On the representation of temporally changing information in OWL

Veruska Zamborlini, Giancarlo Guizzardi
Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espírito Santo, Brazil
{veruska,gguizzardi}@inf.ufes.br

Abstract—**An important challenge in the Knowledge Representation area is on representing and reasoning over temporally changing information. Particularly, a number of authors have been investigating approaches to extend the expressivity beyond what is currently supported by the DL (Description Logics) based languages in order to address this issue, while maintaining compatibility with subclasses of DLs adopted in the Semantic Web. This is mainly due to the increasing popularity of the Semantic Web initiative as well as the role played by DL in that context. In this paper we defend the need of a higher-level foundational framework based on results coming from the discipline of Formal Ontology. We present two complementary proposals for modeling temporally changing information in OWL, based on the most discussed strategy in the literature to address this problem, namely, the use of a perdurantist (or 4D) view of domain entities. Moreover we compare the results with some related work and discuss its limitations and further improvements.**

Keywords: Knowledge Representation, Temporally changing information, OWL, Description Logics, Formal Ontology.

I. INTRODUCTION

Since the Knowledge Representation (KR) area is concerned with representing and reasoning over logically specified conceptual models, one of its fundamental problems is dealing with the tradeoff between expressive power and computational complexity. Indeed, the more expressive is the formal logic underlying a conceptual model, the more characteristics of a given domain can be represented. However, increasing expressivity usually comes with the price of increasing computational complexity [1]. This is especially true when we need to use logical systems more complex than the classical FOL (predicate calculus), for exemple, adopting a temporal one.

One particular class of KR logics is the so-called Description Logics (DL). DL consists of a family of logics typically restricted to a decidable subset of first order logics (FOL). This class of languages has gained much interest in the later years, mainly due to the increasing popularity of the Semantic Web initiative as well as the role played by DL in that context as a logical basis for the W3C recommendation OWL (Web Ontology Languages). As classical FOL, such languages have been designed focusing on the representation of scenarios with immutable truth-values and monotonic information, such that the information about the domain can be completed but cannot be really changed (assuming an open world assumption which entails that what is stated is true but not necessarily the entire truth about the domain).

Therefore, it has been noticed that the limited expressivity of (DL based) OWL is (in principle) insufficient for representing a number of real world situations, for instance, those that deal with temporally changing information. Indeed, a number of authors have been investigating approaches to extend the expressiveness of DL based languages in order to address this issue [2]. On a different research direction, other 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 and keeping a history [3,4,5].

In this paper we present two complementary and alternative proposals of higher-level foundational frameworks for modeling (and keeping a history of) temporally changing information in OWL. They rely on both: (i) the most discussed strategy in the literature to address this problem, namely, the use of a perdurantist (or 4D) view of domain entities; and (ii) some results coming from the discipline of Formal Ontology. While the first approach proposes an adaptation of the 4D view by dividing the domain entities into a static and a dynamic view, the second one (complementary to the first) introduces a more complex (and thus less intuitive) framework addressing some of the previously identified drawbacks. The main goal in this paper is to provide such frameworks and to support modeling decisions in a manner that makes explicit its consequences. Therefore, it is not the goal of this present paper to address other related issues such as time representation and reasoning, temporal cardinality constraints or reuse of existent OWL domain ontologies.

Section 2 presents three 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 on OWL language and some of its characteristics; and (c) a brief explanation on the perdurantist view of domain entities. In the sections 3 and 4 we present the main contribution of this article, which are the aforementioned proposals. In section 5 we review two existing approaches that also employ the perdurantist view with the purpose of representing temporally changing information in OWL, and compare them with our proposals. Finally, in section 6, we make some final considerations, addressing some of the limitations of our proposals and pointing to the direction of possible further improvements.

II. BACKGROUND

A. The Problem of Temporally Changing Information

The UML example model presented in Figure 1 (termed as *running example* in the remainder of this text) illustrates some important change aspects that may be differentiated in order to be properly represented in OWL. This model represents 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 change 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.

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 minimum cardinality must be greater or equal one). For example, the *name* and *age* attributes, as well as all relations in this model are mandatory, while the *ssn* and *nickname* attributes are optional. In contrast, the latter distinction refers to the mutability of the property value, once it is instantiated (mandatorily or not). The immutable case is represented in the model by a *readOnly* label besides the correspondent immutable association end or attribute; otherwise, if no label is used then the property is considered mutable. For instance, the *name* and *ssn* attributes are immutable, as well as the association ends of the *mediates* relations, in the sense that a certain *marriage* instance cannot mediate a *husband* or *wife* different from those to which it is initially linked; in contrast, the *age* and *nickname* attributes are mutable. The same can be said for the *marriedTo* and *isMediatedBy* relations in the sense that a certain husband (or wife) can be married to a different wife (or husband) in different situations by virtue of being mediated by different *marriage* instances.

On the other hand, regarding class changing (though it is not put clearly in UML), some class instantiations must always hold for its individuals, while others do not necessarily have to. For example, the *Person*, *Man*, *Woman* and *Marriage* classes in this model are such that its instances cannot cease to instantiate them without ceasing to exist. In contrast, the instances of *Husband*, *Wife*, Student or Child classes can instantiate them just contingently[1]. Supposing *John* is an instance of the *Person* class, he will never cease to instantiate it unless he ceases to exist. However, *John* can become an instance of the *Husband* class and cease to instantiate it without ceasing to exist, i.e., without loosing his identity. In this example, we assume that being a instance of *Man* and being related to a *Wife* via the *marriedTo* relation are necessary and sufficient conditions for someone to be a

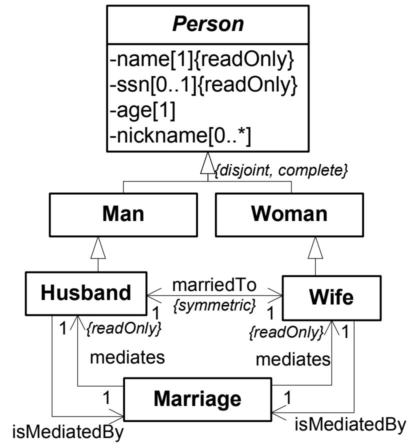


Figure 1. UML example model.

