Using Semantic Web Technologies to Follow the Evolution of Entities in Time and Space

Benjamin Harbelot*, Helbert Arenas†, and Christophe Cruz‡
Laboratoire Le2i, UMR-6302 CNRS,Departement Informatique
University of Burgundy
Dijon, France
*benjamin.harbelot@checksem.fr, †helbert.arenas@checksem.fr, ‡christophe.cruz@u-bourgogne.fr

***Abstract*—In this paper we present the "continuum model". Our work follows a "perdurantism" approach and is designed to handle dynamic phenomena extending the 4D-fluent with the use of semantic web technologies. In our approach we represent dynamic entities as constituted by timeslices each with semantic, geometric, temporal and identity components. Our model is able to link the diverse representations of an entity and allows the inference of qualitative information from quantitative one. The inference results are later added to the ontology in order to improve knowledge about the phenomenon. The model has been implemented using OWL and SWRL. Our preliminary results are promising and we plan to further develop the model in the near future to increase the suitable data sources.**

Keywords-spatio-temporal; semantics; GIS; perdurantism.

I. INTRODUCTION

For the design of a spatio-temporal knowledge system, it is necessary to consider the three components of an entity representation: 1) Spatial: consisting in the geometry, 2) Temporal: which defines the interval of existence of the geometries and finally 3) Semantic: which defines a meaning for the entity beyond the purely geographic one [1] [2]. Most of the current GIS tools focus on analysis and presentation of geographic data. However, nowadays due to the increasing availability of spatial/temporal data, it is necessary to have tools with inference capabilities, capable of assisting researchers in analysing large datasets. This new kind of tools should be able to identify patterns and perform reasoning with datasets corresponding to dynamic phenomena.

Modeling a real dynamic phenomenon can be seen as tracking the transition of phenomenon composing entities from one state to another. This transition is called: filiation relationship. Along time, entities with spatial components, can maintain different spatial and semantic relations with other entities. A natural way to model dynamic phenomena is to represent the evolution as a graph, in which entities and their states are represented as vertices and relations between entities as edges. A phenomenon would then generate a complex graph composed by different types of relations such as: temporal, semantic, spatial or filiation.

An alternative to classic GIS tools are Semantic Web technologies. Using these technologies it is possible to develop data models called ontologies specifically designed for reasoning and inference with software mechanisms. Ontologies allow for any given domain, the representation of relevant high level concepts as well as their properties and the relationships between concepts and entities. In this research we use Semantic Web technologies to develop the "continuum model", an ontology that allows us to represent diverse dynamic entities and analyse their relationships along time. Traditionally ontologies are static in the sense that the information represented in them does not change in time or space. In this paper we introduce the continuum model, an ontology that extends the 4D-fluent. Our ontology provides the mechanisms required to keep track of spatial and semantic evolution of entities along time.

In Section II, we discuss related work in the field of spatio-temporal knowledge representation. In Section III, we introduce the continuum model. In Section IV, we present the model specification using description logics. In Section V, we show some examples of GeoSPARQL used to implement the model. In Section VI, we describe how the model operates using an urban growth example and later we indicate our conclusions and future work.

II. RELATED WORK

The development of a spatial-temporal knowledge system involves two aspects, first the representation of the knowledge and second, the necessary mechanisms to perform analysis and querying.

A. Representing temporal data

The two main philosophical theories concerning the representation of object persistence over time are: *endurantism* and *perdurantism*. The first one, *endurantism*, considers objects as three dimensional entities that exist wholly at any given point of their life. On the other hand, *perdurantism*, also known as the four dimensional view, considers that entities have temporal parts, "timeslices" [3]. From a perdurantism point of view the temporal dimension of an entity is composed by all its timeslices. Therefore, it represents the different properties of an entity over time as *fluent*. A *fluent* is a property valid only during certain intervals or moments in time. From a designer point of view, the *perdurantism* approach offers advantages over the *endurantism* one, allowing richer representations of real world phenomena [4].

The implementation of a *perdurantism* approach within an ontology, requires the conversion of static properties into dynamic ones. The two primary Semantic Web languages are OWL and RDF, unfortunately both of them provide limited support for temporal dynamics [5]. The OWL-Time ontology describes the temporal content of web pages and temporal properties of web services. Moreover, this ontology provides good support for expressing topological relationships between times or time intervals, as well as times or dates [6]. However, OWL allows only binary relations between individuals. In order to overcome this limitation several methodologies have been proposed for the representation of dynamic objects and their properties. Among the most well known are: temporal RDF, versioning, reification, N-ary relationships and the 4D-fluent approach.

Temporal RDF [7] proposes an extension of the standard RDF for naming properties with the corresponding time interval. This allows an explicit management of time in RDF. However, temporal RDF uses only RDF triples; therefore, it does not have all the expressiveness of OWL for instance, it is not possible to employ qualitative relations. Reification is a technique used to represent n-ary relations, extending languages such as OWL that allow only binary relations [8]. In [5], the authors developed a lightweight model using Reification. The model is designed to be deployed on top of existing OWL ontologies extending their temporal capabilities. The model also implements a set of SWRL (Semantic Web Rule Language) operators to query the ontology. Reification allows the use of a triple as object or subject of a property. But this method has also its limitations, for instance the transformation from a static property into a dynamic one increases substantially the complexity of the ontology, reducing the querying and inference capabilities. Additionally reification is prone to redundant objects which reduces its effectiveness. Versioning is described as the ability to handle changes in ontologies by creating and managing multiple variants of them [9]. However, the major drawback of Versioning, is the redundancy generated by the slightest change of an attribute. In addition, any information requests must be performed on multiple versions of the ontology affecting its performance.

An alternative to the previously mentioned approaches is the 4D-fluent, which is an approach based on the *perdurantism* philosophical theory. It considers that the existence of an entity can be expressed with multiple representations, each corresponding to a defined time interval. In the literature, 4D-fluent is the most well known method to handle dynamic properties in an ontology. It has a simple structure allowing to easily transform a static ontology into a dynamic one [10]. Unfortunately, the 4D fluent approach has also some limitations; although it allows the recording of frequent timeslices, it can not handle explicit semantics. This fact causes two problems: 1) It is difficult to maintain a close relationship between geometry and semantics; and 2) It increases the complexity for querying the temporal dynamics and understanding the modelled knowledge. Furthermore, this approach does not define qualitative relations to describe the type of change that has occurred or to describe the temporal relationships between objects. Then we can not know which entities have undergone a change and what entities might be the result of that change. Regardless of its limitations the 4D-fluent approach offers a solid starting point for the representation of temporal information in OWL. An interesting previous work using this approach is [11]. Here the authors developed SOWL, which uses 4D-fluent to extend the ontology OWL-time making it able to handle qualitative relations between intervals, such as "before" or "after" even with intervals with vague ending points.

