mechanism, five different representations for points have been implemented:

– Quantitative Point Representation (**P1**): Relations are extracted by comparing date/time values using SWRL.
– Qualitative Only using SWRL (**P2**): Only qualitative point relations are asserted and reasoning using Path Consistency implemented in SWRL is applied.
– Qualitative Only using Role Inclusion Axioms (**P3**): Only qualitative point relations are asserted and reasoning using Path Consistency is implemented using OWL 2 Role Inclusion Axioms.
– Combined representation using SWRL (**P4**): Both dates and qualitative relations are asserted and reasoning mechanism combines rules from representations P1 and P2.
– Combined representation using OWL Role Inclusion Axioms (**P5**): Both dates and qualitative relations are asserted and reasoning mechanism combines SWRL rules from representations P1 and OWL axioms from P3.

Based on the above representations and rules for extracting interval relations from end-point relations five different interval representations have been implemented[10].

– Allen-based Interval Representation (**I1**): Qualitative Allen relations only are asserted directly between intervals (points are not used, see Figure 9) combined with the reasoning mechanism of Section 4.1.
– Quantitative Only-direct intervals (**I2**): Only dates/times are asserted attached directly to intervals (see Figure 7) and Allen relations are extracted by date/time comparisons.
– Quantitative Only using Points (**I3**): Only dates/times are asserted attached to Points representing endpoints of intervals (see Figure 8) and Allen relations are extracted by date/time comparisons.

---

[10]We have made all point and interval representations available on the Web at: https://github.com/sbatsakis/TemporalRepresentations

- Qualitative Only Point Based Interval representation (**I4**): Only qualitative relations between points (see Figure 8) are asserted and reasoning mechanism is based on Point reasoning rules from Section 4.3 and Allen extraction rules from Section 4).
- Combined qualitative/quantitative Interval representation (**I5**): Both dates and qualitative relations between points are asserted (see Figure 8) and date/time comparisons are combined with SWRL rules of Section 4.3 and Allen extraction rules).

Reasoning is achieved by employing DL-safe rules expressed in SWRL that apply on named individuals in the ontology ABox, thus retaining decidability while offering a sound and complete inference procedure for asserted temporal intervals. Furthermore, computing the rules has polynomial time complexity since tractable sets of relations are supported [24,41].

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 the case of temporal instants. Furthermore, path consistency has $O(n^5)$ time worst case complexity (with n being the number of intervals or instants) and is sound and complete [30].

In the most general case, where disjunctive relations are supported in addition to the basic ones, any interval (or instant) can be related to 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 (point algebra), qualitative relations on time instants form a tractable set [41] (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. Thus, the proposed reasoning method can be extended with disjunctive relations such as $\geq$ denoting that an instant is *after or equals* to another. Applying the *closure method* over temporal Allen relations the minimal tractable sets containing the basic relations consist of 29 relations [6]. For this set the required number of OWL axioms and SWRL rules is 983 [6]. Reasoning over basic point algebra relations does not require additional relations and a total of 20 axioms are adequate for implementing path consistency [8].

The $O(n^5)$ upper limit referred to above is obtained as follows: At most $O(n^2)$ relations can be added in the knowledge base. At each such addition step, the reasoner selects 3 variables among n intervals (or points or regions) which corresponds to $O(n^3)$ possible different choices. Clearly, this upper bound is pessimistic, since the overall number of steps may be lower than $O(n^2)$ because an inconsistency detection may terminate the reasoning process early, or the asserted relations may yield a small number of inferences. Also, forward chaining rule execution engines employ several optimizations (e.g., the Rete algorithm employed at the SWRL implementation of Pellet as presented at [19]), thus the selection of appropriate variables usually involves fewer than $O(n^3)$ trials. Nevertheless, since the end user may use any reasoner supporting SWRL, a worst case selection of variables can be assumed in order to obtain an upper bound for complexity. Nevertheless, retaining control over the order of variable selection and application of rules yields an $O(n^3)$ upper bound for path consistency [37].

Both 4D-fluents and N-ary approaches can be used for representing dynamic properties, and both of them suffer from proliferation of objects. On the other hand, N-ary relations representation is more compact than 4D-fluents since fewer additional objects are required as illustrated in Figure 4 and Figure 3. A detailed comparison of these approaches is presented in [11,33] illustrating the disadvantages of 4D-fluents compared to N-ary approach because of the additional required objects.

