Continuum: A spatiotemporal data model to represent and qualify filiation relationships

Benjamin Harbelot, Helbert Arenas, Christophe Cruz

Continuum: A spatiotemporal data model to represent and qualify filiation relationships

Benjamin Harbelot
Laboratoire Le2i,
UMR-6303 CNRS
Université de Bourgogne
21000 Dijon, France
benjamin.harbelot@u-bourgogne.fr

Helbert Arenas
Laboratoire Le2i,
UMR-6303 CNRS
Université de Bourgogne
21000 Dijon, France
helbert.arenas@checksem.fr

Christophe Cruz
Laboratoire Le2i,
UMR-6303 CNRS
Université de Bourgogne
21000 Dijon, France
christophe.cruz@u-bourgogne.fr

ABSTRACT

This work introduces an ontology-based spatio-temporal data model to represent entities evolving in space and time. A dynamic phenomenon generates a complex relationship network between the entities involved in the process. At the abstract level, the relationships can be identity or topological filiations. The existence of an identity filiation depends on whether the object changes its identity or not. On the other hand, topological filiations are based exclusively on the spatial component, like in the case of growth, reduction, merging or splitting. When combining identity and topological filiations, six filiation relationships are obtained, forming a second abstract level. Upper-level filiation relationships provide better semantic vocabulary to describe the modeled phenomena, thus allowing the implementation of spatial, temporal and identity constraints. In this paper, we present a method based on identity and topological filiation relationships, to improve the capabilities of standard knowledge bases using Semantic Web technologies. Our method enables us to check the consistency of spatio-temporal and semantic data. An example is given in the field of urban growth to show the capabilities of the model.

Keywords

Spatio-temporal modeling, reasoning, semantic, filiation, spatio-temporal evolution, integrity constraints, spatial dynamics

Categories and Subject Descriptors

I.2.4 [**ARTIFICIAL INTELLIGENCE**]: Knowledge Representation Formalisms and Methods – *Semantic networks.*

I.2.3 [**ARTIFICIAL INTELLIGENCE**]: Deduction and Theorem Proving – *Inference engines.*

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications – *Spatial databases and GIS.*

General Terms

Algorithms, Management, Verification.

1. INTRODUCTION

Several spatial and temporal approaches have been proposed to model urban and environmental processes. These approaches allow researchers to study the past and predict trends in the future by addressing the evolution of spatial entities or objects. Considering entities evolving in time, the notions of identity and change have led to the development of several conceptual models. Events and, processes generate complex relationships networks in space and time. There is a need to store a semantic representation of this evolution in the field of GeoInformation Science. A dynamic phenomenon can be defined as a set of entities evolving under the action of a process.

The modeling of entities varying in space and time is a challenge that has received much attention in the community of spatial information. Objects can move or change shape while maintaining their identity, for example, when a city changes its borders. Some objects may change giving rise to new objects, for instance when a land parcel is divided into several sub-parcels. A major concern is the identity of an object through time [**1**,**2**]. The problem is to determine the limits of the object identity through the time, in other words, answer the question: "To what extent can an entity vary before losing its identity?" [**3**,**4**,**5**].

During their existence, the objects maintain relations of various kinds between them. From a spatial perspective, models of topology as 9-intersection [**6**] or Region Connection Calculus [**7**] form the basics of what is known in the literature as qualitative spatial reasoning. By analyzing the topological spatial relationships between the spatial representation of objects co-existing at the same time or at different times, we can better understand the semantics of the relationships between objects. This analysis allows a better understanding of the evolution experienced by an object. An object can continue to exist, originate one or more new objects, and cease to exist. Qualitative spatial reasoning, based on topology, can help determine filiation relationships and help to identify the exact kind of evolution that is occurring.

Since the emergence of GIS, the need to represent real world dynamics, encouraged the scientific community to add a temporal dimension to GIS. Spatio-temporal dynamics generate a large amount of information that must be represented, stored and analyzed. Most of the models proposed in the literature of temporal GIS are based on relational databases [**8**,**1**,**2**,**3**,**4**,**5**,**9**]. While databases provide good capacity of storage and time query response through SQL language, their ability to handle the knowledge stored on the modeled objects is less obvious compared to Semantic Web technologies. The Semantic Web increases the capacity to represent knowledge through classes, relations and properties. In addition, the Semantic Web is widely recognized for its reasoning capabilities allowing to check the consistency of ontologies or to infer new knowledge from existing ones. Because of the previously described reasons the Semantic Web has gradually emerged in the field of GIS [**10**,**11**,**12**,**13**,**14**].

Many languages have been proposed to examine spatio-temporal semantic models. Faced with the heterogeneity of these languages, Open Geospatial Consortium (OGC) has developed the standard GeoSPARQL to represent and query geospatial semantic web data. GeoSPARQL consists in a small ontology that adds the spatial dimension to spatial ontologies. It extends the SPARQL language by providing a set of additional function for handling geometries.

Even though temporal ontologies have been defined [**15**,**16**], there are still elements missing so as to allow an accurate depiction of evolving entities. To solve these limitations the concept of fluent has been introduced, allowing the change of the value of the object's properties over time [**17**,**18**,**19**].

The work presented in this paper introduces a spatio-temporal ontology-based model called the Continuum Model. The Continuum Model tracks the evolution of entities through the time. It combines the spatial functions provided by GeoSPARQL with the temporal capabilities of the fluent ontology. The Continuum Model is additionally enhanced with the definition of filiation relationships that allow to qualify the evolution experimented by the entities. Based on filiation relationships defined in [**1**], the model exploits Semantic Web capabilities defining a filiation relationship hierarchy. By expressing integrity constraints using filiation relationships, it is possible to check consistency in the ontology with respect to the reality modeled. On the other hand, constraints allow inferring filiation relationships or better qualify them with regard to the defined hierarchy. Finally, based on semantic constraints provided in the ontology, the system is able to infer automatically some application dependent upper-level filiation relationships of the modeled field of study.

The paper is organized as follows: Section 2 introduces some basic concepts in order to develop our modeling approach. Section 3 presents spatial and temporal tools available in the Semantic Web literature. Section 4 presents and formalizes the Continuum Model. Section 5 presents integrity constraints for each filiation relationship and presents check-query and inference-query based on the constraints. Section 6 gives an example to switch from general filiation relationships to upper-level filiation relationships on the basis of semantic constraints. Finally, we present our conclusions in section 7.

2. FROM SPATIOTEMPORAL MODELING TO EVOLUTION

Some basic notions of spatio-temporal modeling have to be introduced in order to develop our approach.

2.1 Spatiotemporal entities

A representation of entities of the real world should comprise an identity that describes fixed semantics as well as a dynamic part that represents thematic and spatial properties that can change through time. Another special kind of evolution concerns the identity of the entity. In the literature, we denote two main kinds of spatio-temporal entities: 1) moving object, for example a taxi driving in a town, and 2) changing object, for example, an administrative unit whose borders change over time. In this paper our focus is on the latter.

2.2 Representing the time

The representation of temporal properties of information is a well-researched topic. In [**20**], the authors present an overview of the conceptual modeling of time. In our work, we consider a discrete, linearly-ordered time domain with a focus on absolute time. In our model, time is incorporated by associating intervals with class instances in the ontology. Intervals denote the time during which all the relations and properties of an instance are valid. Each interval is defined by start and end instants. The temporal granularity is application dependent, therefor the time units (e.g. Hours, Days, Weeks, Months, Seasons, Years) must be specified for each instant of time in the ontology.

