Undefined 1 (2009) 1–5
IOS Press

Temporal Representation and Reasoning in OWL 2.0

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Sotiris Batsakis [a,*], and Euripides G.M. Petrakis [b]
[a] *School of Computing and Engineering,University of Huddersfield, Queensgate, Huddersfield, UK*
E-mail: S.Batsakis@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 extents 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" mechanisms. 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. Polynomial time complexity of temporal reasoning is achieved by restricting the supported sets of relations to "tractable" sets. Building upon existing Semantic Web standards (like OWL 2.0, 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 which automatically handle tasks that are typically handled manually 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 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 of their properties form ontologies, which are defined using the OWL

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

language [17]. Ontologies contain definitions of concepts and their properties by means of binary relations. The syntactic restriction of OWL to binary 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 object and the subject. Representing information evolving in time in ontologies, is the problem this work is dealing with.

We introduce an approach for handling temporal information in OWL [17] while being consistent with existing Semantic Web standards (e.g., OWL, SWRL [11]) and tools (e.g., Pellet reasoner [26]). The last is a basic design decision in our work. Earlier work by Welty and Fikes [31] 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 our work, this approach is extended in certain ways: The 4D- fluents mechanism is 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.

In our approach, SWRL and OWL 2.0 constructs (e.g., disjoint properties) are combined, offering a sound and complete reasoning procedure ensuring path consistency [30], an issue which is not examined in the original work by Welty and Fikes. The proposed reasoner handles both quantitative and qualitative information using tractable sets of relations on which path consistency applies. 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 an instant (or point) based representation as well.

Apart from 4D-fluents, a representation of both forms of temporal information (i.e., quantitative, qualitative) based on N-ary relations [19] is also proposed. Both approaches are thorough examined and evaluated. Reasoning is implemented using SWRL rules and is capable of inferring 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 [26]).

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 as well as related work in the field of temporal knowledge representation using description logics are discussed in the following.

2.1. Description Logics and OWL

Description Logics (DLs) are a family of Knowledge Representation languages that form the basis for the Semantic Web standards [3]. The basic components of a Description Logic formalism are the *concepts* or *classes*, their *properties* or *roles* and the *individuals* or *objects*. The expressiveness 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, expressiveness 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.

Description logics are a fragment of First Order Logic and resolution-based approaches (i.e., reasoning methods for first order logic) where initially employed for the required reasoning tasks. Recently, a shift towards the so called "tableaux" based reasoning is observed [3]. 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 triplets of the form *object-predicate-subject* (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 of the object and the subject of a property 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 expressiveness 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 expressiveness. Extending OWL-DL with the additional constructs leaded to the adoption of OWL 2 as the current Semantic Web standard [17].

SWRL[5] is the language for specifying rules applying on Semantic Web ontologies. *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, the rules are restricted to *DL-safe rules* [16] that apply only on named individuals in the ontology ABox.

[1]http://owl.man.ac.uk/factplusplus/
[2]http://clarkparsia.com/pellet/
[3]http://www.hermit-reasoner.com/
[4]http://www.sts.tu-harburg.de/ r.f.moeller/racer/
[5]http://www.w3.org/Submission/SWRL/

2.2. *Representation of Time*

Time can be conceptualized as discrete or continuous, linear or cyclical, absolute or relative, qualitative or quantitative. Also, time can be represented using time instances or intervals. Temporal concepts are represented by the OWL-Time ontology [9]. OWL-Time is an ontology of the concepts of time, but OWL-Time cannot 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 [30]. 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 temporal points or intervals.

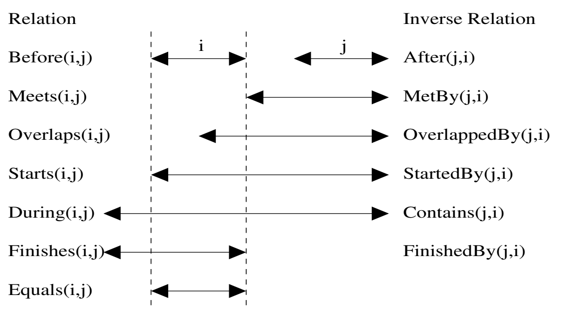

Fig. 1. Allen's Temporal Relations

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, 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 [23].

Inferring implied relations depends on existing relations in the knowledge base and on their semantics. 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 relation (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, compositions of pairs of relations are stored in *composition tables* [23].