B. Querying the ontology

In [12], the authors introduce a model in which spatial-temporal information contained in a database and a spatial-temporal inference system work together. However, no information is given on the Semantic Web technologies, only the Java language is quoted as a component of the inference engine; therefore, the universality and effectiveness of the inference system can be questioned. Another work is [13] in which the authors propose a reasoning system that combines the topological calculus capabilities of a GIS and the inference capabilities of the semantic web field. However, the notion of time is not incorporated in this model.

The capability of switching from quantitative to qualitative data is only possible with a reasoning system. In the case of SOWL this is possible thanks to the implementation of SWRL built-in. In SOWL, the built-ins allow the system to infer topological, directional and metric relations between entities. Qualitative information can be inferred from quantitative one and can be used as an alternative in case of missing quantitative data. In order to query the ontology the developers of SOWL implemented a language similar in syntax to SQL. This language performs simple spatial-temporal querying for both static and dynamic data [11]. However, the work does not support identity relationships. It does not provide mechanisms to follow the changes an entity might experience by analysing its different representations.

Due to the nature of spatial-temporal datasets, we need a system able to handle large datasets. Traditionally, SPARQL has been the most common language to query an ontology. SPARQL is a W3C recommendation that operates at the level of RDF graphs. There are extensions, such as st-SPARQL and geoSPARQL that have been developed in order to allow SPARQL to operate on spatial entities [14].

These extensions define datatypes, functions and operations allowing spatial analysis, however, there is limited temporal support. St-SPARQL is based on an extension of RDF called st-RDF that integrates contact geometries and incorporates time in RDF. GeoSPARQL offers similar capabilities, however, it has the advantage of being an OGC supported standard.

Our previous work [1] introduced the *Continuum* model using Java and SWRL rules to implement it. The rules were executed via a graphical interface using the Jena API to connect to the ontology and JDBC to access a database. The application automatically detects the presence of spatial built-ins in SWRL rules and performs the necessary calculations in the database. The system can automatically rewrite SWRL rules containing spatial built-ins. On one hand, this prevents repeating calculations that have already been done. On the other hand, it also generates SWRL rules without spatial built-ins but rather based on qualitative relationships expressed through properties defined in the ontology. However, we note three limitations to this model: 1) The treatment of a query containing a spatial built-ins can be very long depending on the number of geometries involved in spatial analysis, 2) The execution of SWRL rules containing spatial built-ins currently depends on our application and cannot be executed from other sources, for instance the traditional plugin SQWRL Tab Query of the Protégé tool, 3) There are limitations in the size of the datasets that can be managed by the application.

Although SWRL is a potent inference tool, it is not fully supported in current available triplestores. A triplestore is a software mechanism able to store large datasets with semantic annotations, providing query and retrieval capabilities. Some of the available triplestores support GeoSPARQL allowing users to perform complex spatial queries [15] [16].

In order to overcome this limitations identified in [1], we decided to modify the system architecture and implement a new version using a triplestore with spatial capabilities for spatial calculations and data storage. After evaluating the available options we opted for Parliament. In this paper we present a further development of the model first presented in [1], using in this case GeoSPARQL/SPARQL. In the next section we will describe how we implemented this approach in the continuum model.

III. The Continuum Model

The spatial evolution of an object involves movement or a change of shape [17]. In the case of a movement, it is easy to identify and locate the entity before and after the event. However, when an entity suffers a succession of changes a key question arises: how much can it change before its identity is modified? And if there is a semantic change, then how do we know that this is the same entity at different times?

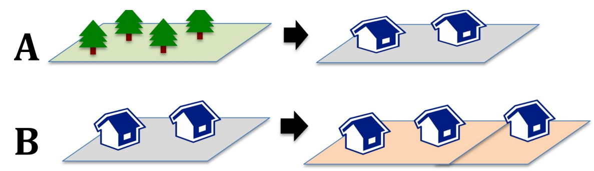

Figure 1. Evolution examples: A)Two different semantic objects for the same geometry. B)Two related geometries for the same semantic object.

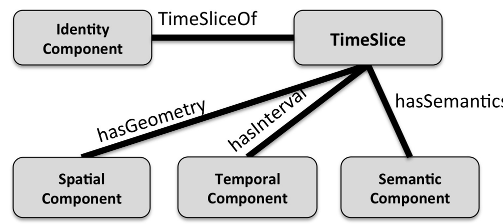

Figure 2. The four components of a timeslice within the continuum model.

The 4D-fluent approach does not allow an entity to change its nature, only allows the change of the value of some of its properties. However, the semantics associated with a geometry may change. For example, a land parcel may change from being forest into being urban. In this example the geometry has not changed, however, a semantic change has occurred (see Figure 1A). It is equally possible that the semantics might not change while the geometry evolves. For instance, a given urban land parcel might expand by purchasing neighbouring parcels (see Figure 1B).

In order to represent a dynamic entity in the continuum model we create a set of timeslices, each corresponding to a representation of the entity during a determined period of time. Each timeslice is constituted by four components as depicted in Figure 2: 1) Semantic: To describe the knowledge associated with the entity, 2) Spatial: It is the graphical representation, 3) Temporal: It represents the interval or time instant that describe the temporal existence, and 4) The identity component, that allows us to group timeslices belonging to the same entity.

The goal of the continuum model is to follow the evolution of entities through time. To achieve this goal the model records the changes that entities might go through in their semantic or spatial components along time. For this purpose the model creates a new representation every time a change occurs (spatial, semantic or identity).There is a parent-child relationship between the resulting timeslices. A resulting child timeslice retains all the unchanged characteristics from the original parent timeslice. Figure 3 depicts how we can represent the evolution of an object in which only the semantic part has changed. Figure 4 depicts the evolution of an object in which the spatial component varies, while the rest of the components remain constant. Each change adds to the genealogy of the spatio-temporal components. The parent-child relation is recorded in the system, allowing

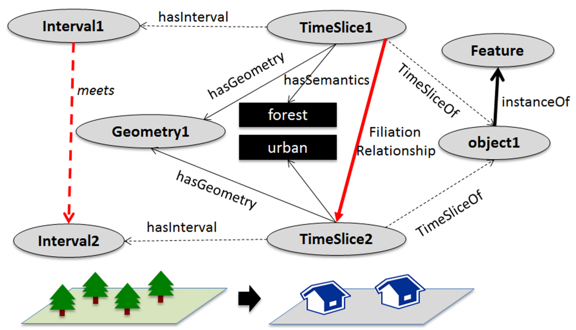

Figure 3. Evolution example: A semantic change with the same geometry.

