LC3: A spatio-temporal and semantic model for knowledge discovery from geospatial datasets

Benjamin Harbelot, Helbert Arenas, Christophe Cruz

LC3: A spatio-temporal and semantic model for knowledge discovery from geospatial datasets

Benjamin Harbelot, Helbert Arenas, Christophe Cruz

Laboratoire Le2i, UMR-6306 CNRS, Checksem , Universite de Bourgogne, 9, rue Alain Savary, 21078 Dijon, France

Abstract

There is a need for decision-makers to be provided with both an overview of existing knowledge, and information which is as complete and up-to-date as possible on changes in certain features of the biosphere. Another objective is to bring together all the many attempts which have been made over the years at various levels (international, Community, national and regional) to obtain more information on the environment and the way it is changing. As a result, remote sensing tools monitor large amount of land cover informations enabling study of dynamic processes. However the size of the dataset require new tools to identify pattern and extract knowledge. We propose a model to discover knowledge on parcel data allowing analysis of dynamic geospatial phenomena using time, spatial and thematic data. The model is called Land Cover Change Continuum (LC3) and is able to track the evolution of spatial entities along time. Based on semantic web technologies, the model allows users to specify and to query spatio-temporal informations based on semantic definitions. The semantic of spatial relationships are of interest to qualify filiation relationships. The result of this process permit to identify evolutive patterns as a basis for studying the dynamics of the geospatial environment. To this end, we use CORINE datasets to study changes in a specific part of France. In our approach, we consider entities as having several representations during their lifecycle. Each representation includes identity, spatial and descriptives properties that evolve over time.

Keywords: Geographic Information System, spatio-temporal dynamics, geospatial analysis, ontology, qualitative spatial modeling and reasoning, decision-support systems

1. Introduction

There is a growing interest in geographic information systems from private organization and governments. Geographic Information Systems (GIS) involves massive quantities of micro- and macro-level spatio-temporal data. Handling massive information of complex spatial-temporal phenomena requires special care on data modeling to facilitate integration and exploitation of these data. The evolution along time of an object with a spatial component, can be represented as a sequence of transitions (change in form, location, or attributes). Each transition represents the evolution of an object within the modeled reality. On the basis of the modeling, one of the crucial development goal in GIS sytems of the future is to provide analysis about the evolution of the geospatial environment. However, with current available tools, it is difficult to provide a full analysis to understand the spatio-temporal dynamics. In conventional GIS system, the analysis consists in linking the data together to bring out information. Therefore, most of the time, the study of the spatio-temporal dynamics is reduced to a summary statistical analysis to quantify the increase or decrease of landcovers in a given territory. However, these lancovers and their causalities are not clearly identified. A large majority of these tools is based on relational databases, which leads to failure to take full account of the geographical environment context. The modeling, analysis and visualisation of dynamic geospatial phenomena has been identified as a key developmental challenge for next-generation GIS [1].

A relevant field of study for spatial dynamics is land use/land cover change (LULCC). Along thousands of years, humans have modified the environment. However, only until recently, scientist have identified a relation between LULCC and weather pattern modifications. In many cases land cover change is the result of a combination of economic opportunities, national policies and markets[2]. There are several models currently

employed to model LULCC, however there is a continuous need for new approaches in order to reevaluate current models and improve them [3].

In this paper, we introduce our model, applied to a Land Use Land Cover case study. The land cover of a region varies along time due to a variety of factors. Being able to understand this process of evolution and identify trends and patterns would be helpful for scientist and policy makers to manage land territory. Our approach is called the Land Cover Change Continuum Model (LC3). It is an extension of the Continuum Model introduced in [4]. In this research, we model the evolution along time of geographical entities associated with alphanumeric properties. Our goal is to model the geometric and attribute changes as well as identity changes, implementing an ontology-based system capable of inferring new implicit knowledge from explicit facts. Conventional GIS systems are able to provide analysis to transform data into information. Thus, the stored data takes on meaning and gives evidence of a tangible fact. So far, datas and informations are explicits and uncertainty about the resulting fact is weak. Our system exceeds this level of representation to go towards the discovery of implicit knowledge. The process of transforming information into knowledge is called "appropriation". Compared to information and data, which are tangible and easy to provide, knowledge belongs to a reasoning process that requires a difficult learning in order to interpret informations. That's why uncertainty about knowledge is higher than for data and information. To overcome the limitations of relational databases and provide a greater level in terms of knowledge modeling about the geographic environment, we used Semantic Web technologies. In addition, these tools allow reasoning on the basis of modeled knowledge. We distinguish two kind of reasoning: data validation in order to check the consistency of the system and inference to generate new knowledge from those already represented in the system. Despite the subjective nature of knowledge, increase the level of representation of the geographical environment via high-level ontological entities and reasoning tools allow to get closer the expert skills in an automated way.

In this paper, we propose a novel approach to study LULCC using Semantic Web technologies. In our approach we analyze LULCC datasets and implement mechanisms that allow intelligent processes to infer new implicit knowledge from explicit facts. In the section 2, we will describe relevant research on this field. Then, the section 3 will introduce our model and its formal semantics. Depending on the dataset, some pre-treatments may be needed to feed our model. The section 4 will be dedicated to illustrate these steps. Then, the section 5 will show how our model is able to gradually increase our knowledge about the geospatial environment and its dynamics. Finally, the section 6 presents our conclusions and future research.

2. Related Research

Spatio-temporal modeling is a very heterogeneous domain whose user needs, data formats and structures are relatively varied. Therefore identifying generic characteristics for modeling an entity over time establishes itself as the preliminary work in the design of any spatio-temporal model. The life cycle of an entity over time can be summed up in a succession of states and transitions. States are intrinsic changes in the entity while transitions are modeling more external factors that have led the passage of the entity from one state to another state. Several models in the literature were defined around these two major parameters specific to spatio-temporal evolution. In addition to the modeling itself, we would also consider modeling support. Although over the years, relational databases have emerged as the preferred support of the spatio-temporal modeling, new requirements for the design of future GIS highlighted the lack of high-level semantic and qualitative analytical capabilities for the study of spatio-temporal dynamics. In this work, we propose the semantic web as a support for modeling. Therefore, this section provides a comparison between the relational approach and the semantic web while designing an information system.

2.1. Representing entities over time

A spatio-temporal entity is a representation of the real world entities composed of an identity, descriptive properties and spatial properties. While identity describes a fixed component of the entity, alphanumeric and spatial properties can vary over time and, thus represents its dynamic part. When the identity of an entity varies, there is a particular type of evolution where the spatial-temporal entity is transformed into a new one. In the literature, there are two main types of spatio-temporal entities: 1) moving objects, like for example a taxi travelling the streets of a city, and 2) changing objects, for example, a region whose administrative boundary evolves in time [5].

An important concept regarding the evolution of entities is the identity. It can be defined as the uniqueness of an object, regardless of its attributes or values. It is the feature that distinguishes one object from all others. The identity is essential in the conceptualization

and modeling of a phenomenon. Its importance while modeling dynamic systems has been identified by previous research such as [6], [7], [8], [9]. However, this concept is very subjective because it depends on the criteria selected by the user to define the identity of an entity. Usually the criteria for the definition of the identity depends on the domain of study.

To palliate this limitation, the filiation relationships define the succession links that exist between different representations of the same object at different instants of time. The filiation relationships reveal their interest in basic changes such as divisions or merge of entities. Other spatial changes more or less complex exist and need to identify the 'parents' and 'children' entities. At this step, a filiation is based only on spatial relationships. Therefore, it can be characterized as spatial filiation in the context of spatial changes. In addition, these spatial changes may reveal changes in the nature of the entity. Because of this, the relationships of a filiation is intimately linked to the notion of identity. This relationship is essential to maintain the identity of an entity that evolves and to follow its evolution along time. In this process, it is also necessary to identify new entities that can result from an evolution.

Previous research have identified two general types of filiation relationship: continuation and derivation [4], [10], [11], [12]. In the first case, the identity remains the same. The entity continues to exist, but undergones a change. While in the second case, a new entity is created from the parent after a certain evolution. Derivation relationships can involve several entities at the same time.

2.2. Study of the differents spatio-temporal modeling approaches

The evolution of a spatial entity over time can be seen either as a succession of states (or representations) of the entity, or as a succession of transitions that involved this entity over time. Some models for spatial dynamics are based on "*discrete approaches*" such as: the snapshot model [13], [14], the Space-Time Composites model (STC) [15] and the Spatial-Temporal Object model [16]. However, there are disadvantages with these approaches. They represent only sudden changes. Consequently, it is difficult to identify processes such as movement of an entity in a geographical environment. Other models focus on the identity of the spatial features and how it evolves along time [12]. Another kind of models use the intersection matrix to identify changes in topological relations between evolving features [17].

Another type of models is the so called "*event and process-based*" approach. This approach considers that spatial entities operate under the impetus of an event or a process, the aim of this approach is to analyze the causes and consequences. An example of this type of models is the Event-Based Spatiotemporal Data Model (ESTDM)[18]. The ESTDM model describes a phenomenon through a list of events; a new event is created at the end of list whenever a change is detected. However, this model takes into account only raster data and the causal links between events are hardly highlighted in this model. An alternative to ESTDM is the composite processes [19]. The composite process model deals with some of the limitations of the ESTDM. It is designed to represent the links between events and their consequences. Moreover, the author argues that the data model must differentiate what is spatial, temporal and thematic. We adhere to this point. Another example is the model of topological changes based on events [20]. This model represents changes of a geographic environment as a set of trees. Each tree is connected to the next and the previous through its nodes. The link between two trees is a topological change that reveals the creation of an entity on the geographical environment, the deletion of an entity, the division or the merge of entities or no change. The succession of these topological changes enables the representation of complex changes. These two models represent the characteristics of spatio-temporal processes. These characteristics can be assimilated to patterns of evolution describing functional links between entities to improve analysis of event sequences. This kind of approach is more suitable to understand evolution as it defines explicitly a dynamic aspect of the entities. However, it is limited in its ability to fully take into account the context of the geospatial environment because models of this kind are based on relational databases. Therefore, the analysis resulting patterns are domain-independent. This implies that the scientists have to add a lot of their personal knowledge to define causal links involved in the spatiotemporal dynamics. This limitation can be overcome by adding semantic information in the system and by providing tools able to take into account such knowledge representing the context of the modeled environment.

A natural way to model dynamic phenomena is through a "*graph*". Using this approach, entities and their states are seen as vertex and the relations as edges. Previous research have employed this approach. For instance, [21] used it to model urban spaces. While in [22] a similar approach was used to study road networks. Another example is [23], in which the authors use graphs to model the structure of a territory. However, all these examples focus on the spatial structure and omit the temporal dimension, which requires addi-

tional modeling procedures.

In [24], the author deals with changing representations using qualitative spatial reasoning. The article propose to study continuous transitions which is an implicit assumed notion for any such understanding of motion. The work formalize continous transition allowing to represent spatio-temporal history. However, there is still a need to represent expert knowledge in order to give concrete meaning to the spatio-temporal dynamics of entities.

[25] and [26] presents a spatio-temporal model able to analyse change types in historic regions such as countries, municipalities, and cities. Over time, regions can be renamed, merged together, split into parts, and annexed or moved to and from other regions. The model is able to detect some pattern mainly based on spatial criteria and some semantic to represent name changes. But theses patterns do not refer clearly to real world phenomena as there is a lack of domain knowledge to be considered in the analysis using ontological hierarchies and structures.

In [27], a system prototype called Progress is proposed in order to process real geographic data and reasoning about changing spatial extensions of geographic features. The system is based on a knowledge base to allow the addition of semantics within the geographical environment and represents a significant contribution in the field of GIS sciences and knowledge representation. However, there is few limitation while dealing with geographic phenomena. In fact, there is a lack of methods for the development of theories of causality for geographic phenomena. Moreover, it reveals some difficulties to deal with entities lifecycle and espacially representing the identity over time.