Husband. Analogously, being an instance of *Woman* and being related to a *Husband* are the sufficient and necessary conditions for someone to instantiate *Wife* class. As a consequence, in the previous example, *John* is part of a pair which instantiates the relation *marriedTo* (together with his wife) iff he is an instance of *Husband*. In this sense, we can observe that a change on an instance of the relation *marriedTo* also implies a change on the instantiation of the classes *Husband* or *Wife* by the related individuals (and vice-versa). To put it simply, if the pair ⟨*John*, *John*′s wife⟩ cease to instantiate the relation *marriedTo*, *John* and *John*′s wife also cease to instantiate the classes *Husband* and *Wife*, respectively (and vice-versa).

Furthermore, we say that when the participation on a relation is necessary/mandatory for an individual, we have a case of a dependence relation. We consider here two cases of dependence, namely, generic and specific dependence. An example of the former is the *isMediatedBy* relation, which defines a generic dependence between a certain instance of *Husband* and a generic instance of *Marriage*. An example of the latter is the existential dependence relation between a particular instance of *Marriage* and the particular instances of *Husband* and *Wife* that it mediates. In other words, a generic dependence can be seen as a relation between an in individual X and a type T such that X is dependent on being related (via a proper relation at hand) with any instance of T. In contrast, a specific dependence relation is one between two individuals X and Y such that X needs to be related to that specific individual Y. In another way of seeing this distinction, the individual in the association end opposite to X in a generically dependence relation can be changed (as long as the new related element instantiates type T), but it cannot be changed in the case of a specific dependence relation. Finally, when the dependence relation holds in both directions between two classes, we say they are mutually dependent. For example, *Husband* and *Wife* classes are mutually dependent. Particularly, if the mutual dependence is existential (specific), we say the classes are mutually existentially dependent.

[1] The latter type of classes is named Phased-Sortals in [6].

Therefore, we can summarize the aspects related to change discussed here in terms of the following meta-properties: (i) **necessity/contingency**: whether the instantiation of a property or class is necessary or contingent for individuals of a certain class; (ii) **mutability**: whether a specific value for a relation (association end) or an attribute can be subject to change (or not) for individuals of a certain class; and (iii) **dependence**: whether the participation of individuals of certain class on a relation is said to be necessary and either (a) the value of its relata is mutable (generic dependence); or (b) the value of its relata is immutable (specific/existential dependence). Once more, when a dependence relation holds between the relata in both directions, it characterizes what is termed a case of mutual dependence, particularly called mutual existential dependence if the dependence is specific in both directions..

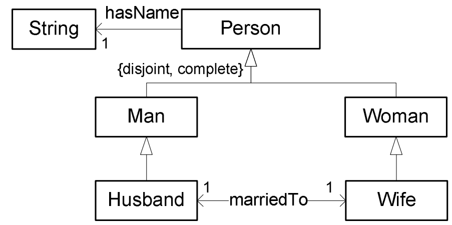
B. OWL (Web Ontology Language)

The OWL (Web Ontology Language) is a well known formal language for representing ontologies on the Semantic Web. In this work we are particularly interested in its DL based variants [7]. Besides its fundamental characteristic of decidability, an important characteristic of this language is monotonicity. Intuitively, in a monotonic logical system the addition of new information/premises must not interfere in the information that has been previously derived. In other words, what is true in one situation must remain true regardless any addition of information to the model. Languages such as OWL are typically designed for representing static situations, such that the information about a domain can be completed but cannot be really changed (assuming the open world assumption which entails that what is stated is true but not necessarily the entire truth about a domain). In particular, the instantiation of a class or relation cannot be retracted, except through external intervention. For example, once an OWL model represent that *John* being 28 years old instantiates the class *Husband*, this information cannot be changed.

In order to favor the understanding of the OWL specifications used along this article, we employ an UML-like visual schema that only covers a fragment of the OWL representation capabilities, but that is enough for the discussion conducted here. The Figure 2 exemplifies how such an UML-like schema can be represented in OWL using the so-called *Manchester Syntax*. We emphasize that a class that has a generalization set characterized as *disjoint* and *complete* must have as equivalent class the disjunction of the correspondent subclasses. Therefore, we consider this class as being an *abstract class*, since each of its instances is supposed to belong to (at least) one of its subclasses.

C. Perdurantism

In the philosophical literature, there are two classical views that reflect different modes of classifying entities with respect to their relation to time. These are named Endurantism and Perdurantism [8]. According to Varzi in [9], the former theory argues that "*things as persons, rocks, and tables are three-dimensional continuants that literally persist through time in spite of the many qualitative changes*



Class: Person
EquivalentTo: Man **or** Woman
SubClassOf: owl:Thing

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: Person

Figure 2. Example of mapping from UML-like shema into OWL.

that they may undergo". In such theory, a continuant such as a person is fundamentally different from what is named a perdurant (or process). The former is said wholly present whenever it exists, whilst the latter has temporal parts that unfold in time. Examples of perdurants include a birthday party, a business process, a football game and the Second World War. Moreover, in this view, perdurants cannot genuinely change. It means that if a perdurant P has a property x in time t1 and another (possibly contradictory) property y in time t2 then there are temporal parts TP1 and TP2 of P such that: (i) TP1 happens in t1 and has property x; (ii) TP2 happens in t2 and has property y.

Contrary to the endurantist stance, a perdurantist[2] view eliminates the distinction between continuants and perdurants by defending that: "*ordinary objects such as persons, rocks, or tables are not continuants; they are perdurants. They have spatial as well as temporal parts, or stages, and to say of such objects that they persist through time is to say that they have different parts that exist at different times. So on this view the person in front of me now is not John Doe in his entirety. It is only a temporal part of John, just as I am not exposed to his whole life but only to its current stage. (In this sense, ordinary objects are not distinct from events, which also extend over time.)*" [9]. Thus, entities that otherwise would be seen as 3D continuants are taken in a perdurantist stance as the so-called perdurantist worm, since in the latter theory all entities can be seen as four dimensional "space-time worms" whose temporal parts are slices (snapshots) of the worm.