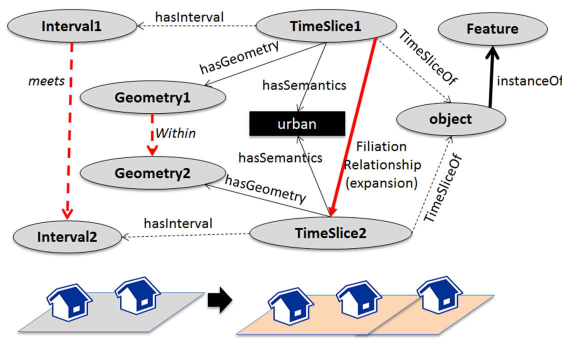

Figure 4. Evolution example: A spatial change with the same semantic

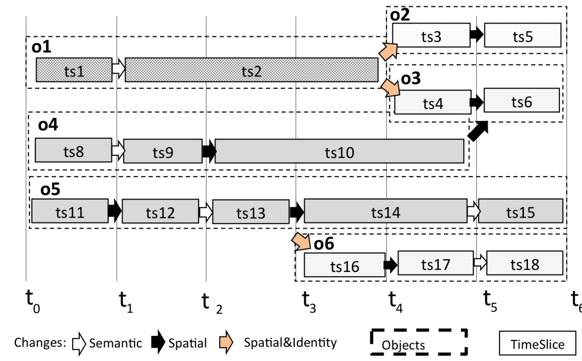

Figure 5. Using the continuum model to represent the evolution of an entity.

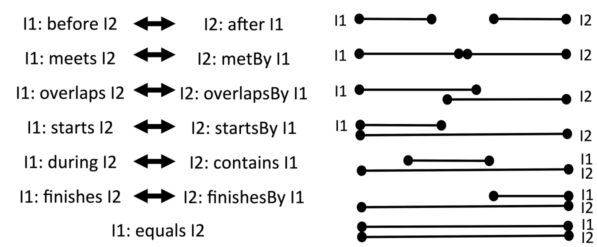

Figure 6. Allen temporal relations.

the analysis and querying of the information. The model enforces a coherency between the time intervals of timeslices contained in the system. By using this representation we are able to establish relationships between the components of two different timeslices.

Figure 5 depicts an example of objects geneology. In this example we have the objects $o_1, o_2, \ldots o_6$. Each of the object evolve along time. A set of timeslices compose the temporal representation of each object, thus $o1 : [ts_1, ts_2]$, $o_2 : [ts_3, ts_5]$, $o_3 : [ts_4, ts_6]$, $o_4 : [ts_8, ts_9, ts_{10}]$,$o_5 : [ts_{11}, ts_{12}, ts_{13}, ts_{14}, ts_{15}]$ and $o_6 : [ts_{16}, ts_{17}, ts_{18}]$. The system enforces temporal coherency, children objects can not occur before the parents.

The continuum groups related timeslices, which have a valid time interval of existence. The model links individual timeslices to their context. For instance, a timeslice can have a child that corresponds to a new object, then the identity component might be different between parent and child. Our system allows the definition of qualitative relations between timeslices, even when the timeslices belong to different objects. Figure 5 depicts the evolution of objects and how the continuum model is used to study them.

In our model we have implemented qualitative temporal relations based on binary and mutually exclusive relations as proposed by Allen [18] (see Figure 6). The addition of Allen relations increase the expressive power of the system by adding qualitative information in addition to the quantitative one. By using defined Allen relations between intervals we can obtain qualitative information even from intervals with vague endpoints in a similar fashion to [10]. For example, Figure 7 depicts intervals "I1", "I2" and "I3". While we know the start and ending points of "I1", we do not know the ending point of "I2", and we do not know the starting point of "I3'. However, we know that "I1" meets "I2" and that "I2" contains "I3". Then we can infer that because "I2" contains "I3",then "I3" must be after "I1", even if the information about start and ending points is incomplete. Lack of knowledge caused by semi closed intervals is largely filled by the integration of Allen relations to the model (see Figure 7).

In GIS, objects or regions are represented by points, lines, polygons or other more complex geometries based on these primitives. All these geometries are defined using the coordinates of points which are quantitative information. There are mainly three types of relationships between geometries: directional, metric, and topological relationships. The topological analysis between two objects is done using the models: Dimensionally Extended Nine-Intersection Model (DE-9IM) or Region Connection Calculus (RCC8)

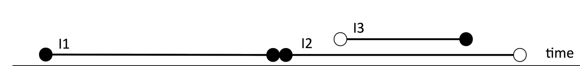

Figure 7. Using Allen temporal relations to infer new knowledge.

Table I
TOPOLOGICAL PREDICATES AND THEIR CORRESPONDING MEANINGS.

Topological	Predicate Meaning
Equals	The Geometries are topologically equal.
Disjoint	The Geometries have no point in common.
Intersects	The Geometries have at least one point in common (the inverse of Disjoint).
Touches	The Geometries have at least one boundary point in common, but no interior points.
Crosses	The Geometries share some but not all interior points, and the dimension of the intersection is less than that of at least one of the Geometries.
Overlaps	The Geometries share some but not all points in common, and the intersection has the same dimension as the Geometries themselves.
Within	Geometry A lies in the interior of Geometry B
Contains	Geometry B lies in the interior of Geometry A (the inverse of Within)

[19]. In both cases, we obtain an equivalent set of topological relationships for specific regions. To calculate the spatial relationships between two geometries the DE-9IM model takes into account the inside, the outside, and the contour of the geometries leading to the analysis of nine intersections as described in [19]. There are eight possible spatial relationships of the resulting analysis-9IM (see Table I).

IV. MODEL SPECIFICATION

The relationships based on quantitative information can be translated later into qualitative data [17]. By analysing the relationships between temporal, spatial and identity components of timeslices it is possible to deduce qualitative topological relationships between them. The results of the analysis can be used to specify more semantically complex constructions. In this section we use Tarski-style formalisms to specify the components of our model.