2.3 Representing the topology

From the human point of view, when we observe a landscape, objects such as forests, buildings, rivers, etc. are "seen" or "perceived" in their context. The concept of neighborhood is implicit. For example, a river runs "through" a parcel, a building "is located in" agglomeration, an agglomeration is "juxtaposed" in a larger settlement. This kind of observation of geographic space has led to the study of topology. From the geographical point of view, the topology is the set of perceived relationships that allow us to situate an object in relation to others. In the literature, several models [**6**,**7**] have been proposed to study the topology of a geographical space. In all cases, we get an equivalent set of 8 spatial topological predicates (equals, disjoint, intersects, touches, crosses, overlaps, within, contains).

From a temporal point of view, topology also exists and has been introduced by [**21**] in order to represent actions and events occurring over time, Allen offers a representation based on temporal intervals. An interval is delimited by defining a start and a later end instant of time. Thus, Allen describes 13 mutually exclusive relationships between time intervals. The first six relations are *before*, *meets, overlaps, during, starts* and *finishes* have reverse, respectively, *after, met-by, overlapped-by, contains, started by* and *finished-by*. The last relationships is *equals* which does not have an inverse.

2.4 The nature of changing

Changes can occur on the position and shape of an object. In addition, knowledge about an object can evolve when changes occur on thematic attributes of objects. These space and semantic changes can occur simultaneously or independently from each other, giving rise to 8 possible different scenarios [**22**] as depicted in figure 1.

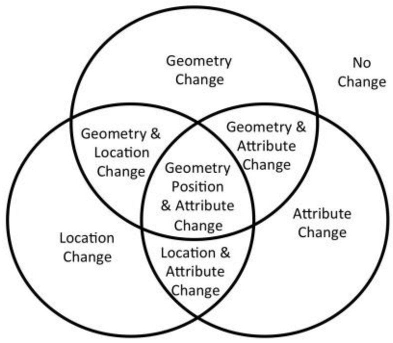

Figure 1: Eights different scenarios of change

The change can be continuous or discrete. We can classify models based on the type of changes they are able to support. In our model, we focus on shape and attribute changes. Moreover, we highlight a particular type of attribute change: identity modification.

2.5 Identity

The identity is a unique feature that distinguishes one object from another, which differentiates it from other properties, values or structures. Objects can change their position and shape while keeping their identity. In other cases new entities may be formed from ancient ones. A major issue then arises: "How far can an entity vary before losing its identity?". Sometimes due to modeling constraints it may be more appropriate to destroy an object and to create a new one due to a large number of changes. The identity is a key point to prove the existence or non-existence of an object as well as to list the similarities or the differences between objects. The existence refers to the physical presence of an object. For the conceptual objects, the existence is the perception of an object. There are generally three states of identity [**4**]: creation, continuation and elimination. There are previous research that identify more states of identity as in [**5**] which defines nine, but it is worth noting that in certain contexts, some of these states of identity seem to be impossible or contradictory to a domain ontology and therefore are not considered.

2.6 Filiation Relationships

Filiation relation defines the succession link that exists between different representations of the same object at different moments of time. This relationship is essential to maintain the identity of an entity that evolves. It is also necessary to identify children entities that might have been created due to some evolution. Filiation relation is intimately linked to the notion of identity. The establishment of a filiation relation induces a dependency in the identity. There are then two general types of filiation relations: continuation and derivation [**1**,**2**,**3**,**4**,**5**]. In the first case, identity is retained. The entity continues to exist but has experienced a change. While in the former, a new entity is created from another after being subjected to an evolution. Unlike the relations of continuation, derivation relations may concern several entities at the same time.

3. AVAILABLE TOOLS FOR SPATIO-TEMPORAL AND SEMANTIC DEFINITION

Several languages and ontologies have been proposed in order to enable the Geospatial Semantic Web [**23**]. Among these we can find GeoSPARQL, an OGC standard designed to provide the foundation for spatial reasoning and manage spatial data in RDF. In the time domain, there exists a proposed ontology to represent time, but the definition of time does not allow changing objects, which has led to the emergence of fluent.

3.1 Handling of spatial properties in the Semantic Web

GeoSPARQL is an emerging standard within the Open Geospatial Consortium (OGC). The main purposes of GeoSPARQL are: 1) provide a standard way to express and query spatial objects in RDF, 2) allow users to exchange data easily, and 3) provide a standard spatial indexing for triple stores.

3.1.1 Overview

Since it became an OGC standard, GeoSPARQL [**24**] has several arguments to be selected as a powerful language in geospatial application. Several spatially-enable SPARQL endpoints like Strabon[1], Parliament[2] or OpenSahara[3] implements GeoSPARQL. Another strength of GeoSPARQL is to be built on existing standards. In fact, it is built on W3C Semantic Web standards like RDF, OWL and SPARQL but also on OGC standards like Simple Features and Spatial Relations. As a result, GeoSPARQL provides a standardized vocabulary for representing linked geodata and for writing SPARQL queries against geospatial RDF data. It also reuses common geometry serialization formats like GML, KML, and WKT stored as strings and encoded as RDF Literals. Besides, it provides a structured vocabulary and semantics for geographic features and relationships. Finally, GeoSPARQL provides the ability to answer queries involving geographic features and relationships.

3.1.2 Enabling the spatial dimension

GeoSPARQL includes a small spatial ontology in RDFS/OWL for representation of spatial entities. There are three main classes in the GeoSPARQL ontology:

• geo:Feature : something which can have a spatial location such as a park or monument, etc.

• geo:Geometry : a representation of the location space, i.e., a set of coordinates

• geo:SpatialObject : the superclass of “Feature” and “Geometry”.

The *geo:hasGeometry* property links the “Features” (a thing) to their spatial representation. GeoSPARQL allows to associate multiple geometries to an object. The resource associated to the geometry then has an RDF literal representation, to which property is linked, named after the type of representation. For example, the *geo:asWKT* property connects geometry resource with a *wktLiteral*. The literal contains the geometry information in the specified format.

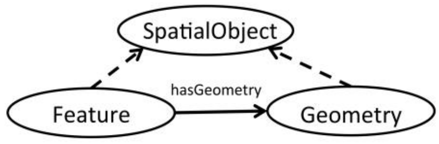

Figure 2: the GeoSPARQL ontology

GeoSPARQL implements a list of spatial concepts described in OGC/ISO Simple Features (ISO 19125), such as Point, Line, Polygon, among others, which could be specialized or generalized in a geometry concept hierarchies. It also provides metadata for each spatial object. Metadata concerns elements such as dimension, SRID (Spatial Reference system Identifier). Additionally GeoSPARQL supports topological relationships based on DE-9IM.

3.1.3 Query data with GeoSPARQL

GeoSPARQL provides a SPARQL query interface using a set of topological SPARQL extension functions for quantitative reasoning. GeoSPARQL covers different types, spatial properties, operations and relationships in order to express spatial queries. Thus, we can find descriptive datatype properties (e.g., dimension), topological relations (e.g., touches, intersects, contains) which can be computed with RCC8, 9-intersection, or Simple Feature. Furthermore, GeoSPARQL provides parameterized relations (e.g., within distance) and operations that produces new objects (e.g., buffer, union, intersect). All these types of spatial capabilities have been encoded into extensible filter functions.

