An Ontologically-Founded Reification Approach for Representing Temporally Changing Information in OWL

Veruska Zamborlini[1, 2], Giancarlo Guizzardi[1,3]

[1]Computer Science Department, Federal University of Espírito Santo (UFES), Vitória, Espírito Santo, Brazil

[2]CR SANTEC, Centre de Recherche Public Henri Tudor, Luxembourg

[3]Laboratory for Applied Ontology (LOA), Institute for Cognitive Science and Technology (ISTC), Trento, Italy

veruska.zamborlini@tudor.lu, gguizzardi@inf.ufes.br

Abstract

In this paper we present an approach for representing temporally changing information in OWL. This approach is based on a reification strategy founded on results from the philosophical discipline of Formal Ontology. These results grant ontological meaning to the reified individuals (intrinsic and relational properties) and provide an ontological semantics to the resulting specification. We also propose here some methodological guidelines for guiding the use of the proposed framework in supporting modeling decisions in OWL. By using the proposed framework, one can represent domain information regarding sources of temporal change such as the distinction between necessary versus contingent properties, or mutable versus immutable ones. Finally, we compare the proposed approach with another commonly used strategy for circumventing OWL´s limitation w.r.t. temporally changing information.

Introduction

Since the Web Ontology Language (OWL) was recommended by the W3C as the standard language for representing knowledge and information in the context of Semantic Web (SW), it has been widely adopted in diverse areas as medicine, biology, geography, astronomy, defense and the automotive and aerospace industries (Grau et al., 2008). However, it has been noticed that the limited expressivity of OWL as a Description Logics (DL) based language is (in principle) insufficient for representing a number of real world situations, for instance, those that deal with temporally changing information (Welty & Fikes, 2006). Indeed in DL-based models the information can be completed but cannot be in fact changed. It is because DL is designed focusing on decidable reasoning over the representation of static scenarios, assuming immutable truth-values and monotonic information given an open world assumption.

On one hand a number of authors have been investigating approaches to extend the expressiveness of classical DL to include modal (temporal) operators (Lutz, 2008). On the other hand, some authors are especially interested in maintaining compatibility with the classical DLs adopted in the Semantic Web by creating frameworks for representing changeable information in OWL (Welty & Fikes, 2008; Zamborlini & Guizzardi, 2010).

In this paper we present a proposal of a higher-level foundational framework for representing temporally changing information in OWL that relies on both: (i) the reification strategy, which is widely applied in the literature for addressing this issue, and (ii) some results coming from the discipline of Formal Ontology. We use the ontological notion of *moment* (or *trope*) (Guizzardi, 2005) for giving an ontological interpretation for the reification of attributes, material relations and role instantiations. Our purpose is to address the aforementioned issue providing guidelines for modeling decisions, while still preserving OWL standard semantics, so that the derived solutions could be reused in every tool that is compatible with the W3C recommendation. We highlight that it is not the goal of this particular paper to address related issues such as the representation of the time domain itself or the representation of temporal cardinality constraints.

The second section of the paper presents four relevant backgrounds for this work as follows: (a) a simplified view on the problem of temporally changing information, highlighting and classifying some important aspects related to change; (b) a brief introduction to the OWL language and some of its characteristics which are relevant to the issues of this paper; (c) a brief explanation on reification and (d) an overview of the ontological notion of *objects* and *moments*. In the third section, we present the main contribution of this article, which is the aforementioned framework. In section 4, we review another approach for the same issue based on a perdurantist (or 4D) view. Finally, the paper is finished with some final considerations pointing to directions of future work.

Background

The Problem of Temporally Changing Information

With the example model presented in Figure 1 (termed as running example in the remainder of this text) we intend to illustrate some important change aspects that may be differentiated in order to be properly represented in OWL. We use here the Unified Modeling Language (UML) to represent a situation in which a person, who can be a man or a woman, is identified by its name. Moreover, he/she can have a social security number (ssn) that cannot change. He/she has an age that changes annually, and can also be referred by one or more nicknames that may change along his/her life. Finally, a man can get married to only one woman per time (and vice-versa), thus, becoming husband and wife, respectively.