A. Temporal components

To represent time intervals we follow the semantics suggested by Artale and Franconi (1998). We can think of the temporal domain as a linear structure $\mathcal{T}$ composed by a set of temporal points $\mathcal{P}$. The components of $\mathcal{P}$ follow a strict order $<$, which forces all points between two temporal points t_1 and t_2 to be ordered. By selecting a pair $[t_1, t_2]$ we can limit a closed interval of ordered points. The set of interval structures in $\mathcal{T}$ is represented by $\mathcal{T}^{\star}_{<}$ [20].

Temporal Points ($\mathcal{P}$):

$$\mathcal{P}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \tag{1}$$

Time Intervals ($\mathcal{T}^{\star}_{<}$):

$$[to, tf] \doteq \{x \in \mathcal{P} | to \leq x \leq tf, to \neq tf\} in \mathcal{T} \tag{2}$$

where to and tf are the initial and ending points of the interval respectively.

To define the relations identified by Allen [18] (see Figure 6) we first define two intervals $i1$ and $i2$: $\mathcal{T}^{\star}_{<}(i1)$, $\mathcal{T}^{\star}_{<}(i2)$, being i_{to} the starting point and i_{tf} the ending point of the intervals:

$$Before(i1, i2) \rightarrow (i1_{tf} < i2_{to}) \tag{3}$$
$$Meets(i1, i2) \rightarrow (i1_{tf} = i2_{to}) \tag{4}$$
$$Overlaps(i1, i2) \rightarrow (i1_{tf} > i2_{to}) \wedge (i1_{tf} < i2_{tf}) \tag{5}$$
$$Starts(i1, i2) \rightarrow (i1_{to} = i2_{to}) \wedge (i1_{tf} < i2_{tf}) \tag{6}$$
$$During(i1, i2) \rightarrow (i1_{to} > i2_{to}) \wedge (i1_{tf} < i2_{tf}) \tag{7}$$
$$Finishes(i1, i2) \rightarrow (i1_{to} > i2_{to}) \wedge (i1_{tf} = i2_{tf}) \tag{8}$$
$$Equals(i1, i2) \rightarrow (i1_{to} = i2_{to}) \wedge (i1_{tf} = i2_{tf}) \tag{9}$$

B. Spatial Components

The spatial representation of an object is given by the coordinates representing its geometry and characteristics associated to it (Spatial reference system, accuracy, precision, format, etc). It is represented by $\mathcal{G}$. The spatial topological relations between geometries are defined by the Extended Nine-Intersection model (DE-9IM) (see Table I [19]).

Additionally, we can implement the operation *Union* valid for geometries:

$$[x_g, y_g, z_g] \in \mathcal{G} | Equals(z_g, Union(x_g, y_g)) \tag{10}$$

In this case, the combination of geometries x_g and y_g will result in a new geometry z_g.

C. Semantic Component

The semantic component of the objects describes the nature of the entities and can be composed by one or more alphanumeric properties.

D. Timeslices

An object representation in time is composed by a set of timeslices. Each timeslice $\mathcal{TS}$ in the model has four components: 1) A time interval $\mathcal{T}^{\star}_{<}$ 2) A geometry $\mathcal{G}$, 3) An identity $\mathcal{O}$ and 4) A semantic component representing all other potential alphanumeric properties associated to a timeslice. We represent all these properties as $\overline{\mathcal{TS}}$, as suggested in [21]. $\overline{TS}$ represents all the qualities that distinguish the class timeslice from other classes,.

$$\begin{aligned} &\mathcal{TS} \equiv \forall hasGeometry.\mathcal{G} \sqcap \forall hasTime.\mathcal{T}^{\star}_{<} \\ &\sqcap \overline{\mathcal{TS}} \sqcap \forall isTimeSliceOf.\mathcal{O} \end{aligned} \tag{11}$$

E. Filiation relationships between timeslices

In the continuum model the existence of an object is defined by a set of timeslices representing the state of the object during a defined period of time. In the model, a new timeslice is generated when a original timeslice suffers a change in any of its components. The relation between the original and the new timeslice follows a *parent - child*, filiation pattern. For this relation to exist the interval of the *parent* timeslice must *meets* the interval of the *child* timeslice (see Figure 6). In order to exist the filiation *parent-child* relationship between timeslices at least one of the components (geometry, semantics or identity) must remain constant. The filiation relationship is specified as:

$$\begin{array}{l}\forall hasFiliation.\mathcal{TS}\\ \{ts_1 \in \mathcal{TS}^I | \forall ts_2.(ts_1, ts_2) \in hasFiliation^I\\ \rightarrow ts_2 \in \mathcal{TS}^I \wedge \exists_{\leq 2}((ts_{1g} \neq ts_{2g}) \vee (ts_{1s} \neq ts_{2s}) \vee\\ (ts_{1o} \neq ts_{2o})) \wedge (meets(ts_{1i}, ts_{2i}))\}\end{array} \tag{12}$$

where: $\{ts_1, ts_2\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}\} \in \mathcal{G}$, $\{ts_{1s}, ts_{2s}\} \in \overline{\mathcal{TS}}$ and $\{ts_{1i}, ts_{2i}\} \in \mathcal{I}$

The filiation relationship can be further specialized by setting or not constraints in the identity ($\mathcal{O}$) component, then we have two possible filiation relationships: *hasContinuation* and *hasDerivation* [22] [23].

1) Continuation relationship: In this case, the identity component of parent and child timeslices is the same.

$$\begin{array}{l}\forall hasContinuation.\mathcal{TS}\\ \{ts_1 \in \mathcal{TS}^I | \forall ts_2.(ts_1, ts_2) \in hasContinuation^I\\ \rightarrow ts_2 \in \mathcal{TS}^I \wedge ((ts_{1g} \neq ts_{2g}) \vee (ts_{1s} \neq ts_{2s})) \wedge\\ (ts_{1o} = ts_{2o}) \wedge (meets(ts_{1i}, ts_{2i}))\}\end{array} \tag{13}$$

where: $\{ts_1, ts_2\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}\} \in \mathcal{G}$, $\{ts_{1s}, ts_{2s}\} \in \overline{\mathcal{TS}}$ and $\{ts_{1i}, ts_{2i}\} \in \mathcal{I}$

2) Derivation relationship: In this case, there is a difference between the identity component of parent and child, while there is at least one component (geometry or semantic) that remains constant.

$$\begin{array}{l}\forall hasDerivation.\mathcal{TS}\\ \{ts_1 \in \mathcal{TS}^I | \forall ts_2.(ts_1, ts_2) \in hasDerivation^I\\ \rightarrow ts_2 \in \mathcal{TS}^I \wedge \exists((ts_{1g} = ts_{2g}) \vee (ts_{1s} = ts_{2s})) \wedge\\ (ts_{1o} \neq ts_{2o}) \wedge (meets(ts_{1i}, ts_{2i}))\}\end{array} \tag{14}$$

where: $\{ts_1, ts_2\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}\} \in \mathcal{G}$, $\{ts_{1s}, ts_{2s}\} \in \overline{\mathcal{TS}}$ and $\{ts_{1i}, ts_{2i}\} \in \mathcal{I}$

F. Topological filiation relationships

By identifying the topological relationships between the geometric component of the timeslices we can define specific filiation relationships in which the spatial components evolve (see Figure 8).