The required expressiveness of the proposed representations is within the limits of OWL 2 expressiveness combined with SWRL and date/time datatypes. Thus, reasoners such as Pellet and HermiT can be used for reasoning. Reasoning mechanism is tractable since it consists of date/time comparisons and/or path consistency using SWRL [24]. Orthogonal to the problem of representing dynamic properties using 4D-fluents or N-ary relation is the representation of points and intervals. A summary of all proposed representations is presented in Table 3.

Notice that quantitative only approaches don't need to perform consistency checking since date/time assertions represent a valid instantiation of such values, while qualitative assertions my impose restrictions that cannot be satisfied. To the best of our knowledge, HermiT and Pellet are the only reasoners currently supporting SWRL, while only Pellet currently supports date/time comparisons needed for SWRL rules used by quantitative approaches.

Table 3
Comparison of Point and Interval Representations

Representation	P1	P2	P3	P4	P5	I1	I2	I3	I4	I5
Qualitative	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Quantitative	Yes	No	No	Yes	Yes	No	Yes	Yes	No	Yes
Reasoning Support: HermiT (H), Pellet (P)	P	H,P	H,P	P	P	H,P	P	P	H,P	P
Consistency Checking	N/A	Yes	No	Yes	No	Yes	N/A	N/A	Yes	Yes

5. Evaluation

Measuring the efficiency of the proposed representations requires temporal intervals and points as defined in Section 3, containing instances. Evaluation of quantitative representations was done by asserting actual dates/times and reasoning using Pellet[11]. Qualitative approaches required qualitative assertions which were extracted using dates from quantitative representations and retaining n such extracted relations for datasets of size n. Thus, datasets of various sizes containing points and intervals, both qualitative (using relations) and quantitative (using dates) or both were used for the experimental evaluation. Reasoning response times of the temporal reasoning rules were measured as the average over 10 runs. HermiT 1.3.8 and Pellet 2.3.0 running as a Java library were the reasoners used in the experiments. All experiments were run on a PC, with Intel Core CPU at 2.4 GHz, 6 GB RAM, and Windows 7.
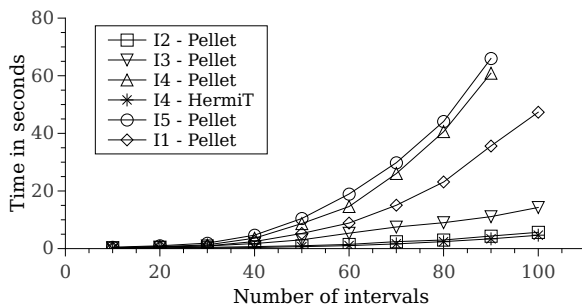

Fig. 10. Average reasoning time as a function of the number of intervals

Measurements illustrate that there are major differences in performance between various approaches, and reasoners. Interval representations can be used for reasoning over 100 intervals, while qualitative represen-
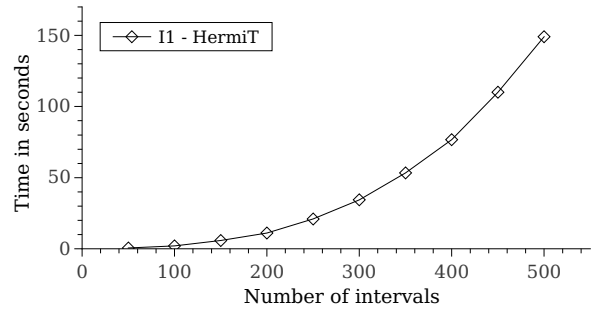