[2] There are two different views in the perdurantist theory, namely worm and stage view. The description presented here addresses the former, since the latter is out of the scope of this work. For a more elaborated discussion on this issue, the reader can refer to [9].

Figure 3 illustrates this idea of representing a 3D continuant with a fourth temporal dimension. In this example an individual called John is represented as being extended in an extra time dimension, composing a 4D object also called space-time worm whose slices are snapshots of John during his life (also said its temporal parts or time slices).

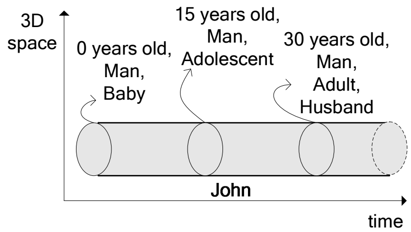


Figure 3. Ilustrative schema of a 4D object (space-time worm).

III. First Proposal – Individual Concepts and Rigidity

In this first proposal we take into account the ontological notions of **individual concept** and **rigidity**. The former, first introduced by Leibniz, refers to a singleton characteristic (or set of characteristics) that only holds for one individual, and also holds for it whenever it exists [6]. These characteristics, said essential (i.e., necessary and immutable), define the identity of an individual. For example, suppose the proper name of a person (in a Kripkean sense) carries his/her identity [10]. Thus, the property of being John holds for a single instance and also holds for him whenever he exists. Moreover, an individual concept allows mapping to its snapshots (or time slices) whenever it exists. In the previous example, the John's name allows one to refer to him in his childhood or adolescence.

On the other hand, rigidity refers to a meta-property of a concept/class such that (i) a rigid class is that whose instances must always instantiate it (e.g., an instance of *Person* cannot cease to be a person without ceasing to exist), whilst a non-rigid class is that whose instances contingently instantiate it (e.g., an instance of *Husband* can cease to be its instance and still remain existing) [6]. In the running example, *Person*, *Man*, *Woman* and *Marriage* are rigid classes while *Husband* and *Wife* are non-rigid ones.

Although we adopt the perdurantist worm view (in which the whole individual is considered as a four dimensional "space-time worm" whose temporal parts are slices of the worm) we consider here that an individual is divided into two views: a static and a dynamic one. While the former regards the individual concept (i.e. the immutable set of properties that defines the individual's identity), the latter regards the changeable part of the individual's snapshots or time slices, that represent different moments in its history[3]. These two views for individual reflect two levels of information representation: (i) a static level that comprises the rigid classes, the necessary and immutable attributes, and the relations that determine mutual existential dependence; and (ii) a dynamic level that comprises the non-rigid classes, the contingent or mutable properties, and also the relations that do not determine mutual existential dependence. For example, even though the *mediates* relation is necessary and immutable for every instance of *Marriage*, it cannot be represented in the static level, since it is not immutable in the inverse direction, i.e. an instance of *Husband* can change the instance of *Marriage* it is related to.

Figure 4 presents a situation that illustrates the two views or levels aforementioned. It represents that John gets married to Mary when they are 27 and 26 years old respectively, and that they get divorced one year later. The ellipses at the top represent the individual concepts while the cylinders represent the changeable part of the space-time worm. The darker ellipses represent time slices in the moments when some change takes place. According to the running example, the horizontal arrows represent the *isMediatedBy* property that holds during the entire existence of the *Marriage* instance (the others are omitted for sake of simplicity).

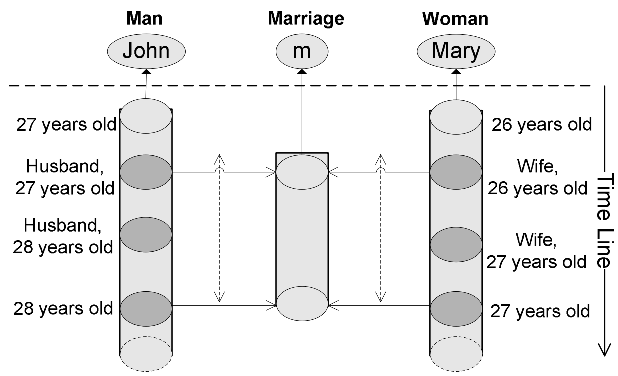


Figure 4. Ilustrative schema of individual concepts mapping to (the changeable part of) its 4D objects (space-time worms).

For representing such situation in OWL we use the UML-like schema proposed in the Figure 5. The OWL structure is then divided into two levels, the static one called Individual Concept level (IC level) and the dynamic one called Time Slice level (TS level).

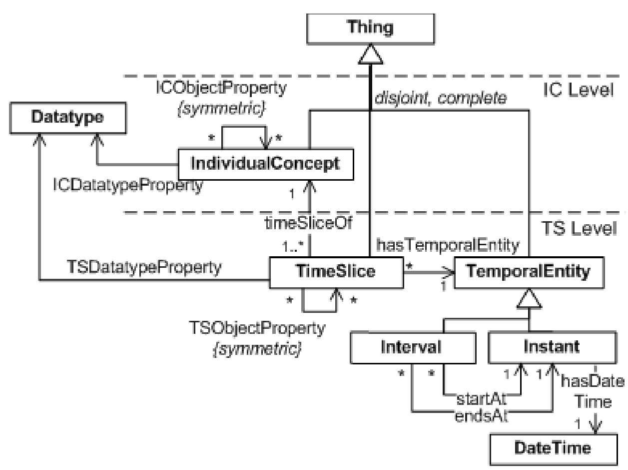


Figure 5. UML-like schema of the OWL structure for the first proposal.

[3] For simplicity of the presentation, we are considering here a linear time structure.

Although both levels are static in practice, since any OWL instantiation will always hold (due to the DL's monotonic characteristic), this structure allows the instantiations in a time slice context to be interpreted as holding for the respective individual concept just within a specified time interval.

The *timeSliceOf* relation, which connects both levels, is such that each instance of *IndividualConcept* maps to one or more instance of *TimeSlice* and a time slice refers to one and only one individual concept. Thus, the life-time of an individual concept can be determined by the initial instant of the first time slice and the final instant of the last one. Therefore, every time slice must have exactly one interval with initial and final instants and every individual concept must have at least one time slice for representing its life-time.