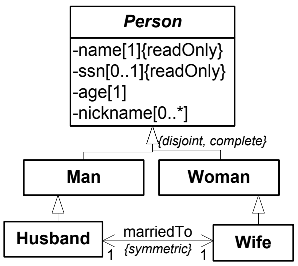

Figure 1. UML example model

We distinguish here three sources of changes: attributes, relations, and class instantiation. Regarding attributes and relations (also called properties), we can (roughly) classify them under two orthogonal dimensions: *necessary (mandatory)* versus *contingent (optional)*; *mutable* versus *immutable*. The former distinction refers to the need for an object to bear that property regardless its value. It is represented in the model by the cardinality restriction (the necessity case requires the minimum cardinality to be at least one). For example, the *name* attribute is mandatory for instances of *Person*. while the attribute *ssn* is optional. The second distinction refers to the mutability of the property value once it is settled (mandatorily or not). In UML, the immutable case is represented by a label *readOnly* next to the attribute or association end of the relation; otherwise, if no label is used then the property is considered mutable. For instance, the attribute *name* is immutable while the attribute is not.

Regarding class instantiation there is also an aspect of change said necessary versus contingent (which cannot be represented in UML): while some class instantiations must always hold for its individuals (i.e., are necessary), others are contingent. For example, the classe *Person* (in this model) is such that their instances cannot cease to instantiate it without ceasing to exist. In contrast, the instances of the classe *Husband* can move in and out of the extension of these classes without ceasing to exist.

The Web Ontology Language (OWL)

In this article, we simply use the term OWL (Web Ontology Language) when referring to its DL based variants. DL consists of a family of subsets of classical first order logics that is designed focusing on decidable reasoning. By using DL-based languages, one is able to represent static scenarios with immutable truth-values such that the information about the domain can be completed but cannot be really changed. In particular, the instantiation of a class or property cannot be retracted, except through external intervention. For example, once a model represents that John being 28 years old instantiates the class Husband, this information cannot be changed. DLs has three important characteristics to be taken into account here: (i) Open World Assumption (OWA); (ii) non-Unique Name Assumption (nUNA); (iii) monotonicity. The OWA entails that what is stated in the model is true but not necessarily the complete truth about a domain. The nUNA allows each entity of the domain to be referenced by more then one identifier. Finally, a monotonic logical system is such that the addition of new information/premises must not interfere with the information that has been previously derived. In the end, the combination of these features is such that what is stated true must remain true regardless the addition of information to the model. In other words, the information can be completed but cannot be in fact changed.

Finally, we present in Figure 2 a simple mapping of the model presented in Figure 1 into OWL using the so-called Manchester Syntax: UML classes are mapped to OWL classes, UML attributes to OWL datatype properties and UML associations to OWL object properties. When a class is specialized by two or more classes in a generalization set labeled as *disjoint and complete*, it is equivalent to the union of the correspondent subclasses. The minimum cardinality restrictions are also mapped as functional properties. However, it can be observed that the temporal semantics cannot be properly represented due to the limitations inherent to OWL.

Class: Person
  EquivalentTo: Man **or** Woman
  SubClassOf: hasName some **String**

Class: Woman
  SubClassOf: Person,
   marriedTo **only** Man
  DisjointWith: Man

Class: Wife
  EquivalentTo: marriedTo **some** Man
  SubClassOf: Woman

Class: Man
  SubClassOf: Person,
   marriedTo **only** Woman
  DisjointWith: Woman

Class: Husband
  EquivalentTo: marriedTo **some** Woman
  SubClassOf: Man

ObjectProperty: marriedTo
  Characteristics: Functional,
   Symmetric
  Domain: Person **Range**: Person

DataProperty: hasName
  Characteristics: Functional
  Domain: Person **Range**: String

Figure 2. A mapping of the model of fig.1 into OWL

On the remainder of this paper we adopt the UML-like notation for sake of simplicity.

Reification

In the philosophical literature, the reification technique is been used as means for treating abstract things like an event, an attribute, a relation or a type as a concrete entity in a way that it can be referenced, qualified and quantified. Quine (1985) presents reification as a strategy for forging links between sentences represented in a first order logic

(FOL) language. The author extends Daividson's theory of adverbs (Daividson, 1980 apud Quine, 1985) for reifying events, proposing to generalize it to what he called syncategorematic adjectives. An example of reification for the sentence 'Sebastian walked slowly and aimless in Bologna at t' is reifying the event "to walk" as ∃x(x is a walk and x is slow and x is aimless and x is in Bologna and x is at t and x is by Sebastian) where x is the objective reference that connect all clauses.