Fig. 11. Average reasoning time as a function of the number of intervals

tation combined with HermiT reasoner (representation I1 with HermiT, presented in Figure 11) can reason over 500 intervals in 149.07 seconds when using Allen relations directly (representation I1). For 100 intervals corresponding time using I1 and HermiT is 2.03 seconds respectively (see Figure 11), clearly outperforming all interval representations (I2-I5 and I1 with Pellet) of Figure 10. Notice that the direct representation using Allen intervals (I1) is faster using both Pellet and HermiT than the qualitative representation involving points (I4), while I4 is more complex than I1 and supports both points and intervals.
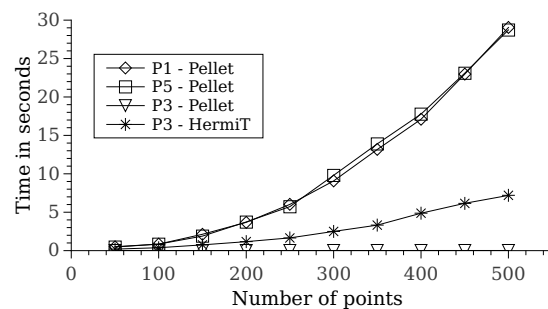

Fig. 12. Average reasoning time as a function of the number of points

Point representations can be used for reasoning over 500 points efficiently (see Figure 12), except the qualitative representations using SWRL -P2 and P4- and

---

[11]Dates were extracted from the dataset representing marriages at freebase (Retrieved in March 2015): http://www.freebase.com/people/marriage?instances=
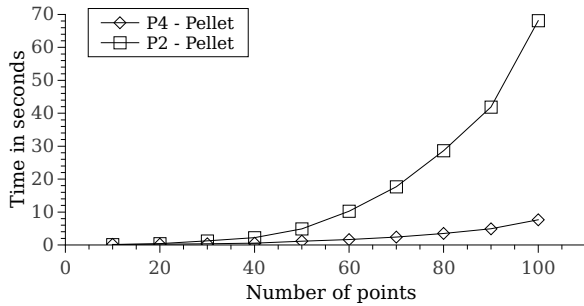
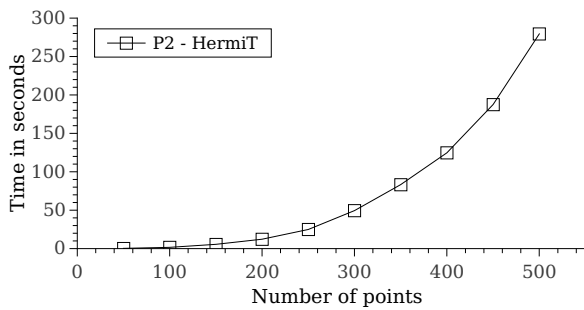Fig. 13. Average reasoning time as a function of the number of points

Fig. 14. Average reasoning time as a function of the number of points

Pellet, which can be practically used for at most 100 points. Reasoning time over these two representations is presented in Figure 13. When combining representation P2 with HermiT instead of Pellet reasoning over 500 points can be achieved in 279.47 seconds (see Figure 14). This is slower than all measurements for representations presented in Figure 12.
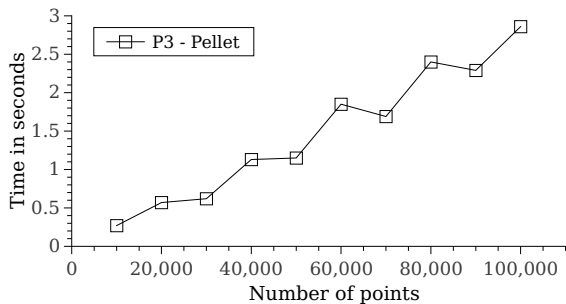

Fig. 15. Average reasoning time as a function of the number of points

An interesting case is the representation based on Role Inclusion Axioms (P3) that can be used for reasoning over 100K points in less than 3 seconds when using Pellet (see Figure 15), but not when using Her-

miT (see Figure 12), being orders of magnitude faster than all other approaches. This illustrates that there is clearly room for optimization on SWRL implementations of current reasoners.

This result indicates that when inconsistency detection is not required (i.e., assertions are guaranteed to be correct, thus only inference and not inconsistency detection is required) the implementation based on OWL axioms is faster and can be preferred. Optimizations employed in reasoning engines such as Pellet over OWL axioms result in faster reasoning times than reasoning using SWRL when the two approaches are directly comparable (i.e., when inconsistency detection is not required). Also, the OWL axioms-based approach applies on the TBox of the ontology, thus on implied anonymous individuals and concept definitions, and is not restricted to asserted named individuals as the SWRL-based reasoning mechanism.