3.2 The temporal component in the Semantic Web

One of the goals of the Semantic Web is to treat the temporal aspects of entities and perform reasoning with this information. However it is common to find ontologies representing the world as static, while in reality, the world is constantly evolving. In this section we describe previous research on the semantic representation of the temporal nature of objects.

3.2.1 Ontology of time

Several ontologies have been proposed in the literature in order to represent time. The most famous is OWL-Time. OWL-Time is an ontology of time designed for describing the temporal content of web pages and the temporal properties of

[1] http://www.strabon.di.uoa.gr/

[2] http://parliament.semwebcentral.org/

[3] https://dev.opensahara.com/

web services [**15**]. OWL-Time represents both instants and intervals of time. Properties are defined with specific time intervals. It supports the 13 Allen relationships to specify relations between intervals. Ontology OWL-Time also allows the use of time units such as seconds, minutes, hours, days, etc. Finally OWL-Time is the only known ontology to enable temporal aggregates, i.e. to represent expressions such as "every first Tuesday of the week". OWL-Time consists of a main class named *TemporalEntity*, which has two subclasses *Instant* and *Interval*. *Begins* and *ends* properties have been defined to specify the beginning and end of a temporal entity. Other properties have been defined in order to enrich the relations between temporal entities: *inside* relation applies on instants and allows to define the latter as part of an interval. The class *Interval* has a subclass *ProperInterval* that corresponds to a range whose start and end times are not equal. Another ontology of time used to represent temporal information in OWL based on intervals of time is SWRL Temporal Ontology [**16**]. It represents time similar to OWL-Time, except that it does not allow to establish a topological relationship between instants or time intervals. However, SWRL Temporal Ontology offers some temporal built-ins for SWRL, a rule-based language. These built-ins allow checking of Allen relationships between different temporal entities when querying.

3.2.2 Ontology of Fluent

Ontologies of time are very similar in the way they represent time. But representing time is not sufficient to represent the evolution of an object. In order to do this we need upper-level temporal ontology based on ontologies of time. In [**17**], the authors introduce ontology of fluent to represent the fact that the properties of objects vary over time. The approach, called 4D-fluent, considers objects as having temporal parts, called *timeslices*, and the representation of change in the properties of an object is done through the use of *fluent*, which are properties that are valid only during a certain time interval. When the property of an object changes, a new *timeslice* is established, holding the new property of the object. The 4D-fluent approach was intended to model perdurant objects which are objects having several temporal parts representing them during a certain interval of time. An entity is defined by its temporal parts: *timeslices*. The properties that are valid only during a certain time interval are called *fluent*. In the 4D-Fluent approach, the used *TimeInterval* class corresponds to the *Interval* class defined in OWL-Time. Considering that the time interval associated with the *timeslice* represents the period during which the associated *fluent* is valid. If a *timeslice* is associated with many *fluent* then these *fluent* must be valid on the same interval of time.

A more recent development is TOWL (Time-determined Ontology Web Language) [**18**,**19**]. It is a language to extend the OWL language with a temporal dimension allowing the representation of time, change and state transitions. TOWL proposes a four layer architecture in order to expand the capabilities of OWL-DL as depicted in the figure 3.

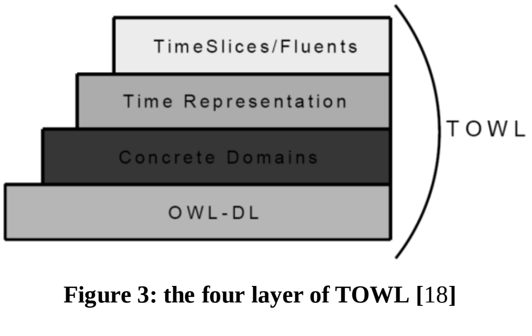

Figure 3: the four layer of TOWL [18**]**

The first layer of TOWL *OWL-DL* does not require further explanation. The second layer entitled *Concrete Domain*s allows the representation of intervals and supports the 13 Allen relationships between intervals. The next layer is used to represent the time in a quantitative manner. Finally, the last layer uses the idea of perdurant objects using *timeslices* and *fluent* from the 4D-fluent approach.

However, some differences exist between 4D-fluent and TOWL. First, in TOWL, the *timeslices* are not necessarily associated with a time interval but can also be associated with an instant of time. When defining a *fluent* between two *timeslices*, the 4D-fluent approach requires the two *timeslices* to be associated with the same interval of time. With TOWL, the *timeslices* can have their own time interval, however, when a fluent is defined between two *timeslices* the equal relationship must be defined in order to ensure that these *timeslices* are valid on the same interval of time.

Another contribution of TOWL is to specify type of fluent. While with 4D-fluent, we had only a *fluentProperty* property, TOWL distinguishes between the *fluentObjectProperty* and the *fluentDatatypeProperty*. A sub-property of *fluentObjectProperty* connects two *timeslices* while a *fluentDatatypeProperty* property connects a *timeslice* to an object of type *rdfs:Literal* and thus indicates changes in values.

3.3 Discussion

GeoSPARQL is an efficient standard that enables geospatial Semantic Web by providing a small reusable spatial ontology. GeoSPARQL is implemented in efficient triplestores allowing topology based calculations as well as other geospatial operations. However the temporal aspect is still excluded from this standard. To represent time some well-known ontologies such as OWL-Time and SWRL Temporal Ontology have been proposed in the literature. Representing time differs from representing evolution. That's why upper-level ontology of time called ontology of fluent has recently emerged. But this ontology of fluent is rarely used in the field of spatio-temporal modeling. A notable exception is [**25**], however, this research does not implement filiation relationships in order to track entities over time and understand the kind of evolution that objects might have suffered. Additionally, the query language proposed is not a standard and suffers from lack of tools implementing it.

4. THE CONTINUUM MODEL

In this section, we present our spatio-temporal model inspired in GeoSPARQL and ontology of fluent in order to represent entities over time.

4.1 Overview

The Continuum Model extends GeoSPARQL allowing it to represent spatio-temporal dynamic objects. This extension is achieved by combining the GeoSPARQL Ontology with an ontology of fluents. Therefore, The *feature* class corresponding to entities with spatial component is now associated with *timeslices*. In this way we are able to create multiple representations of the object, each, corresponding to different instants or time intervals. Unlike GeoSPARQL, the *hasGeometry* relationship is no longer established between the *feature* and the *geometry* class but rather between a *timeslice* and a *geometry* allowing an object of the class feature to change its geometry (See Figure 4).

As in a fluent approach, the timeslice represents the state of an object during a certain interval or instant of time. When a property or relationship changes a new timeslice is created.

In the case of a relationship between two different objects, a relationship is valid only between timeslices of both objects coexisting at the same time. Therefore, this interval must be the same for each of the timeslices of objects affected by this relationship.

The design of our model has been influenced by [**1**,**2**]. In addition to *hasGeometry* and relations *hasTime* respectively for linking geometry and an interval of time to a timeslice, the model presented in this report implements the relationships 1) *hasSpatialRelation*, 2) *hasFiliation*. Figure 4 depicts the classes and relations implemented in our model.