The modeling of a phenomenon generates a complex graph of relationships, such as temporal, spatial and semantic relations. The ontology is presented as a data model capable of representing the concepts of a domain with a dynamic nature, and the relationships between their concepts. The concepts in an ontology are organized in a graph to model all of the relationships necessary to spatial-temporal and thematic modeling.

Ontologies are based on notions of individuals, classes, attributes, relations and events. In an ontology, entities can be treated as individuals and grouped in defined classes. The definition of a class can be further specialized, creating subclasses. Attributes are modeled as properties that represent characteristics, or parameters that entities may possess. Relations are defined as links between entities. Finally, the events correspond to changes in attributes or relationships. The definition of relationships between entities can also be further gener-

alized or specialized. In cases of generalization or specialization for both relations and class definitions, the ontology enables the construction of a hierarchy based on subsumption relationships. The hierarchy allows the inference of new knowledge from explicit information.

2.3. The Semantic Web and relational approach

The concept of an information system is to store data in an organized way to model a domain. While modeling a domain application, two different vision can be applied to represent knowledge. The first one is Closed World Assumption (CWA). In this vision, what is not currently known to be true, is false. The opposite of the closed-world assumption is the open-world assumption (OWA), stating that lack of knowledge does not imply falsity. Each of CWA and OWA are strongly related to specific technologies. OWA is frequently related to semantic Web while CWA is traditionnaly associated to relational databases. Today, much information is stored in relational databases. In this work, we wished to study the potential of the semantic web as support for the spatio-temporal modeling. The stored information is called "complete" or "incomplete ". The relational model is a paradigm where the information must be complete and it must be described by a rigid schema while semantic Web considers incomplete informations described by a flexible schema. Traditional databases require an agreement on a schema, which must be made before data can be stored and queried. While semantic Web allows to define it incrementally, sequentially asserting new statements or conditions. Moreover, the relational model assumes that the only objects that are explicitly represented in the database exist in the domain of interest. Conversely, semantic Web allows different synonym labels to be used for the same object, and same names may refer to different objects. Finally, CWA and OWA directly impact the reasoning that can be applied to the system. In this section, we discuss about CWA and OWA specificities.

2.3.1. Completeness of informations

The relational approach is recognized as complete [28]. The missing information is often managed with the assertion of the value " null " as defined in SQL. In the literature, this situation is known as domain closure assumption [29] and defined that there can not be other objects in the universe than those designated by constants in the database. For example, in a travel application that fly between two destinations on a certain date. If the user searches for a flight between Paris and New York on January 15, 2014 and the application returns

no results then the conclusion is that there is no flight between Paris and New York for the day. However, the completeness of the closed world is not realistic in practice because most databases have information that may be incomplete [30] [31]. One of the main points of the open world assumption is to work on incomplete information. Therefore, the open world allows attributes of a specific object or instance to be incomplete or partially known. The open world assumption adopts the open-domain assumption [29] which defines that there can be in the universe more objects than those defined in the knowledge base unless a constraint in the database prevents this. Taking example of a medical application where the user enters a series of symptoms and the system provides a set of disease or illness for these symptoms return. In case where the system does not return any corresponding diseases symptoms, it does not necessarily mean that the patient is not sick. Consequently the answer of a system under the open world assumption would be " I don't know ". The notion of completeness directly affects how to design the layout of the data management system whether it concern relational databases or ontologies.

2.3.2. Flexibility of the system schema

Since knowledge is a representation of a given understanding by users and domain experts, it is important that both are active in its collection, organization (structure) and its use [32]. Incomplete approaches in open world, such as semantic web approaches, creates the need to extend the information model later in the life cycle of the application. The flexibility of the schema becomes an issue facing the need to adapt their models. Schema changes of a relational system require a fundamental revision of the architecture of the database. Therefore, the relational approach is not fundamentally extensible but provides good capabilities to define the boundaries of a domain model. The Semantic Web was created to reuse existing ontologies and to be an extensible approach. Design and management of semantic web databases can be more flexible through a scheme able to evolve incrementally. In addition, entities of various ontologies can be merged to extend the capabilities of representation of an application. However, the reuse and the merging of ontologies raise the problem of the uniqueness of entities represented when the same entity is represented in two ontologies with different conceptualisations.

2.3.3. Management of the entity uniqueness

Representating entities raises the need to identify them in order to fill the gap between system representa-

tion and reality. The identification is done via the use of labels or names. The unique name assumption is often used to define that different names in the system refer to different entities in the real world [29] [30]. This assumption is often linked to the closed world assumption where data are considered as complete and the system is not supposed to be extended thereafter by merging data from another relational database. Contrary to the relational approach, OWL (Ontology Web language) is used to define different labels to represent same objects as the information is accepted as being incomplete in this kind of application. Therefore a graph fusion process is frequently implemented. So it is common to see same names that refer to different entities. Similarly, different names may refer to a same entity. These assertions about the identity must be explicitly defined through relations *owl* : *sameAs* and *owl* : *differentFrom*.

2.3.4. Interpretation of the world represented

Knowledge of a domain is contextual. Therefore, depending on the context and perspective, the meaning or importance of information can have more or less importance. Thus, the interpretation of constraints under CWA or OWA have a direct impact on the reasoning results that will be applied on the system. In relational databases, a single schema is used to define the scope and give an interpretation of the world. This uniqueness of the schema makes the closed world particularly suitable for data validation as it leaves only one interpretation of the world, therefore facts are true or false, and no other explanation can justify integrity constraint violation. However, within the ontology, schema and data are defined separately respectively in the TBox and ABox [32]. Thus, several interpretations (world) for the same data are possible. Therefore, open world reasoners will generate knowledge to explain the violation of a constraint. For example, considering a simple constraint like "a child can have only one mother". Now, consider the two following assertions: "Brittany is the child of Amy" and "Brittany is the child of Amanda". Closed world reasoners will raise an error assuming that Brittany have two mothers which violates the constraint. Meanwhile, open world reasoners will not raise any errors but infer the fact that Amy and Amanda refer to the same person who is the only one mother of Brittany. As a result, the infered fact avoid violating the constraint.

2.3.5. Discussion

The closed world assumption invites to define "what is possible". Conversely, the open world assumption allows to decide "what is not possible". When OWL ontology is empty then everything is possible. It is only

when the ontology is gradually constrained that it becomes more restrictive.

In well-defined (seats in an airplane, the books in a library) domains, the relational model is a suitable approach. The closed world assumption is effective to facilitate the validation data for the trading operations. The number of negative facts about a given area is usually much larger than the number of positive developments. Thus, in many applications, the number of adverse events is so great that their explicit representation may become impossible. In such cases, it is easier and faster to set all the "real" facts known to enumerate all the conditions of "false". However, the relational model is a paradigm where the information must be complete and described by a single scheme. The traditional databases require agreement on a scheme to be done before data storage and querying. The relational model assumes that all objects and relationships in the field are those that are explicitly represented in the database, which uniquely identifies the object names in this field. This makes the closed world assumption and its related assumptions a very bad choice when attempting to combine information from multiple sources, to deal with uncertainty or incompleteness of the world. The main advantage of the open world assumption and the Semantic Web (usually associated with this assumption) is to allow information to be reusable. Reuse of an ontology enables to assemble, extend, specialize or adapt knowledge defined from other ontologies. In this way, the Semantic Web offers a good flexibility to allow the integration of new knowledge when the application requires specialized knowledge. Finally, the approach allows the Semantic Web to provide inference mechanisms to generate knowledge within an application.

To conclude, the relational approach is a very suitable candidate for data validation. However, irregularities and incompleteness is a limit to the design of relational model. The Semantic Web approach goes beyond these limitations by providing a flexible structure of the data schema. In addition, the explicit separation between the schema (TBox) and data (ABox) provides an environment for the interpretation of the data represented. Therefore, the incompleteness of the open world can be partially filled with reasoners capable of self-feeding system based on constraints. In many applications of the Semantic Web, it should be interesting to use OWL to define integrity constraints [33]. However, the open world assumption and rejection of the unique name assumtion contradict the development of integrity constraints in OWL [34] [35]. Indeed, the constraints, defined for data validation in closed world, generate new knowledge in reasoning applications based on

OWL. Therefore, in this work, we provide hybrid applications that combine both arguments with the open world reasoning capabilities and closed world data validation. Consequently, we adopt the unique name assumption to allow the coexistence of both approaches.

3. The LC3 Model

LC3 model takes advantage of GeoSPARQL spatial capabilities as well as methods proposed in ontologies *fluent* to represent the evolution. In addition, this model allows to represent the filiation relationship between consecutive *timeslices* associated with an entity. LC3 model was proposed to meet the need of modeling changes that occur on spatial entities. These changes are characterized by spatial changes (related to position or spatial footprint of the entity), related to the semantic or identity-related entities. Therefore, the model integrates and distinguished relations related to space (topological relations), time (Allen Relations [36]), semantics and identity (parent-child relationship).

In order to specify our model, we use Description logics [37], later we provide information regarding how the model has been implemented.

3.1. Defining spatio-temporal layers

Our model deals with dynamic entities evolving in time called "timeslice". Each of them can be defined along four components that are identity, spatial, temporal and semantic as depicted in Figure 1.
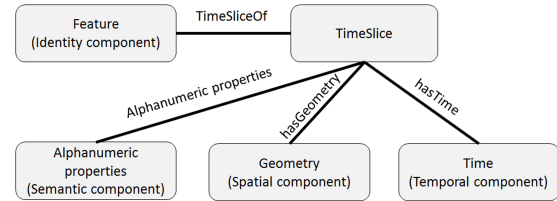

Figure 1: Components of an entity

The identity is the most important component of the model. Each timeslice has an identity defined using a class. Traditionally, dealing with landparcels, a class corresponds to a specific land cover, but in the case of timeslice correspond to individual object. Other semantic can be used to underline the uniqueness of a timeslice. Ontologies are useful to organize classes in different semantic levels using a taxonomy. Each class describes a concept and the taxonomy allows the same

timeslice to be associated to concepts more or less specific.

$$Timeslice \sqsubseteq \top \qquad (1)$$

In our model, the *Timeslice* class is the more general concept, as shown in equation 1, and can be specialized in a hierarchy. Then, specific concepts are useful for discriminating entities represented while general concepts allows to group entities. Such an arrangement of concepts created a depth index of the hierarchy to evaluate different semantic levels which is possible to exploit. Equation 2 formalizes the hierarchy.

$$C_n \sqsubseteq ... \sqsubseteq C_i \sqsubseteq ... \sqsubseteq \mathcal{TS} \qquad (2)$$

where $\mathcal{TS}$ correspond to the Timeslice class and n is the depth of the hierarchy.