1) Expansion: In this case, the geometric component of the child timeslice contains the geometry of the parent timeslice. There is no change in the identity component of the timeslice, both parent and child timeslices belong to the same object (see Figure 8).

$$\begin{array}{l}\forall hasExpansion.\mathcal{TS}\\ \{ts_1 \in \mathcal{TS}^I | \forall ts_2.(ts_1, ts_2) \in hasExpansion^I\\ \rightarrow ts_2 \in \mathcal{TS}^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1o} = ts_{2o}) \wedge\\ meets(ts_{1i}, ts_{2i}) \wedge hasWithin((ts_{1g}, ts_{2g})\}\end{array} \tag{15}$$

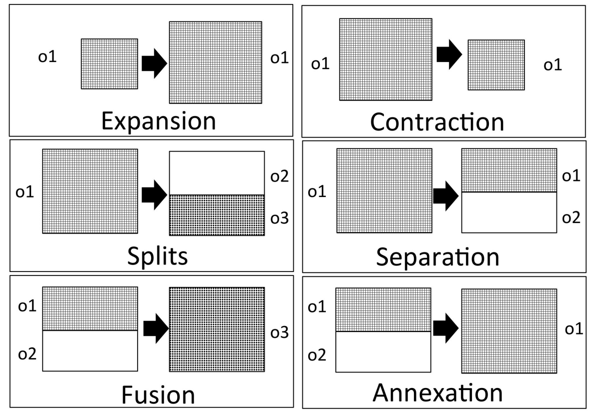

Figure 8. Topological filiation relationships.

where: $\{ts_1, ts_2\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}\} \in \mathcal{I}$

2) Contraction: This process is the opposite to *expansion* (see Figure 8).

$$\begin{array}{l}\forall hasContraction.\mathcal{TS}\\ \{ts_1 \in \mathcal{TS}^I | \forall ts_2.(ts_1, ts_2) \in hasContraction^I\\ \rightarrow ts_2 \in \mathcal{TS}^I \wedge (ts_{1g} \neq ts_{2g}) \wedge (ts_{1o} = ts_{2o}) \wedge\\ meets(ts_{1i}, ts_{2i}) \wedge hasContains((ts_{1g}, ts_{2g})\}\end{array} \tag{16}$$

where: $\{ts_1, ts_2\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}\} \in \mathcal{I}$

3) Splits: In this relationship, the object identified as the parent timeslice identity (O) ceases to exist. The geometry of the parent timeslice is then the origin of two new geometries corresponding to timeslices whose identity is new. The union of the geometries of the resulting children timeslices is equal to the geometry of the parent timeslice (see Figure 8).

$$\begin{array}{l}\forall hasSplits.\mathcal{TS}\\ \{ts_1 \in \mathcal{TS}^I | \forall (ts_2, ts_3).(ts_1, (ts_2, ts_3)) \in hasSplits^I\\ \rightarrow (ts_2, ts_3) \in \mathcal{TS}^I \wedge\\ (ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge\\ (ts_{1o} \neq ts_{2o}) \wedge (ts_{1o} \neq ts_{3o}) \wedge\\ meets(ts_{1i}, ts_{2i}) \wedge meets(ts_{1i}, ts_{3i}) \wedge\\ equals(ts_{1g}, Union(ts_{2g}, ts_{3g}))\}\end{array} \tag{17}$$

where: $\{ts_1, ts_2, ts_3\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}, ts_{3g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}, ts_{3i}\} \in \mathcal{I}$

4) Separation: 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* at least one of the children timeslices must have the same entity as the parent timeslice (see Figure 8).

$$\begin{aligned}&\forall hasSeparation.\mathcal{TS}\\&\{ts_1 \in \mathcal{TS}^I | \forall (ts_2, ts_3).(ts_1, (ts_2, ts_3) \in hasSeparation^I\\&\rightarrow (ts_2, ts_3) \in \mathcal{TS}^I \wedge\\&(ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge\\&\exists_{=1}((ts_{1o} = ts_{2o}) \vee (ts_{1o} = ts_{3o})) \wedge\\&meets(ts_{1i}, ts_{2i}) \wedge meets(ts_{1i}, ts_{3i}) \wedge\\&equals(ts_{1g}, Union(ts_{2g}, ts_{3g}))\}\end{aligned} \tag{18}$$

where: $\{ts_1, ts_2, ts_3\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}, ts_{3g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}, ts_{3i}\} \in \mathcal{I}$

5) Fusion: 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.

$$\begin{aligned}&\forall hasFusion.\mathcal{TS}\\&\{ts_1 \in \mathcal{TS}^I | \forall (ts_2, ts_3).(ts_1, (ts_2, ts_3) \in hasFusion^I\\&\rightarrow (ts_2, ts_3) \in \mathcal{TS}^I \wedge\\&(ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge\\&(ts_{1o} \neq ts_{2o}) \wedge (ts_{1o} \neq ts_{3o}) \wedge (ts_{2o} \neq ts_{3o}) \wedge\\&meets(ts_{1i}, ts_{3i}) \wedge meets(ts_{2i}, ts_{3i}) \wedge\\&equals(Union(ts_{1g}, ts_{2g}), ts_{3g})\}\end{aligned} \tag{19}$$

where: $\{ts_1, ts_2, ts_3\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}, ts_{3g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}, ts_{3i}\} \in \mathcal{I}$

6) Annexation: In this case, the two parent entities merge but the resulting entity keeps the identity of one of its parents.