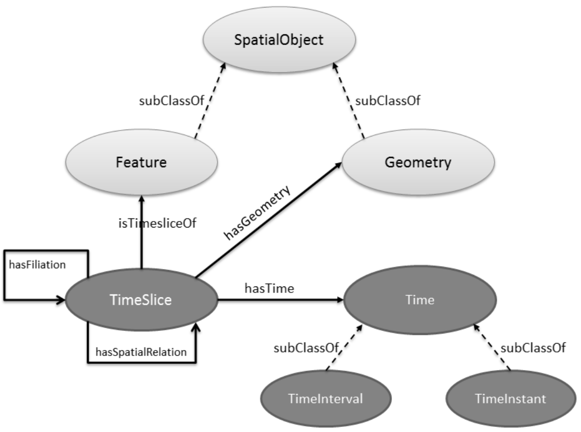

Figure 4: The core architecture of our proposition

The relationship *hasSpatialRelation* allows expressing the spatial topology between objects and must be established between two *timeslices* of different objects valid on the same interval or instant of time. Finally, our model focuses on the *hasFiliation* relationship. This relation is established between two *timeslices*, it allows to track and define the evolution of an object in the case of a continuation, and to track and define the transformation of one object to another in the case of a derivation. This relationship is designed to translate the semantic of the transition between two consecutive *timeslices*. This relationship can be specified more precisely combining identity-filiation and topological-filiation. Our model exploits the capabilities of specialization inherent in Semantic Web.

Figure 5 depicts the hierarchy of filiation-relation defined in the model. Details about each relationship are given in the section "Filiation relation integrity constraints".

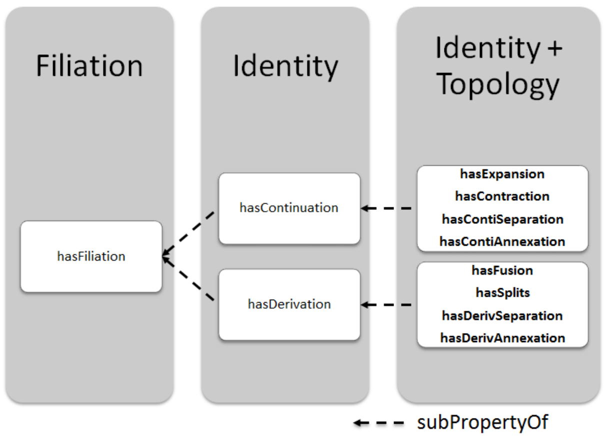

Figure 5: different level of filiation relationships

4.2 OWL DL Continuum Model definition

To present our model, we will adhere to the Karlsruhe Ontology Model [**26**]. An OWL DL ontology is a structure O:=(SO, σO, FO) consisting of:

- The underlying set SO containing:

Six disjoint sets sC, sT, sR, sA, sI, sV, sK_R and sK_A called concepts, datatypes, relations, attributes, instances, data values, relation characteristics (among Symmetric, Functional, Inverse Functional, Transitive) and attribute characteristics (Functional),

Four partial orders $\leq_C$ $\leq_T$, $\leq_R$ and $\leq_A$, respectively on sC called concept hierarchy or taxonomy, on sT called type hierarchy, on sR called relation hierarchy and on sA called attribute hierarchy,

such that SO :={(sC, $\leq_C$), (sT, $\leq_T$), (sR, $\leq_R$), (sA, $\leq_A$), sI, sV, sK_R, sK_A},

- The signature σO containing:

Two functions σ_R:sR→sC^2 called relation signature and σ_A:sA→sC × sT called attribute signature, such that σO :={σ_R, σ_A},

- The interpretation function FO containing the functions:
  - ι_C:sC→ 2^{sI} called concept instantiation,
  - ι_T:sA→ 2^{sV} called data type instantiation,
  - ι_R:sC→ $2^{sI \times sI}$ called relation instantiation,
  - ι_A:sC→ $2^{sI \times sV}$ called attribute instantiation,
  - κ_R:sR→ 2^{sKR} called relation characterization,
  - κ_A:sA→2^{sKA} called attribute characterization,
  - $-_R$:sR→2^{sR} called relation inverse specification,

- sC={ TopConcept, SpatialObject, Feature, Geometry, TimeSlice, Time, TimeInterval, TimeInstant},
- sT={TopType, dateTime, String},
- $\leq_C$ ={(TopConcept, SpatialObject), (TopConcept, TimeSlice), (TopConcept, Geometry), (TopConcept, Time), (SpatialObject, Feature), (SpatialObject, Feature), (Time, TimeInterval), (Time, TimeInstant)},
- $\leq_T$ ={(TopType, Float) (TopType, String)},
- sR={ TopRelation, hasFiliation, hasSpatialRelation, hasTemporalRelation, **isTimesliceOf**, **hasGeometry**, **hasTime**[4]**,** asWKT, hasDisjoint , hasOverlaps, hasTouches, hasContains, hasWithin, hasEqual, hasContinuation, hasDerivation, hasExpansion, hasContraction, hasSplits, hasFusion, hasDerivSeparation, hasContiSeparation, hasDerivAnnexation, hasContiAnnexation, before, meets, overlaps, during, starts, finishes, after, metBy, overlappedBy, contains, startedBy, finishedBy, Equal },
- sA={TopAttribute, hasBeginInstant, hasEndInstant, hasTimeInstant },
- σ_R={(hasFiliation (TimeSlice, TimeSlice)), (hasSpatialRelation, (TimeSlice, TimeSlice)), (isTimesliceOf,(TimeSlice, Feature)), (hasGeometry,(TimeSlice, Geometry)), (hasTime,(TimeSlice, Time)) , (hasDisjoint,(TimeSlice, TimeSlice)),(hasOverlaps,(TimeSlice, TimeSlice)), (hasTouches,(TimeSlice, TimeSlice)), (hasWithin,(TimeSlice, TimeSlice)),(isWithin,(TimeSlice, TimeSlice)), (hasDisjoint,(Geometry, Geometry)),(hasOverlaps,(Geometry, Geometry)), (hasTouches,(Geometry, Geometry)), (contains,(Geometry, Geometry)),(within,(Geometry, Geometry)) ,(equal,(Geometry, Geometry)) ,(hasContinuation,(TimeSlice, TimeSlice)), (hasDerivation,(TimeSlice, TimeSlice)), (hasExpansion,(TimeSlice, TimeSlice)), (hasContraction,(TimeSlice, TimeSlice)), (hasSplits,(TimeSlice, TimeSlice)), (hasFusion,(TimeSlice,

[4] Relations in bold represent the core of our model. More explanations are given in the next section.