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 can not 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 [23]. 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* [23]. Tractable subsets for point algebra have been identified in [30] and tractable sets of Allen interval algebra have been identified in [18].

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)* [2], *Concrete domains* [15], *Reification* [19], *Temporal RDF* [8], *Versioning* [13], *named graphs* [28] and *4D-fluents* [31].

Temporal Description Logics (TDLs) [2] 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 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 [8]).

Concrete Domains [15] 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 [7] 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 [13] 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, (c) it is not clear how the relation between evolving classes is represented.

Using an improved form of reification, the N-ary relations approach [19] suggests representing an n-ary relation as two properties each related with a new object (rather than as the object of a property). 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). This is illustrated in Figure 2. In the case of transitive properties additional triples are introduced as well.

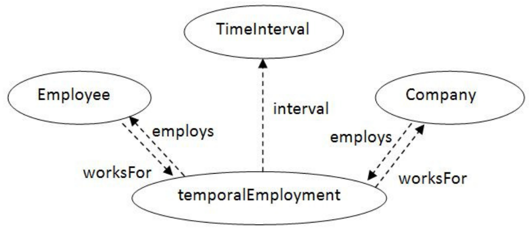

Fig. 2. Example of N-ary Relations

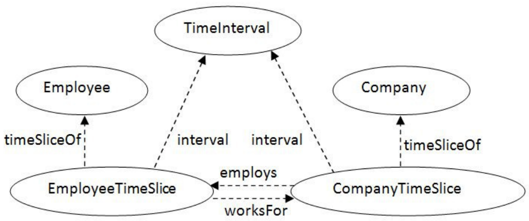

Fig. 3. Example of 4D fluents

The *4D-fluents* (perdurantist) approach [31] shows how temporal information and the evolution of temporal 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 3. 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 considered into this work.

3. Temporal Representation

We propose an ontology for representing and reasoning over dynamic information in OWL. Building upon well established standards of the semantic web (OWL 2.0, SWRL) the proposed ontology enables representation of static as well as of dynamic information based on the 4D-fluents [31] (or, equivalently, on the N-ary [19]) 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-fuents and the N-ary relations approaches are expanded to accommodate this information. The corresponding reasoner implements path consistency [23], 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 [31], to add time dimension to an ontology, classes *TimeSlice* and *TimeInterval* with properties *tsTimeSliceOf* 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*) connects an instance of class *TimeSlice* with an entity, and property *tsTimeInterval* (or the equivalent property *interval*) 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*.

Figure 4 illustrates a temporal ontology with classes *Company* (with datatype property *companyName*), *Product* (with datatype properties *price* and *product-*

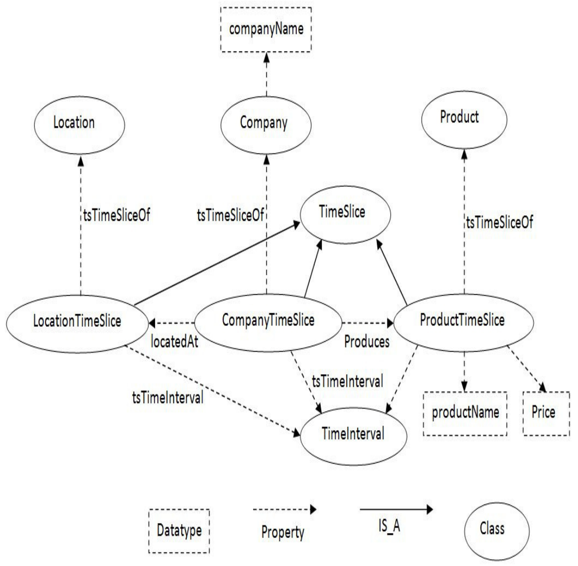

Fig. 4. Dynamic Enterprise Ontology

Name), and *Location* which represents spatial information. In this example, *CompanyName* is static property (its value does not change in time), while properties *produces*, *productName*, *locatedAt* and *price* are dynamic (fluent) 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* and are provided to denote that the domain of properties *produces*, *locatedAt*, *productName* and *price* are time slices restricted to be slices of a specific class. For example, the domain of property *productName* is not class *TimeSlice* but it is restricted to instances that are time slices of class *Product*. All fluent properties are defined as $subproperties$ of the property $fluent$.