Although this technique has been used in the Artificial Intelligence community in the last decades, Galton (2006) has pointed out (when reviewing some reification strategies using FOL) that "*we lack clear criteria for regarding a formalism as reified,* [for deciding whether something] *can be quantified over, and if so, whether we have clear identity criteria for them*". If we take, for example, the statement *'John is married to Mary at t'* reified as ∃x(isRelatedTo(x, John) ∧ isRelatedTo(x, Mary) ∧ holds(x, t)) some questions arise such as: (i) what is this thing that is related to *John* and *Mary?*, or (ii) can this thing keep existing (holding) without being related to both *John* and *Mary?*, or yet (iii) are the relata (i.e., *John* and *Mary)* related to each other in this context in the very same way?

For addressing this issue, we introduce in the next section some ontological notions that we use for answering the aforementioned questions and giving some ontological meaning for the reified temporal knowledge. More specifically, we are interested on reifying the individuals' properties and then qualifying them with the time interval during which they hold having a certain value. For example, we can reify John's age the time interval during which John has the age of 27 years old, or that one during which he is married to Mary. We are particularly interested in reification as a strategy for representing temporal knowledge using OWL. It means that we are restricted here to a subset of FOL whose predicates are at most binary.

Objects and Moments

For reifying the properties of an individual of the domain we take into account the ontological notions of *Objects* and *Moments* (for a detailed and formalized description see Guizzardi, 2005; Guizzardi et al., 2006;). The term *Moment* denotes, in general terms, what is sometimes named trope, abstract particular, individual accident, or property instance (e.g. a kiss, an enrollment). In this work we regard a *moment* as a reified individual that is existentially dependent on other individuals (their bearers). Existential dependence is a modal relation (i.e., a relation holding between individuals across possible situations). So, if x is existentially dependent on y then y must exist in every possible situation in which x exists. In contrast, an *Object* is an (not-reified) individual that is not (necessarily) existentially dependent of other individuals (e.g. a person, a house, a car).

The notion of *moment* employed here comprises: (a) *Quality*: a *moment* that is existentially dependent on a single individual reifying its internal properties (e.g., a particular electric charge of a conductor). In this case, the *moment* is said to *inhere in* that individual; (b) *Relator*: a *moment* that is existentially dependent on two or more individuals (e.g., a marriage, an enrollment). In this case, the moment is said to *mediate* its bearers and capturing all the relational properties that individuals bear in the scope a relation (Masolo et al, 2005); and (c) Qua-Individual: it represents (or reify) the way an individual participates in a relation or the role it plays in that context. The name qua-individual comes from considering an individual only w.r.t. certain aspects (e.g., John-qua-student). Qua-individuals inhere in a single individual but they are also existentially dependent of other individuals disjoint from theirs bearers. Indeed, the relator is an aggregation of *qua-individuals*

The figure 3 illustrated these notions. While the figure 3a presents a traditional representation of the individual *John* directly related to a value in the age's concrete domain, the figure 3b presents the (reified) quality *John's Age* that inheres in *John*. This quality is an individual to which is attributed a value in the age's concrete domain. In an analogous manner, in figure 3c, we present the individual *John* directly related to the individual *Mary*. They also instantiate the respective role concepts (*Husband* and *Wife*). In contrast, in figure 3d there exists a (reified) relator *JMMarriage* that mediates the individuals *John* and *Mary*. . There are also the (reified) qua-individuals *JohnQuaHusbandOfMary* and *MaryQuaWifeOfJohn* that respectively inhere in the objects *John* and *Mary*, and are part of the relator *JMMarriage*. In this case, despite inhering in *John*, the qua-individual *JohnQuaHusbandOfMary* is also existentially dependent on *MaryQuaWifeJohn* and, due to transitivity of existential dependence, it is also dependent on *Mary*. In other words, the properties that *John* acquires by virtue of being married to *Mary* constitute an aspectual slice of the *Marriage* between *John* and *Mary*. Moreover, these properties are existentially dependent on the properties that *Mary* acquires by virtue of being married to John (ibid.).