In order to represent the time in the evolution, we take advantage of the approach suggested by [38]. We can think of the temporal domain as a linear structure composed by a set of temporal points (*TemporalPoint*).

$$TemporalPoint \sqsubseteq \top \qquad (3)$$

All elements of type *TemporalPoint* follow a strict order, which forces all points between two temporal points t_1 and t_2 to be ordered. By selecting a pair of temporal points $[t_o, t_f]$ we can limit a closed set of ordered points that defines time intervals (*Interval*).

$$\begin{aligned} Interval &\sqsubseteq \top \\ Interval &\equiv \exists hasStartPoint.TemporalPoint \sqcap \qquad (4) \\ &\exists hasEndPoint.TemporalPoint \end{aligned}$$

In some studies, we have land cover datasets at different instant of time. These information datasets can be interpreted as snapshots of a dynamic process. Then, each datasets corresponds to a temporal point (*TemporalPoint*). However, when a geographic entity does not change between two time points, we can infer that the land cover remains static during that interval. In order to represent both time intervals and time points we define the concept *Time* ($\mathcal{T}$).

$$\mathcal{T} \equiv TemporalPoint \sqcup Interval \qquad (5)$$

In our model, we define the property *hasTime* which has as a domain elements of the class *Time* ($\mathcal{T}$), in this way, we can use both *TemporalPoints* and *Intervals*:

$$\forall hasTime.\mathcal{T} \qquad (6)$$

In a perdurantism approach the evolution of an entity is described by a set of transitory constructions called timeslices. They are transitory in the sense that they are valid only for a defined finite time. In our work, the concept of timeslices is represented by the class *TimeSlice* ($\mathcal{TS}$). This class has four components: 1) Spatial, which is the geometric representation ($\mathcal{G}$) of the feature; 2) Identity, to associate each timeslice to the object ($\mathcal{O}$) they represent; 3) Temporal, to describe the time ($\mathcal{T}$) in which the timeslice is valid; 4) A set of alphanumeric properties are used to describe characteristics of features during the timeslice valid time, we call it the semantic component ($\mathcal{S}$). Equation 7 depicts the formalization of the class *TimeSlice* in our research.

$$\begin{aligned} \mathcal{TS} \equiv\ &\exists hasGeometry.\mathcal{G} \sqcap \\ &\exists TimeSliceOf.\mathcal{O} \sqcap \\ &\exists hasTime.\mathcal{T} \sqcap \\ &\exists hasProperties.\mathcal{S} \end{aligned} \qquad (7)$$

3.2. Defining spatio-temporal transitions

In a geographic area with a dynamic land cover, the same region can be associated with different timeslices at different points of time. In order to represent spatial association in time, we define a *filiation* property in the LC3 model. This property enables us to link two timeslices of consecutive times. As a result, the class *Timeslice* should be defined as *domain* (see Equation 8) and *range* (see Equation 9).

$$\exists hasFiliation \sqsubseteq \mathcal{TS} \qquad (8)$$

$$\top \sqsubseteq \forall hasFiliation.\mathcal{TS} \qquad (9)$$

This property is essential to retrieve linkage between two entities. But, this knowledge requires further processing to properly understand the evolution. In our model, we propose to specialize this property in a different layer of knowledge.

Equation 10, 11 and 12 formalize the hierarchy using Description Logic.

$$\begin{aligned} hasFiliation \equiv\ \\ hasContinuation \sqcup hasDerivation \end{aligned} \qquad (10)$$
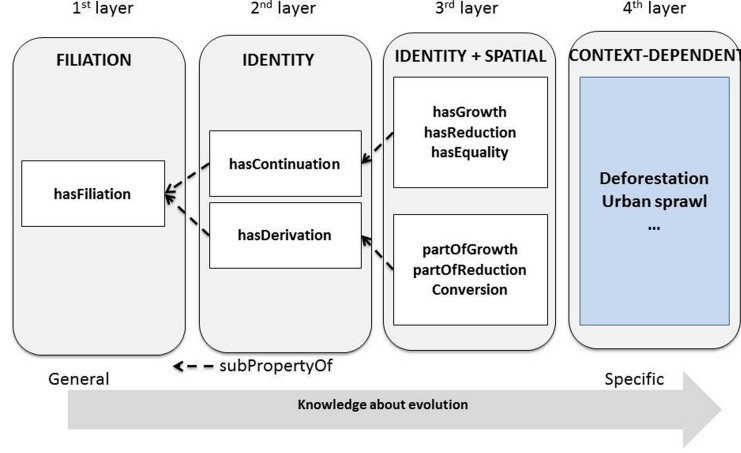
Figure 2: Different layers to qualify an evolution

In order to specialize our filiation relationships, we integrate identity constraints to distinguish objects that have changed their natures. Thus, filiation relationships are divided into continuation or derivation relationships.

$$hasContinuation \equiv$$
$$hasEquality \sqcup hasGrowth \sqcup hasReduction \sqcup \quad (11)$$
$$hasAnnexation \sqcup hasSeparation$$

$$hasDerivation \equiv$$
$$Conversion \sqcup hasSplit \sqcup hasFusion \sqcup \quad (12)$$
$$partOfAnnexation \sqcup partOfSeparation$$

Then, we can add also spatial contraints to further specialize each of these two relationships into domain independant pattern which can be identified within the dataset. For the definition of the different types of evolution, we use the relations defined in DE-9IM (Equal, Within, Contains) to define spatial constraints [39]. Moreover, we use the relations defined by Allen in [36] to define temporal constraints (Meets, Equal, ...). In order to distinguish spatial and temporal equality, we decided to adopt *hasEqual* term in case of spatial equality.

In the LC3 Model a change on the spatial representation or on the semantic component generates a new timeslice establishing 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 timeslices ts1 and ts2 is defined by the relationships between their spatial representations (g_p and g_c), their semantic definitions (s_p and s_c), their identity (o_p and o_c)

and their time intervals (i_p and i_c). A filiation relationship is defined when a change occurs on the geometry, the semantic component or the identity.

In order to define our constraints, we consider interpretations that consist of a non-empty set Δ^I (the domain of the interpretation) and an interpretation function, which assigns to every atomic concept A a set $A^I \subseteq \Delta^I$, to every atomic role R an Object Properties (R) $R^I \subseteq \Delta^I \times \Delta^I$ and to every datatype properties (U) a set $U^I \subseteq \Delta^I \times \Delta^I_{\mathcal{D}}$ with $\mathcal{D}$ the Data Ranges defined by $D^{\mathcal{D}} \subseteq \Delta^I_{\mathcal{D}}$. In order to have a better expressiveness, we define our constraints in First Order Logic (FOL). As a first step, we express some operators needed for expressing our spatio-temporal constraints in Equation 14.

$$Meets^I = \{\langle \mathcal{T}_i, \mathcal{T}_{i+1}\rangle | i \in \mathbb{N}\} \ with \begin{cases} if \ \mathcal{T}_i, \mathcal{T}_{i+1} \in TemporalPoint \\ \forall \mathcal{T}_i, \mathcal{T}_{i+1} \to succ(\mathcal{T}_i) = \mathcal{T}_{i+1} \\ \\ if \ \forall a, b \in TemporalPoint^I \\ if \ \forall I, I' \in Interval^I \\ if \ (I, a) \in hasEndPoint \\ and \ (I', b) \in hasStartPoint \\ \to a = b \end{cases} \quad (13)$$

$$hasEqualGeometry^I = \{\langle g, g'\rangle | (\forall x(x \in g \to x \in g')) \wedge$$
$$(\forall x(x \in g' \to x \in g)), g, g' \in \mathcal{G}\}$$

$$Within^I = \{\langle g, g'\rangle | (\forall x(x \in g \to x \in g')) \wedge$$
$$(\exists x(x \in g' \wedge x \notin g)), g, g' \in \mathcal{G}\} \quad (14)$$

$$Union^I = \{\langle g_1, g_2, g'\rangle | (\forall x(x \in g_1 \vee x \in g_2 \to x \in g')),$$
$$g_1, g_2, g' \in \mathcal{G}\}$$

Moreover, we defined the property *hasSemanticProperty* representing all semantic properties (*Datatype* or

Object) of a timeslice which is not hasGeometry, has-Time or isTimesliceOf. This semantic component ($\mathcal{S}$) is defined by Equation 15.

$$
\begin{aligned}
SemanticProperties^I = \{\langle s, s'\rangle \,|\, \forall s (\forall s' (&(\langle s, s'\rangle \in \mathcal{R} \vee \\
&\langle s, s'\rangle \in \mathcal{U}) \wedge (\neg hasGeometry(s, s') \wedge \\
&\neg hasTime(s, s') \wedge \neg isTimesliceOf(s, s'))))\}
\end{aligned}
\tag{15}
$$

Now we can define the constraints for each relationship of our model. Equation 16 expresses the most generic of them: the *hasFiliation* relationship.

$$
\begin{aligned}
\big[\forall p (\forall c (&hasFiliation(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow \neg hasEqual(g_p, g_c))) \\
&\vee (\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p \neq o_c) \vee \\
&(\exists s_p \exists s_c (SemanticProperties(p, s_p) \wedge \\
&SemanticProperties(c, s_c) \rightarrow s_p \neq s_c)) \\
&\wedge (\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{16}
$$

Then, we can further specialize the *hasFiliation* to define more complex relationships. **hasContinuation**: The *continuation* relationship specializes the filiation relationship. In this case a change may occur only on the geometry or the semantic component but the identity remains the same. A filiation relationship is defined when a change occurs on the geometry, the semantic component or the identity.

$$
\begin{aligned}
\big[\forall p (\forall c (&hasContinuation(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow \neg hasEqual(g_p, g_c))) \vee \\
&(\exists s_p \exists s_c (SemanticProperties(p, s_p) \wedge \\
&SemanticProperties(c, s_c) \rightarrow s_p \neq s_c) \wedge \\
&(\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p = o_c) \wedge \\
&(\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{17}
$$

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

$$
\begin{aligned}
\big[\forall p (\forall c (&hasDerivation(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow hasEqual(g_p, g_c))) \vee \\
&(\exists s_p \exists s_c (hasSemanticProperties(p, s_p) \wedge \\
&hasSemanticProperties(c, s_c) \rightarrow s_p = s_c) \wedge \\
&(\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p \neq o_c) \wedge \\
&(\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{18}
$$

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

$$
\begin{aligned}
\big[\forall p (\forall c (&hasGrowth(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow Within(g_p, g_c))) \wedge \\
&(\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p = o_c) \wedge \\
&(\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{19}
$$

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

$$
\begin{aligned}
\big[\forall p (\forall c (&hasReduction(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow Within(g_c, g_p))) \wedge \\
&(\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p = o_c) \wedge \\
&(\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{20}
$$

hasEquality: In this case, there is no change on any component.

$$
\begin{aligned}
\big[\forall p (\forall c (&hasEquality(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow hasEqual(g_c, g_p))) \wedge \\
&(\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p = o_c) \wedge \\
&(\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{21}
$$

hasConversion: In this case, there is no change on geometry but the identity changes.

$$
\begin{aligned}
\big[\forall p (\forall c (&hasConversion(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \\
&\rightarrow hasEqual(g_c, g_p))) \wedge \\
&(\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow o_p \neq o_c) \wedge \\
&(\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \\
&\rightarrow meets(i_p, i_c))))))\big]
\end{aligned}
\tag{22}
$$

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

$$
\begin{aligned}
\big[\forall p (\forall c \forall c' (&hasSplits(p, c) \wedge hasSplits(p, c') \wedge \\
&\mathcal{TS}(p) \wedge \mathcal{TS}(c) \wedge \mathcal{TS}(c')) \rightarrow \\
&((\exists g_p \exists g_c \exists g_{c'} (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \wedge \\
&hasGeometry(c', g_{c'}) \rightarrow Within(Union(g_c, g_{c'}), g_p))) \wedge \\
&(\exists o_p \exists o_c \exists o_{c'} (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \wedge \\
&isTimesliceOf(c', o_{c'}) \rightarrow (o_p \neq o_c) \wedge (o_p \neq o_{c'}))) \wedge \\
&(\exists i_p \exists i_c \exists i_{c'} (hasTime(p, i_p) \wedge hasTime(c, i_c) \wedge \\
&hasTime(c', i_{c'}) \rightarrow (Meets(i_p, i_c) \wedge Meets(i_p, i_{c'}))))))\big]
\end{aligned}
\tag{23}
$$

hasSeparation : 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.