Moreover we propose some methodological guidelines for properly using this proposal. These guidelines are illustrated with the UML-like model[4] depicted in Figure 6, which implements the running example[5]:

(i) every domain concept or property must specialize one of the structural elements in figure 4 according to the level they fit. The concepts specialize the *IndividualConcept* or *TimeSlice* classes; domain relations specialize the *ICObjectProperty* or *TSObjectProperty* properties, and attributes specialize the *ICDatatypeProperty* or *TSDatatypeProperty* datatype properties. These elements must not be directly instantiated, i.e., they are *abstract classes*. For example, the *Person* rigid concept specializes the *IndividualConcept* class, the *Husband* anti-rigid concept (indirectly) specializes the *TimeSlice* class, the *Name* necessary and immutable attribute is represented as the *hasName* datatype property that specializes the *ICDatatypeProperty*, while the *marriedTo* contingent relation specializes the *TSObjectProperty*.

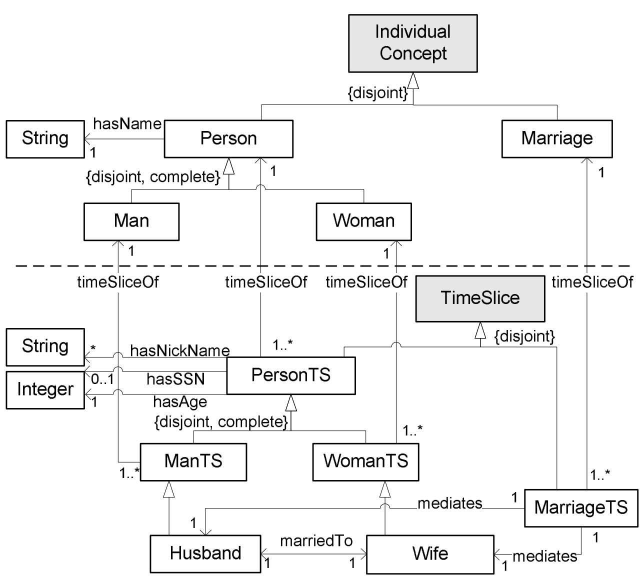


Figure 6. The UML-like schema of the running example implemented according to the first proposal.

(ii) each subclass of *IndividualConcept* must have as a counterpart a subclass of *TimeSlice*. Thus, the domain and the range of the *timeSliceOf* property must be restricted between the correspondent pair of classes. For example, the *Person* rigid concept, besides specializing the *IndividualConcept* class, must also have a *PersonTS* counterpart class that specializes the *TimeSlice* class. Moreover, the *timeSliceOf* relation must be restricted in such a way that (a) any instance of *Person* must instantiate it with at least one instance of *PersonTS* and only with its instances; and (b) any instance of *PersonTS* must instantiate it with exactly one instance of *Person* and only with its instances.

(iii) each direct subclass of *TimeSlice* must be a counterpart of exactly one class at IC level. In other words, the contingent concepts must not directly specialize the *TimeSlice* class, but must specialize the counterpart class of its correspondent super-concept. For example, the *Husband* contingent concept must be represented as subclass of *ManTS* that is the time slice counterpart class of its super-concept, namely the *Man* concept.

A simplified instantiation schema[6] for this example model is presented in Figure 7 (which represents the same situation represented in Figure 4). The ellipses at the top represent the individual concepts, more specifically instances of some classes of the IC level (*Man*, *Woman* and *Marriage*). A cylinder represents the temporal projection of the individual concept to which it is connected by a vertical arrow. Each division in the cylinder is a new (contiguous) time slice of the connected individual concept, and thus the vertical arrow represents the instantiation of the *timeSliceOf* 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

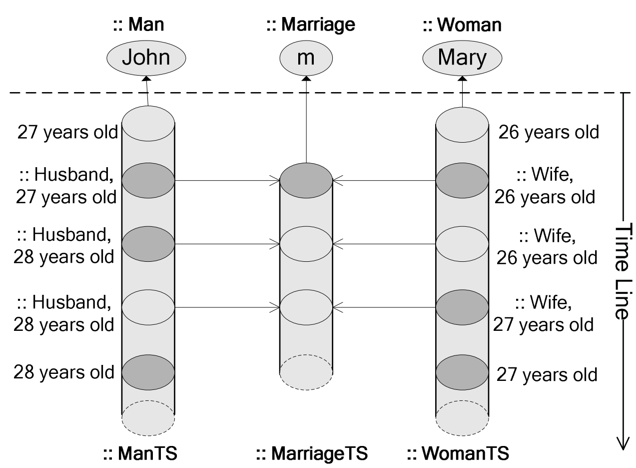


Figure 7. Instantiation schema of the model presented in Figure 6.

[4] For the sake of simplicity, from now on we omit from the UML-like models some concepts and relations from the original example.

[5] Although we just exemplify one type of non rigid concepts (called roles in [6]), other types (e.g., phases, role mixins [6]) can also be represented following these guidelines.

[6] From now on we omit from the instantiation schemas some class and property instantiations that are irrelevant to the present discussion.

the *isMediatedBy* property. The labels beside the time slices indicate the age value and the contingent class instantiation that hold within its time extent. The darker ellipses represent that some change occur w.r.t. the previous time slice.

Although this result is quite close to what we intended to represent, some drawbacks can be observed: (i) any change occurred in a certain time slice leads to what we call 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; (ii) in this view, rigid concepts exist both as 3D entities (individual concepts) and as time slices. This is, however, not the case for anti-rigid concepts. This makes the ontological interpretation for the contingent concepts (like *Husband* and *Wife*) rather uncanny; and finally, regarding immutable properties, when they must be represented at the TS (Time Slice) level we both (iii) cannot guarantee their immutability (iv) if we could, we would have a tedious repetition of these immutable properties across the time slices of the same individual concept. Section IV presents a complementary proposal in order to address some of these issues.

