Empirically Evaluating Three Proposals for Representing Changes in OWL2

J.-R. Bourguet [a] G. Guizzardi [a,b] A. Botti Benevides [a] and V. Zamborlini [c]
[a] *Ontology & Conceptual Modeling Research Group (NEMO)*
Federal University of Esprito Santo (UFES), Vitória – Brazil
[b] *Research Centre for Knowledge and Data (KRDB)*
Free University of Bozen-Bolzano (UNIBZ), Bozen-Bolzano – Italy
[c] *Institute for Logic, Language and Computation (ILLC)*
Universiteit van Amsterdam (UvA), Amsterdam – The Netherlands

Abstract. In almost all domains in practice, it is fundamental to properly represent entities amenable to changes. For instance, in business analytics, we must be able to reason with large amounts of time-changing KPI (key performance indicators) data. For this reason, general-purpose practical knowledge representation frameworks must be able to support the representation of temporally changing information and in a way that affords decidable automated reasoning. In this paper, we address the issue of representing entities amenable to intrinsic or extrinsic changes in OWL2. These sources of change are illustrated in a simplified model of the scholar domain. We then propose three strategies to represent entities amenable to changes as well as their changes. In particular, we do that by employing strategies that are based on a philosophical stance called perdurantism, which sees all individuals as 4D entities, i.e., as individuals that unfold in time as well as in space. Finally, we compare these three alternatives by generating synthetic instances and performing an empirical evaluation of reasoning tasks.

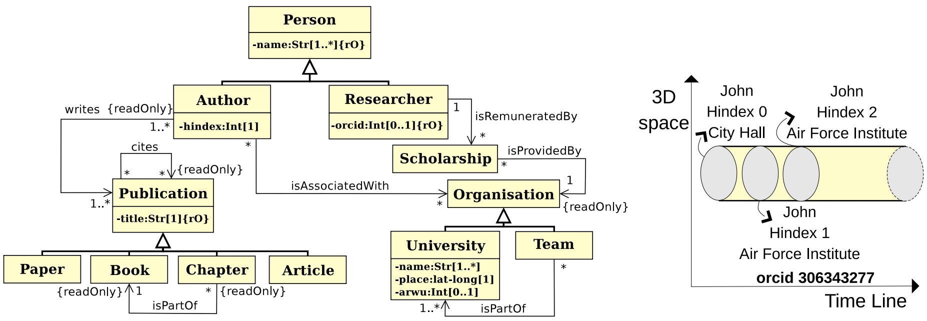
Keywords. Perdurantist representation, Temporally changing information, Formal Ontology, OWL, Empirical Assessment of Ontology Codification Alternatives

1. Introduction

It is of crucial importance for the Knowledge Representation community to provide means for the modeler to explicitly represent information in a declarative form that is suitable for performing reasoning tasks. For instance, there are attempts to apply time changing information-based models in a range of domains, including enterprise contracts [1], environnemental data integrations [2], stock analysis (buy-hold-sell) [3] or court proceedings, as litigations [4]. A classical problem is the trade-off between expressivity and time/space computational complexity of reasoning tasks. The OWL2 ontology representation language [5] is underpinned by restricted *DL* fragments, the majority of which were designed to preserve decidability and provide tractability. Nevertheless, OWL2 has been designed focusing on the representation of scenarios with immutable truth-values and unchangeable information about the world. Different approaches to cope with this issue have been propose in the literature. These include concrete domains, reifications, annotations, versioning, named graphs and perdurantism-based representations (see [6] and [7] for overviews). The perdurantist view claims that all entities have temporal parts and can be intuitively represented as four dimensional space-time worms whose temporal parts are time slices of the worm. This representation is relevant in business analytics for example where it is frequent to reason with large amounts of historical time-changing data. A series of works adopt an OWL-based perdurantist representation of the world. Welty and Fikes [8] introduced the idea of 4D fluents that provide temporal parts to each instance (extended toward ND fluents in [9]). Krieger et al [10,11] proposed a total perdurantist view by introducing the class of time slices as a superclass of both contingent and mandatory classes. Finally in [12,13], the authors present a proposal in relation with the ways properties and relations can evolve in time (e.g., (im)mutability). In this paper, we propose three alternatives based on the approach presented in [12,13] and perform an empirical evaluation to compare them. Our evaluation is based on a model representing the scholar domain, which illustrates some mutable aspects. Thus, we formalized this model in the three alternatives and generated synthetic instances in order to perform an empirical evaluation of some reasoning tasks. The remainder of this paper is structured as follows. In Section 2, we introduce the notions of mandatoriness, contingency, (im)mutability, and dependence, illustrated in a purely illustrative UML-like diagram of the scholar domain. Section 3 describes our three proposals for mapping UML-like diagram to TBoxes in OWL2. In Section 4, we present the results of our empirical evaluation of the aforementioned alternatives. Section 5 presents some related works. Finally, Section 6 presents some final considerations.