In this work, the 4D-fluents representation is enhanced with qualitative temporal relations 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.

By allowing for qualitative relations the expressive power of the representation increases. Typically, the 4D-fluents model (similarly to other approaches such as Temporal RDF [8]), assume closed temporal intervals for the representation of temporal information, while semi-closed and open intervals can not 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 interval's 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 expressiveness compared to previous approaches [28,7] by combining 4D-fluents with Allen's temporal relations, their formal semantics and composition rules as defined in [1]. 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 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 it's starting and ending points using the $hasBegining$ and $hasEnd$ object properties respectively. 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 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 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" [30]. 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. 1 are rewritten as follows:

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

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.

In the original work by Welty and Fikes [31], 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 imposes the required restriction:

$$fluent(x, y) \wedge tsTimeInterval(y, z) \wedge tsTimeInterval(x, w) \rightarrow equals(w, z) \tag{1}$$

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 [25]. In our work, the temporal property remains a property relating the additional object with both the objects (e.g., an $Employee$ and a $Company$) involved in a temporal relation. This is illustrated in Figure 2. The representation of qualitative relations between temporal intervals or instants (and the corresponding reasoning mechanisms) remain identical to the 4D-fluents based version of the model.

Enforcing transitive properties is rather involved since the equality of the related intervals must also hold when a transitive property applies. This can be achieved using an SWRL rule such as in the case of 4D-fluents.

N-ary relations (similarly to 4D-fluents) require modifying the domains and ranges of fluent properties. Specifically, when a property is temporal, if the domain of property is $ClassA$ and the range is $ClassB$ (where domains and ranges can be composite class definitions or atomic concepts), then using the N-ary representation the domain becomes *ClassA OR Event* and the range *ClassB OR Event*. Compared to 4D-fluents, the disjunction of concepts appearing both in domain and ranges of properties limits specificity of references of the N-ary representation.

4. Temporal Reasoning

Temporal reasoning in this work is realized by introducing a set of SWRL [11] 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 [26] can be used for inference and consistency checking over temporal relations.

Specifically, reasoning is applied either on temporal intervals directly [4] or by applying point-based reasoning [6] 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 *Starts, Overlaps and During*. In our proposed approach, reasoning is realized by sets of rules corresponding to compositions of relations R_1, R_2 [6]. Rules yielding a unique relation R_3 as a result can be represented using SWRL as follows:

$$R_1(x,y) \wedge R_2(y,z) \rightarrow R_3(x,z) \tag{2}$$

An example of temporal inference rule is the following:

$$DURING(x,y) \wedge MEETS(y,z) \rightarrow BEFORE(x,z)$$

Rules yielding a set of possible relations cannot be represented directly in SWRL since, disjunctions of 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*) as a result:

$$OVERLAPS(x,y) \wedge DURING(y,z) \rightarrow During \vee Starts \vee Overlaps \tag{3}$$

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:

$$OVERLAPS(x,y) \wedge DURING(y,z) \rightarrow DOS(x,z)$$

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 [18,30]. 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 a result as follows:

$$\begin{aligned}
&DOS \circ During \rightarrow \\
&(During \vee Overlaps \vee Starts) \circ During \rightarrow \\
&(During \circ During) \vee (Overlaps \circ During) \\
&\vee (Starts \circ During) \rightarrow \\
&(During) \vee (During \vee Overlaps \vee Starts) \vee (During) \\
&\rightarrow During \vee Starts \vee Overlaps \rightarrow DOS
\end{aligned} \tag{4}$$

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 [23,18,30]. Path consistency is implemented by consecutive application of the formula:

$$\forall x,y,k\, R_s(x,y) \leftarrow R_i(x,y) \cap (R_j(x,k) \circ R_k(k,y)) \tag{5}$$

[6]We have made this representation available on the Web at http://www.intelligence.tuc.gr/prototypes.php

representing intersection of compositions of relations with existing relations. Symbol $\cap$ denotes intersection, symbol $\circ$ denotes composition and symbols R_i, R_j, R_k, R_s denote Allen relations. 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:

$$R_1(x,y) \wedge R_2(x,y) \rightarrow R_3(x,y), \tag{6}$$

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:

$$DOS(x,y) \wedge During(x,y) \rightarrow During(x,y).$$

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

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

The maximal tractable subset of Allen relations containing all basic relations when applying path consistency comprises of 868 relations [18]. Tractable subsets of Allen relations containing 83 or 188 relations [30] can be used instead, offering reduced expressiveness but increased efficiency over the maximal subset of [18]. 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 [30]. 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 [24]. 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 [12]. 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., [22]) 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 contain the required relations). For example, implementing path consistency over the maximal tractable set of Allen relations [22], 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 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 the Section 4.2.