$$
\begin{aligned}
&[\forall p(\forall c \forall c'(hasSeparation(p,c) \land hasSeparation(p,c') \\
&\land \mathcal{TS}(p) \land \mathcal{TS}(c) \land \mathcal{TS}(c')) \rightarrow \\
&((\exists g_p \exists g_c \exists g_{c'}(hasGeometry(p,g_p) \land hasGeometry(c,g_c) \\
&\land hasGeometry(c',g_{c'}) \rightarrow Within(Union(g_c,g_{c'}),g_p)))\land \\
&(\exists o_p \exists o_c \exists o_{c'}(isTimesliceOf(p,o_p) \land isTimesliceOf(c,o_c) \\
&\land isTimesliceOf(c',o_{c'}) \rightarrow (o_p = o_c) \land (o_p \neq o_{c'})))\land \\
&(\exists i_p \exists i_c \exists i_{c'}(hasTime(p,i_p) \land hasTime(c,i_c)\land \\
&hasTime(c',i_{c'}) \rightarrow (Meets(i_p,i_c) \land Meets(i_p,i_{c'}))))))]
\end{aligned}
\tag{24}
$$

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.

$$
\begin{aligned}
&[\forall p \forall p'(\forall c(hasFusion(p,c) \land hasFusion(p',c) \\
&\land \mathcal{TS}(p) \land \mathcal{TS}(p') \land \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_{p'} \exists g_c(hasGeometry(p,g_p) \land hasGeometry(p',g_{p'}) \\
&\land hasGeometry(c,g_c) \rightarrow Within(Union(g_p,g_{p'}),g_c))\land \\
&(\exists o_p \exists o_{p'} \exists o_c(isTimesliceOf(p,o_p) \land isTimesliceOf(p',o_{p'}) \\
&\land isTimesliceOf(c,o_c) \rightarrow (o_p \neq o_c) \land (o_{p'} \neq o_c)))\land \\
&(\exists i_p \exists i_{p'} \exists i_c(hasTime(p,i_p) \land hasTime(p',i_{p'})\land \\
&hasTime(c,i_c) \rightarrow (Meets(i_p,i_c) \land Meets(i_{p'},i_c))))))]
\end{aligned}
\tag{25}
$$

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

$$
\begin{aligned}
&[\forall p \forall p'(\forall c(hasAnnexation(p,c) \land hasAnnexation(p',c) \\
&\land \mathcal{TS}(p) \land \mathcal{TS}(p') \land \mathcal{TS}(c)) \rightarrow \\
&((\exists g_p \exists g_{p'} \exists g_c(hasGeometry(p,g_p) \land hasGeometry(p',g_{p'}) \\
&\land hasGeometry(c,g_c) \rightarrow Within(Union(g_p,g_{p'}),g_c))\land \\
&(\exists o_p \exists o_{p'} \exists o_c(isTimesliceOf(p,o_p) \land isTimesliceOf(p',o_{p'}) \\
&\land isTimesliceOf(c,o_c) \rightarrow (o_p = o_c) \land (o_{p'} \neq o_c)))\land \\
&(\exists i_p \exists i_{p'} \exists i_c(hasTime(p,i_p) \land hasTime(p',i_{p'})\land \\
&hasTime(c,i_c) \rightarrow (Meets(i_p,i_c) \land Meets(i_{p'},i_c))))))]
\end{aligned}
\tag{26}
$$

Figure 3 depicts the list of pattern detected in the LC3 Model that we have presented using the FOL.

Before developing a system capable of reasoning and increase the knowledge about the geographical environment, it is necessary to apply a pretreatment in order to detect filiation relationships and to deal with some uncertainty in the dataset. To this end, some specifics tools have been used in this project. Then, we would introduce Corine Land Cover dataset. Finally, our LC3 model was defined in compliance with certain integrity constraints. Traditionally, the semantic web approach is not intended for the management of integrity constraints because this approach follows the open world assumption. In last section, we present a hybrid solution allowing to manage both reasoning capabilities under the open world assumption and data validation under closed world assumption.
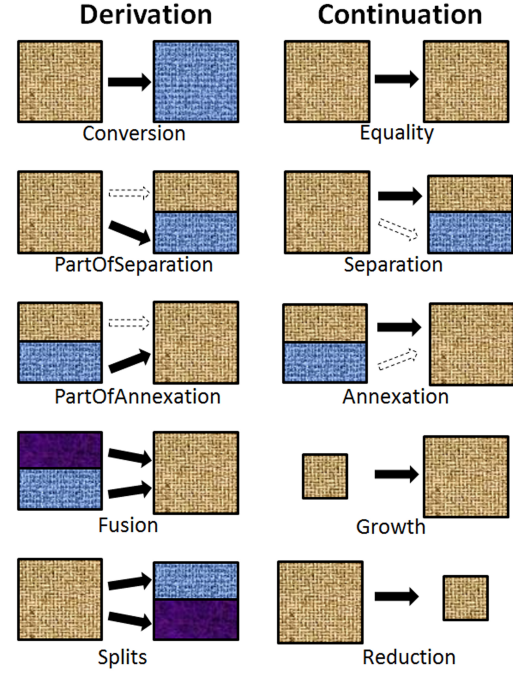

Figure 3: List of patterns detected in the LC3 Model

4. Feeding the LC3 Model

Corine Land Cover dataset defines three layers of an area at different time. Each of them is composed of multiple landparcel that evolve between each layer. In the previous section, we presented our model assuming that filiation relationships were already known. However, Corine Land Cover does not provide any knowledge about filiation relationships and requires a new methodology and some adjustments in order to identify the relationships between timeslices from scratch. Determining corresponding landparcel from one layer to the next one is a big challenge due to heterogeneity of change. Change may occur on identity or on the spatial component. First, we introduce the basics of our implementation before presenting the new methodology to deal with Corine Land Cover specificities. Later, we increase knowledge about filiation relationships to achieve the qualification of dynamic phenomena.

4.1. Architecture overview and data integration

In order to test our model, we opted for using LULCC information from Corine Land Cover [40]. The Corine dataset covers multiple countries. Each one of them are represented to three time points being the years 1990,

2000 and 2006. Corine Land Cover provide objects encoded as shapefiles. In order to integrate these data in our system, we translated it into RDF triples using a custom made JAVA program using the library GeoTools [41]. The information in triple format was then uploaded into a Stardog [42] triplestore. The main reason to opt for this triplestore concern reasoning capabilities. More specifically, Stardog supports OWL 2 and SWRL for reasoning and great possibilities to deal with constraints under Open World Assumption and Closed World Assumption. We detail these specific capabilities in the remainder of this article.

Stardog does not offer support for GeoSPARQL. Therefore spatial analysis has to be computed with external tools. In our case, we developed a JAVA/Geotools application to perform all the required spatial analysis. In our research, polygons for each time point were identified and encoded as a timeslices. Then, the LC3 application queries the triplestore and retrieves the timeslices using a spatial index. Next, it proceeds to identify the filiation relationships by taking into consideration the overlapping between timeslices of consecutive time periods. Our application also identifies the adjacency relations for timeslices that coexist in the time interval. Once the relationships are identified, they are translated into triples and uploaded into the triplestore.

4.2. CORINE Land Cover dataset

Corine Land Cover (CLC) is a map of the European environmental landscape based on interpretation of satellite images. It provides comparable digital maps of land cover for each country for much of Europe. CORINE datasets defines a land cover hierarchy of classes composed of three level to classify each feature contained in the dataset. the first level has five categories, the second level fifteen and the third has forty four land cover categories. The five broader concepts at level 1 are: 1) Artificial Surfaces, 2) Agricultural Areas, 3) Forest and seminatural areas, 4) Wetlands and 5) Water bodies.

In addition, Corine Land Cover defines three datasets corresponding to years 1990, 2000, 2006. For testing our model, we selected France area and worked on the department of Gironde.

Figure 4 depicts an example of evolution we would like to model with timeslices. At time *t1*, the first part of the area has a "*Continuous urban fabric*" land cover and the rest is a "*Coniferous forest*". At time *t2*, a large part of the forest transforms into "*Continuous urban fabric*". As a result "*Continuous urban fabric*" expands on the area while "*Coniferous forest*" reduces. At time *t3*, "*Continuous urban fabric*" divided into two part. One
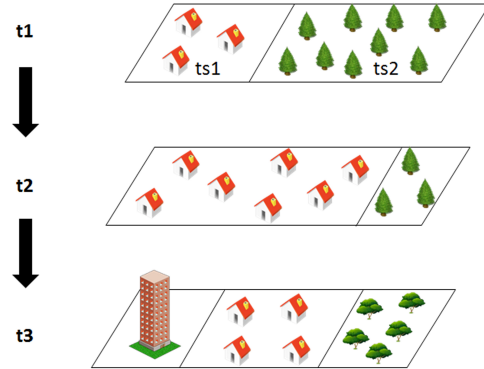

Figure 4: Example of evolution of the land cover modeled with timeslice

of them remains unchanged while the other become "*Industrial or commercial units*". Finally, the "*Coniferous forest*" transform into "*Broad-leaved forest*". The study of such an evolution requires an effective methodology for handling spatio-temporal entities.

4.3. Integrity Constraint Validation

Spatiotemporal data is the most likely to be updated due to their evolutionary nature. Therefore, data integration is an intrinsic component of the spatio-temporal modeling. Most of information systems are designed in accordance with a data model in order to structure, manipulate and search data. In addition, data models should provide a language or implicit rules for maintaining data integrity. Therefore, data integration is strongly related to data validation to ensure consistency of the information system. Traditionnally, constraints are interpreted differently depending on whether they respect the open world assumption (OWA) or the closed world assumption (CWA). Contraints are interpreted as integrity constraints under relational approach and as logical axioms according to Semantic Web approach. In the first approach, constraints raises errors when data inserted violate these constraints whereas in the second one, constraints are the basis for reasoning inference.

For example, let us consider an information system with the following statements:

Subject	Predicate	Object
hasGrowth	range	Timeslice
Timeslice_01_1990	hasGrowth	Timeslice_01_2000

11

Table 1: Corine Land Cover class hierarchy

Level 1	Level2	Level 3
Artificial surfaces	Urban fabric	Continuous urban fabric
		Discontinuous urban fabric
	Industrial, commercial and transport units	Industrial or commercial units
		Road and rail networks and associated land
		Port areas
		Airports
	Mine, dump and construction sites	Mineral extraction sites
		Dump sites
		Construction sites
	Artificial, non-agricultural vegetated areas	Green urban areas
		Sport and leisure facilities
Agricultural areas	Arable land	Non-irrigated arable land
		Permanently irrigated land
		Rice fields
	Permanent crops	Vineyards
		Fruit trees and berry plantations
		Olive groves
	Pastures	Pastures
	Heterogeneous agricultural areas	Annual crops associated with permanent crops
		Complex cultivation patterns
		Land principally occupied by agriculture, with significant areas of natural vegetation
		Agro-forestry areas
Forest and semi natural areas	Forests	Broad-leaved forest
		Coniferous forest
		Mixed forest
	Scrub and/or herbaceous vegetation associations	Natural grasslands
		Moors and heathland
		Sclerophyllous vegetation
		Transitional woodland-shrub
	Open spaces with little or no vegetation	Beaches, dunes, sands
		Bare rocks
		Sparsely vegetated areas
		Burnt areas
		Glaciers and perpetual snow
Wetlands	Inland wetlands	Inland marshes
		Peat bogs
	Maritime wetlands	Salt marshes
		Salines
		Intertidal flats
Water bodies	Inland waters	Water courses
		Water bodies
	Marine waters	Coastal lagoons
		Estuaries
		Sea and ocean

The first statement defines that a triple using the *has-Growth* relationship as a predicate should have an instance of *Timeslice* class. The second statement is an example of triple using the *hasGrowth* relationship as a predicate. As we can see, according to the first statement *Timeslice_01_2000* should be an instance of *Timeslice* class as range. But this fact is not explicitly defined in the information system. Under the open world assumption, the system will consider that the database is consistent infering that *Timeslice_01_2000* is an instance of the class *Timeslice*. In other words, it will create knowledge in order to provide an explanation to avoid the constraint violation. Now, considering the closed world assumption, there is no fact in the knowledge base defining *Timeslice_01_2000* as an instance of the *Timeslice* class, then the system cannot infer it. Consequently, an error will be raised considering that the database is not consistent.