2. Modeling dynamic phenomena

Most Description Logics (*DLs*) (except the temporal DLs [14] for example) have been designed to represent immutable and unchangeable information capturing snapshots of the world. However, some applications require modelers to represent and reason over different kinds of information; for instance, decision support systems like business analytics usually require keeping track of historical changes in order to compute crucial indicators. We illustrate some kinds of mutable information by means of a model of scholar activities represented in Figure 1(a). In this domain, `Author`s (which are `Person`s) write `Publication`s, which can be classified into `Paper`s, `Article`s, `Chapter` or `Book`s. A `Publication` can be cited by another `Publication`, and `Author`s have also an hindex. Since 2012, orcid (open researcher and contributor id) can provide a persistent digital identifier to academic `Researcher`s. orcid was created as a response to several problems: authors may be called by different names through time (e.g. a marriage can append a name for an author); cultural differences can exist in naming people, in the ordering of names and surnames; and names can be written in different alphabets. Going back to our example, a `Researcher` can be remunerated by `Scholarship`s provided by one `Organisation` in such a way that an `Organisation` can provide several `Scholarship`s, and a `Scholarship` remunerates one `Researcher`. `Author`s and `Organisation`s can be associated. We specify two kinds of `Organisation`s: a `Team` and a `University`. A `Team` is part of one or more `Universitie`s, and a `University` can be ranked with regard to arwu (academic ranking of world universities), also known as Shanghai ranking, an annual publication of `University` rank. This model includes time changing and obsolete information that should be properly represented. For example, once a name form for a `Person` is used in the head of a `Publication`, this occurrence will always refer to the specific `Person`; this situation is different for an email address, which can be replaced or removed. Moreover, indicators such as hindex and arwu are typically volatile information and also seems pertinent to keep track of their historical changes to support reasoning on issues such as causality or correlation. Also, hindex is an attribute that is always available (0 by default and monotonic afterwards), while arwu is non mandatory for a university (there can be universities not ranked by this metric), and a `University` can change its name or place. Finally, an association between an `Author` and an `Organisation` can also cease to hold. One can notice three sources of changes in Figure 1(a): attributes, relations and class instantiations.



(a) Domain (b) Individuals

Figure 1. Application case

Concerning attributes and relations, we highlight two characteristics: (i) *mandatoriness* vs. *contingency*, and (ii) *mutability* vs. *immutability*. (i) is usually represented in a class diagram by means of cardinality constraints, where a cardinality greater or equal to one ensures that the attribute or relation is mandatory, otherwise it is contingent (optional). For example, if an `Author` exists, her name and hindex are mandatory, while her orcid is optional. Also, it is mandatory for a `Scholarship` to remunerate one `Researcher`, while it is contingent for a `Researcher` to be remunerated by a `Scholarship`. (ii) can be represented in a class diagram by placing or not {readOnly} close to the corresponding immutable attribute or association end. If an attribute is immutable, once the value of the attribute is set, it cannot change; and given a relation R from a class A to a class B and s.t. the association end near B is tagged as immutable, once an instance x of A starts to relate to $y_1, \ldots, y_n$ via R, then x cannot start an R-relationship with any other y_i, and no $xRy_1, \ldots, xRy_n$ can cease to hold until x ceases to exist. For example, if an `Author` exists, orcid is immutable, while hindex is mutable. Similarly, a `Scholarship` cannot change its isRemuneratedBy relation from a `Researcher` to another (it must always remunerate the same `Researcher`), while a `Researcher` can cease to be remunerated by a `Scholarship` without ceasing himself to exist.

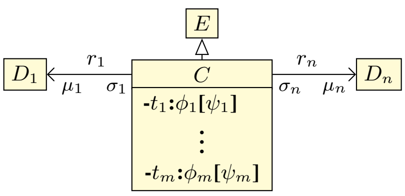
One can ground such notions on Formal Ontology (see for example [15]). A *generic dependence* holds between an individual x via relation R to a type T when, in order to exist, an individual x has to be R-related with an instance of T. A *specific depen-*

dence holds from an individual x to y iff x cannot exist when y does not exist. Generic dependence is expressed by attributes/relations having a cardinality greater or equal to one, i.e., mandatory attributes/relations. On the other hand, by assuming that whenever a relationship xRy holds, x and y must exist, specific dependence is entailed by means of attributes/relations that are tagged as {readOnly} and having a cardinality greater or equal to one, i.e., mandatory immutable attributes/relations. Moreover, some class instantiations must always hold for its individuals, while others do not necessarily have to hold. For example, an instance of `Person` cannot cease to instantiate it without ceasing to exist, while a `Researcher` can cease to be a `Researcher` without ceasing to exist. This property of the class `Person` is called *rigidity* [15, Ch. 4]. The property of the class `Researcher` is called *anti-rigidity*[1] as it requires an entity that instantiates `Researcher` at a time t to not instantiate this class at a different time t'. We highlight here that the ontological choices made at this example are intuitive, but arguable. Our aim was to illustrate ontological notions, not to propose an ontological analysis of the scholar domain.

3. Representing changes in OWL

In this section, we present some alternatives to represent changes in *DL*. First, in order to clarify the choices in the following sections, we show a static-world mapping that is incapable of dealing with changes. We introduce here a UML interpretation for $\mathcal{SROIQ}^{\mathcal{D}}$, the fragment of *DL* underpinning OWL2. We denote $\mathbf{C}$ a set of concepts or classes, $\mathbf{R}$ a set of object properties or relations, $\mathbf{R_T}$ a set of data properties or attributes, Φ_T a set of datatypes or attributes types, S a set of symbols from the alphabet of *DL*s (see [16] for their interpretations in first order logic), Ω a set of UML cardinalities and a function $\ell(\Omega \rightarrow S)$ such that $\ell(*) \mapsto \forall$, $\ell(n) \mapsto \forall_{=n}$, $\ell(0..n) \mapsto \forall_{\leq n}$ and $\ell(n..*) \mapsto \forall_{\geq n}$ (with $n > 0$).[2]

Definition 1. Let $\{C, D_1, \ldots, D_n, E\} \subseteq \mathbf{C}$, $\{r_1, \ldots, r_n\} \subseteq \mathbf{R}$, $\{t_1, \ldots, t_m\} \subseteq \mathbf{R_T}$, $\{\phi_1, \ldots, \phi_m\} \subseteq \Phi_T$ and $\{\mu_1, \ldots, \mu_n, \sigma_1, \ldots, \sigma_n, \psi_1, \ldots, \psi_m\} \subseteq \Omega$, an UML interpretation U is defined below:



$$\Leftrightarrow \quad \begin{gathered} \Big(C \sqsubseteq E \sqcap \prod_{i=1}^{n} \ell(\mu_i) r_i.D_i \sqcap \prod_{j=1}^{m} \ell(\psi_j) t_j.\phi_j\Big)^{\mathrm{U}} \\ (D_1 \equiv \ell(\sigma_1) r_1^{-1}.C)^{\mathrm{U}} \\ \vdots \\ (D_n \equiv \ell(\sigma_n) r_n^{-1}.C)^{\mathrm{U}} \end{gathered}$$