IV. Second Proposal – Objects and Moments

In this section, we extend our first proposal considering the ontological distinction between the notions of **Objects** and **Moments** [6]. A moment[7] is an individual that only exists in other individuals, and thus is existentially dependent of them. This term is derived from the german *Momente* in the writings of E. Husserl and it denotes, in general terms, what is sometimes named trope, abstract particular, individual accident, or property instance [11]. In this work we are restricted to two special types of moments, namely **Relators** and **Qua Individuals**. The former is also called a relational moment since it *mediates* more than one individual. Moreover, since this relation is necessary and immutable for the relator, it is said existentially dependent of the objects it mediates. Examples are a kiss, a handshake, a covalent bond, a medical treatment, but also social objects such as a flight connection, a purchase order and a commitment or a marriage.

Furthermore, a Qua Individual is a special kind of Intrinsic Moment, since it *inheres in*[8] only one individual. Intuitively, a qua individual is the way an object participates in a certain relation [12]. This name comes from considering an individual only w.r.t. certain aspects (e.g., John qua student; Mary qua musician) [13,14]. The notion of (relational) qua individual employed here is that of a moment that inhere in an object participating in a material relation. A *material relation* is one that needs a truthmaker to hold, namely a relator. In other words, when two individuals participate in such relation, there must be a relator mediating them. The relator is then composed by the qua individuals that inheres in the mediated individuals due to its participation in the relation. For example, when the objects *John* and *Mary* are married to each other, there is a relator *Marriage* mediating them. This relator is composed by two qua individuals, namely *JohnQuaHusbandOfMary* and *MaryQuaWifeOfJohn*, which inheres in John and Mary respectively. The qua individual *JohnQuaHusbandOfMary*, represents all properties that *John* acquires by virtue of being related to *Mary* via the *Marriage* relator (e.g., all the commitments and claims). Mutatis Mutandis, the same can be said about *MaryQuaWifeOfJohn*.

The Figure 8 presents the extension of the structure of the first proposal including the aforementioned concepts and relations as abstract structural elements.

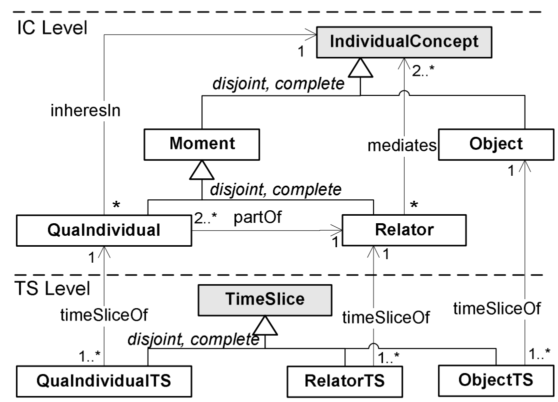


Figure 8. UML-like schema of the OWL structure for the 2nd proposal.

The idea is to replace the representation of material relations and the contingent concepts connected to their association ends, with the representation of the elements from which these material relations are derived, namely, the qua individuals and their corresponding inherence and mediation relations. Both inherence and mediation relations immutably hold for the involved moments whenever they exist, thus characterizing that they are existentially dependents of the individuals they are related to. Although these relations do not characterize mutual existential dependence, in this approach they are represented at the IC level and then interpreted as holding for both related individuals during the lifetime of the dependent one. Thereby we avoid their (possibly tedious) repetition at the TS level, guaranteeing their immutability and further reducing the risk of proliferation to the (non-material) relations that hold at the TS level. For instance, the inherence relation between the qua individual *JohnQuaHusbandOfMary* and the object *John*, once holding at the IC level, is constantly represented, immutable and interpreted as holding just during the lifetime of the former.

Once more, a number of methodological guidelines can be put forth for this second proposal. Following Figure 8, we have here that the leaf classes of each level are those that will be specialized by the domain concepts according to the level they fit. For example, the *Person* concept must specialize the *Object* class, as well as its *PersonTS* counterpart class must specialize the *ObjectTS* class.

[7] In the scope of this work, the term bears no relation to the notion of time instant in colloquial language.

[8] Inherence is a special type of functional and intransitive existential dependence relation. Mediation is a non-functional type of inherence [6].

Moreover, since the material relation and the related non-rigid concepts are not directly represented, we now represent the respective relator and qua individuals specializing the *Relator* and *QuaIndividual* classes at both levels, and properly restrict the domain, range and cardinality of the *mediates*, *inheresIn* and *partOf* relations. For instance, the *marriedTo* material relation and the *Husband* and *Wife* non-rigid concepts are no longer represented, but the respective *QuaHusband* and *QuaWife* qua individual classes. These classes are connected to the *Marriage* relator class via a *partOf* relation and to the *Man* and *Woman* object classes via an *inheresIn* relation. The UML-like model[9] depicted in Figure 9 follows this approach for implementing the running example.

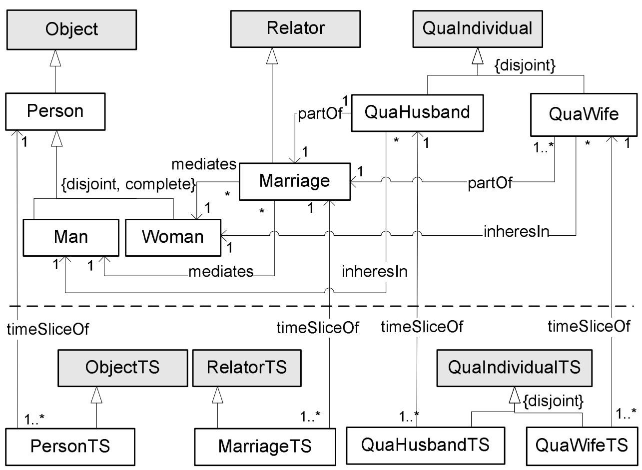


Figure 9. The UML-like schema of the running example implemented according to the 2nd proposal.