Considering that both approaches have their own advantages, we propose an hybrid system in which certain well-defined statements are interpreted in a closed world to maintain data consistency while other statements will be left to the interpretation of the open world to generate knowledge automatically from existing data in the information system. Therefore, the constraints that must be interpreted in a closed world are those used to define our data model and have been formalized in the previous section. Few tools of literature allow coexistence of open world and closed world. To our knowledge, the only one able to provide this functionality is *Stardog* triplestore. For this, the constraints that have to be interpreted in a closed world (integrity constraints) are defined in a specific file. Then, Stardog simply launches a data validation to check the consistency of the system based on all the constraints in the file. The contents of this file is given in the code 1:

Code 1: The integrity Constraints file used for data validation

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix : <http://www.w3.org/2012/12/rdf-val/SOTA-ex#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/'> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix cs: <http://www.checksem.fr#> .


cs:TimeSlice a owl:Class ;
    rdfs:subClassOf
        [ owl:onProperty cs:hasGeometry   ; owl:cardinality 1 ] ,
        [ owl:onProperty cs:hastime       ; owl:cardinality 1 ] ,
    [ owl:onProperty cs:hasTSArea   ; owl:cardinality 1 ] .

cs:hasGeometry  a owl:DatatypeProperty ; rdfs:domain cs:TimeSlice; rdfs:domain cs:SpatialTemporalRelation;
 rdfs:range xsd:String .
cs:hasTime  a owl:ObjectProperty ; rdfs:domain cs:TimeSlice ; rdfs:range checksem:Time   .
cs:hasTSArea  a owl:DatatypeProperty ; rdfs:domain cs:TimeSlice; rdfs:range xsd:double .


cs:SpatialTemporalRelation a owl:Class ;
    rdfs:subClassOf
        [ owl:onProperty cs:hasGeometry    ; owl:cardinality 1 ] ,
        [ owl:onProperty cs:hasTSInitial    ; owl:cardinality 1 ] ,
    [ owl:onProperty cs:hasTSResulting   ; owl:cardinality 1 ] ,
    [ owl:onProperty cs:hasArea   ; owl:cardinality 1 ] ,
    [ owl:onProperty cs:hasOverlappingChild   ; owl:cardinality 1 ] ,
    [ owl:onProperty cs:hasOverlappingParent    ; owl:cardinality 1 ] .

cs:hasTSInitial  a owl:ObjectProperty ; rdfs:domain cs:SpatialTemporalRelation ; rdfs:range cs:TimeSlice   .
cs:hasTSResulting  a owl:ObjectProperty ; rdfs:domain cs:SpatialTemporalRelation ; rdfs:range cs:TimeSlice   .
cs:hasArea  a owl:DatatypeProperty ; rdfs:domain cs:SpatialTemporalRelation; rdfs:range xsd:double .
cs:hasOverlappingChild  a owl:DatatypeProperty ; rdfs:domain cs:SpatialTemporalRelation;
rdfs:range xsd:integer .
cs:hasOverlappingParent  a owl:DatatypeProperty ; rdfs:domain cs:SpatialTemporalRelation;
rdfs:range xsd:integer .
```

The file shows integrity constraints of the LC3 Model that have to be interpreted in a closed world. In the LC3 model, primitives describing our entities are timeslices. Each timeslice is required to have a geometry, a timestamp and an identity. OWL defines identity through a class that can be substituted for the *isTimesliceof* relationship. Thus, identity is defined by a class and this class is a subclass of the class *Timeslice*. Therefore, the class corresponding to the identity inherits the constraints defined on the *Timeslice* class.

Lines 9-12 are used to define integrity constraints about timeslices. The relationships of the model must also be defined by integrity constraints. Indeed, the datatype property *hasGeometry* can apply only on timeslices. Consequently, the *domain* for this property must be the *Timeslice* class. The coordinates of geometries of our model is defined and stored in the format *WKT* which is a string format. And the range property *hasGeometry* is a *xsd:string* value.

Similarly, the object property *hasTime* applies on a timeslice and can set a time instant or a time interval for each of them. Thus, the *Timeslice* class is defined as *domain* and the *Time* class as *range*. Lines 14 and 15 define the integrity constraints for the properties *hasGeometry* and *hasTime* of our model.

The following lines of the file concern the preprocessing work needed when filiation relationships are not defined in the dataset. Specifically, these integrity constraints help ensure that all elements required for calculations of filiation relationships are included in the information system and for all landparcels.

4.4. Detect filiation relationships

Filiation relationships correlate entities at different instants of time. It can represent both identity and spatial changes. Spatial changes include division, fusion, growth, reduction or just shape modifications making difficult the establishment of links between two entities at different time. Some spatial changes are very small and not significant. Thus, they have to be removed to maintain relevance of the system in respect of the reality.

4.4.1. Spatio-temporal transitions as a basic to study filiation relationship

In Corine Land Cover datasets, we have information for three times (1990, 2000 and 2006). All timeslices are represented, but filiation relationships are missing preventing proper understanding of evolution. Thus, the first step involves identifying filiation relationships in respect of the reality. Without any previous knowledge, we decided to study filiation on the

basis of spatio-temporal overlap between timeslices at consecutive time. To this end, we make two comparisons, 1990 with 2000 and 2000 with 2006. The result of the overlapping analysis is stored as an instance of a class called *SpatialTemporalTransition*. Equation 27 shows the formal specification of the class *SpatialTemporalTransition*.

$$
\begin{aligned}
SpatialTemporalTransition &\equiv \\
\exists hasParentTimeSlice.\mathcal{TS} &\sqcap \\
\exists hasChildTimeSlice.\mathcal{TS} &\sqcap \\
\exists hasOverlappingParent.Double &\sqcap \quad (27) \\
\exists hasOverlappingChild.Double &\sqcap \\
\exists hasArea.Double &\sqcap \\
\exists hasGeometry.\mathcal{G}
\end{aligned}
$$

Each instance of the spatio-temporal transitions is connected to the parent and the child timeslices respectively through *hasParentTimeSlice* and *hasChildTimeSlice* relationships. In addition, the area of intersection is stored as a geometry, through the *hasGeometry* property. The area value of the intersection is also stored through an *hasArea* property. Finally, comparing intersection areas with parent timeslice areas and child timeslice areas, we obtain a percentage in the range 0 - 100 representing the overlapping rate. This rate is defined by the properties *hasOverlappingParent* and *hasOverlappingChild*. More explanation about the calculation and the usefulness of these values are given in the following. As a result of a spatio-temporal transition analysis, we obtain all pairs of entities having an overlap. Some of them represents a filiation relationships while others should be considered as noise due to inaccuracies or negligible changes. Therefore, a methodology must be established to determine what relationships should be stored or eliminated.

4.4.2. Compute spatial filiation relationships

The simplest idea to retrieve the filiation relationships is to considere that child entities having an overlap with a parent entity should have a link with the latter one. Figure 5 depicts the spatial graph resulting from this approach. Figure 5 represents the same land cover evolution as depicted in Figure 4. However, this time each polygon is associated with a timeslice ($ts1, ts2 \ldots ts7$). Then, it is possible to compare timeslices of consecutive time points. In this example, we compare the timeslices corresponding to time $t1$ ($ts1, ts2$) with the timeslices corresponding to time $t2$ ($ts3, ts4$) to detect the evolution. Finally, we compare the timeslices of time $t2$ with the timeslices corresponding to time $t3$ ($ts5, ts6, ts7$).
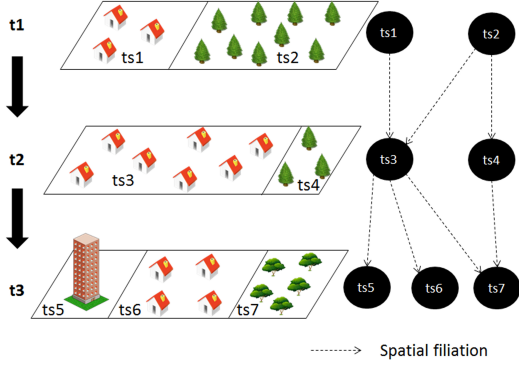
Figure 5: The computation of the spatial filiations

In this example, most of the spatial filiation seems correct, but it remains many cases where this naive approach is unsatisfactory. This weakness is highlighted between T2 and T3 as shown in Figure 6. The system retrieve a correct filiation between *ts4* and *ts7*, but we notice a very slight growth of *ts4* to become *ts7*. As a result, *ts3* has an overlap with *ts7* and a filiation is retrieve. Consequently, the system will considered that *ts4* has grown while *ts3* divide into *ts5*, *ts6* and also *ts7*. Although, these findings are with respect to the method used. They do not reflect the reality of the situation because the growth is too slight to be considered as significant. Ultimately, no spatial evolution should be take into account between *ts4* and *ts7* in respect of the reality. Accordingly, the filiation relationship between *ts3* and *ts7* is wrong.

Example depicted by Figure 6 is one of many cases that reflect the need for flexibility of the system when computing filiation relationships. Thus, there is a need for methods enabling user to define a threshold in order to filter spatial filiation.
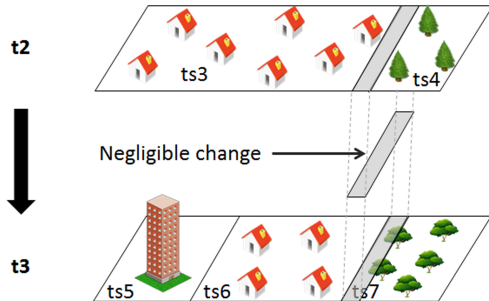

Figure 6: An example of a negligible change

4.4.3. Eliminate negligible change

To enhance the fiability of the retrieval process, we use also identity component considering that a spatial filiation occurring between two timeslices refering to the same class is more likely to be correct than between timeslices refering to different classes. In other word, continuation relationships are more likely to be correct than derivation based on the assumption that an entity having a spatio-temporal overlap and a same identity reflects more probability to be linked in time than any other cases. A second assumption considers that the most common area, the greater the likelihood that the entities are linked is high whether a continuation or derivation.

In Equation 27, we saw the definition of the concept *SpatialTemporalTransition*. The values of properties *hasOverlappingParent* and *hasOverlappingChild* give us information about the type of spatial relationships that exists between *parent* and *child* timeslices. The domain for both properties are double with values from 0 to 100. To calculate the values of these properties, we intersect the geometries of the timeslices and obtain a new geometry as shown in Equation 28.

$$timeSliceParent_{area} \bigcap timeSliceChild_{area} = Intersection_{area} \qquad (28)$$

Then, we compare the area of the resulting geometry against the areas of the parent and child timeslices. Equation 29 shows the procedure to calculate the value for the property *hasOverlappingChild*, while Equation 30 calculates the value for the property *hasOverlappingParent*.

$$\left(\frac{Intersection_{area}}{timeSliceChild_{area}} \right) * 100 = OverlappingChild \qquad (29)$$

$$\left(\frac{Intersection_{area}}{timeSliceParent_{area}} \right) * 100 = OverlappingParent \qquad (30)$$

The values of the properties *hasOverlappingParent* and *hasOverlappingChild* allow us to quantify how much of the geometry of the parent is part of the geometry of the child and vice versa. In an evolution, a child timeslice can have multiple parents, while the geometry of a parent can generate multiple children. But, some timeslice can have slight overlap that should not reveal a real filiation relationship. In our work, we propose the use of the properties *hasOverlappingParent* and *hasOverlappingChild* to determine when two timeslices should be associated by a filiation relationships.

15

Thus, we decided to accept a correct filiation relationships only when parent and child overlaps exceed a certain threshold. We defined two threshold value ρ and χ for which respectively *op* and *oc* must be greater in order to validate a correct filiation relationship. Thus, Equation 31 specifies a real filiation relationship constraint:

$$
\begin{aligned}
[\forall p \forall c (\mathcal{T}S(p) \wedge \mathcal{T}S(c) \wedge hasFiliation(p,c) \rightarrow \\
(\exists st(SpatialTemporalTransition(st) \wedge \\
hasParentTimeSlice(st,p) \wedge \\
hasChildTimeSlice(st,c) \wedge \\
hasOverlappingParent(p,op) \wedge \\
hasOverlappingChild(c,oc) \wedge \\
(OverlappingParent > \rho_{Filiation}) \wedge \\
(OverlappingChild > \chi_{Filiation}))))]
\end{aligned} \quad (31)
$$

with $op, oc \in \mathbb{Q}$

The same principle is applied to all relationships inherited from the filiation relationship. Equation 32 and 33 depict our proposed rules to determine if a spatial filiation should be accepted as continuation, derivation or rejected from the graph. The defined threshold can be adjusted by users depending on the desired degree of tolerance while determining filiation relationships. In this work, we defined two differents thresholds to distinguish continuation and derivation in respect of the fact that derivation detected are more error prone than continuation. Consequently, we decided to use a higher threshold in case of derivation. The values of $\rho_{Continuation}$ and $\chi_{Continuation}$ both have been set to 5 to detect *Continuation* while the values of $\rho_{Derivation}$ and $\chi_{Derivation}$ both have been set to 15 to detect *Derivation*. Table 2 show an example of correct filiation.

In order to present our implementation, in the remainder of this article, we will show the mathematical formulas and the query used for each relationships inherited from the filiation relationship. For this purpose, we use SPARQL 1.1 implemented on Stardog as the query language. More specifically, we use SPARQL 1.1 Update as it provides operations to update, insert, and remove RDF graphs in a Graph Store.

$$
\begin{aligned}
[\forall p \forall c (hasFiliation(p,c) \wedge isTimesliceOf(p,classP) \\
\wedge isTimesliceOf(c,classC) \wedge (classP = classC) \\
\rightarrow hasContinuation(p,c))]
\end{aligned} \quad (32)
$$

Code 2: A SPARQL Query to detect continuation relationship