This mapping cannot represent any change, thus assuming a world in which everything is static and immutable, i.e., attributes and relations cannot change, including the relation of instantiation between individuals and classes. The fundamental dichotomy Endurant vs. Perdurant, between types of individuals, appears in the systems of categories of Foundational Ontologies like UFO [17], BFO [18] and DOLCE [19] and have already been employed to address the issue of changing information in DL (e.g. Descriptions and Situations ontology [20]). Figure 1(b) shows the representation of a 3D en-

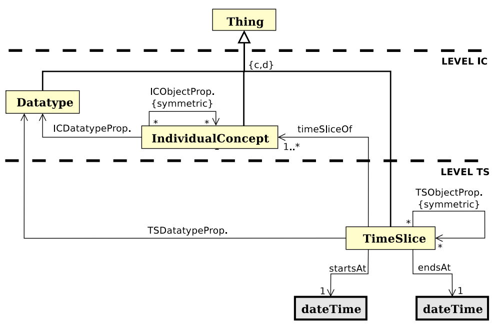
[1] *Anti-rigidity* is stronger than *non-rigidity*, the logical negation of *rigidity*.

[2] We denote: $\forall_{=n} r.C \triangleq \forall r.C \sqcap {=}nr.C$, $\forall_{\geq n} r.C \triangleq \forall r.C \sqcap {\geq}nr.C$ and $\forall_{\leq n} r.C \triangleq \forall r.C \sqcap {\leq}nr.C$.

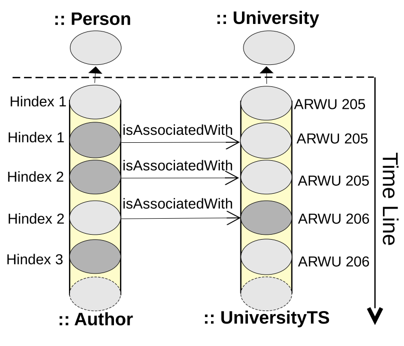
durant with a fourth temporal dimension, where an individual named John is represented as a 4D object—also called a *space-time worm*—whose slices are snapshots of John's orcid 0000-0003-0634-3277 during his life as an author. An endurant, such as a person, is fundamentally different from what is called a perdurant (or process), which has temporal parts unfolding in time, e.g., a flight, a conference, or a PhD defense. Intuitively, endurants *exist* at times, while perdurants *happen* at times.

Contrarily to endurantism, the perdurantist approach removes the distinction between endurants and perdurants by defending that "objects are composed of so-called temporal parts. When we see an object here and now, we are seeing the parts of it that are now — but there are other parts of it at other times that we might have encountered or might yet encounter." [21]. While objects are seen as 3D endurants through the endurantist approach, they appear otherwise as perdurantist worms, i.e., four dimensional "space-time worms" whose temporal parts are slices (snapshots) of the worms. In the following, we present the temporally changing information frameworks proposed in [12,13]. We illustrate this framework in our domain by snapshots of Mary's life during her existence as an author.

First introduced by Leibniz [22], the notion of "individual concept" allows the mapping of an individual to all its snapshots (or time slices), whenever it exists, by referring to a single characteristic (or set of characteristics). These characteristics, said *essential* (i.e., necessary and immutable), define the identity of an individual [15] (e.g., the proper name of an individual in the Kripkean sense [23]). In [12,13], the authors use the UML diagram pattern depicted in the Figure 2(a) as a framework to capture temporally changing information. The main idea is to partition the domain in two levels: the static level (**IC level**), regarding individual concepts; and the dynamic level (**TS level**), concerning changeable parts of individual's snapshots. The timeSliceOf relation connects both levels such that each instance of `IndividualConcept` maps to one or more instances of `TimeSlice`, while an instance of `TimeSlice` refers to exactly one instance of `IndividualConcept`. Indeed, the life-time of an instance of `IndividualConcept` can be determined by the initial instant (the value of the startsAt dataproperty) of its first `TimeSlice` and the final instant of its last `TimeSlice` (the value of the endsAt dataproperty). Every instance of `IndividualConcept` must have at least one time slice for representing its life-time.



(a) Framework



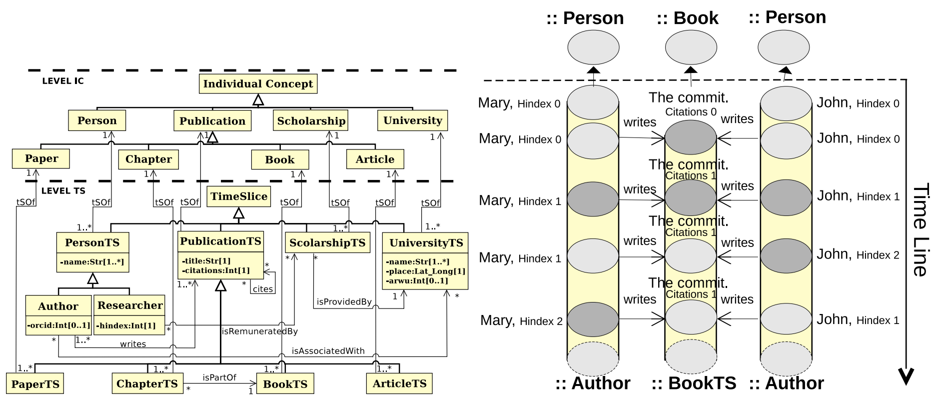
(b) Individuals

Figure 2. 4D representation

Figure 2(b) illustrates the 4D approach by presenting a situation in which Mary is temporally associated with a `University`. The value of the Mary's hindex and the value of the rank of the University evolves through time. The ellipses at the top represent the individual concepts, which are instances of some classes of the **IC level** (`Person` and `University`). Inside the **TS level**, a cylinder represents the temporal projection of the individual concept to which it is connected. Each division in the cylinder is a new (contiguous) `TimeSlice` of the connected individual concept, and thus the inter-level vertical arrows (at the top) represent instantiations of the timeSliceOf (shortened with tSlOf) property. The temporal extension of each time slice goes until the next one (or until the end of the cylinder for the last division). The horizontal arrows represent the instantiation of the isAssociatedWith property. The darker ellipses represent that some change occur w.r.t. the previous time slice. Thereafter, we present some alternatives (namely, A0, A1 and A2) to design a TBox based on the framework introduced in the Figure 2(a), in order to capture the perdurantist and endurantist notions together.