$$\begin{aligned}&\forall hasAnnexation.\mathcal{TS}\\&\{ts_1 \in \mathcal{TS}^I | \forall (ts_2, ts_3).(ts_1, (ts_2, ts_3) \in hasAnnexation^I\\&\rightarrow (ts_2, ts_3) \in \mathcal{TS}^I \wedge\\&(ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge\\&((ts_{1o} = ts_{3o}) \vee (ts_{2o} = ts_{3o})) \wedge (ts_{1o} \neq ts_{2o}) \wedge\\&meets(ts_{1i}, ts_{3i}) \wedge meets(ts_{2i}, ts_{3i}) \wedge\\&equals(Union(ts_{1g}, ts_{2g}), ts_{3g})\}\end{aligned} \tag{20}$$

where: $\{ts_1, ts_2, ts_3\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}, ts_{3g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}, ts_{3i}\} \in \mathcal{I}$

V. Implementation

This is an evolving work, continuously we are adding new capabilities to the continuum model. In our latest implementation we have deployed our ontology in a Parliament triplestore. In order to populate our ontology we have developed customized tools able to read information stored in shapefiles, GML, WFS and postgreSQL/postGIS data repositories and upload it into our triplestore. The harvesting tools have been developed using Java with Jena and Geotools libraries.

In this section we show how we can identify some of the filiation relationships between timeslices using GeoSPARQL.

Continuation:

```
SELECT
?ts1 ?ts2
WHERE{
?ts1 a abc:TimeSlice.
?ts2 a abc:TimeSlice.
?o1 a abc:Object.
?ts1 abc:isTimeSliceOf ?o1.
?ts2 abc:isTimeSliceOf ?o1.
?ts1 abc:hasInterval ?i1.
?ts2 abc:hasInterval ?i2.
?ts1 geo:hasGeometry ?geo1.
?ts2 geo:hasGeometry ?geo2.
?geo1 geo:asWKT ?geo1wkt.
?geo2 geo:asWKT ?geo2wkt.
?ts1 abc:hasSemantic ?s1.
?ts2 abc:hasSemantic ?s2.
FILTER (
((!geof:sfEquals(?geo1wkt,?geo2wkt)) &&
(?s1=?s2))) ||
((geof:sfEquals(?geo1wkt,?geo2wkt)) &&
(?s1!=?s2))) &&
 (temporal:meets(i1,i2)) )
}
```

Derivation:

```
SELECT
?ts1 ?ts2
WHERE{
?ts1 a abc:TimeSlice.
?ts2 a abc:TimeSlice.
?o1 a abc:Object.
?o2 a abc:Object.
?ts1 abc:isTimeSliceOf ?o1.
?ts2 abc:isTimeSliceOf ?o2.
?ts1 abc:hasInterval ?i1.
?ts2 abc:hasInterval ?i2.
?ts1 geo:hasGeometry ?geo1.
?ts2 geo:hasGeometry ?geo2.
?geo1 geo:asWKT ?geo1wkt.
?geo2 geo:asWKT ?geo2wkt.
?ts1 abc:hasSemantic ?s1.
?ts2 abc:hasSemantic ?s2.
FILTER (
((!geof:sfEquals(?geo1wkt,?geo2wkt)) &&
(?s1=?s2))) ||
((geof:sfEquals(?geo1wkt,?geo2wkt)) &&
(?s1!=?s2))) &&
 (temporal:meets(i1,i2)) &&
(?o1!=?o2) )
```

```
}
```

Topological filiation relationships: Splits:

```
SELECT
?ts1 ?ts2 ?ts3
WHERE{
?ts1 a abc:TimeSlice.
?ts2 a abc:TimeSlice.
?ts3 a abc:TimeSlice.
?o1 a abc:Object.
?o2 a abc:Object.
?o3 a abc:Object.
?ts1 abc:isTimeSliceOf ?o1.
?ts2 abc:isTimeSliceOf ?o2.
?ts3 abc:isTimeSliceOf ?o3.
?ts1 abc:hasInterval ?i1.
?ts2 abc:hasInterval ?i2.
?ts3 abc:hasInterval ?i3.
?ts1 geo:hasGeometry ?geo1.
?ts2 geo:hasGeometry ?geo2.
?ts3 geo:hasGeometry ?geo3.
?geo1 geo:asWKT ?geo1wkt.
?geo2 geo:asWKT ?geo2wkt.
?geo3 geo:asWKT ?geo3wkt.
FILTER (
(?s1!=?s2)&&
(?o1!=?o3) &&
(?o2!=?o3) &&
(temporal:meets(i1,i2)) &&
 (temporal:meets(i1,i3)) &&
 (!geof:sfEquals
(?geo1wkt,geof:union(?geo2wkt,?geo3wkt))))
}
```

VI. Example Continuum

The continuum model is flexible enough to be adapted in multiple fields. In this example, we use it to represent the urban evolution of the city of New Orleans. This city is the largest in Louisiana. It is located between the Mississippi river and the lake Pontchatrain. The oldest part of the city is placed on the banks of the Mississippi river, on the natural levees of the river. Since its beginning it was the main settlement in the area, however, New Orleans was not the only one. In the vicinity, other cities such as Jefferson, Lafayette or Greenville were established. These cities were absorbed into New Orleans in the XIXth. century. From 1718 to 1900 the urban growth was only into areas that were by nature *high and dry*. However, after 1900, technical developments allowed municipal authorities to drain the swamps located between the river and the lake Pontchatrain, creating artificially dry land for urban development. The final result is a city that occupies an area that resembles a bowl, with large neighborhoods lying on the bottom, in areas with low elevation, some of them even below sea level, which make them vulnerable to floods. If we add to this, the subsiding soil phenomenon occurring in the area, and the erosion of the coast line, we end up with a city in a particularly vulnerable location [24]. Figure 9 depicts different stages of the urban evolution of New Orleans. Figure 9D depicts the extension of the city around 2005, when it was flooded by Hurricane Katrina.

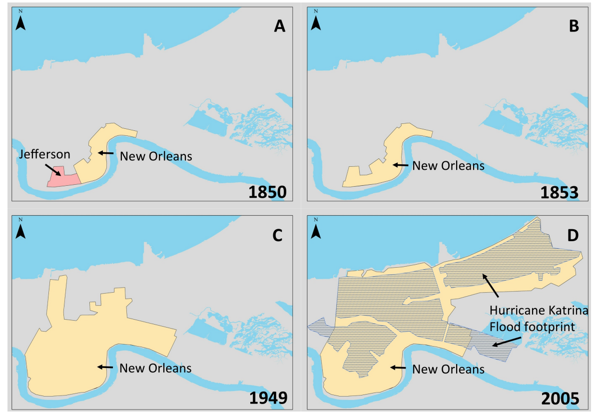

Figure 9. City of New Orleans along time.