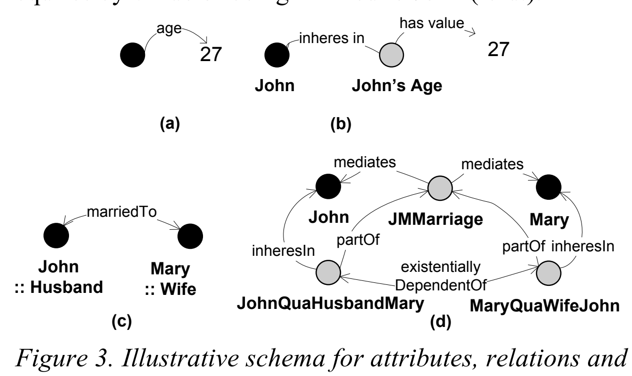

Figure 3. Illustrative schema for attributes, relations and roles representation in the traditional way (a,c) and in the reified way (b,d).

The Reification Approach

As previously mentioned, reifying the properties of an individual allows predicating and quantifying over them. It includes attributing to them a time interval during which they are held to be true. In figure 4 we present two illustrative instantiation schemas of applying an ontologically-founded reification approach to the running example in a temporal view. The object and moment individuals are represented by graphical elements in different shapes, whose projection onto the timeline corresponds to the individual's temporal extension. Moreover, the spatial inclusion of elements represents the *inherence* relation, i.e., the spatially included elements inhere in the container object, but they also reflect the temporal inclusion imposed by the existential dependence. The mandatory properties are represented as rectangles, while the optional properties are represented as rounded corner rectangles. Moreover, the mutable properties are in a lighter grey than those immutable ones.

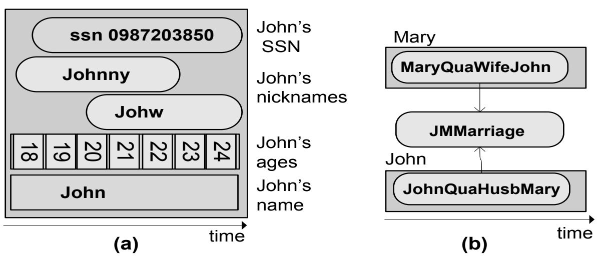

Figure 4. Illustrative schema for the reification approach.

In figure 4a, the big rectangle represents the object individual *John* that is an instance of the class *Person*, whilst the other elements inside it represent the qualities corresponding to the reification of its attributes. Particularly, the quality *John's name* has the same width extension than the individual *John*, since it represents the attribute *name* that is necessary and immutable. In contrast, the necessary and mutable attribute *age* is represented by many qualities *John's ages* that together must have the same width extension as the individual *John*. Likewise, the figure 4b represents the reification of the relation *marriedTo* between the object individuals *John* and *Mary*, as well as the reification of the correspondent role instantiations. The relator that mediates the couple is represented by the rounded corner rectangle identified as *JMMarriage*, and the qua-individuals that compose it are represented by the elements connected to it by an arrow.

In figure 5, we propose a framework that reflects the ontological notions presented in the previous section and which allows representing the aforementioned situation in OWL. Every individual (either an object or moment) has a temporal extent. A moment is existentially dependent on at least one individual, and can be either a relator or an intrinsic moment. The former mediates two or more individuals, whilst the latter inheres in exactly one individual and can be either a quality or a qua-individual. A quality has one datatype value whilst the latter is part of one relator and is *existentiallyDependentOf* at least another qua-individual. The relations *inheresIn*, *mediates* and *partOf* are specializations of *existentiallyDependentOf*.