3.1. The mapping alternative A0

In the mapping alternative A0 (Figure 3(a), exemplified in Figure 3(b)), the **IC level** comprises rigid classes, while the **TS level** concerns all the others classes, relations and attributes.



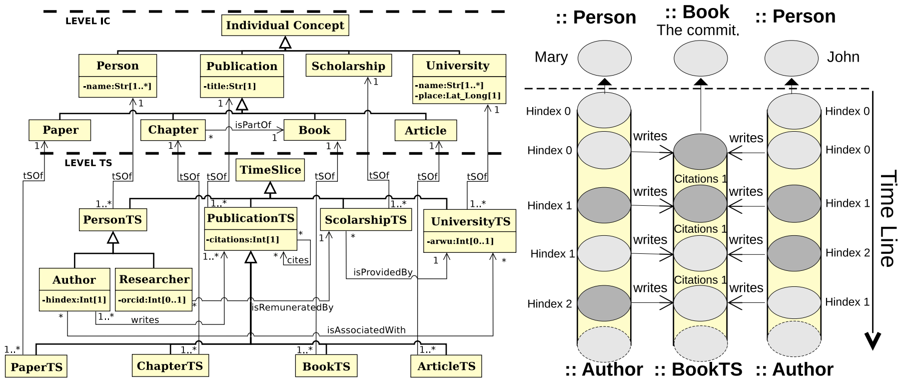
(a) Domain

(b) Individuals

Figure 3. Alternative A0

3.2. The mapping alternative A1

In the mapping alternative A1, while the **IC level** level comprises rigid classes, simultaneously mandatory and immutable attributes, and relations determining mutual existential dependencies; the **TS level** concerns the non-rigid classes, properties and relations that do not configure mutual existential dependencies (see Figure 4(a)). Figure 4(b) exemplifies the alternative A1. The main difference in using the alternative A1 w.r.t. A0 is the decreasing of redundancy (by using A0, all the attributes, including immutable and mandatory, are duplicated).

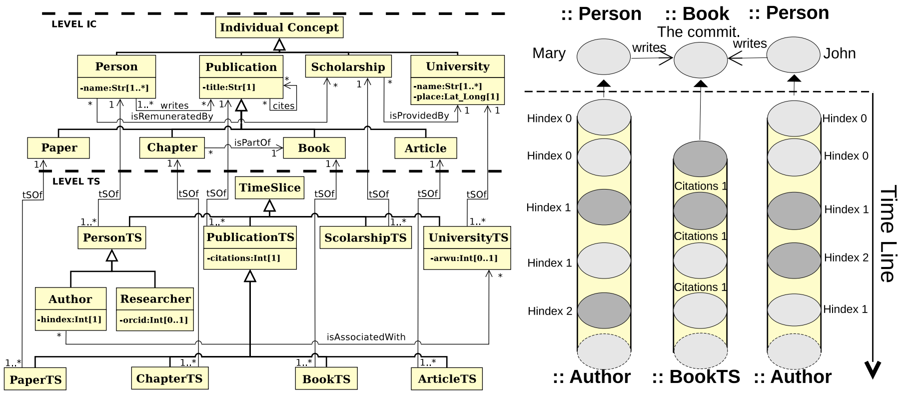


(a) Domain (b) Individuals

Figure 4. Alternative A1

3.3. The mapping alternative A2

In the mapping alternative A2, the **IC level** comprises rigid classes, simultaneously mandatory and immutable attributes, while the **TS level** concerns non-rigid classes, contingent and mutable properties (see Figure 5(a)). Figure 5(b) exemplifies the alternative A2. The main difference in using the alternative A2 instead of the alternative A1 is the decrease in the proliferation of time slices. Differently from the alternative A2, by using the alternative A1, every time slice in a chain of connected instances is duplicated when a new time slice is created. Note that the relations implying unilateral existential dependence are represented in the **IC level** and interpreted as valid through the whole lifetime of the dependent entity.



(a) Domain (b) Individuals

Figure 5. Alternative A2

The generic dependences require relaxing the maximum cardinality at the side of the dependent entity, as the relationship is changeable w.r.t. the independent individual.

For example, an instance of `Scholarship` can participate in a relationship isRemuneratedBy with different instances of `Researcher` during its existence (it can be the case that one researcher is hired to complete the scholarship of another researcher). The maximum cardinality constraint should be relaxed in order to allow this kind of change.

In next section, we report on an experimental comparison of A0, A1 and A2.