In conclusion, experimental evaluation indicates that there are differences in performance between reasoners such as Pellet and HermiT (see for example Figures 10 and 11), which means that the proposed representations will directly benefit from future optimizations in rule engines. This is also illustrated by the fact that the OWL axiom based representation can support fast reasoning over 100K points (see Figure 15). An alternative approach instead of optimizing rule engines of reasoners such as Pellet is to build specialized standalone temporal reasoners that offer increased performance over existing SWRL based approaches. CHRONOS reasoner [2] is such a standalone reasoner based on path consistency. CHRONOS can reason over 10K Allen interval relations in less than 10 seconds [3].

6. Applications

Ontology editors, such as Protégé[12] are particularly well suited for crafting (creating, editing) static ontologies with binary relations, but have no means for dealing with temporal entities and temporal (ternary) relations. In [26,27] we presented CHRONOS-Ed, a Tab widget plug-in for the Protégé editor that facilitates handling of temporal ontologies such as definition of temporal classes and of temporal properties. It is portable and easy to use (i.e., handles temporal ontologies similarly to the way static ontologies are cre-

---

[12]http://protege.stanford.edu/

ated and handled in Protégé) and does not require the user to be familiar with the peculiarities of the underlying representation model of temporal information (i.e., the N-ary relations model). Temporal ontologies, can still be exported in OWL and handled (i.e., viewed or modified) by standard OWL editors (although much more difficult to handle in this case). CHRONOS-Ed interface is consistent with the layout of the default Protégé Tabs. We have made CHRONOS-Ed available on the Web[13].

CHRONOS-Ed supports adding restrictions on temporal properties, classes and individuals (e.g., "an employee can't work for two different companies at the same time"). Notice that if there are inconsistencies within a set of defined temporal relations, normally, these will not be detected by a conventional OWL reasoner (i.e., a reasoner for static ontologies such as Pellet in Protégé) or, an OWL reasoner might not compute all temporal inferences. The problem is that property restrictions defined on temporal classes now refer to the new classes introduced by the N-ary relations model rather than to the classes on which they were meant to be defined. Dealing with such issues calls for reasoning rules capable of handling temporal information in OWL with the N-ary relations model as the one we presented in [7], where we proposed a mechanism for handling OWL property restrictions and semantics over temporal representations in conjunction with the 4D-fluents and the N-ary relations approaches. Property semantics are expressed by a set of SWRL rules defined over temporal relations (rather than by OWL axioms as it is typical in DL ontologies). To the best of our knowledge, this is the only known solution to this problem.

CHRONOS-Ed plug-in was used for the development of SybillaTUC [40], a recommendation system for monitoring the condition of patients suffering from the Bipolar Disorder. It is designed to represent and manage the information about patient's medical record and the modelling of the disease evolution. Combining the clinical guidelines for Bipolar Disorder with a patient's medical record, SybillaTUC can predict the evolution of each patient, alert the clinician on the possibility of a critical incident and propose the best treatment suggested in the clinical practice guidelines asserted into the system, using the N-ary representation for the implementation of a dynamic ontology encoding experts knowledge for the management of patients

along with a SWRL reasoner for inferring recommendations for best treatment of patients based on their current condition and examination tests.

7. Conclusions and Future Work

We introduce a framework for handling temporal information in ontologies. The proposed framework handles both, time instants and time intervals (and also semi-closed intervals) equally well using a sound and complete inference procedure based on path consistency. Two alternative representations based on the 4D-fluents and the N-ary relations respectively are presented. It is compliant with existing Semantic Web standards (OWL 2) and W3C member submissions (SWRL) which increases its applicability. Being compatible with W3C standards and member submissions the proposed framework can be used in conjunction with existing editors, reasoners and querying tools such as Protégé and Pellet without requiring specific additional software.

Directions for future work include: Addressing scalability issues by applying optimizations tailored for specific datasets in large scale applications. Optimizations (e.g., parallelization) can apply on both reasoning and querying process. For example, indexing mechanisms for quantitative datasets can be applied in certain applications following the example of [39,28].

Also, proposing extensions on the OWL specification (e.g., by combining them with Temporal Description Logics) that will increase expressivity and compactness of temporal representations is a direction for future work. An example of this approach is TOWL[14] [10] which handles only quantitative defined temporal information by means of concrete domains. Integrating expressions such as "always" or "until" into OWL syntax and semantics will offer direct representation of qualitative temporal information. Finally, developing tools for providing reasoning and querying support for such extensions is a direction for future work.