A query language can be used for deriving from a model based on this approach the indirectly represented material relations and contingent concepts. On what follows we present a SQWRL[10] [15] query that allows retrieving from a population of this example model which persons are married to each other in which period of time or, in other words, who contingently participates in the material relation *marriedTo* and in which way they do (qua *Husband* or qua *Wife*).

```
Person(?x)  ^ Person(?y) ^
QuaHusband(?qx) ^ inheresIn(?qx, ?x) ^
QuaWife(?qy) ^ inheresIn(?qy, ?y) ^
Marriage(?m) ^ partOf(?qx, ?m) ^ partOf(?qy, ?m) ^
MarriageTS(?tsm) ^ timeSliceOf(?tsm, ?m) ^
hasTemporalEntity(?tsm, ?te) ^
startsAt(?te, ?ti) ^ endsAt(?te, ?tf) ^
hasDateTime(?ti, ?dti) ^ hasDateTime(?tf, ?dtf)
  → sqwrl:selectDistinct(?x, ?y, ?dti, ?dtf)
```

A simplified instantiation schema for this UML-like model is presented in Figure 10, which represents the same situation presented in Figure 4. However, in this case, the model refrains from directly representing the *marriedTo* material relation as well as the classes *Husband* and *Wife* whose instances participate in this relation.

In the example depicted in Figure 10, it can be noticed that the changes (of the age property) in the individuals' time slices no longer cause proliferation of time slices. This is due to the elimination of the explicit representation of the material relation. In comparison with the instantiation example of the first proposal, in which 13 time slices are necessary, in this approach just 9 time slices are enough, plus 2 extra individual concepts. Although the difference does not seem to be substantial at a first glance, it increases at the pace that we consider changes in other properties (rather than age), or more material relations involving the individuals.

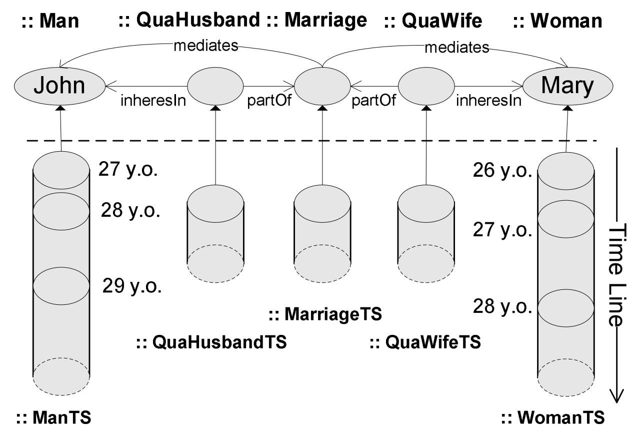


Figure 10. Instantiation schema of the model presented in Figure 9.

Furthermore, the ontological interpretation of (a particular kind of) contingent classes (e.g., *Husband* and *Wife*) is no longer restricted to the time slice context, but it can now be individualized in a qua individual that can have its own characteristics. Indeed, the notion of qua individuals has been used in the literature to ascribe even incompatible characteristics to the same individual, which result from its participation in different relations. This is illustrated by the classical example "*Nixon qua quaker is a pacifist, while Nixon qua republican is not*" [13, 16]. In a model that countenances the notion of qua individuals, this situation can be modeled by having two different moments *NixonQuaQuaker* and *NixonQuaRepublican* inhering in the object *Nixon*. In this case, we take that whilst the former exemplifies pacifism, the latter does not. The Figure 11

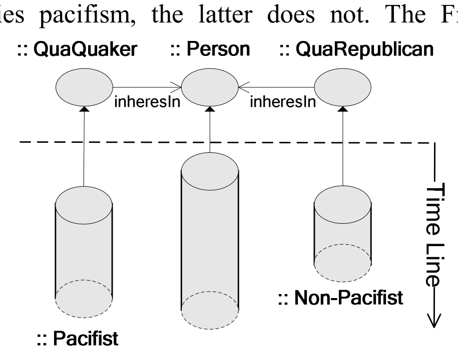


Figure 11. Instantiation schema of Nixon classical example.

[9] For the sake of simplicity, we once more omit from this model the representation of attributes. We should emphasize, nonetheless, that their representation in this proposal is analogous to the solution just discussed for the case of relations.

[10] SQWRL is a query language based on the rule language SWRL (see: http://www.w3.org/Submission/SWRL

illustrates an instantiation of this situation following this proposal (we omit the relators and other supposedly involved individuals).

V. RELATED WORK

In this section we discuss two works in literature that also apply the perdurantist worm view for representing temporally changing information in OWL. These are the works of Welty & Fikes [3] and Krieger [4,5].

A. Ontology for fluents – Welty and Fikes

Welty and Fikes [3] propose the introduction of fluents in an OWL model, where a fluent is considered a relation that holds within a certain time interval. The main notions involved in this approach are depicted (in a white background) in Figure 12. In this proposal, each instance of an ontology concept can be related to one or more of its temporal slices (or temporal parts) via a *temporalPartOf* relation. The domain of this relation is the *TemporalPart* class whilst the range is defined as the complement of the *TimeInterval* class (which we denote as ~*TimeInterval*). A domain ontology should specialize this structure as follows: (i) all domain specific classes are represented as specializations of the ~*TimeInterval* class and (ii) the properties that can change with time are represented between temporal parts via a *fluentProperty* relation whose domain and range are both defined as the anonymous classes that restrict the *TemporalPart* class.

By strictly following these guidelines, it is not obvious how the running example could be properly represented, since the contingent classes *Husband* and *Wife* would have to be placed specializing ~*TimeInterval*, i.e. they would not be treated as contingent. An approximately equivalent representation is presented in Figure 12 (gray shade classes), excluding the aforementioned contingent concepts. Now, Welty and Fikes´ proposal has been conceived at high level of generality and with the aim of achieving reusability. In fact, it can be demonstrated that not only both modeling

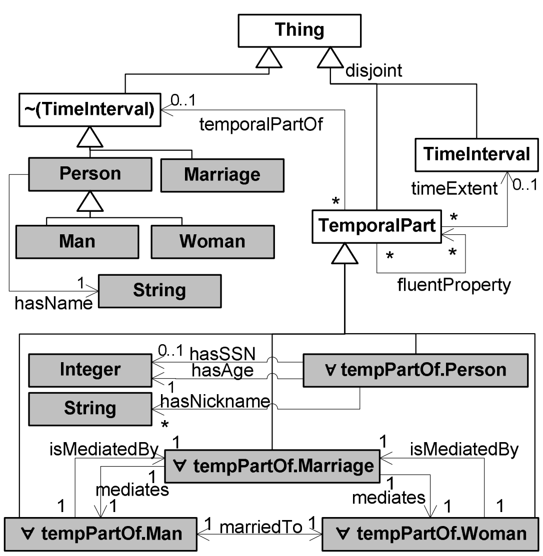


Figure 12. UML-like schema of the OWL structure for the Welty and Fikes's proposal.

strategies presented in our approach can be mapped to their proposal but this mapping is performed in a systematic manner once the corresponding methodological guidelines are followed.