TimeSlice)), (hasDerivSeparation,(TimeSlice, TimeSlice)), (hasContiSeparation,(TimeSlice, TimeSlice)), (hasDerivAnnexation,(TimeSlice, TimeSlice)), (hasContiAnnexation,(TimeSlice, TimeSlice)),(equal,(TimeInterval, TimeInterval)), (before,(TimeInterval, TimeInterval)), (after,(TimeInterval, TimeInterval)), (meets,(TimeInterval, TimeInterval)), (metBy,(TimeInterval, TimeInterval)), (overlaps,(TimeInterval, TimeInterval)), (overlappedBy,(TimeInterval, TimeInterval)), (during,(TimeInterval, TimeInterval)), (contains,(TimeInterval, TimeInterval)), (starts,(TimeInterval, TimeInterval)), (startedBy,(TimeInterval, TimeInterval)), (finishes,(TimeInterval, TimeInterval)), (finishedBy,(TimeInterval, TimeInterval))},

- σ_A={(hasBeginInstant (TimeInterval, dateTime)), (hasEndInstant (TimeInterval, dateTime)), (hasTimeInstant (TimeInstant, dateTime)), (asWKT (Geometry, String))},
- $\leq_R$={(TopRelation, hasFiliation), (TopRelation, hasSpatialRelation), (TopRelation, isTimesliceOf), (TopRelation, hasGeometry), (TopRelation, asWKT), (TopRelation, hasTime), (hasSpatialRelation, hasDisjoint), (hasSpatialRelation, hasOverlaps), (hasSpatialRelation, hasTouches), (hasSpatialRelation, contains), (hasSpatialRelation, within), (TopRelation, hasTemporalRelation), (hasTemporalRelation, before) , (hasTemporalRelation, after) , (hasTemporalRelation, meets) , (hasTemporalRelation, metBy) , (hasTemporalRelation, overlaps) , (hasTemporalRelation, overlappedBy) , (hasTemporalRelation, during) , (hasTemporalRelation, contains) , (hasTemporalRelation, starts) , (hasTemporalRelation, startedBy) , (hasTemporalRelation, finishes) , (hasTemporalRelation, finishedBy) , (hasTemporalRelation, equal) }
- $\leq_A$={(TopAttribute, hasBeginInstant),(TopAttribute, hasEndInstant) ,(TopAttribute, hasTimeInstant)},
- sK_R={Transitive, Symmetric, Functional},
- sK_A={Functional},
- κ_R={(hasDisjoint, {Symmetric}), (hasOverlaps, {Symmetric}), (hasTouches, {Symmetric}) , (hasEqual, {Symmetric}), (hasContains, {Transitive}), (hasWithin,{Transitive}), (hasContinuation,{Functional}), (hasExpansion,{Functional}), (hasContraction,{Functional}), (hasContiSeparation,{Functional}), (hasContiAnnexation,{Functional}), (equal, {Symmetric}), (during, {Transitive}), (contains,{Transitive}),
- κ_A={(hasBeginInstant, {Functional}), (hasEndInstant, {Functional}), (hasEndInstant, {Functional}), (asWKT, {Functional}) },
- $-_R$={(hasContains,{hasWithin}), (before,{after}), (meets,{metBy}), (overlaps,{overlappedBy}), (during,{contains}), (starts,{startedBy}), (finishes,{finishedBy}) },

5. QUALIFICATION OF FILIATION AND REASONING WITH THE CONTINUUM MODEL

This section aims at formalizing integrity constraints for our spatio-temporal ontology. Based on [**1**], we provide a set of integrity constraints and SPARQL queries for checking consistency of filiation relation in the ontology. Finally, we propose to infer filiation relation on the basis of integrity constraints and increase the knowledge combining the filiation relation with semantic defined in the ontology.

5.1 Filiation relation integrity constraints

In this section we use a Tarski-style specification to describe the model filiation-relations. To represent time intervals we follow the semantics suggested in [**27**]. We can think of the temporal domain as a linear structure T composed by a set of temporal points P. The components of P follow a strict order <, which forces all points between two temporal points t1 and t2 to be ordered. By selecting a pair [t1; t2] we can limit a closed interval of ordered points. The set of interval structures in *T* is represented by $T^*_<$.

Temporal Points:

$P \qquad P^I \subseteq \Delta^I$

Time Intervals:

$T^*_< \qquad [t_0,t_f] = \{x\epsilon\, P | t_0 \leq x \leq t_f, t_0 \neq t_f\}\ in\ T$

In the Continuum Model, t_0 is defined by the datatype property *hasBeginInstant* and t_f is defined by the datatype property *hasEndInstant*. The spatial representation of the *timeslice* of an object is given through its geometry (*G*). The semantic component of the timeslice of an object is represented by *S*. It describes the nature of the entities and can be made up of one or more alphanumeric properties. Finally, a timeslice has an identity held by the property *isTimesliceOf* which connects it to an object (*O*).

Each timeslice (*TS*) in the model has four components: 1) a time interval ($T^*_<$), 2) a geometry (*G*), 3) an identity (*O*) and 4) a semantic component (*S*) representing all other potential properties associated to a timeslice. We define all these properties using $\overline{TS}$ base symbol, as defined in [**28**], which stands for the qualities that distinguish a timeslice from another apart from its interval of existence, identity, and its geometry (note that $S \equiv \overline{TS}$):

$TS \equiv \forall hasGeometry.G \sqcap \forall hasTime.T^*_< \sqcap \overline{TS} \sqcap \forall isTimesliceOf.O$

In the Continuum Model a change on the spatial representation or on the semantic component generates a new timeslice which has a filiation relationship with the original timeslice, additionally we know that the time interval of the parent timeslice meets the time interval of the child timeslice. The filiation relationship between timeslice ts_1 and ts_2 is defined by the relationships between their spatial representations (ts_{1g} and ts_{2g}), their semantic definitions (ts_{1s} and ts_{2s}), their identity (ts_{1o} and ts_{2o}) and their time intervals (ts_{1i} and ts_{2i}). A filiation relationship is defined when a change occurs on the geometry, the semantic component or the identity.

$\forall hasFiliation.TS\ \{ ts_1 \in TS^I \mid \forall ts_2.(ts_1, ts_2) \in hasFiliation^I \rightarrow ts_2 \in TS^I \wedge \exists((ts_{1g} \neq ts_{2g}) \vee (ts_{1s} \neq ts_{2s}) \vee (ts_{1o} \neq ts_{2o})) \wedge (meets(ts_{1i}, ts_{2i}))\}$

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

We can further specialize the *hasFiliation* to define more complex relationships:

hasContinuation : In this case a change may occur only on the geometry or the semantic component but the identity is maintained.

$\forall hasContinuation.TS\ \{ ts_1 \in TS^I \mid \forall ts_2.(ts_1, ts_2) \in hasContinuation^I \rightarrow ts_2 \in TS^I \wedge \exists((ts_{1g} \neq ts_{2g}) \vee (ts_{1s} \neq ts_{2s})) \wedge (ts_{1o} = ts_{2o}) \wedge (meets(ts_{1i}, ts_{2i}))\}$

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasDerivation: In this case the change may occur only on the geometry or the semantic component, while the identity must be different.

$\forall$*hasDerivation*.TS { $ts_1 \in TS^I$ | $\forall ts2.(ts_1,ts_2) \in hasDerivation^I \rightarrow ts_2 \in TS^I \wedge \exists((ts_{1g} \neq ts_{2g}) \vee (ts_{1s} \neq ts_{2s})) \wedge (ts_{1o} \neq ts_{2o}) \wedge (meets(ts_{1i}, ts_{2i}))$}

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasExpansion: In this relationship the entity continues to exist but the geometry grows.

$\forall$*hasExpansion*.TS { $ts_1 \in TS^I$ | $\forall ts_2.(ts_1,ts_2) \in hasExpansion^I \rightarrow ts_2 \in TS^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1o} = ts_{2o}) \wedge meets(ts_{1i}, ts_{2i}) \wedge hasWithin(ts_{1g}, ts_{2g})$ }

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasContraction: In this case there is a reduction in the geometry size.