```
insert
{
```

```
  ?p checksem:hasContinuation ?c
}
where
{
  ?st a checksem:SpatiotemporalTransition.
  ?st checksem:hasParentTimeSlice ?p.
  ?st checksem:hasChildTimeSlice ?c.
  ?p a ?landcoverP .
  ?c a ?landcoverC .
  ?p checksem:hasOverlappingParent ?rho.
  ?c checksem:hasOverlappingChild ?xhi.
  FILTER(?landcoverP=?landcoverC &&
      ?rho>5 && ?xhi>5)
}
```

$$
\begin{aligned}
[\forall p \forall c (hasFiliation(p,c) \wedge isTimesliceOf(p,classP) \\
\wedge isTimesliceOf(c,classC) \wedge (classP \neq classC) \\
\rightarrow hasDerivation(p,c))]
\end{aligned} \quad (33)
$$

Code 3: A SPARQL Query to detect derivation relationship

```
insert
{
  ?p checksem:hasDerivation ?c
}
where
{
  ?st a checksem:SpatiotemporalTransition.
  ?st checksem:hasParentTimeSlice ?p.
  ?st checksem:hasChildTimeSlice ?c.
  ?p a ?landcoverP .
  ?c a ?landcoverC .
  ?f checksem:hasOverlappingParent ?rho.
  ?f checksem:hasOverlappingChild ?xhi.
  FILTER(?landcoverP != ?landcoverC &&
      ?rho>15 && ?xhi>15)
}
```

Table 2: Filiation relationships between timeslices

Time	Parent	Child	Overlapping Parent	Overlapping Child	Filiation
t1 to t2	ts1	ts3	100	45	TRUE
	ts2	ts3	70	55	TRUE
	ts2	ts4	30	100	TRUE
t2 to t3	ts3	ts5	40	100	TRUE
	ts3	ts6	60	100	TRUE
	ts3	ts7	2	4	FALSE
	ts4	ts7	100	96	TRUE

Using the values of *hasOverlappingParent* and *hasOverlappingChild* with the rule in Equation 32 and 33, we obtain the graph depicted by Figure 7 composed from continuation and derivation.

Finally, only one spatial filiation has not been retain thresholding the spatial graph as depicted by Figure 8.

5. Knowledge discovery

At this step, we obtain correct filiation relationships which can serve as basics to compute more complexe
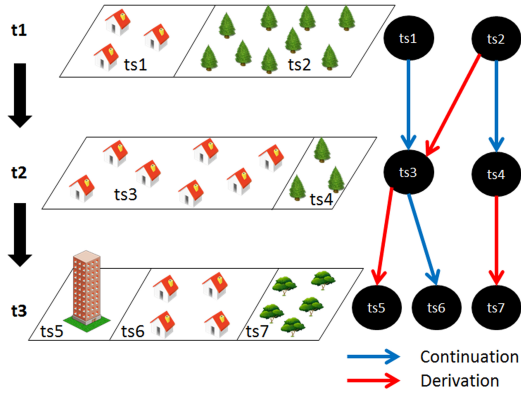
Figure 7: Continuation and derivation relationships after the removal of negligible changes
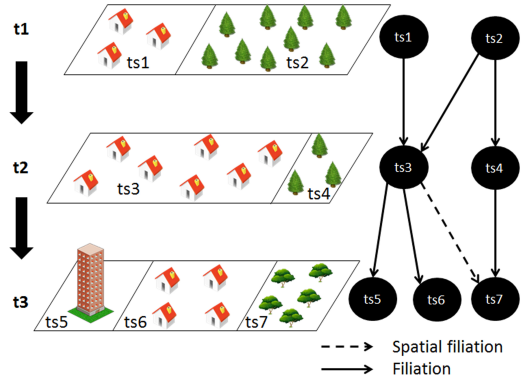

Figure 8: Filiation graph in respect of reality

Table 3: Summary of areas for each CORINE Land Cover class, at level 3.

	Area 1990 (Km^2)	Area 2000 (Km^2)	Area 2006 (Km^2)	% Change 1990 to 2000	% Change 2000 to 2006
BeachDune	30.8	61.1	60.6	98.4	-0.8
ConiferousForest	7366.4	7862.5	6187.6	6.7	-21.3
Vineyards	1395	1481.2	1558.4	6.2	5.2
NaturalGrassLands	15.3	112.7	20.2	636.6	-82.1
Pastures	597.1	644	658.1	7.9	2
Ports	1.6	1.6	2.9	0	81.3
UrbanGreen	10.3	10	8.3	-2.9	-17
Marshes	49.6	51.2	61.5	3.2	20.1
Salines	4.7	4.7	0	0	-100
IndCommercial	71.5	83.7	87	17.1	3.9
BroadLeavedForest	434.5	639	642.2	47.1	0.5
FruitPlantation	14.8	16	15.7	8.1	-1.9
VegMoors	1.8	4.2	121.7	133.3	2797.6
Estuaries	380.2	380.2	389.2	0	2.4
WaterCourses	250.8	250.8	251	0	0.1
ShrubWoodland	513.7	665.9	1830.5	29.6	174.9
Airports	6.9	9.5	11.4	37.7	20
IntertidalFlats	114.9	136.2	136.7	18.5	0.4
ArableNonIrrigated	2163.6	2492.2	2087.3	15.2	-16.2
UrbanContinuous	25.4	26.2	23.6	3.1	-9.9
RoadsRails	7.3	7.7	8.5	5.5	10.4
Construction	0.6	9.2	0	1433.3	-100
SeasOceans	24042.6	24046.4	24046.7	0	0
Sports	28.5	35.2	36.3	23.5	3.1
Mines	16.2	23.6	24.5	45.7	3.8
UrbanDiscontinuous	397.3	483.8	513.4	21.8	6.1
AgricAndNatural	72.3	109.9	115.5	52	5.1
ComplexCultivation	853.1	1018.9	1126.8	19.4	10.6
InlandWaterBodies	80.6	139.2	143.8	72.7	3.3
MixedForest	432.9	622.1	631.6	43.7	1.5
SaltMarshes	30.8	31.3	36.8	1.6	17.6
Dumps	0	0	1.1	0	0
ArablePermIrrigated	0	0	10.5	0	0

analysis. To this end, to increase knowledge about evolution become an essential. Two complementary ways allows studying evolution: quantitative and qualitative. The former reveals how much land territory have been converted between two different time point while the latter try to understand how depth is the change.

5.1. Determine rate of change

A first way to study evolution area consists in evaluate changes on each land cover. Table 3 summaries computations of areas using CORINE land cover classes at level 3.

Two kinds of information can be retrieved from this trivial analysis. On one hand, we easily detect a gain or loss for each the land cover. On the other hand, we can determine how much a gain or loss occurs between two time points. But, this approach does not

fully explain the reality of an evolution. First, the results obtained only describe the evolution of an area at a global scale excluding any more detailed analysis. Second, the area studied can be seen as a unique screenshot composed from multiple timeslices each belonging to a land cover class. As a result, the gain or loss of a certain land cover respectively correspond to a loss or gain of another one. Thus, information and knowledge are missing to understand causality between timeslices implied into a land cover evolution. Finally, a more detailled analysis makes it possible to precisely localize specific phenomena within a region rather than recording a global trend. To switch from a global scale to a detailed scale, we try to find the evolution of each timeslice on the basis of filiation relationships.

5.2. Qualify filiation relationships

Table 3 illustrates a global approach to detect changes on land cover but fails to fully explain it. To overcome these weaknesses, we work at a more detailed level using filiation relationships. Filiation relationships can be either continuation or derivation. Then, continuation

and derivation can involve in the evolution of a single entity (1-1 relationship) or in spatial structures involving several entities (0-n or n-0 relationships) like merging and division patterns. In the case of single entity, a continuation could correspond to a growth, reduction or equality while a derivation could correspond to a conversion. In this work, we call it a single continuation and a single derivation. In case of several entities, we distinguish four different patterns: split, fusion, annexation, separation.

5.2.1. Single continuation

Continuations are those relationships for which identity remains unchanged between two consecutive time points. We distinguish three different kinds of continuation refering to growth, reduction or equality that we need to detect to enhance knowledge about evolution. Prima facie, detecting theses continuations is a trivial process. Given two overlapping timeslices *ts1* and *ts2* respectively corresponding to two consecutive time points, then based on spatial analysis:

- Growth: *ts1* is within *ts2*

- Reduction: *ts1* contains *ts2*

- Equal: *ts1* is equal *ts2*

Two problems arise from the implementation of this analysis. The first one concerns the time to perform spatial analysis for each filiation relationships of the dataset. The second is due once again to negligible changes that should be excluded from the analysis result. Examples are slight growth or reduction which should not be construed as such or slight modification of the timeslice boundary in the case of equality which should be construed as such. Once again, a threshold is needed in order to deal with theses vaguenesses. In our approach, we decided to use the values of overlapping parent and overlapping child. The first issue raised previously implied to find a way to determine spatial operation like *contains*, *within* or *equal*. Traditional spatial operations are able to solve the problem, but are time-consuming while overlapping values allow a faster calculation for theses operations. An overlapping value defined to 0 determines disjointness whereas a 100 value leads to three distinct options:

- A child timeslice is within parent timeslice

- A child timeslice contains parent timeslice

- A child timeslice is equal to parent timeslice

To distinguish each case, it is necessary to compare child and parent overlapping value. Thus, if an overlapping parent value is 100 and an overlapping child value is lower than 100, then we can infer a growth between the two timeslices linked by a continuation filiation relationships. Similarly, if an overlapping child value is 100 and an overlapping parent value is lower than 100 then, we can infer a reduction. Finally, if an overlapping parent and an overlapping child values are 100, then we can infer an equality. In a theoretical way, this method is valid but in practice, it does not deal with negligible changes in order to reveal much more realistic informations. For example, a continuation with an overlapping parent value defined at 100 and an overlapping child value defined at 97 will be interpreted as a growth while common sense rather denote an equality. To address this second issue, we define threshold defining a minimum rate of growth or reduction and an acceptable gap between the two values in case of equality.

Figure 9 depicts different examples of wrong and correct situations.

In order to define a minimum rate of growth or reduction we calculate the difference between parent and child overlapping values and retains only relationships for which the results obtained exceed a certain threshold φ as shown in Equation 34 and 35. In case of equality, only relationships for which overlapping child and overlapping parent are greater than or equal to $100 - \varphi$ as shown in Equation 36. In this work, the value of φ has been set to 10.

$$[\forall p \forall c (hasContinuation(p,c) \wedge hasOverlappingParent(p,op) \wedge \\ hasOverlappingChild(c,oc) \wedge (op = 100) \wedge (op > oc) \wedge \\ ((op - oc) > \varphi) \rightarrow hasGrowth(p,c))] \quad (34)$$

$$[\forall p \forall c (hasContinuation(p,c) \wedge hasOverlappingParent(p,op) \wedge \\ hasOverlappingChild(c,oc) \wedge (oc = 100) \wedge (oc > op) \wedge \\ ((oc - op) > \varphi) \rightarrow hasReduction(p,c))] \quad (35)$$

$$[\forall p \forall c (hasContinuation(p,c) \wedge hasOverlappingParent(p,op) \wedge \\ hasOverlappingChild(c,oc) \wedge (oc \geq (100 - \varphi)) \wedge \\ (op \geq (100 - \varphi)) \rightarrow hasEquality(p,c))] \quad (36)$$

Logically, equations 34, 35 and 36 are expressed as follows in SPARQL:

Code 4: A SPARQL Query to detect a hasGrowth relationship