4. Empirical comparison

We evaluated the three alternatives by performing reasoning tasks. Firstly, we developed a TBox populator that generates random consistent ABoxes for the three alternatives.[3] The populator was developed in Java and is supported by the OWLAPI 4. The population starts with initial simple assertions: *John* and *Mary* are co-authors of the book *Commitments* when each one was associated with the organisation *City Hall*, after what the random process of axioms creation begins revolving around these assertions about John and Mary. We included a parameter k for the total number of ABoxes axioms (assertions of person, publications and affiliations changes and inherent roles). We made 5 populations P_k for each alternatives and $k \in \{$10,100,1000,5000,10000, 15000,20000, 30000,35000,45000,50000$\}$. Table 1 presents the median w.r.t. the 5 populations of the total number of time slices created in function of the alternatives and the parameter k. Note that a "-" in the table does not mean that the population could not be performed, but that the reasoning task on the ABox was impossible due to a heap space limit as explained thereafter.

k	A_0	A_1	A_2	k	A_0	A_1	A_2
10	17	14	6	25000	149278	149207	129989
100	409	362	262	30000	180300	178723	156201
1000	5526	5404	4755	35000	210436	209279	182756
5000	30167	28822	25491	40000	-	237732	209315
10000	59350	59090	52117	45000	-	268551	234595
15000	89735	89664	77115	50000	-	-	260493
20000	119221	119576	104165	55000	-	-	287296

Table 1. Number of time slices created in the Aboxes

Once the population was performed, we launched some queries listed in Figure 6 over the generated KBoxes[4] (TBoxes + ABoxes) to represent the evolution of some key indicators. We designed the SPARQL Query 1 in order to retrieve the evolution of the hindex for each `Author` present in the ABoxes, while the Query 2 was designed to retrieve the evolution of the arwu index (based on the average of hindex in our simulation) of all the `Organisations` present in the ABoxes for the alternatives $A0$, $A1$ and $A2$. The Query 3 was designed to retrieve the number of citations for each `Publication` present in an ABox. Query 3 was designed for $A2$, while Query 3' is the same query adapted for $A0$ and $A1$. The Query 4 was designed to output all the Mary's co-`Authors` that are present in an ABox. Query 4 was designed for $A2$, while Query 4' is the same query adapted for $A0$ and $A1$. The Query 5 was designed to output all the

[3] We consider here an ABOX as a finite set of concept and role (abstract and concrete assertions).
[4] Some samples are available at `https://ontohub.org/repositories/linkedun`

Query 1.
```
SELECT ?α ?β ?γ
  WHERE {
    {?α a lu:Person .
     ?α lu:timeSlice ?δ .
     ?δ lu:hindex ?β .
     ?δ lu:startsAt ?γ .}
    UNION
    {?α a lu:Person .
     ?α lu:timeSlice ?δ .
     ?δ lu:hindex ?β .
     ?δ lu:endsAt ?γ .}
  }
```

Query 2.
```
SELECT ?α ?β ?γ
  WHERE {
    {?α a lu:Organisation .
     ?α lu:timeSlice ?δ .
     ?δ lu:arwu ?β .
     ?δ lu:startsAt ?γ .}
    UNION
    {?α a lu:Organisation .
     ?α lu:timeSlice ?δ .
     ?δ lu:arwu ?β .
     ?δ lu:endsAt ?γ .}
  }
```

Query 3.
```
SELECT ?α ?β ?γ ?ϵ
  WHERE {
   ?α a lu:Author .
   ?α lu:writes ?β .
   ?β lu:timeSlice ?δ .
   ?δ lu:startsAt ?γ .
   ?δ lu:citations ?ϵ .
  }
```

Query 3'.
```
SELECT ?α ?β ?γ ?ϵ
  WHERE {
   ?α a lu:Author .
   ?α lu:timeSlice ?δ .
   ?δ lu:writes ?ζ .
   ?ζ lu:tSOf ?β .
   ?ζ lu:startsAt ?γ .
   ?ζ lu:citations ?ϵ .
  }
```

Query 4.
```
SELECT DISTINCT ?α
  WHERE {
   ?α lu:writes ?β .
   ?β lu:isWrittenBy :Mary .
  }
FILTER(?α != :Mary )
```

Query 4'.
```
SELECT DISTINCT ?α
  WHERE {
   ?α lu:timeSlice ?δ .
   ?δ lu:writes ?β .
   ?β lu:isWrittenBy :Mary .
  }
FILTER(?α != :Mary )
```

Query 5.
```
SELECT ?α
  WHERE {
   ?α lu:cites :Commitments .
  }
```

Query 5'.
```
SELECT DISTINCT ?α
  WHERE {
   ?α lu:timeSlice ?δ .
   ?δ lu:cites ?δ.
   ?β lu:tSOf :Commitments .
  }
```

Query 6.
```
SELECT DISTINCT ?β
 WHERE {
  ?α lu:isPartOf ?β .
  ?α lu:cites ?γ .
  :John lu:writes ?γ .
 }
```

Query 6'.
```
SELECT DISTINCT ?β
 WHERE {
  ?α lu:isPartOf ?β .
  ?α lu:timeSlice ?δ .
  ?δ lu:cites ?γ .
  ?ϵ lu:writes ?γ .
  ?ϵ lu:tSOf :John .
 }
```

Query 6".
```
SELECT DISTINCT ?β
 WHERE {
  ?α lu:isPartOf ?δ .
  ?δ lu:tSOf ?β .
  ?α lu:cites ?γ .
  ?ϵ lu:writes ?γ .
  ?ϵ lu:tSOf :John .
 }
```

Figure 6. Different SPARQL queries designed for A0, A1 and A2

`Publications`, present in an ABox, and that cite the book of John and Mary. Query 5 was designed for $A2$, while Query 5' is the same query adapted for $A0$ and $A1$. Finally, the Query 6 was designed to output all the `Books` present in an ABox having a chapter that cites a `Publication` of John. Query 6 was designed for $A2$, while Query 6' is the same query adapted for $A1$, and Query 6" for $A0$.