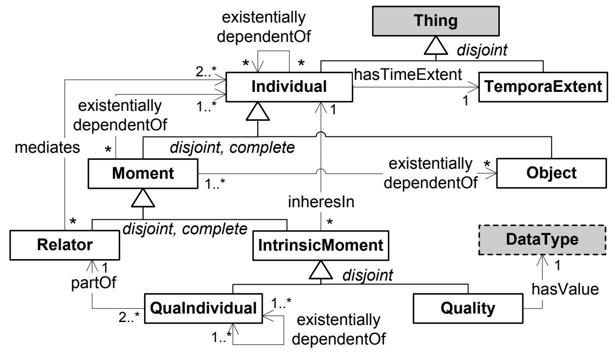

Figure 5. UML-like schema of the OWL reification approach framework

This framework should be used according to the following methodological guidelines (see example in figure 6):

a. The necessary classes in the domain (e.g. Person) must specialize the class *Object*;

b. The contingent classes (roles) are represented as subclasses of the class QuaIndividual. The latter class groups all the qua-individuals resulting from the reification of the participation of individuals of a same object class in a same relation. For example, the class Husband is represented as the class QuaHusband, which group all the qua-individuals resulting from the reification of the participation of instance of Man in the relation *marriedTo*;

c. Domain relations are represented as subclasses of the class Relator, which group all the relator individuals resulting from the reification of a same domain relation. For example, the domain relation *marriedTo* is represented as the *Marriage* class, which group all the relator individuals resulting from the reification of the instantiation of the relation *marriedTo*;

d. Attributes are represented as subclasses of the class *Quality*, which groups all the qualities resulting from the reification of a certain attribute of individuals of a same class. For example, the attribute *name* of the class Person is represented by the class *Name*, which groups all the quality individuals resulting from the reification of the instantiation of the attribute *name* of individuals of the class Person.

Moreover, we must restrict which and how properties can be or must be applied over the classes. We use the terms *minC*, *maxC* and *exactC* for, respectively, referring to the minimum, maximum and exact values of cardinality holding for attributes or relations.

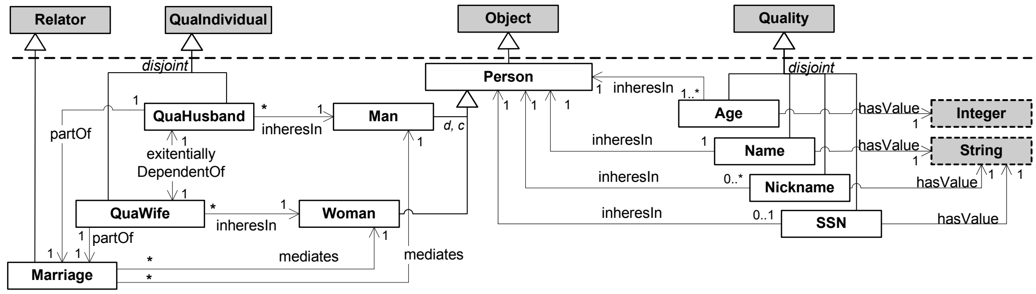

Figure 6. UML-like schema of the running example implemented according to the reification approach.

e. every instance of a qua-individual class must *inheresIn* exactly one individual of the correspondent object class. For example, any individual quaHusband *inheresIn* exactly one instance of Man;

f. every instance of a qua-individual class must be a *partOf* exactly one individual of the correspondent relator class and only be *partOf* it. For example, any individual *quaHusband* must be *partOf* exactly one instance of *Marriage* and cannot be *partOf* anything else;

g. every instance of a qua-individual class must be *existentiallyDependentOf* all other qua-individuals participating in the same relation. For example, any individual *quaHusband* must be *existentiallyDependentOf* all other qua-individuals that are part of the relator *Marriage* and cannot be *existentiallyDependentOf* any other qua-individual;

h. every instance of a relator class must *mediate* only individuals of the correspondent object classes (e.g. an individual of the class *Marriage* must mediates only instances of the classes *Man* or *Woman*);

i. every instance of a relator class must have as part (*inverse partOf*) only individuals of the qua-individual classes that inhere in the individuals of object mediated by that relator. For example, any individual of the class *Marriage* must have as part only instances of the classes *QuaHusband* or *QuaWife.* The latter instances, in turn, inhere in individuals of the classes *Man* and *Woman*, exactly those mediated by that *Marriage* relator;

j. every instance of a relator class must have as part (*inverse partOf*) at least *minC*, at most *maxC* or exactly *exactC* instances of the correspondent qua-individual classes. For instance, any individual of the class *Marriage* must be *partOf* exactly one instance of the class *Man* and exactly one instance of the classes *Woman*;