References

[1] J.F. Allen, Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26[11]:pp. 832–843, 1983.

---

[13] Available at: http://www.intelligence.tuc.gr/prototypes.php

[14] TOWL (IST-STREP FP6, No. 026896): "Time-determined ontology based information system for real time stock market analysis (2006 - 2008)", Project Coordinator Euripides G.M. Petrakis, http://cordis.europa.eu/ist/kct/towl_synopsis.htm

[2] E. Anagnostopoulos, E.G. Petrakis, and S. Batsakis,. Chronos: A Reasoning Engine for Qualitative Temporal Information in OWL. Procedia Computer Science 22 (2013): pp. 70–77, 2013.

[3] E. Anagnostopoulos, E.G. Petrakis, and S. Batsakis, . CHRONOS: Improving the Performance of Qualitative Temporal Reasoning in OWL. In Tools with Artificial Intelligence (ICTAI), IEEE 26th International Conference on, pp. 309–315. IEEE, 2014.

[4] A. Artale and E. Franconi, A Survey of Temporal Extensions of Description Logics. Annals of Mathematics and Artificial Intelligence, 30[1]:pp. 171–210, 2000.

[5] F. Baader, Description Logics. In Reasoning Web: Semantic Technologies for Information Systems, 5th International Summer School 2009, 5689 of Lecture Notes in Computer Science, pp. 1–39. Springer–Verlag, 2009.

[6] S. Batsakis and E.G.M. Petrakis, SOWL: A Framework for Handling Spatio-Temporal Information in OWL 2.0. 5th International Symposium on Rules: Research Based and Industry Focused (RuleML' 2011), pp. 242–249, 2011.

[7] S. Batsakis and E.G.M. Petrakis, Imposing Restrictions over Temporal Properties in OWL: A Rule-Based Approach. 6th International Symposium on Rules: Research Based and Industry Focused (RuleML' 2012), pp. 240–247, 2012.

[8] S. Batsakis, K. Stravoskoufos, and E.G.M. Petrakis, Temporal Reasoning for Supporting Temporal Queries in OWL 2. 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES' 2011), [6881]:pp. 558–567, 2011.

[9] V. Ermolayev, S. Batsakis, N. Keberle, O. Tatarintseva and G. Antoniou, Ontologies of Time: Review and Trends. In International Journal of Computer Science and Applications, Vol. 11, No. 3, pp. 57–115, 2014.

[10] F. Frasincar, V. Milea, and U. Kaymak, tOWL: Integrating Time in OWL. Semantic Web Information Management: A Model-Based Perspective, pp. 225–246, 2010.

[11] A. Gangemi and V. Presutti, A multi-dimensional comparison of ontology design patterns for representing n-ary relations. In SOFSEM 2013: Theory and Practice of Computer Science, pp. 86–105. Springer Berlin Heidelberg, 2013.

[12] C. Gutierrez, C. Hurtado, and A. Vaisman, Introducing Time into RDF In Knowledge and Data Engineering, IEEE Transactions on 19.2:pp. 207–218, 2007.

[13] J.R. Hobbs and F. Pan, Time Ontology in OWL. W3C Working Draft, September 2006, 2006.

[14] I. Horrocks, O. Kutz, and U. Sattler, The Even More Irresistible SROIQ. In Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), pp. 57–67, 2006.

[15] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member submission, (21), 2004.

[16] I. Horrocks, U. Sattler, and S. Tobies, Practical Reasoning for Expressive Description Logics. In Logic for Programming and Automated Reasoning, pp. 161–180. Springer, 1999.

[17] M. Kifer, and H. Boley. RIF overview. W3C Working Group Note, 2010.

[18] M. Klein and D. Fensel, Ontology Versioning on the Semantic Web. In Proceedings of the International Semantic Web Working Symposium (SWWS), pages 75–91. Citeseer, 2001.

[19] V. Kolovski, B. Parsia, and E. Sirin, Extending the SHOIQ (D) Tableaux with DL-Safe Rules: First Results. In Proceedings International Workshop on Description Logic, DL-2006. Citeseer, 2006.