$\forall$*hasContraction*.TS { $ts_1 \in TS^I$ | $\forall ts_2.(ts_1,ts_2) \in hasContraction^I \rightarrow ts_2 \in TS^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1o} = ts_{2o}) \wedge meets(ts_{1i}, ts_{2i}) \wedge hasContains(ts_{1g}, ts_{2g})$ }

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasSplits: In this relationship, the parent entity ceases existing. While its geometry is divided, generating two new geometries corresponding each to a new entity. The union of the new geometry is equal to the former geometry

$\forall$*hasSplits*.TS { $ts_1 \in TS^I$ | $\forall ts_2.(ts_1,ts_2) \in hasSplits^I$ | $\forall ts_3.(ts_1,ts_3) \in hasSplits^I \rightarrow ts_2 \in TS^I \wedge ts_3 \in TS^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge (ts_{1o} \neq ts_{2o}) \wedge (ts_{1o} \neq ts_{3o}) \wedge (ts_{2o} \neq ts_{3o}) \wedge meets(ts_{1i}, ts_{2i}) \wedge meets(ts_{1i}, ts_{3i}) \wedge equal(ts_{2i}, ts_{3i}) \wedge hasEqual(ts_{1g}, (ts_{2g} \cup ts_{3g}))$ }

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasSeparation[5]: In this case the parent entity continues existing, however its geometry originates a new geometry corresponding to a new entity. A "hasSeparation" relationship is similar to a "hasSplits" relationship with the difference that in *hasSeparation* the original entity remains.

$\forall$*hasSeparation*.TS { $ts_1 \in TS^I$ | $\forall ts_2.(ts_1,ts_2) \in hasSeparation^I$ | $\forall ts_3.(ts_1,ts_3) \in hasSeparation^I \rightarrow ts_2 \in TS^I \wedge ts_3 \in TS^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge (ts_{1o} = ts_{2o}) \wedge (ts_{1o} \neq ts_{3o}) \wedge (ts_{2o} \neq ts_{3o}) \wedge meets(ts_{1i}, ts_{2i}) \wedge meets(ts_{1i}, ts_{3i}) \wedge equal(ts_{2i}, ts_{3i}) \wedge hasEqual(ts_{1g}, (ts_{2g} \cup ts_{3g}))$ }

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasFusion: In this relationship the two parent entities merged and cease to exist to give rise to a new geometry corresponding to a new entity. Inverse to a "hasSplits" relationship. The resulting geometry is equal to the union of the former geometries.

$\forall$*hasFusion*.TS { $ts_1 \in TS^I$ | $\forall ts_2.(ts_1,ts_2) \in hasFusion^I$ | $\forall ts_3.(ts_1,ts_3) \in hasFusion^I \rightarrow ts_2 \in TS^I \wedge ts_3 \in TS^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge (ts_{1o} \neq ts_{2o}) \wedge (ts_{1o} \neq ts_{3o}) \wedge (ts_{2o} \neq ts_{3o}) \wedge meets(ts_{1i}, ts_{2i}) \wedge meets(ts_{2i}, ts_{3i}) \wedge equal(ts_{1i}, ts_{2i}) \wedge hasEqual((ts_{1g} \cup ts_{2g}), ts_{3g})$ }

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

hasAnnexation[6]: In this case the two parent entities merge but the resulting entity keeps the identity of one of its parents.

$\forall$*hasAnnexation*.TS { $ts_1 \in TS^I$ | $\forall ts_2.(ts_1,ts_2) \in hasAnnexation^I$ | $\forall ts3.(ts1,ts3) \in hasAnnexation^I \rightarrow ts_2 \in TS^I \wedge ts_3 \in TS^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge (ts_{1o} \neq ts_{2o}) \wedge (ts_{1o} = ts_{3o}) \wedge (ts_{2o} \neq ts_{3o}) \wedge meets(ts_{1i}, ts_{2i}) \wedge meets(ts_{2i}, ts_{3i}) \wedge equal(ts_{1i}, ts_{2i}) \wedge hasEqual((ts_{1g} \cup ts_{2g}), ts_{3g})$ }

Where : $\{ts_1, ts_2\} \in TS$, $\{ts_{1g}, ts_{2g}\} \in G$, $\{ts_{1s}, ts_{2s}\} \in S$, $\{ts_{1o}, ts_{2o}\} \in O$, $\{ts_{1i}, ts_{2i}\} \in I$

5.2 Checking consistency of the filiation relation

A given dynamic process involving multiple entities generates large amounts of spatiotemporal relationships. The handling of this large amount of information is prone to errors. Therefore there is a need to enforce integrity constraints and develop mechanisms to detect inconsistencies. In this section, we present some SPARQL check-queries designed to detect filiation relations that do not satisfy all the corresponding integrity constraints. All queries follow the same strategy. We retrieve all filiation relations in the ontology and we subtract from all relations which satisfy the corresponding integrity constraints. As a result, we obtain a filiation relation defined in the ontology which does not satisfy integrity constraints. If the answer to the query is empty, it implies that the ontology satisfies the constraints. Note that we show only some examples to prove the feasibility of our purpose.

Example 1: Check-query for filiation relation

```
SELECT  ?ts1 ?ts2 WHERE {
        ?ts1 ex:hasFiliation ?ts2 .
        MINUS{
                ?ts1 ex:isTimesliceOf ?o1  .
                ?ts1 geo:hasGeometry ?g1 .
                ?ts1 time:hasTime ?t1 .
                ?ts1 ?otherProperties ?attribut1.
                ?ts2 ex:isTimesliceOf ?o2  .
                ?ts2 geo:hasGeometry ?g2 .
                ?ts2 time:hasTime ?t2 .
                ?ts2 ?otherProperties ?attribut2.
                FILTER ((?g1 != ?g2 || ?o1 != ?o2 ||
                ?attribut1 != ?attribut2) &&
                temporal:meets(t1, t2) &&
                ?otherProperties != geo:hasGeometry
                && ?otherProperties != time:hasTime
                && ?otherProperties !=
                ex:isTimesliceOf )
        }
}
```

Example 2: Check-query for the continuation relation

```
SELECT  ?ts1 ?ts2 WHERE {
        ?ts1 ex:hasContinuation ?ts2 .
        MINUS{
                ?ts1 ex:isTimesliceOf ?o1  .
                ?ts1 geo:hasGeometry ?g1 .
                ?ts1 time:hasTime ?t1 .
                ?ts1 ?otherProperties ?attribut1.
                ?ts2 ex:isTimesliceOf ?o2  .
                ?ts2 geo:hasGeometry ?g2 .
                ?ts2 time:hasTime ?t2 .
                ?ts2 ?otherProperties ?attribut2.
                FILTER ((?g1 != ?g2  || ?attribut1
                != ?attribut2) && ?o1 = ?o2 &&
                temporal:meets(t1, t2) &&
                ?otherProperties != geo:hasGeometry
```

[5] In our ontology, we define two separation relations which are *hasContiSeparation* and *hasDerivSeparation* in order to distinguish easily entities which continues existing and entities ceasing to exist when a separation occurs.