k. every instance of a relator class must *mediate* at least *minC*, at most *maxC* or exactly *exactC* instances of the correspondent object classes (e.g. any individual of the class *Marriage* must mediate exactly one instance of the class *Man* and exactly one instance of the classes *Woman*);

l. for the case of immutable relations, the domain individuals must be mediated by (*inverse mediates*) at most *maxC* or *exactC* instances of the relator class. In contrast, if a relation is mutable, no cardinality restrictions are imposed to the number of relators mediating the domain individuals (*inverse mediates*);

m. every instance of a quality class must *inheresIn* exactly one individual of the correspondent object class and only *inheresIn* it. For example, any individual *Name* must *inheresIn* exactly one instance of *Person* and cannot *inheresIn* anything else;

n. every instance of a quality class must have as value (*hasValue)* exactly one value of the correspondent DataType. For example, any individual *Name* must have as value (*hasValue)* exactly one *String* value and cannot have as value anything else;

o. for necessary attributes, every instance of the correspondent object class must bear (*inverse inheresIn*) at least one instance of the quality class. In contrast, for contingent attributes, the minimum cardinality is not restricted. For example, every instance of *Person* must have at least one instance of the quality *Age* inhering in it; such restriction does not hold for the quality *SSN*.

p. for the case of immutable attributes, every instance of the correspondent object class must bear (*inverse inheresIn*) at most *maxC* or exactly *exactC* instances of the quality class. In contrast, for mutable attributes, the maximum cardinality is not restricted. It means that every time that the attribute changes, a new *quality* individual is necessary for holding the new value. For example, every instance of the class *Person* must have at most one instance of the quality *SSN* inhering in it; such restriction does not hold, for instance, for the quality *Age*.

In figure 6 we present the UML-like schema of the OWL implementation of the running example following the proposed reification approach. Indeed, a possible instantiation of these models is the situation presented in figure 4.

A Comparison to the Perdurantistic View

A commonly employed alternative to solve the problem of representing temporal information in OWL (while maintaining a well-defined ontological interpretation) relies in an approach named the *Perdurantistic* (or 4D) view. According to this view, a domain individual is seen as a four dimensional "space-time worm" whose temporal parts are slices (snapshots) of the worm.

In (Zamborlini & Guizzardi, 2010), a number of alternative proposals for addressing the problem of representing temporal information in OWL while following the 4D paradigm are systematically reviewed. Moreover, the paper puts forth a proposal comprising the most significant features of this paradigm while addressing some problems with the proposals reviewed there. In the sequel, for comparison, we refer to this approach.

The approach presented in (ibid.) combines the perdurantistic 4D view with the notion of *individual concepts*. Individual concepts can be seen as aggregation of essential properties of individuals. This notion is employed for formulating a conceptual structure that allows one to separate the information that (essentially) define the individuals of those that can eventually change. It presents: (i) an *individual concept* level that comprises the necessary and immutable information about the domain individuals; and (ii) a time-slice level that comprises the contingent and/or mutable information about the domain individuals. Although that proposal allows one to reasonably represent the intended models, it presents the following drawbacks:

(i) *proliferation of time slices:* any change occurred in a certain time slice leads to what is called a *proliferation of time slices*, which means that every time slice in a chain of connected instances (which includes the one initiating the change) must be duplicated. In contrast, in the reification approach, changes occurred in contingent properties do not cause proliferation of objects. We do need in this case reified individuals for each of the changing properties. However, since the number of reified objects do not increase for each change, we consider the reification proposal more scalable than the perdurantistic 4D one;

(ii) *obscure ontological interpretation of contingent concepts:* in 4D approaches, the contingent classes are classes that apply just for time-slices, while the necessary classes apply both for the objects and time-slices. This makes the ontological interpretation for the contingent classes (e.g., *Husband* and *Wife*) rather obscure. In contrast, in the reification view, we have homogeneous ontological interpretation for necessary and contingent classes;

(iii) *repetition of the immutable information on time slice level:* the properties that are immutable but not necessary are represented at the time slice level, which leads to their tedious repetition across the time slices of the same individual concept. In contrast, except for the mutable properties, no other property is repeated in the reification approach;

(iv) *lack of guarantee of immutability of contingent properties at the time slice level:* since the immutable properties represented at time slice level must be repeated across the time slices of the same individual concept, we cannot guarantee that these property value do not change. In the reification view, since the immutable properties are represented just once, their value cannot change and, hence, the immutability of contingent properties is guaranteed.