Notice that, implementing path consistency using rules of the form of Equation 5 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 6 (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

```
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
```

Table 1

Closure method

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:

$$All(x,y) \wedge R_x(x,y) \rightarrow R_x(x,y)$$
$$All(x,y) \wedge R_x(y,z) \rightarrow All(x,z)$$
$$R_x(x,y) \wedge All(y,z) \rightarrow All(x,z) \quad (7)$$

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}:

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

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, ENDS, ENDEDBY 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.

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

Table 2

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 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):

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

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

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

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 checking is achieved by imposing path consistency [30]. Path consistency is implemented by iteratively applying formula of Equation 5. 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 6:

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

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 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 [30].

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 2.0 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[7] disallow the combination of transitivity and disjointness (or asymmetry) axioms on a property since they can lead to undecidability [10]. 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:

$$\begin{aligned} &Instant(x) \wedge Instant(z) \wedge inXSDDateTime(x, y) \\ &\wedge inXSDDateTime(z, w) \wedge lessThan(y, w) \\ &\rightarrow before(z, x) \end{aligned} \tag{8}$$

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

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

endpoint relations is the following:

$$ProperInterval(a) \wedge ProperInterval(x) \wedge before(b,y) \wedge before(z,c) \wedge hasBeginning(a,b) \wedge hasBeginning(x,y) \wedge hasEnd(a,c) \wedge hasEnd(x,z) \rightarrow intervalDuring(x,a) \tag{9}$$

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 supported 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 $IntervalOverlaps$:

$$hasBeginning(a,b) \wedge hasBeginning(x,y) \wedge hasEnd(a,c) \wedge hasEnd(x,z) \wedge intervalOverlaps(x,a) \rightarrow before(z,c) \tag{10}$$

$$hasBeginning(a,b) \wedge hasBeginning(x,y) \wedge hasEnd(a,c) \wedge hasEnd(x,z) \wedge intervalOverlaps(x,a) \rightarrow before(b,z) \tag{11}$$

$$hasBeginning(a,b) \wedge hasBeginning(x,y) \wedge hasEnd(a,c) \wedge hasEnd(x,z) \wedge intervalOverlaps(x,a) \rightarrow before(y,b) \tag{12}$$

$$hasBeginning(a,b) \wedge hasBeginning(x,y) \wedge hasEnd(a,c) \wedge hasEnd(x,z) \wedge intervalOverlaps(x,a) \rightarrow before(y,c) \tag{13}$$

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 does 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:

$$before \circ equals \sqsubset before$$

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.

5. Evaluation

In the following, the efficiency of our approach reasoning is assessed both, theoretically and experimentally. The purpose of the theoretical analysis is to show that reasoning retains soundness, completeness and tractability over the supported sets of relations.

5.1. Theoretical Evaluation

The required expressiveness of the proposed representation is within the limits of OWL 2 expressiveness. Notice that, an application might require additional expressiveness which can be evaluated once the application and its respective ontological representation has been analysed. This type of evaluation is outside the scope of the analysis discussed below.

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 [18,30].

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 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 [23]. In the most general case where disjunctive relations are supported in addition to the basic ones, any interval (or 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 (point algebra), qualitative relations on time instants form a tractable set [30] (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 [4]. For this set the required number of OWL axioms and SWRL rules is 983 [4]. Reasoning over basic point algebra relations does not require additional relations and a total of 20 axioms are adequate for implementing path consistency [6].

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 [14]), 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 [27].

5.2. Experimental Evaluation

To evaluate the performance of reasoning, we run several sets of experiments whose purpose is to demonstrate the run-time efficiency of the proposed reasoning approach as a function of the size of the data sets and its dependence on the type and peculiarities of the underlying ontological representation. The efficiency tests of the proposed reasoner require a temporal ontology. Because such an ontology is not available to us for these experiments, as a test-bed, we used data-set of 10 to 100 individuals generated randomly. In fact, these randomly generated data are used to populate 10 ontologies with random instances. Reasoning response times of the temporal reasoner, are measured as the average over 10 runs. Pellet 2.1.2 running as a plug-in of Protégé 4.1 beta was the reasoner used in the experiments. All experiments run on a PC, with Intel Core 2 CPU at 2.13 GHz and 1 GB RAM.