```
insert
{
    ?p checksem:hasGrowth ?c
}
where
{
```
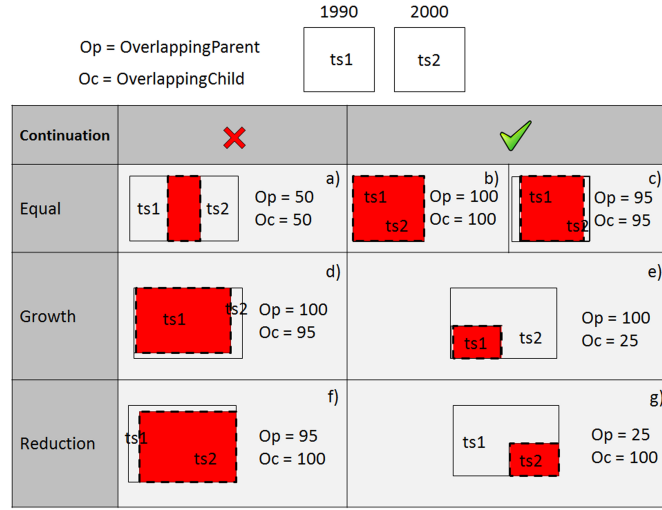
Figure 9: The detection and qualification of continuation relationships

```
  ?p checksem:hasContinuation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Op=100 && ?Op>?Oc && (?Op−?Oc)>10)
}
```

Code 5: A SPARQL Query to detect a hasReduction relationship

```
insert
{
  ?p checksem:hasReduction ?c
}
where
{
  ?p checksem:hasContinuation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Oc=100 && ?Oc>?Op && (?Oc−?Op)>10)
}
```

Code 6: A SPARQL Query to detect a hasEquality relationship

```
insert
{
  ?p checksem:hasEquality ?c
}
where
{
  ?p checksem:hasContinuation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Op>90 && ?Oc>90)
}
```

5.2.2. A single derivation

Derivations are those relationships for which identity varies between two consecutive time points. In our model, we define only one single derivation corresponding to a total landcover change of a unique timeslice

called *Conversion*. This kind of relation is very similar to the *Equality* relationship except that the identity child timeslice is different from the parent one. Then, the equation to infer *Conversion* relationship using overlapping values is:

$$[\forall p \forall c (hasDerivation(p,c) \wedge hasOverlappingParent(p,op) \wedge \\ hasOverlappingChild(c,oc) \wedge (oc \geq (100 - \varphi)) \wedge \\ (op \geq (100 - \varphi)) \rightarrow hasConversion(p,c))] \quad (37)$$

Code 7: A SPARQL Query to detect a hasConversion relationship

```
insert
{
  ?p checksem:hasConversion ?c
}
where
{
  ?p checksem:hasDerivation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Op>90 && ?Oc>90)
}
```

5.2.3. Complex pattern

In our approach, some other patterns can be detected based on multiple filiation relationships. Theses complex patterns correspond to a division and a merge. A division or a merge can involve one continuation and at least one derivation, or it can involve only derivations giving rise to four main complex patterns: splits, fusion, annexation, separation. In order to describe our patterns, we introduce another operator *Sum* to sum overlapping values.

19

$$Sum^I = \{\langle val, val' \rangle \mid val + val' = res, val, val', res \in \mathbb{Q}\} \quad (38)$$

$$[\forall p \forall c \forall c'(hasContinuation(p,c) \wedge hasDerivation(p,c') \wedge$$
$$hasOverlappingParent(p,op) \wedge hasOverlappingChild(c,oc) \wedge$$
$$hasOverlappingChild(c',oc') \wedge (op = 100) \wedge (Sum(oc,oc') = 100) \quad (39)$$
$$\rightarrow (hasSeparation(p,c) \wedge PartOfSeparation(p,c')))]$$

$$[\forall p \forall p' \forall c(hasContinuation(p,c) \wedge hasDerivation(p',c) \wedge$$
$$hasOverlappingParent(p,op) \wedge hasOverlappingChild(p',op') \wedge$$
$$hasOverlappingChild(c,oc) \wedge (oc = 100) \wedge (Sum(op,op') = 100) \quad (40)$$
$$\rightarrow (hasAnnexation(p,c) \wedge PartOfAnnexation(p',c)))]$$

$$[\forall p \forall c \forall c'(hasDerivation(p,c) \wedge hasDerivation(p,c') \wedge$$
$$hasOverlappingParent(p,op) \wedge hasOverlappingChild(c,oc) \wedge$$
$$hasOverlappingChild(c',oc') \wedge (op = 100) \wedge (Sum(oc,oc') = 100) \quad (41)$$
$$\rightarrow (hasSplits(p,c) \wedge hasSplits(p,c')))]$$

$$[\forall p \forall p' \forall c(hasDerivation(p,c) \wedge hasDerivation(p',c) \wedge$$
$$hasOverlappingParent(p,op) \wedge hasOverlappingChild(p',op') \wedge$$
$$hasOverlappingChild(c,oc) \wedge (oc = 100) \wedge (Sum(op,op') = 100) \quad (42)$$
$$\rightarrow (hasMerging(p,c) \wedge hasMerging(p',c)))]$$

In order to detect complex patterns in the graph, we defined two steps. The first one focuses only on the spatial component of timeslices by searching division or merging pattern. Then, the second step relies on the identity component in order to distinguish a merge from an annexation or a splitting from a separation. Notice that all could be done in one big step, but two steps allow us to explain more clearly how our system is implemented.

To perform our first step, we decided to use and insert temporary relations into our triple store representing division and merging patterns based on overlapping values. The following queries show how these relationships were computed using informations in the triple store.

Code 8: SPARQL Query to detect TemporaryMerger relationship

```
insert
{
  ?p checksem:TemporaryMerger ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
    (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?st a checksem:SpatiotemporalTransition.
      ?st checksem:hasParentTimeSlice ?p.
      ?st checksem:hasChildTimeSlice ?c.
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Oc=100)
    }
    group by ?c
  }
  FILTER((?SumOp=100)&&(?countC>1))
}
```

To calculate the patterns of merge, the main idea is to have a child entity that contains all his parents in its geometry. In other words, we look for all the filiation relationships involving the same child entity and multiple parent entities. Moreover, the sum of parents overlaps for all these relationships must be equal to 100 while the child overlap must be worth 100 for each one of them.

Code 9: SPARQL Query to detect TemporaryDivision relationship

```
insert
{
  ?p checksem:TemporaryDivision ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
    (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?st a checksem:SpatiotemporalTransition.
      ?st checksem:hasParentTimeSlice ?p.
      ?st checksem:hasChildTimeSlice ?c.
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Op=100)
    }
    group by ?c
  }
  FILTER((?SumOc=100)&&(?countP>1))
}
```

Conversely, to calculate the patterns of division, the main idea is to have a parent entity that contains all its childs in its geometry. In other words, we look for all the filiation relationships involving the same parent entity and multiple child entities. Moreover, the sum of child overlaps for all these relationships must be equal to 100 while the parent overlap must be worth 100 for each one of them.

After computing the first step, we focus on the identity component to complete the previous queries and distinguish complex patterns. To add identity information in our query, we use timeslice classes stored in our ontology. Each class represents the identity of timeslice and correspond to a specific landcover defined in the Corine Land Cover dataset. Initially, we focus on the fusion and split patterns, because they are composed entirely of derivation relationships which simplifies the query compared to separation and annexation patterns. The following queries show how to integrate the notion of identity in the query.

Code 10: A SPARQL Query to detect a hasFusion relationship

```
insert
{
  ?p checksem:hasFusion ?c
}
where
{
```

```
{
  select ?c (sum(?Oc) as ?SumOc)
  (count(?p) as ?countP) (sum(?Op) as ?sumOp)
  where{
    ?p checksem:TemporaryMerger ?c .
    ?p a ?landcoverP .
    ?c a ?landcoverC .
    ?p checksem:hasOverlappingParent ?Op.
    ?c checksem:hasOverlappingChild ?Oc.
    FILTER(?Oc=100 && ?landcoverP!=?landcoverC)
  }
  group by ?c
}
FILTER((?SumOp=100)&&(?countC>1))
}
```

Code 11: A SPARQL Query to detect a hasSplit relationship

```
insert
{
  ?p checksem:hasSplit ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
    (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?p checksem:TemporaryDivision ?c .
      ?p a ?landcoverP .
      ?c a ?landcoverC .
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Op=100 && ?landcoverP!=?landcoverC)
    }
    group by ?c
  }
  FILTER(((?SumOc)=100)&&(?countC>1))
}
```

In both cases, we simply compare the identity of the parent timeslice with child timeslices, and we look for relationships where the identity is different. In cases of annexation and separation, we need to detect two different relationships. One corresponding to a continuation and the other corresponding to a derivation. Thus, the query must browse the graph of our triple store trying to find patterns of a division or a merge involving at least one continuation and one derivation. The following queries show how to find the relationships related to separation and annexation patterns respecting the constraints defined.

Code 12: A SPARQL Query to detect a hasAnnexation and a PartOfAnnexation relationship