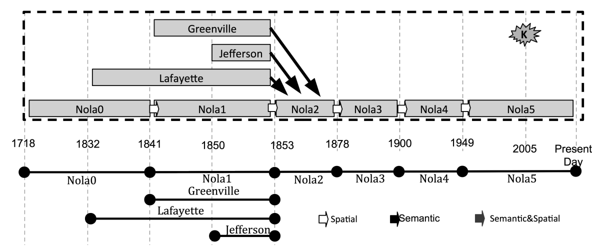

Figure 10. Time frame or urban evolution

In order to use the continuum model to represent the urban evolution, first we define the class *Human Settlement* ($\mathcal{HS}$) which represents cities that evolve through time. The temporal existence of each of the entities belonging to this class is represented by a set of timeslices which have the four components: 1) Semantic: representing properties associated with the entity, valid for the specific time interval, 2) Spatial: It is the graphical representation, in this case, the footprint of the human settlement, 3) Temporal: It represents the valid interval of existence for this timeslice, and 4) The identity component, that allow us to group timeslices belonging to the same human settlement.

By using the continuum model we are able to identify processes such as *conurbation*. The conurbation process involves two cities merging. Using the model we can represent the process as:

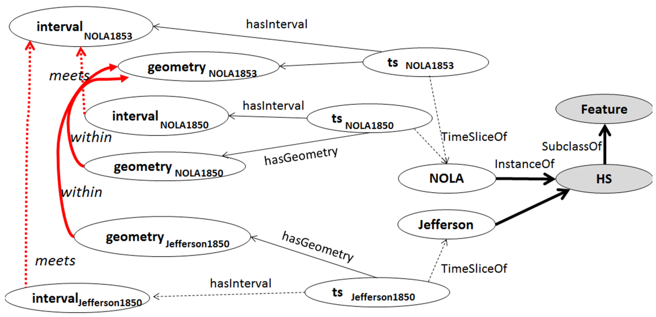

Figure 11. Representation of a conurbation process using the continuum model

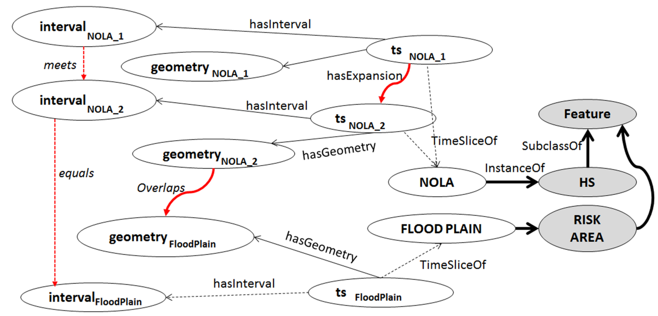

Figure 12. Representation urban growth in risk areas using the continuum model

$$\begin{aligned}
&\forall hasConurbation.\mathcal{TS}\\
&\{ts_1 \in \mathcal{TS}^I | \forall(ts_2, ts_3).(ts_1, (ts_2, ts_3) \in hasConurbation^I\\
&\rightarrow (ts_2, ts_3) \in \mathcal{TS}^I \wedge\\
&(ts_{1g} \neq ts_{2g}) \wedge (ts_{1g} \neq ts_{3g}) \wedge (ts_{2g} \neq ts_{3g}) \wedge\\
&((ts_{1o} = ts_{3o}) \vee (ts_{2o} = ts_{3o})) \wedge (ts_{1o} \neq ts_{2o}) \wedge\\
&meets(ts_{1i}, ts_{3i}) \wedge meets(ts_{2i}, ts_{3i}) \wedge\\
&equals(Union(ts_{1g}, ts_{2g}), ts_{3g}) \wedge ([ts_{1o}, ts_{2o}, ts_{3o}] \in \mathcal{HS})\}
\end{aligned} \tag{21}$$

where: $\{ts_1, ts_2, ts_3\} \in \mathcal{TS}$, $\{ts_{1g}, ts_{2g}, ts_{3g}\} \in \mathcal{G}$ and $\{ts_{1i}, ts_{2i}, ts_{3i}\} \in \mathcal{I}$. This can be expressed in a more compact form as:

$$\begin{aligned}
&hasConurbation((ts_1, ts_2), ts_3) \equiv\\
&hasAnnexation((ts_1, ts_2), ts_3) | ([ts_{1o}, ts_{2o}, ts_{3o}] \in \mathcal{HS})
\end{aligned} \tag{22}$$

Figure 11 depicts how the model is used in the conurbation New Orleans example. $NOLA$ (New Orleans) and $Jefferson$ are instances of the class *human settlements* $\mathcal{HS}$. $ts_{NOLA1850}$ and $ts_{NOLA1853}$ are timeslices of the entity $NOLA$, while $ts_{Jefferson1850}$ is a timeslice of the entity $Jefferson$. In the graphic we can see the relationships that can be established between the geometries and intervals of the different timeslices. Based on the analysis of the spatial-temporal relationships of the components of the timeslices we can infer qualitative information such as the identification of *Conurbation* processes.

Using the continuum model it is also possible to combine timeslices of objects of different nature. For instance, we can model the process *urban growth in risk area*. In order to represent the risk area we will use the footprint of the flood caused by Hurricane Katrina in 2005 (see Figure 9D). We create a new class *risk areas* as $\mathcal{RA}$. Then we can identify the process *growth in risk area* as:

$$\begin{aligned}
&UrbanGrowthInRiskArea(hs_1, hs_2, ra_1) \equiv\\
&hasExpansion(hs_1, hs_2) \wedge\\
&\neg Overlaps(hs_1, ra_1) \wedge Overlaps(hs_2, ra_1)
\end{aligned} \tag{23}$$

where: $\{hs_1, hs_2\} \in \mathcal{HS}$ and $ra_1 \in \mathcal{RA}$

Figure 12 depicts the relationships that are necessary to analyse to determine the urban growth process in risk areas.

VII. Conclusion

In Figures 11 and 12 we represent the relationships between geometries and intervals used in our analysis. By using these relations we can detect complex transitions between timeslices. Understanding data semantics is at the core of our work providing an easier way to manage data and reduce queries complexity. When using reasoning capabilities specific to the Semantic Web, the system may increase the knowledge stored in the ontology.