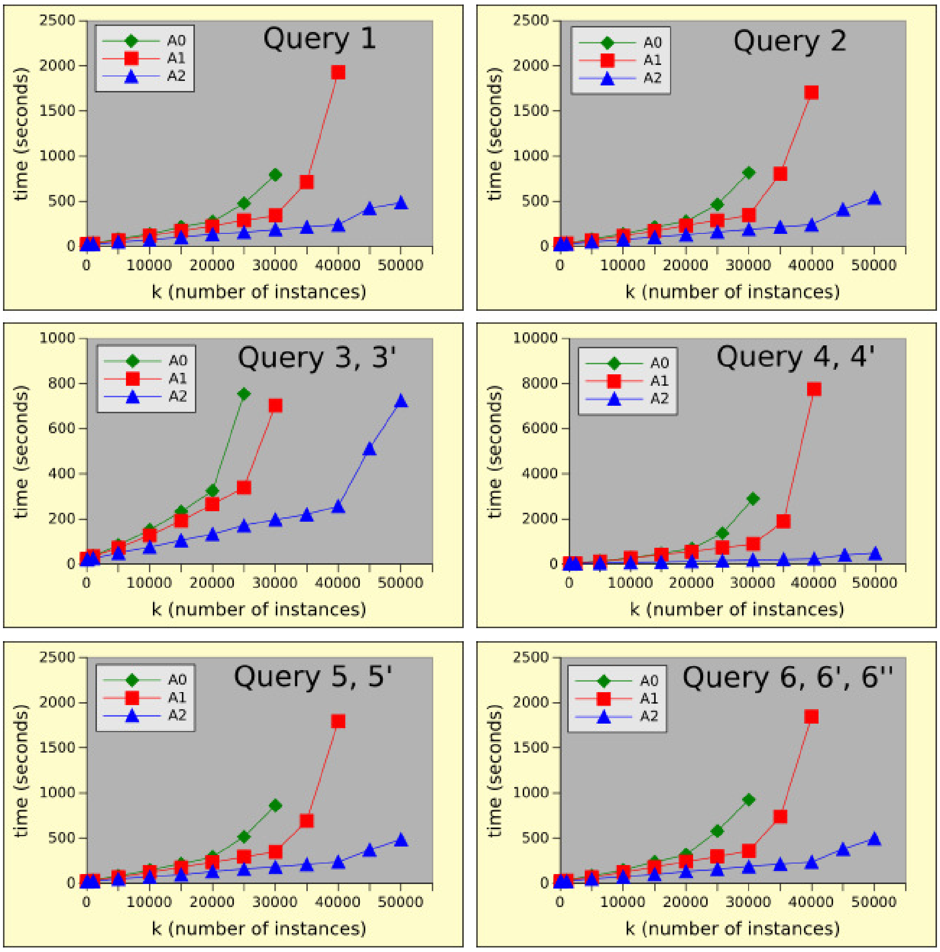


Figure 7. Experimental comparison between the alternatives.

We performed an empirical analysis on a machine equipped with an Intel Core at 3.30GHz and Ubuntu 15.04. We ran the Java-based reasoner Pellet with Sun Java 1.8, and we set the maximum heap space to 7.5 GB. Figure 7 shows a comparison performed by launching queries with Pellet and measuring the elapsed CPU times. For each query and each alternative, we performed 5 query answering tasks on the ABoxes corresponding to each population P_k, after what we retained the median value of the CPU times.

We make the following observations from Figure 7. For all the queries, for a number of instances $\gtrapprox$5000, the alternative A2 is always the fastest model w.r.t. CPU time. A2 succeeds to compute the results until ∼50000 instances for all the queries, after what a heap space limit occurs and precludes the computation. Note that this heap space limit occurs earlier w.r.t. the number of instances for the alternatives A1 ($\gtrapprox$40000 instances, except for the Query 3') and again earlier for A0 ($\gtrapprox$30000 instances, except for the Query 3'). Generally, the Queries 3 / 3' are the queries for which the heap space limit occurs earlier (for A0 and A1) and require more CPU time to finish (around twice more). The Query 4' is the query requiring the maximum amount of time to output the results for a large number of instances ($\gtrapprox$35000 instances for A1, and $\gtrapprox$25000 instances for A0). The proliferated time slices fill the memory space, eventually reaching the heap space limit and precluding the query task. We noticed that for the alternative A1 and close to the heap space limit, the task takes a longer time to finish. We hypothesize that, due to the non proliferation of the immutable attributes (e.g. name), the remaining memory enables the computation for a higher number of instances; while the proliferation of some dependent relations (those that are not mutually dependent) among the time slices makes the higher density of the graph of instances to slow down the computation.

On the Queries1 / 2, starting with all the instances of an individual concept (e.g., Person), the reasoner explores all their time slice to output the evolution in the time of the mutable attributes (e.g., hindex or arwu). For these queries (and also for their alternatives) the node length amplitude of the matching graph pattern is 2.

On the Queries 3 / 3' (and their alternatives) the node length amplitude of the matching graph pattern is 3. Nevertheless, the Query 3' designed for the alternatives A0 and A1 differs from the Query 3 designed for A2. The latter launches a task where the matching graph pattern has parts in both the static and dynamic levels, while the matching graph pattern of A0 and A1 is only in the dynamic level. For the alternative A2, the computation is 1.5 times slower than the times spent with the other queries for the same alternative, what suggests that dealing with a mix of static and dynamic entities increased the computational times. For A0 and A1, the computation is also slower than the times spent with the other queries for the same alternatives, what suggests that dealing with both object properties and data properties (i.e. writes, citations and startsAt) among the time slices can be also much more greedy.

The Queries 4 / 4' and 5 / 5' confront the speed of exploring (i) only in the static level (A2), and (ii) only in the dynamic level (A0, A1). For the Queries 4 / 4', the node length amplitude of the matching graph pattern is: 4 for the alternatives A0 and A1, and 2 for A2. For the Queries 5 / 5', the node length amplitude of the matching graph pattern is: 3 for the alternatives A0 and A1, and 1 for A2. We encoded in our populating algorithm a random draw for the authors of publications (1∼5) and the citations of publications (1∼15). Thus, the lower number of instances involving the relation writes could explain the performance of the reasoner for the Query 4 being better than for the Query 5. Nevertheless, for A0 and A1, it seems that due to the node length amplitude of the matching graph pattern, the reasoner performed the Query 5' in a shorter time.