[6] As in separation-filiation, we provide two annexation-relations which are *hasContiAnnexation* and *hasDerivAnnexation*..

```
            && ?otherProperties != time:hasTime
            && ?otherProperties !=
            ex:isTimesliceOf )
            }
}
```

Example 3: Check-query for the derivation relation

```
SELECT  ?ts1 ?ts2 WHERE {
        ?ts1 ex:hasDerivation ?ts2 .
        MINUS{
            ?ts1 ex:isTimesliceOf ?o1  .
            ?ts1 geo:hasGeometry ?g1 .
            ?ts1 time:hasTime ?t1 .
            ?ts1 ?otherProperties ?attribut1.
            ?ts2 ex:isTimesliceOf ?o2  .
            ?ts2 geo:hasGeometry ?g2 .
            ?ts2 time:hasTime ?t2 .
            ?ts2 ?otherProperties ?attribut2.
            FILTER ((?g1 != ?g2  || ?attribut1
            != ?attribut2) && ?o1 != ?o2 &&
            temporal:meets(t1, t2) &&
            ?otherProperties != geo:hasGeometry
            && ?otherProperties != time:hasTime
            && ?otherProperties !=
            ex:isTimesliceOf )
            }
}
```

Example 4: check-query for the expansion relation

```
SELECT  ?ts1 ?ts2 WHERE {
        ?ts1 ex:hasExpansion ?ts2 .
        MINUS{
            ?ts1 ex:isTimesliceOf ?o1  .
            ?ts1 geo:hasGeometry ?g1 .
            ?g1 geo:asWKT ?gwkt1 .
            ?ts1 time:hasTime ?t1 .
            ?ts2 ex:isTimesliceOf ?o2  .
            ?ts2 geo:hasGeometry ?g2 .
            ?g2 geo:asWKT ?gwkt2 .
            ?ts2 time:hasTime ?t2 .
            FILTER (?g1 != ?g2 && ?o1 = ?o2 &&
            temporal:meets(t1, t2) &&
            geof:sfWithin(?gwkt1, ?gwkt2))
            }
}
```

Example 5: Check-query for the separation relation

```
SELECT  ?ts1 ?ts2 ?ts3 WHERE {
        ?ts1 ex:hasContiSeparation ?ts2 .
        ?ts1 ex:hasDerivSeparation ?ts3 .

        MINUS{
            ?ts1 ex:isTimesliceOf ?o1  .
            ?ts1 geo:hasGeometry ?g1 .
            ?g1 geo:asWKT ?gwkt1 .
            ?ts1 time:hasTime ?t1 .
            ?ts2 ex:isTimesliceOf ?o2  .
            ?ts2 geo:hasGeometry ?g2 .
            ?g2 geo:asWKT ?gwkt2 .
            ?ts2 time:hasTime ?t2 .
            ?ts3 ex:isTimesliceOf ?o3  .
            ?ts3 geo:hasGeometry ?g3 .
            ?g3 geo:asWKT ?gwkt3 .
            ?ts3 time:hasTime ?t3 .
            FILTER
            !=geof:sfEquals(?gwkt1,?gwkt2)   &&
            !=geof:sfEquals(?gwkt1,?gwkt3)   &&
            !=geof:sfEquals(?gwkt2,?gwkt3)   &&
            ?o1 = ?o2 && ?o1 != ?o3 && ?o2 !=
            ?o3 && temporal:meets(t1, t2) &&
            temporal:meets(t1, t3) &&
            temporal:equal(t2, t3) &&
            geof:sfEquals(?gwkt1,geof:union(?gwk
            t2, ?gwkt3)))
        }
}
```

The approach is naive because several check-queries have some common constraints. To check all filiation relationships, we run each of the check-queries. Thus, some constraints may be verified several times during this process. An optimized approach can be considered grouping some constraints as in [**1**].

5.3 Inferring filiation relationship and knowledge

As we have seen, it is possible to check the consistency of the spatio-temporal ontology based on relations already defined. But sometimes, information may be incomplete and alter the semantic and understanding of the spatio-temporal phenomena modeled. In this section, on the basis of integrity constraints, we show how to infer filiation relation using SPARQL Update.

5.3.1 Inferring basic filiation relationships

The transformation from a SPARQL check-query to SPARQL Update inference-query is trivial, so we show only one example of inference (see).

Example 6: Inference-query for expansion relation

```
INSERT {?ts1 ex:hasExpansion ?ts2 }
WHERE {

        ?ts1 ex:isTimesliceOf ?o1  .
        ?ts1 geo:hasGeometry ?g1 .
        ?g1 geo:asWKT ?gwkt1 .
        ?ts1 time:hasTime ?t1 .
        ?ts2 ex:isTimesliceOf ?o2  .
        ?ts2 geo:hasGeometry ?g2 .
        ?g2 geo:asWKT ?gwkt2 .
        ?ts2 time:hasTime ?t2 .
        FILTER (?g1 != ?g2 && ?o1 = ?o2 &&
        temporal:meets(t1, t2) &&
        geof:sfWithin(?gwkt1, ?gwkt2))
}
```

All the filiation relationships discussed in the previous sections form the basis of spatio-temporal qualification of evolution. If these relations are standardized for all applications dealing with change on a geographical space, the semantic brought by these relations may be increased using contextual information defined in the ontology.

5.3.2 Increasing semantic of filiation relationships with contextual information

So far, filiation relationships of expansion, contraction, splits, separation, fusion and annexation were based on spatial, temporal and identity constraints. The Continuum Model uses ontology capabilities to define upper-level definitions based on filiation relationships. These upper-level relationships provide a better understanding of the evolution adding semantic constraints to previous filiation relationships. For example, the Dissolution of Czechoslovakia, which took effect on 1 January 1993, was an event that saw the self-determined division of the federal state of Czechoslovakia. The Czech Republic and Slovakia, entities which in 1969 within the framework of the Czechoslovak federalization, became immediate subjects of international law in 1993. Considering the representation of the phenomenon, we note that the state “Czechoslovakia” ceases to exist and its geometry splits into two new geometries, each corresponding to a new state: the “Czech Republic” and “Slovakia”. Finally, the union of the new geometry is equal to the former geometry. Thus, we notice that the phenomenon corresponds to a “split” filiation relationship. Consequently, it is possible to infer automatically that involving the “*hasSplits*” filiation relationships occurring between entities whose type is “State” is a “Dissolution” process.

6. EXAMPLE

For this example, we will use our model to study the urban evolution of the city of New Orleans, Louisiana. Figure 6 depicts the urban evolution of the entity "city of New Orleans" with timeslice, time interval and filiation relationships.

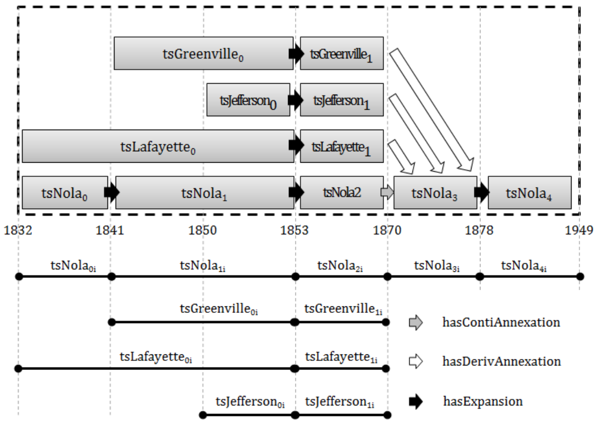

Figure 6: Time frame of urban evolution

Until 1870, Greenville, Jefferson, Lafayette and New Orleans cities grew, each having a *hasExpansion* relationship with its previous timeslice. In 1870, Greeville, Lafayette and Jefferson cities are absorbed by the city of New Orleans. This phenomenon is called a conurbation. We note that the parent entities merged, but one continues to exist meaning that the corresponding filiation relationship is *hasAnnexation*. Thus, we consider that an annexation occurring between cities is a conurbation. The Continuum model distinguishes the relation *hasContiAnnexation* meaning that the entity continues to exist after the process. The model also implements the relation *hasDerivAnnexation* implying that the entity ceases to exist after the annexation process. In this example, we will infer that a *hasContiAnnexation* relation between entities whose type is "city" leads to a *isConurbation* (Example 8: Inference-query for *hasConurbation* relation) upper-level filiation relationship. A *hasDerivAnnexation* relation between entities whose type is "city" leads to a *hasConurbation* (Example 8: Inference-query for *hasConurbation* relation) upper-level filiation relationships.