5.3. Temporal Reasoning

In this work, as we show in the experiments, the point-based representation is the preferred one for handling both temporal instants and intervals. Relations between intervals are expressed as a function of relations between their end-points. A representation relying on intervals is also implemented. However, since the number of basic relations is 13 (Figure 1) and because all possible disjunctions appearing in the supported tractable set must also be supported, the representation may become particularly involved. Notice also that, temporal instants, the same as semi-closed temporal intervals (i.e., intervals whose one of their end-points is undefined) cannot be represented efficiently in an interval-based representation. On the other hand, the interval based representation directly handles temporal intervals without the need of an intermediate translation to instants as the the point-based approach does. Also, the definition of n intervals requires $2n$ points (although this number can be reduced in the case of temporal intervals with end-points in common). A point-based representation has the advantage that, if intervals share end-points then, a reduction in the number of instances is achieved, since using n points (depending on the number of shared end-points) the number of derived intervals ranges from $n * (n-1)/2$ to $n/2$ in the worst case.

In the following experiment, we measure the performance of reasoning in the cases of both, an interval-based and a point-based representation and their performance is discussed. In both cases, n random instances with n random Allen or point relations defined between them were asserted, and reasoning times using Pellet are measured. This measurement does not provide a direct comparison between interval and point-based reasoning since, less than or exactly $2n$ points are required to define n intervals. Measurements of the time required by each approach for producing all inferred relations from a data set of random intervals or points are reported in Table 3. Each entry in the table is the average over 10 runs of the reasoner corresponding to 10 random instantiations of the ontology. Assertions were selected to form a consistent set of axioms, thus the reasoning procedure did not terminate early due to inconsistencies. If an inconsistency is detected by the reasoner, the inconsistent random instance is removed.

The evaluation indicated that the point-based approach is faster, although the number of temporal instants involved is typically larger than the number if intervals from which they are derived. The point-based representation requires fewer rules applied on a largest set of points (in the worst case) compared to the interval-based representation which requires 29 relations (the minimal tractable subset containing basic relations) and 983 OWL axioms and SWRL rules as opposed to just 20 axioms in the case of point algebra. Overall, point-based representations are more flexible (facilitate representation of time instants and semi-closed intervals and they are space efficient involving only 3 relations) and are preferred. Also, if consistency checking is not needed, (in case instance assertions does not contain implied or direct conflicts) then, temporal properties need not be declared disjoint, and reasoning can be achieved using OWL role inclusion axioms that apply to the ontology TBox instead of SWRL rules.

Reasoning over instant-based representations is faster due to the small number of axioms involved and the smaller number of inferences over a set of random relations.

Number of Individuals	Reasoning Time (ms)	
	Points	Intervals
10	134.9	793.8
20	162.9	1558.4
30	207.9	2340.6
40	245.2	4523.0
50	310.1	4728.1
60	282.5	5294.9
70	401.0	5558.2
80	497.2	8045.7
90	476.3	8098.7
100	496.8	11329.3

Table 3

Average reasoning time for point and interval algebra as a function of the number of points and intervals

5.4. *Comparative Evaluation of 4D-Fluent and N-ary representations.*

Table 4 represents the comparison of the number of assertions required for expressing a number of temporal relations using the 4D-fluents and the N-ary relations approach. Table 4 represents the average time (for 5 random generated ontologies for every instance set size) required for reasoning over temporal properties using the N-ary and the 4D-fluents approach as in Figure 3 and Fig. 2. Reasoning does not involve qualitative temporal relations as in Table 3. The representation involve both simple properties and properties that are transitive, symmetric or subproperties of other properties.

Number of fluent properties	4D-fluents		N-ary	
	Assertions number	Reasoning time (ms)	Assertions number	Reasoning time (ms)
100	1039	89.2	732	47.8
200	2039	156.0	1432	144.0
300	3039	169.2	2132	145.2
400	4039	191.4	2832	156.6
500	5039	241.6	3532	163.4
600	6039	313.0	4232	205.0
700	7039	317.6	4932	261.8
800	8039	365.8	5632	281.2
900	9039	435.8	6332	305.6
1000	1039	453.8	7032	358.0

Table 4

Reasoning time and required assertions for the 4D-fluents and N-ary representations as a function of the number of dynamic properties defined.