The Queries 6 / 6' / 6" deal with mutual dependencies (e.g., partOf). The Query 6 (A2) only explores the static level, and the node length amplitude of the matching graph pattern is 3. The Query 6' (A1) explores both the static and the dynamic levels, and the node length amplitude of the matching graph pattern is 5. The Query 6" (A0) explores only the dynamic level, the node length amplitude of the matching graph pattern is 5.

To summarize, the relative similitude between the behaviors of the reasoner confronted to the same alternatives for the Query 1, the Query 2, the Queries 5 / 5' and the

Queries 6 / 6’ / 6” suggests little difference between exploring only in the dynamic level: (i) compared to an exploration in both the static and the dynamic levels for A0 and A1; or (ii) compared to an exploration only in static level for A2. The notable difference occurs when the reasoner performs an exploration both in the static and the dynamic levels for the alternative A2 (corresponding to the Query 3), or an exploration tackling object properties and data properties among the time slices (i.e., writes, citations and startsAt) for the alternatives A0 and A1 (corresponding to the Query 3’). Finally, if it could be in one sense unsurprising that these different queries have different performances in their executions, note that the comparison touched upon the performance of different mapping frameworks to retrieve the same kind of domain information.

5. Related Works

Four-dimensionalism is a significant school of thought, particularly in the field of Formal Ontology (see [24]). Krieger et al [10,11] proposed to reinterpret the 4D view by introducing the class of time slices as a superclass of both contingent and mandatory classes. In a sense, this proposal is represented by our alternative A0 where all the class, relations and attributes are encoded in a dynamic level.

As we mentioned in the introduction, the first attempt to deal with a four dimensional approach in OWL was proposed by Welty and Fikes [8], who introduced the idea of using 4D fluents to deal with relationships that change over time. Nevertheless, the nature of the relations or attributes were not considered, and the dependence between individuals was not evoked in their model. That is why in [12,13], the authors introduced a static level in which immutable properties could be encoded in order to optimize the memory requirements during the reasoning task. This proposal corresponds to the alternative A1. A 4D-based analysis of “Roles” (a specific kind of non-rigid classes) has also been performed in [25]. Later, in [26], some foundational ontologies were compared also considering the scholar domain (with a particular perdurantist view of the behavior of the role student).

Concerning the experimental analysis on temporally changing information-based models, such validations or comparisons have been attempted in very few cases. In [27,28], the authors proposed alternatives (based on a combination of qualitative and quantitative representation for interval and point relations) to represent in OWL and/or in SWRL the so called Allen’s temporal relations. Note that in [29], the authors point out the potential usage of such alternatives to express relations between time intervals of 4D-fluents in OWL. Thus, the authors performed an experimental comparison of the alternatives w.r.t. consistency tests using the reasoners Hermit and Pellet on a data-set with a relative small amounts of instances (100 to 1000 intervals generated randomly). The authors claimed it was the first such experimental evaluation of both qualitative and quantitative Semantic Web temporal representations.

Finally, Gutierrez et al. [30] were the first to propose a formal extension of the RDF data model to integrate a consideration of time validity. Thus, they introduced graphs containing quads of the form (s;p;o)[t] where t is a timestamp during which the triple (s;p;o) is valid. In [31], the authors implemented a solution to query such quad stored and experimentally demonstrated that their implementation (based on the system Strabon) outperforms all other existing implementations (e.g. AnQL, AllegroGraph).

6. Conclusion

It is very important to properly represent entities amenable to changes in terms of a knowledge representation language that could support decidable automated reasoning. In this paper, after introducing some notions that can describe the qualities of attributes and relations subject to change, we presented three strategies to map a sterotyped UML-like class diagram into TBoxes. We also performed an experimental comparison to observe in real reasoning tasks how the aforementioned alternatives would behave. The comparison showed that the alternative A2 had the best performance for all the queries. The empirical studies reported here serve the purpose of stress testing these mapping frameworks as practical alternatives to represent large instance datasets. In a sense, the observed differences between these frameworks (some of which are analogous to well-know proposals in the literature [12,9]) could be expected from an analytical study of how each of these frameworks structures information. However, the study reported here is in a much better position to analyze and quantify these differences in terms of the performance of execution of representative queries. A future work would be to perform an experimental comparison between the alternative A2 and a reification-based model [12,13], in order to assess in which situation one alternative is more suitable than the other.

References

[1] S. de Cesare and G.L. Geerts, Toward a Perdurantist Ontology of Contracts, in: *Advanced Information Systems Engineering Workshops - CAiSE 2012 International Workshops, Gdańsk, Poland, June 25-26, 2012. Proceedings*, M. Bajec and J. Eder, eds, Lecture Notes in Business Information Processing, Vol. 112, Springer, 2012, pp. 85–96.

[2] B. Tran, C. Plumejeaud-Perreau, A. Bouju and V. Bretagnolle, A Semantic Mediator for Handling Heterogeneity of Spatio-Temporal Environment Data, in: *Proceedings of the 9th Research Conference on Metadata and Semantics Research*, E. Garoufallou, R.J. Hartley and P. Gaitanou, eds, Communications in Computer and Information Science, Vol. 544, Springer, 2015, pp. 381–392.

[3] V. Milea, F. Frasincar, U. Kaymak and T. Di Noia, An OWL-based approach towards representing time in web information systems, in: *The 4th International Workshop of Web Information Systems Modeling Workshop (WISM 2007)*, 2007, pp. 791–802.

[4] B. McBride and M. Butler, Representing and querying historical information in RDF with application to E-discovery, *HP Laboratories Technical Report, HPL-2009-261* (2009).

[5] O.W. Group, Web Ontology Language 2(OWL2), 2012. `www.w3.org/TR/owl2-overview/`.

[6] F. Grandi, Introducing an annotated bibliography on temporal and evolution aspects in the semantic web, *SIGMOD Record* **41**(4) (2012), 18–21.

[7] V. Ermolayev, S. Batsakis, N. Keberle, O. Tatarintseva and G. Antoniou, Ontologies of time: review and trends, *Int. J. Comput. Sci. Appl* **11**(3) (2014), 57–115.