A simplified instantiation schema for the UML-like model of Figure 12 is presented in Figure 13, which intends to represent the same situation presented in Figure 4. Notice that the result is quite similar to that one of our first proposal, except for the non rigid concepts that are not explicitly represented (actually they are represented via the instances of anonymous classes in Figure 12).

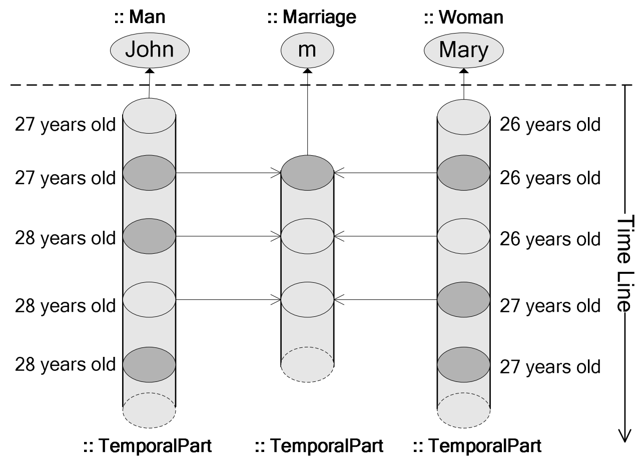


Figure 13. Instantiation schema of the model presented in Figure 12.

B. 4D Reinterpretation - Krieger

Krieger also proposes a perdurantist approach in [4,5], the so-called 4D interpretation, which is based on the ontology for fluents with the intention of reusing existent ontologies. He suggests that "*what has been an entity now becomes a time slice*", i.e. the original classes of an existent ontology become specializations of *TimeSlice* class, thus the original relations are considered fluent properties, and the instances of these classes can compose an entity called *Perdurant*. The main notions involved in this approach are depicted (in a white background) in Figure 14. Moreover he suggests the immutable relations to be represented between perdurants (instead of between the original classes). Besides representing changeable relations, this approach allows representing concepts that an individual contingently instantiates and presents contributions specifically related to time representation and reasoning.

In contrast with the previous approach, a more natural representation of our running example can be obtained by strictly following this one. In this case, the entire model would be placed as a specialization of *TimeSlice*. Although this is the reason why the contingent classes can now be represented, it gives rise to the problem of arbitrary combination of time slices. It means that a perdurant can be composed by, for example, time slices of table, dog, husband, etc, thus opening up the possibility of creating absurd individuals from an ontological point of view [6]. This undesirable situation is prevented in our proposal, since

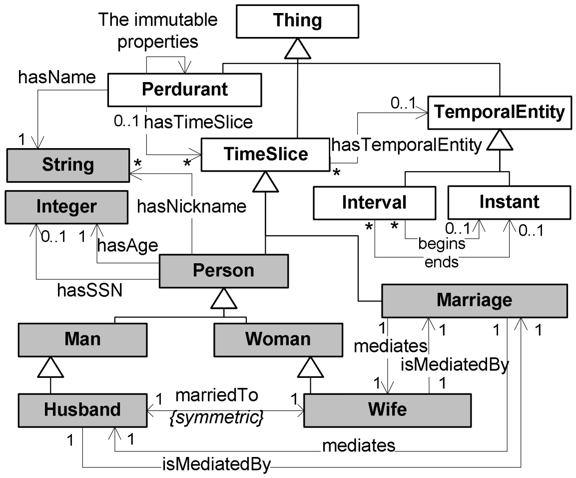


Figure 14. UML-like schema of the OWL structure for the Krieger's proposal.

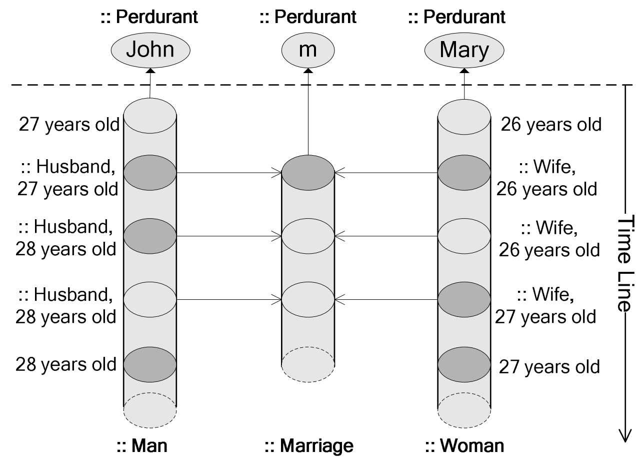


Figure 15. Instantiation schema of the model presented in Figure 14.

there every time slice must be an instance of a counterpart class of exactly one individual concept class. In other words, instances of a certain time slice class must always compose individual concepts of the same class type. Figure 14 presents the representation of the running example following this approach (gray shade classes). Once more, it is possible to map both of our proposals to this approach, and again this mapping is performed in a systematic manner once the corresponding methodological guidelines are followed .

A simplified instantiation schema for this UML-like model is presented in Figure 15, which intends to represent the same situation presented in Figure 4. Notice that the result is quite similar to that one of our first proposal, except for the rigid classes that are not represented at the IC level in the Krieger's proposal.

VI. Final Considerations

In this work we present two complementary and alternative proposals that address the problem of representing temporally changing information in OWL. They are based on the perdurantist theory and benefit from results coming from the discipline of Formal Ontology, in order to restrict the appropriate use of the proposed frameworks.