[20] H.U. Krieger, A Temporal Extension of the Hayes/ter Horst Entailment Rules and an Alternative to W3C's N-ary Relations. In FOIS, pp. 323–335, 2012.

[21] C. Lutz,. Description logics with concrete domains-a survey. In Advances in Modal Logics, volume 4. King's College Publications, 2003.

[22] B. Motik, I. Horrocks, R. Rosati, and U. Sattler, Can OWL and Logic Programming Live Together Happily Ever After? 5th International Semantic Web Conference, ISWC 2006, pp. 501–514, 2006.

[23] B. Motik, P.F. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, et al., OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Recommendation, (27), 2009.

[24] B. Nebel and H.J. Burckert. Reasoning About Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. Journal of the ACM (JACM), 42[1]:pp. 43–66, 1995.

[25] N. Noy, A. Rector, P. Hayes, and C. Welty, Defining N-ary Relations on the Semantic Web. W3C Working Group Note, (12), 2006.

[26] A. Preventis ,X. Makri, E. Petrakis, and S. Batsakis, CHRONOS: A Tool for Handling Temporal Ontologies in Protege. In Proceedings of 24th International Conference on Tools with Artificial Intelligence (ICTAI 2012), pp. 460–467, 2012.

[27] A. Preventis ,X. Makri, E. Petrakis, and S. Batsakis, CHRONOS Ed: A Tool for Handling Temporal Ontologies in Protégé In International Journal on Artificial Intelligence Tools, Vol 23(4), 2014

[28] A. Pugliese, O. Udrea, and VS Subrahmanian, Scaling RDF with Time. In Proceeding of the 17th international conference on World Wide Web, pp. 605–614. ACM, 2008.

[29] J. Renz, Maximal Tractable Fragments of the Region Connection Calculus: A Complete Analysis. In International Joint Conference On Artificial Intelligence, (16), pp. 448–455. Citeseer, 1999.

[30] J. Renz and B. Nebel, Qualitative Spatial Reasoning using Constraint Calculi. Handbook of Spatial Logics, pp. 161–215, 2007.

[31] A. Rula, M. Palmonari, A. Harth, S. Stadtmuller and A. Maurino, On the diversity and availability of temporal information in linked open data. In The Semantic Web (ISWC) 2012, pp. 492–507. Springer Berlin Heidelberg, 2012.

[32] K. Schild, Towards a Theory of Frames and Rules. PhD thesis, Technische Universitaet Berlin, Berlin, Germany, 1989.

[33] A. Scheuermann, E. Motta, P. Mulholland, A. Gangemi and V. Presutti, An empirical perspective on representing time. In Proceedings of the seventh international conference on Knowledge capture, pp. 89–96. ACM, 2013.

[34] R. Shaw, R. Troncy, and L. Hardman, Lode: Linking Open Descriptions of Events. The Semantic Web, pp. 153–167, 2009.

[35] R. Shearer,B. Motik, and I. Horrocks, HermiT: A Highly-Efficient OWL Reasoner. In OWLED, vol. 432. 2008.

[36] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz, Pellet: A Practical OWL-DL Reasoner. Web Semantics: Science, Services and Agents on the World Wide Web, 5[2]:51–53, 2007.

[37] M. Stocker and E. Sirin, Pelletspatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. In CEUR Workshop Proceedings, 529. Citeseer, 2009.

[38] C. Tao, H.R. Solbrig, D.K. Sharma, W.Q. Wei, G.K. Savova, and C.G. Chute, Time-oriented question answering from clinical narratives using semantic-web techniques. In The Semantic Web (ISWC), pp. 241–256. Springer Berlin Heidelberg, 2010.

[39] J. Tappolet and A. Bernstein, Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, pages 308–322. Springer-Verlag, 2009.

[40] C. Thermolia, , E. Bei, and E.G.M. Petrakis, TeleCare of mental disorders by applying Semantic Web Technology. In Bioin-

formatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on, pp. 1–4. IEEE, 2013.

[41] P. Van Beek and R. Cohen, Exact and Approximate Reasoning about Temporal Relations. *Computational Intelligence*, 6[3]:132–144, 1990.

[42] C. Welty and R. Fikes. A Reusable Ontology for Fluents in OWL, In *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, pages 226–336, 2006.