[8] C.A. 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, Baltimore, Maryland, USA, November 9-11, 2006*, B. Bennett and C. Fellbaum, eds, Frontiers in Artificial Intelligence and Applications, Vol. 150, IOS Press, 2006, pp. 226–236.

[9] J.M. Giménez-García, A. Zimmermann and P. Maret, NdFluents: An Ontology for Annotated Statements with Inference Preservation, in: *European Semantic Web Conference*, Springer, 2017, pp. 638–654.

[10] H.-U. Krieger, B. Kiefer and T. Declerck, A Framework for Temporal Representation and Reasoning in Business Intelligence Applications, in: *AI Meets Business Rules and Process Management, Papers from the 2008 AAAI Spring Symposium, Technical Report SS-08-01, Stanford, California, USA, March 26-28, 2008*, AAAI, 2008, pp. 59–70.

[11] H.-U. Krieger, Where Temporal Description Logics Fail: Representing Temporally-Changing Relationships, in: *KI 2008: Advances in Artificial Intelligence, 31st Annual German Conference on AI, KI 2008, Kaiserslautern, Germany, September 23-26, 2008. Proceedings*, A. Dengel, K. Berns, T.M. Breuel,

F. Bomarius and T. Roth-Berghofer, eds, Lecture Notes in Computer Science, Vol. 5243, Springer, 2008, pp. 249–257.
[12] V. Zamborlini and G. Guizzardi, On the Representation of Temporally Changing Information in OWL, in: *Workshops Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference*, IEEE Computer Society, 2010, pp. 283–292.
[13] V. Zamborlini, Estudo de Alternativas de Mapeamento de Ontologias da Linguagem OntoUML para OWL: Abordagens para Representação de Informação Temporal, Master's thesis, Federal University of Espirito Santo, Vitória, Brazil, 2011.
[14] A. Artale and E. Franconi, A temporal description logic for reasoning about actions and plans, *Journal of Artificial Intelligence Research* **9** (1998), 463–506.
[15] G. Guizzardi, Ontological Foundations for Structural Conceptual Models, PhD thesis, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 2005.
[16] F. Baader and W. Nutt, Basic Description Logics, in: *The Description Logic Handbook: Theory, Implementation, and Applications*, F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Patel-Schneider, eds, Cambridge University Press, 2003, pp. 43–95.
[17] G. Guizzardi, G. Wagner, R. de Almeida Falbo, R.S.S. Guizzardi and J.P.A. Almeida, Towards Ontological Foundations for the Conceptual Modeling of Events, in: *Proceedings of the 32th International Conference on Conceptual Modeling - ER 2013*, W. Ng, V.C. Storey and J. Trujillo, eds, Lecture Notes in Computer Science, Vol. 8217, Springer, 2013, pp. 327–341.
[18] R. Arp, B. Smith and A.D. Spear, *Building ontologies with basic formal ontology*, Mit Press, 2015.
[19] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari and L. Schneider, Sweetening ontologies with DOLCE, in: *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 2002, pp. 166–181.
[20] A. Gangemi and P. Mika, Understanding the semantic web through descriptions and situations, in: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer, 2003, pp. 689–706.
[21] S.D.H.a.T.A. Johnson, Endurantism, Perdurantism and Special Relativity, *Philosophical Quarterly* **53**(213) (2003), 524–539.
[22] G.W. von Leibniz, *New Essays on Human Understanding*, Cambridge University Press, 1981 (1705).
[23] S.A. Kripke, *Naming and Necessity*, Harvard University Press, 1980.
[24] T. Sider, *Four-dimensionalism: An ontology of persistence and time*, Oxford University Press on Demand, 2001.
[25] M. West, Roles: a four-dimensional analysis, *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS* **174** (2008), 45.
[26] S. de Cesare, B. Henderson-Sellers, C. Partridge and M. Lycett, Improving model quality through foundational ontologies: two contrasting approaches to the representation of roles, in: *International Conference on Conceptual Modeling*, Springer, 2015, pp. 304–314.
[27] S. Batsakis, G. Antoniou and I. Tachmazidis, Integrated Representation of Temporal Intervals and Durations for the Semantic Web, in: *New Trends in Database and Information Systems II - Selected papers of the 18th East European Conference on Advances in Databases and Information Systems and Associated Satellite Events*, N. Bassiliades, M. Ivanovic, M. Kon-Popovska, Y. Manolopoulos, T. Palpanas, G. Trajcevski and A. Vakali, eds, Advances in Intelligent Systems and Computing, Vol. 312, Springer, 2014, pp. 147–158.
[28] S. Batsakis, I. Tachmazidis and G. Antoniou, Representing Time for the Semantic Web, in: *Multidisciplinary Trends in Artificial Intelligence - 9th International Workshop, MIWAI 2015, Fuzhou, China, November 13-15, 2015, Proceedings*, A. Bikakis and X. Zheng, eds, Lecture Notes in Computer Science, Vol. 9426, Springer, 2015, pp. 3–15.
[29] S. Batsakis, K. Stravoskoufos and E. Petrakis, Temporal reasoning for supporting temporal queries in OWL 2.0, *Knowledge-based and intelligent information and engineering systems* (2011), 558–567.
[30] C. Gutierrez, C. Hurtado and A. Vaisman, Temporal RDF, in: *European Semantic Web Conference*, Springer, 2005, pp. 93–107.
[31] K. Bereta, P. Smeros and M. Koubarakis, Representation and Querying of Valid Time of Triples in Linked Geospatial Data, in: *Proceedings of 10th International Conference on The Semantic Web: Semantics and Big Data,, ESWC 2013*, P. Cimiano, Ó. Corcho, V. Presutti, L. Hollink and S. Rudolph, eds, Lecture Notes in Computer Science, Vol. 7882, Springer, 2013, pp. 259–274.