In the first proposal we combine the perdurantist worm view with the notion of individual concepts for formulating a conceptual structure that allows one to separate the information that (essentially) define the individuals of those that can eventually change. Thereby, we propose an OWL representation for this conceptual structure as well as some methodological guidelines for properly mapping the domain concepts, relations and attributes into the OWL framework. Although this proposal allows one to reasonably represent the intended models, it presents the following drawbacks: (i) proliferation of time slices; (ii) uncanny ontological interpretation of contingent concepts; (iii) repetition of the immutable information on time slice context and (iv) not guaranteeing immutability in the time slice level.

Hereafter, in the second proposal we address some of these drawbacks extending the first proposal with the distinction between objects and moments and the notion of qua individuals. Thereby we reduce the proliferation of time slices, provide an ontological interpretation for the contextually contingent concepts and finally eliminate the repetition and guarantee the immutability of the relations that involve the relators and qua individuals.

As discussed in the previous section, the related works of Welty & Fikes and Krieger are similar to our proposals (even more the first one). However, the lack of commitment to a higher-level foundational framework for guiding the modeling decisions (which is intentional in the first case) has the undesirable side effect of place all the responsibility of making sound modeling choices in the user´s hands.

The table I summarizes the comparison among the two proposals put forth in this paper and the two related works

TABLE I. Addressed Problems

Approach	proliferation of time slices	arbitrary combination of time slices	uncanny ontological interpretation of contingent concepts	repetition of immutable contingent information	not guaranteeing immutability of contingent properties
Ontology for fluents	-	*	-	-	-
4D reinterpretation	-	-	-	-	-
1st proposal	-	+	-	-	-
2nd proposal	+/-	+	+/-	+/-	+/-

* it is not considered a problem since contingent concepts are not addressed

aforementioned. The table compares these four approaches highlighting the existent (-) or addressed (+) problems for each of the approaches. We emphasize that all the approaches address the issue of representing temporally changing information in OWL in a simpler or more complex way, and are applicable depending on the relevance of the associated problems.

We are aware that other strategies have been used for addressing this issue. In particular, in [17], an approach is presented which makes of use of a specific tool to manage information change in the represented models. In that case, the semantic of the change aspects are external to the model, thus becoming dependent on the tool that controls the deletion and insertion of information into the model. In this paper, our intention is to address the representation of temporally changing information in OWL preserving its standard semantics, so that the derived solutions could in principle be reused in all the tools that are compatible with the recommendation.

As future work, we intend to study other approaches based on different strategies, and so apply these theoretical results on proposing a mapping from a modeling language that countenances the referred ontological distinctions [6] (called OntoUML) into OWL. Moreover, we intend to present a generalization of these results to other DL or frame based languages.

ACKNOWLEDGMENT

This research is funded by the Brazilian Research Funding Agencies FAPES (Grant Number 45444080/09) and CNPq (Grant Number 481906/2009-6). V.Z. thanks her NEMO colleagues for many fruitful discussions.

REFERENCES

[1] R. J. Brachman and H. J. Levesque, “Knowledge representation and reasoning,” Morgan Kaufmann, Amsterdam; Boston, 2004.

[2] C. Lutz, F. Wolter, e M. Zakharyaschev, “Temporal Description Logics: A Survey,” Proceedings of the 2008 15th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society, 2008, pp. 3-14.

[3] C. A. Welty and R. Fikes, “A reusable ontology for fluents in OWL,” In: Brandon Bennett and Christiane Fellbaum, editors, FOIS, vol. 150 of Frontiers in Artificial Intelligence and Applications, pp. 226--236. IOS Press (2006)

[4] H. U. Krieger, “Where Temporal Description Logics Fail: Representing Temporally-Changing Relationships,” In: A. Dengel, K. Berns, T.M. Breuel, F. Bomarius, T.R. Roth-Berghofer (eds.): KI 2008: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence number 5243, Pages 249-257, Kaiserslautern, Springer, Berlin (2008)

[5] H. U. Krieger, B. Kiefer and T. Declerck, “A framework for temporal representation and reasoning in business intelligence applications,” In: AAAI 2008 Spring Symposium on AI Meets Business Rules and Process Management . (2008) 59–70

[6] G. Guizzardi, “Ontological foundations for structural conceptual models,” PhD thesis, Enschede, 2005.

[7] I. Horrocks, P. F. Patel-Schneider, F. van Harmelen, From SHIQ and RDF to OWL: The Making of a Web Ontology Language, Journal of Web Semantics 1 (1) (2003) 7-26.

[8] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari. “Ontology Library’, WonderWeb Deliverable D18, 2003a.

[9] A. Varzi, “Naming the Stages,” Dialectica 57: 387–412, 2003

[10] S. Kripke, ‘Naming and Necessity’, Harvard University Press, 1982.

[11] G. Guizzardi, C. Masolo, S. Borgo, “In the Defense of a Trope-Based Ontology for Conceptual Modeling: An Example with the Foundations of Attributes, Weak Entities and Datatypes,” 25th International Conference on Conceptual Modeling (ER’2006), Arizona, USA, 2006.

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

[13] C. Masolo, G. Guizzardi, L., Vieu, E., Botazzi, R. Ferrario, “Relational Roles and Qua Individuals,” AAAI Fall Symposium on Roles, an Interdisciplinary Perspective, Virginia, USA, 2005.

[14] G. Guizzardi, “Agent Roles, Qua Individuals and The Counting Problem,” Software Engineering of Multi-Agent Systems, vol. IV, P. Giorgini, A.Garcia, C. Lucena, R. Choren (eds.), Springer-Verlag, 2006.

[15] M. J. O’Connor and A. K. Das, “SQWRL: a query language for OWL,” In OWL: Experiences and Directions (OWLED), Fifth International Workshop, 2009.

[16] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, N. Guarino, “Social Roles and their Descriptions”, in D. Dubois, C. Welty, M.A. Williams (eds.), Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004), Whistler, Canada, June 2-5, 2004, pp. 267-277, 2004

[17] R. Mizoguchi, E. Sunagawa, K. Kozaki, e Y. Kitamura, “The model of roles within an ontology development tool: Hozo,” Appl. Ontol., vol. 2, 2007, pp. 159-179.