The continuum model is based on the 4D-fluent representation and develops the continuum concept in the context of a spatial-temporal GIS in order to preserve best understandable semantics for the objects represented. The continuum model handles time and space independently for each object allowing the inclusion or not of time and space in queries of spatial, temporal or spatial-temporal nature. Currently, the system is capable of tracking the evolution of objects along time. This model introduces a novel approach for the handling of properties and attributes for each object. The semantic management of the properties and attributes for each object will be part of further research in order to develop a complete system for the semantics of spatial-temporal information.

Our model offers explicit semantic and flexibility for semantics interoperability between information systems and data sharing. Currently, we are doing research in the field of *smart queries*, a term coined by [25]. The term refers to the combination of heterogeneous datasources in order to solve complex proplems. Using the continuum model we will be able to integrate vector data sources available on the web [26]. We plan to apply these capabilities to complex modelling scenarios such as Land Use/Land Cover change.

References

[1] B. Harbelot, H. Arenas, and C. Cruz, “The spatio-temporal semantics from a perdurantism perspective,” in *Proceedings of the Fifth International Conference on Advanced Geographic*

Information Systems, Applications, and Services GEOProcessing, Nice, France, February-March 2013, pp. 114–119.

[2] M. Yuan, "Use of a three-domain representation to enhance GIS support for complex spatial-temporal queries," *Transactions in GIS*, vol. 3, pp. 137–159, March 1999.

[3] C. Welty and R. Fikes, "A reusable ontology for fluents in OWL," in *Proceedings of 2006 conference on Formal Ontology in Information Systems (FOIS 2006)*, 2006, pp. 226–236.

[4] M. Al-Debei, M. Mourhaf Al Asswad, S. de Cesar, and M. Lycett, "Conceptual modelling and the quality of ontologies: Endurantism vs. perdurantism," *International Journal of Database Management Systems*, vol. 4, no. 3, June 2012.

[5] M. O'Connor and A. Das, "A method for representation and querying temporal information in OWL," in *Proceedings of Biomedical Engineering Systems and Technologies BIOSTEC 2010*, 2010, pp. 97–110.

[6] J. Hobbs and F. Pan, "Time ontology in OWL," (Online) http://www.w3.org/TR/owl-time/, (Accessed on November 2012).

[7] C. Gutierrez, A. Hurtado, and A. Vaisman, "Introducing time into RDF," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 207–218, February 2007.

[8] P. Hayes, "RDF semantics, W3C Recomendation, 10 february 2004," (Online) http://www.w3.org/TR/rdf-mt/, 2004, (Accessed on November 2012).

[9] M. Klein and D. Fensel, "Ontology versioning on the Semantic Web," in *Proceedings of the First International Semantic Web Working Symposium SWWS'01*, Stanford, July 2001, pp. 75–91.

[10] S. Batsakis and E. Petrakis, "SOWL: Spatio-temporal representation reasoning and querying over the semantic web," in *Proceedings of the 6th. International Conference on Semantic Systems I–SEMANTICS 2010*, Graz, Austria, September 2010, pp. 15:1–15:9.

[11] ——, "SOWL: a framework for handling spatio-temporal information in OWL2.0," *Rule Based Reasoning, Programming, and Applications Lecture Notes in Computer Science*, vol. 6826, pp. 242–249, 2011.

[12] K. Ryu and Y. Ahn, "Application of moving objects and spatiotemporal reasoning," 2001, a TIMECENTER Technical Report.

[13] A. Karmacharya, C. Cruz, F. Boochs, and F. Marzani, "Integration of spatial processing and knowledge processing through the semantic web stack," in *GeoSpatial Semantics*, ser. Lecture Notes in Computer Science, C. Claramunt, S. Levashkin, and M. Bertolotto, Eds. Springer Berlin Heidelberg, 2011, vol. 6631, pp. 200–216. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20630-6_13

[14] M. Koubarakis and K. Kyzirakos, "Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL," in *The Semantic Web: Research and Applications*, ser. Lecture Notes in Computer Science, L. Aroyo, G. Antoniou, E. Hyvnen, A. Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, Eds. Springer Berlin Heidelberg, 2010, vol. 6088, pp. 425–439. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13486-9_29

[15] I. Emmons, *Parliament User Guide*, Raytheon BBN Technologies, 2012.

[16] R. Battle and D. Kolas, "Enabling the geospatial semantic web with parliament and GeoSPARQL," *Semantic Web*, 2012.

[17] N. Brisaboa, I. Mirbel, and B. Pernici, "Constraints in spatio-temporal databases: A proposal for classification," in *Proceedings of the 3th. International Workshop on Evaluation of Modeling Methods in System Analysis and Design*, Pisa, 1998.

[18] J. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, Nov. 1983. [Online]. Available: http://doi.acm.org/10.1145/182.358434

[19] C. Strobl, *Encyclopedia of GIS Springer*. Springer, 2008, ch. Dimensionality Extended Nine–Intersection Model (DE–9IM), pp. 240–245.

[20] A. Artale and E. Franconi, "A temporal description logic for reasoning about actions and plans," *Journal of Artificial Intelligence Research*, vol. 9, pp. 463–506, 1998.

[21] F. Baader and W. Nutt, "The description logic handbook," F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds. New York, NY, USA: Cambridge University Press, 2003, ch. Basic description logics, pp. 43–95. [Online]. Available: http://dl.acm.org/citation.cfm?id=885746.885749

[22] G. D. Mondo, M. Rodrguez, C. Claramunt, L. Bravo, and R. Thibaud, "Modeling consistency of spatio-temporal graphs," *Data & Knowledge Engineering*, vol. 84, no. 0, pp. 59 – 80, 2013.

[23] G. D. Mondo, J. G. Stell, C. Claramunt, and R. Thibaud, "A graph model for spatio-temporal evolution," vol. 16, no. 11, pp. 1452–1477, jun 2010.

[24] R. Campanella, *Geographies of New Orleans: Urban fabrics before the storm*. Center for Louisiana Studies, 2006.

[25] J. Goodwin, "What have ontologies ever done for us - potential applications at a national mapping agency," in *OWL: Experiences and Directions (OWLED)*, 2005.

[26] H. Arenas, B. Harbelot, and C. Cruz, "A Semantic Web Approach for Geodata Discovery," in *Proceedings of 7th. SecoGIS Workshop*, Hong Kong, PRC, November 2013.