Example 7: Inference-query for *isConurbation* relation

```
INSERT {?ts1 ex:isConurbation ?ts2 }
WHERE {

        ?ts1 ex:hasContiAnnexation ?ts2  .
        ?ts1 ex:isTimesliceOf ?o1  .
        ?ts2 ex:isTimesliceOf ?o2  .
        ?o1 rdf:type ex:City .
        ?o2 rdf:type ex:City .
}
```

Example 8: Inference-query for *hasConurbation* relation

```
INSERT {?ts1 ex:hasConurbation ?ts2 }
WHERE {

        ?ts1 ex:hasDerivAnnexation ?ts2  .
        ?ts1 ex:isTimesliceOf ?o1  .
        ?ts2 ex:isTimesliceOf ?o2  .
        ?o1 rdf:type ex:City .
        ?o2 rdf:type ex:City .
}
```

7. CONCLUSION

The research presented in this paper introduces the Continuum Model whose goal is to track the identity of spatio-temporal entities through time and capture the semantics of the modeled phenomena. The model is ontology-based and provides a solution that combines GeoSPARQL and ontology of fluent.

Additionally, the model focuses on filiation relationships. On the basis of filiation relationships presented in [**1**], we defined a hierarchy of these relations. Basic filiation relationships are subjected to integrity constraints affecting spatiality, temporality and identity.

The model makes it possible to check consistency in the ontology with respect to the modeled phenomena. It can happen that data sets are partially incomplete. Using the integrity constraints model, it is possible to infer filiation relationships or qualify them more specifically with respect to the filiation hierarchy defined in the Continuum Model.

Semantic depicted by basic filiation relationships may be increased to improve understanding of the phenomenon modeled. Therefore, we proposed adding semantic constraints which are application dependent to obtain upper-level filiation relationships. An example of semantics constraints is introduced in order to infer some upper-level filiation relationship on the basis of general ones.

8. ACKNOWLEDGMENTS

This research is supported by: 1) Conseil regional de Bourgogne. 2) Direction Générale de l'Armement, see: http://www.defense.gouv.fr/dga/. We would like to thank the cooperation of Perrine Pittet who helped us in the formalization of the Continuum Model.

9. REFERENCES

[1] Del Mondo, G., Rodríguez, M. A. , Claramunt, C. , Bravo, L. , Thibaud, R. *Modeling consistency of spatio-temporal graphs*. Data & Knowledge Engineering, 2013.

[2] Del Mondo, G., Stell, J.G.,Claramunt, C., Thibaud, R. A Graph Model for Spatio-temporal Evolution. *Journal of Universal Computer Science* (2010), 1452-1477.

[3] Muller, P. Topological spatio-temporal reasoning and representation. *Computational Intelligence* (2002), 420-450.

[4] Stell, J.G., Del Mondo, G., Thibaud, R., Claramunt, C. *Spatio-temporal Evolution as Bigraph Dynamics*. Belfast, Maine, USA, 2011.

[5] Hornsby K., Egenhofer, M. Identity-based change: a foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science* (2000), 207-224.

[6] Strobl, C. Dimensionally Extended Nine-Intersection Model (DE-9IM). In *Encyclopedia of GIS Springer*. 2008.

[7] Randell, D. A., Cui, Z. and Cohn, A. G. A spatial logic based on regions and connection. In *Knowledge Representation and Reasoning* (San Mateo 1992), 165-176.

[8] Yuan, M. Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatiotemporal Queries. *Transaction in GIS* (1999), 137-159.

[9] Worboys, M. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science* (2005).

[10] Egenhofer, M. J. Toward the semantic geospatial web. In *In Proceedings of the 10th ACM international symposium on Advances in geographic information*

systems (2002), 1-4.

[11] Koubarakis, M., Kyzirakos, K. Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL. In *Proceedings of the Extended Semantic Web Conference* (2010), 425-439.

[12] Perry, M. *A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data*. 2008.

[13] Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y. Literature review of spatio-temporal database models. *The Knowledge Engineering Review* (2004), 235-274.

[14] Peuquet, D.J. *Making space for time: Issues in space-time data representation*. GeoInformatica, 2001.

[15] Hobbs, J.R., Pan, F. An ontology of time for the semantic web. In *ACM Transactions on Asian Language Information Processing* (2004), 66-85.

[16] O'Connor, M. J., Das, A. K. A Method for Representing and Querying Temporal Information in OWL. *Biomedical Engineering Systems and Technologies* (2011), 97-110.

[17] Welty, C., Fikes, R. A Reusable Ontology for Fluents in OWL. In *In Formal Ontology in Information Systems: Proceedings of the Fourth International Conference* (2006), 226-236.

[18] Frasincar, F., Milea, V., Kaymak, U. tOWL: Integrating Time in OWL. In *Semantic Web Information Management* (2009), 225-246.

[19] Milea, V.,Frasincar, F., Kaymak, U. The tOWL Temporal Web Ontology Language. In *The 20th Belgian-Netherlands Conference on Artificial Intelligence* (2008).

[20] Theodoulidis, C.I. and Loucopoulos, P. The Time Dimension in Conceptual Modelling. In *Information Systems*. 1991.

[21] Allen, J. F. Maintaining knowledge about temporal intervals. In *Communications of the ACM* (1983), 832-843.

[22] Nadi, S. Delavar, M.R. Spatio-Temporal Modeling of Dynamic Phenomena in GIS. In *ScanGIS* (2003), 215-225.

[23] Karmacharya, A., Cruz, C., Boochs, F., Marzani, F. Use of Geospatial Analyses for Semantic Reasoning. *Knowledge-Based and Intelligent Information and Engineering Systems*, 6276 (2010), 576-586.

[24] Battle, R., Kolas, D. Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web* (2012), 355-370.

[25] Batsakis, S., Petrakis, E. G. M. SOWL: A Framework for Handling Spatio-temporal Information in OWL 2.0. *Rule-Based Reasoning, Programming, and Applications*, 6826 (2011), 242-249.

[26] Stumme, G., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Sure, Y., Volz, R. & Zacharias, V. *The Karlsruhe view on ontologies*. University of Karlsruhe, Institute AIFB., 2003.

[27] Artale, A., Franconi, E. Temporal Description Logics. In Dov Gabbay, Michael Fisher and Lluis Vila, ed., *Handbook of time and temporal Reasoning in Artifical intelligence*. MIT Press, 2000.

[28] Baader, F., Nutt, W. Basic description logics. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, ed., *In The description logic handbook*. Cambridge University Press, New York, NY, USA, 2003.