The quantitative comparison of the two approaches indicates that the N-ary approach outperforms the 4D-fluents representation in terms of required assertions (axioms) and consequently in reasoning time. This is attributed to the fact that fewer additional objects are required as illustrated in Figure 3 and Figure 2.

5.5. OWL Axioms-Based Implementation

In the case of basic point relations, when the requirement of inconsistency detection is dropped, an implementation based on OWL axioms is feasible (See Section 4.3). Table 5 represents the average time (for 10 random generated ontologies for every instance set size) required for point relations using SWRL rules and OWL axioms (only for inference and without inconsistency detection in the later case).

Number of temporal instants	Reasoning Time (ms)	
	SWRL	OWL axioms
10	134.9	14.9
20	162.9	19.9
30	207.9	18.2
40	245.2	23.0
50	310.1	19.4
60	282.5	21.8
70	401.0	36.1
80	497.2	25.5
90	476.3	33.9
100	496.8	32.5

Table 5

Average response time for reasoning using SWRL rules and OWL axioms as a function of the number of temporal instants.

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 much 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.

6. Applications

Ontology editors, such as Protégé[8] 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 [20] we present CHRONOS, 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 created 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 in this work). 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 interface is consistent with the layout of the default Protégé Tabs. We have made CHRONOS available on the Web[9].

CHRONOS 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 [5], where we propose 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 static ontologies). To the best of our knowledge, this is the only known solution to this problem.

CHRONOS plug-in was used for the development of SybillaTUC [29], 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

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

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

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 temporal 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 and evaluated. It is fully compliant with existing Semantic Web standards and specifications which increases its applicability. Being compatible with W3C specifications the proposed framework is compatible and can be used in conjunction with existing editors, reasoners and querying tools such as Protégé and Pellet without requiring specific additional software. Therefore, information can be distributed and shared without specific 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 the reasoning and the querying process. For example, indexing mechanisms for quantitative datasets can be applied in certain applications following the example of [28,21].

Also proposing extensions on the OWL specification (e.g., by combining them with Temporal Description Logics) that will increase expressiveness and compactness of temporal representations is a direction for future work. An example of this approach is TOWL[10] [7] 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.

[10]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

References

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

[2] A. ARTALE AND E. FRANCONI. A Survey of Temporal Extensions of Description Logics. *Annals of Mathematics and Artificial Intelligence*, **30**[1]:171–210, 2000.

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

[4] 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)*, pages 242–249, 2011.

[5] 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)*, pages 240–247, 2012.

[6] S. BATSAKIS, K. STRAVOSKOUFOS, AND E.G.M. PETRAKIS. Temporal Reasoning for Supporting Temporal Queries in OWL 2.0. *15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES' 2011)*, **6881**:558–567, 2011.

[7] F. FRASINCAR, V. MILEA, AND U. KAYMAK. tOWL: Integrating Time in OWL. *Semantic Web Information Management: A Model-Based Perspective*, pages 225–246, 2010.

[8] C. GUTIERREZ, C. HURTADO, AND A. VAISMAN. Temporal RDF. In *Second European Semantic Web Conference (ESWC 2005)*, pages 93–107, 2005.

[9] J.R. HOBBS AND F. PAN. Time Ontology in OWL. *W3C Working Draft, September 2006*, 2006.

[10] 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)*, pages 57–67, 2006.

[11] 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.

[12] I. HORROCKS, U. SATTLER, AND S. TOBIES. Practical Reasoning for Expressive Description Logics. In *Logic for Programming and Automated Reasoning*, pages 161–180. Springer, 1999.

[13] 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.

[14] 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.

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

[16] 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*, pages 501–514, 2006.

[17] 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.

[18] 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]:43–66, 1995.

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

[20] 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)*, Athens, Greece, November 7-9, 2012.

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

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

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

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

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

[26] 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.

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

[28] 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.

[29] C. Thermolia. Prediction of the Evolution of Bipolar Disorder using Semantic Web Technologies. *Diploma Thesis, Department of Electronic and Computer Engineering, Technical University of Crete*, 2012.

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

[31] 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.