Final Considerations

In this work we present a proposal that addresses the problem of representing temporally changing information in OWL. It is based on the well-known reification strategy and it benefits from results coming from the discipline of Formal Ontology in order to provide support for modeling decisions. The main ontological distinction employed here is the one between object and moment, in which the latter is used to represent the reified (intrinsic and relational) properties of an individual. Our approach allows representing attributes as quality individuals, domain relations as relator individuals, and roles as qua-individuals. Thus, we attribute to these individual objects or moments the time interval during which they hold. Thereby, we can represent the domain information regarding attributes, classes and relations according to the dimensions of necessary versus contingent, as well as mutable versus immutable.

In summary, we consider the main contributions of this work to be: (i) providing a higher-level foundational framework for guiding the modeling decisions on representing temporally changing information in OWL; (ii) a discussion about the consequences of adopting the reification strategy using OWL.

We are aware of initiatives for addressing the time domain representation and reasoning in OWL (Hobbs & Pan, 2004). Taking into account a representation for the time domain is indeed necessary for imposing the temporal restrictions pointed out in our reification proposal, namely: (i) the existential dependence relation must imply temporal inclusion of the dependent individual in the time-extent of the individual(s) it depends on; (ii) a reified necessary and immutable property must have exactly the same time-extent as the individual it depends on; and (iii) a reified necessary and mutable property must have the temporal projection of all its individuals equal to the time-extent of the individual they depend on (i.e. the property age). A full treatment of these issues in OWL is non-trivial in the presence of OWA, given that information about individuals and their lifetime can be incomplete in the model. These concerns also must be addressed for temporal cardinality restrictions in OWL. For instance, if a functional mutable relation relates one individual to temporally overlapping individuals in the range, then one can infer that both individuals must represent the same entity due to non-UNA. In contrast, if the mutable relation is not functional, one could not infer anything (except if the temporally overlapping individuals in the range could be differentiated by other means). For example, suppose a husband can be married to two wives per time, and John is said to be married to three temporally overlapping individuals. In this case, one cannot decide whether they represent just one wife, or yet two wives but one which is represented twice. These issues will receive a fuller treatment in a future version of this framework.

References

Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., e Sattler, U. OWL 2: The next step for OWL. *Web Semant. 6*, 4 (2008), 309-322.

Welty, C. and Fikes, R. A Reusable Ontology for Fluents in OWL. *Proceeding of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, IOS Press (2006), 226-236.

Lutz, C., Wolter, F., and Zakharyaschev, M. Temporal Description Logics: A Survey. Proceedings of the 2008 15th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society (2008).

Zamborlini, V., Guizzardi, G., On the representation of temporally changing information in OWL. *EDOC Workshops - VORTE'10*, IEEE Computer Society (2010).

Guizzardi, G., Masolo, C., Borgo, S., In Defense of a Trope-Based Ontology for Conceptual Modeling: An Example with the Foundations of Attributes, Weak Entities and Datatypes. *Conceptual Modeling - ER 2006*, (2006), 112-125.

Guizzardi, G., Ontological foundations for structural conceptual models. Telematics Institute Fundamental Research Series, volume 15, ISBN 90-75176-81-3, The Netherlands, (2005).

Quine, W.V. Events and reification. In Actions and Events: Perspectives on the Philosophy of Davidson. Blackwell, 1985, 162-71.

Davidson, D. (1967). The logical form of action sentences. *Essays on actions and events*, *5*, 105-148.

Galton, A. Operators vs. Arguments: The Ins and Outs of Reification. Synthese 150, (2006), 415-441.

Masolo, C., Guizzardi, G., Vieu, L. Bottazzi, E., Ferrario, R.. Relational roles and qua-individuals. *Procs. of AAAI Fall Symposium Roles*, (2005).

Loebe, F. An Analysis of Roles. Master thesis in Computer Science, University of Leipzig, 2003.

Hobbs, J.R. e Pan, F. An ontology of time for the semantic web. ACM Transactions on Asian Language Information Processing (TALIP) 3, 1 (2004), 66-85.