```
insert
{
  ?p1 checksem:hasAnnexation ?c .
  ?p2 checksem:PartOfSeparation ?c .
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
    (count(?p) as ?countP) (sum(?Op1) as ?sumOp1)
    (sum(?Op2) as ?SumOp2)
```

```
  where{
    ?p1 checksem:TemporaryDivision ?c .
    ?p2 checksem:TemporaryDivision ?c .
    ?p1 a ?landcoverP1 .
    ?p2 a ?landcoverP2 .
    ?c a ?landcoverC .
    ?p1 checksem:hasOverlappingParent ?Op1.
    ?p2 checksem:hasOverlappingParent ?Op2.
    ?c checksem:hasOverlappingChild ?Oc.
    FILTER(?Op=100 && ?landcoverP1=?landcoverC &&
    ?landcoverP2!=?landcoverC2 )
  }
  group by ?c
}
FILTER(((?SumOp1+?SumOp2)=100)&&(?countC>1))
}
```

Code 13: A SPARQL Query to detect a hasSeparation and a PartOf-Separation relationship

```
insert
{
  ?p checksem:hasSeparation ?c1 .
  ?p checksem:PartOfSeparation ?c2 .
}
where
{
  {
    select ?c1 ?c2 (sum(?Oc1) as ?SumOc1)
    (sum(?Oc2) as ?SumOc2)
    (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?p checksem:TemporaryDivision ?c1 .
      ?p checksem:TemporaryDivision ?c2 .
      ?p a ?landcoverP .
      ?c1 a ?landcoverC1 .
      ?c2 a ?landcoverC2 .
      ?p checksem:hasOverlappingParent ?Op.
      ?c1 checksem:hasOverlappingChild ?Oc1.
      ?c2 checksem:hasOverlappingChild ?Oc2.
      FILTER(?Op=100 && ?landcoverP=?landcoverC1
      && ?landcoverP!=?landcoverC2)
    }
    group by ?c1 ?c2
  }
  FILTER(((?SumOc1+?SumOc2)=100)&&(?countC>1))
}
```

Figure 10 depict the result of the pattern analysis. Between T1 and T2, the process has retrieved an annexation involving *ts1*, *ts2* and *ts3* and a separation involving *ts2*, *ts3* and *ts4* pattern while between T2 and T3, the process has retrieved a separation involving *ts3*, *ts5* and *ts6* and a conversion involving *ts4* and *ts7* pattern.

5.3. How deep is change

Derivations are of a particular importance in case of an annexation and a separation. Actually, we consider that when an entity is annexed, the new area acquired replaces one or more other entities with a different land cover. Conversely, when an entity shrinks, a part of the previous area is replaced by one or more other entities. As a result, annexations and separations are inevitably related to one or more derivations and theses derivations can help for a better understanding of the
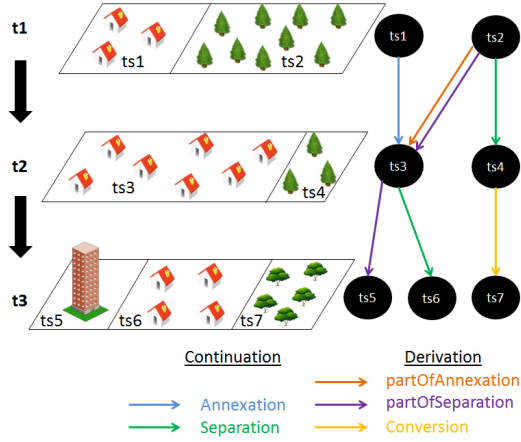
Figure 10: Result of pattern analysis

evolution phenomena. For example, if we observe that the forest areas frequently experience reductions in their territory, and land cover in place are *UrbanContinuous*, then we may infer that deforestation is strongly related to urbanisation. In this example, the superseded area has undergone a strong land cover change, because both old and new land cover are very different. But, sometimes land cover changes may be soft. For example, when a coniferous forest land cover become a mixed forest land cover, semantic of both land cover remains very similar. Consequently, if *coniferous forest* experiences a separation because a part of its area became a *mixed forest*, this information is correct only if user study the evolution of different kinds of forests. In addition, if user's field study is more general, for instance the study of forest whatever the kind of forest, the information is not correct because the forest land cover did not experienced any separation regarding the field of study. Previous examples highlights different semantic level to study a land area. In this work, we naturally use the Corine Land Cover hierarchy modeled with an ontology in order to deal with theses semantics granularities.

The depth of a change that can be analyzed depends entirely on the depth of the landcover hierarchy of classification as depicted in Figure 11. As the Corine Land Cover dataset offers three levels of classification, our system is capable of detecting three levels of change. As a result, we distinguish three kinds of derivation. For a clearer understanding, we called them: weak, medium and strong for transcribing the depth of change.

Figure 12 depicts the result of the analysis to increase knowledge about the graph.

Using depth information, it is interesting to note that patterns can lead to very different interpretations. In our example, we detected two separation patterns, one annexation pattern and one conversion pattern. The first separation pattern occurs between t1 and t2 with a part of *Coniferous forest* converting into *Continuous urban fabric*. This derivation reveals a strong change as it involves a change in the class at level 1 of the Corine Land Cover hierarchy. Thus, this forest land parcel loss might be interpreted as a part of a deforestation process due to an urban sprawl. Now, considering the second separation pattern where a part of *Continuous urban fabric* became *Industrial or commercial units*, one could thinks of urban loss because of *Continuous urban fabric* landcover reduction. But, analysing depth of change, we realize that this derivation is moderate which means an *urban* area that has been replaced by a semantically closed landcover. Therefore, contrary to the first case, this change gives a reason to suppose the existence of a phenomenon submitted in a logical evolutionary process. The most plausible hypothesis leads us to believe that this is a Urban intensification process. Figure 13 depicts the result of the analysis.

Table 4 shows an example of analysis that can be performed using the depth of a change. Here, we computed the number of timeslices and the surface affected by conversion pattern. Moreover, all pre-changed landcovers and post-changed landcovers are given. Finally, each change is also distributed according to whether it is weak, medium or strong.

In this example, we rapidly notice a strange phenomenon of conversion from *ArtificialSurfaces* to *WaterBodies* resulting from an *Urban flooding* risk management in this highly affected area of Gironde [43], [44] mainly because of the frequent flooding of the Garonne river that crosses this region. Figure 14 shows that we can easily confirm this theory using our application.

5.4. Trend patterns of evolution

Evolution can be summarized through different patterns that appear frequently between two consecutive instants. But, evolution understanding is not limited to occasional patterns occurring only between two consecutive times. These fields of study should also be extended to the recurrence of these patterns over time. For example, a forest landcover undergoing *reduction* of its area between a time *t*1 and a time *t*2 may be an isolated event for which an adjustment in policy land use is not necessarily required. Now, considering the same landcover experiencing a second *reduction* of its
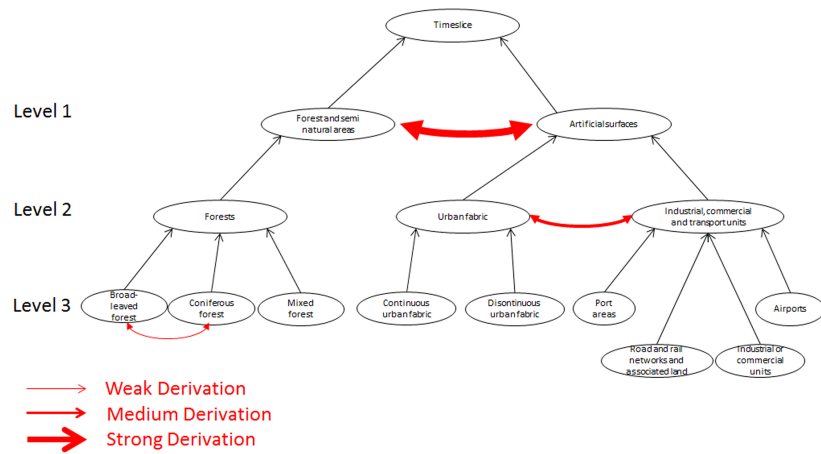
22

Figure 11: An example of three kinds of derivation: weak, medium and strong derivation
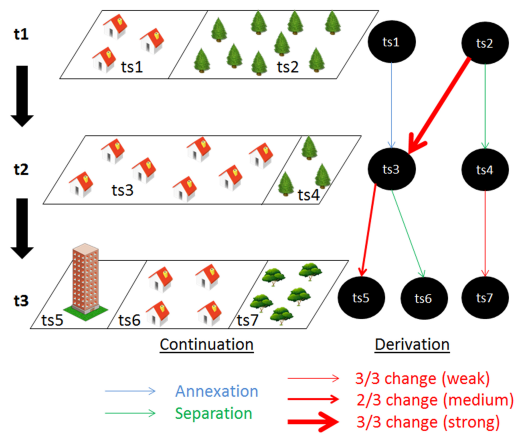

Figure 12: Detection of patterns based on the taxonomy analysis
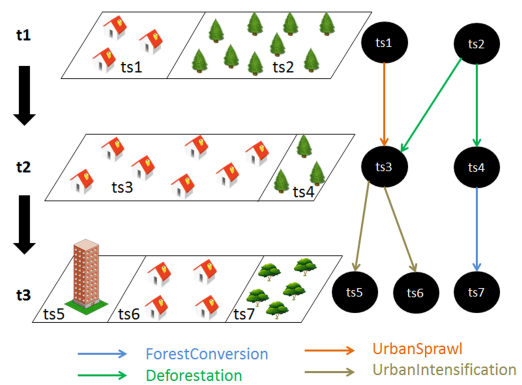

Figure 13: Interpretation of phenomena

Figure 14: Timeslices transformed from Artificial Surfaces to Waterbodies along the Garonne

Table 4: Conversions

(weak) Conversion	1990 to 2000		2000 to 2006	
	Nb	Area (dam^2)	Nb	Area (dam^2)
Construction to Dumps	-	-	2	8338
NaturalGrassLands to VegMoors	-	-	1	902162
NaturalGrassLands to ShrubWoodland	-	-	1	3189
ArableNonirrigated to ArablePermIrrigated	-	-	2	105443
ConiferousForest to MixedForest	-	-	1	37667
AgricAndNatural to complexCultivation	-	-	1	4336
IndCommercial to Ports	-	-	2	13008
Mines to Dumps	-	-	1	2731
(medium) Conversion	**1990 to 2000**		**2000 to 2006**	
	Nb	Area (dam^2)	Nb	Area (dam^2)
Forest to ShrubVegetation	5	22403	10	133433
ShrubVegetation to Forest	23	113225	-	-
ArableLand to HeterogenousAgric	1	2607	5	22885
Pasture to HeterogeneousAgric	-	-	11	69725
HeterogeneousAgric to PermanentCrops	-	-	1	5374
MinesDumpConstruction to IndustrialCommercialTransportation	-	-	3	8501
ArableLand to pasture	-	-	7	54473
MinesDumpConstruction to ArtificialVegetated	-	-	1	2965
ArableLand to PermanentCrops	-	-	1	2516
HeterogeneousAgric to Pasture	-	-	3	13392
(Strong) Conversion	**1990 to 2000**		**2000 to 2006**	
	Nb	Area (dam^2)	Nb	Area (dam^2)
ForestSemiNaturalAreas to AgriculturalAreas	1	19080	2	11962
AgriculturalAreas to ArtificialSurfaces	2	6871	4	18761
ArtificialSurfaces to WaterBodies	**2**	**7600**	**3**	**17248**
ArtificialSurfaces to AgriculturalAreas	-	-	1	4000
ForestSemiNaturalAreas to ArtificialSurfaces	-	-	1	3865
AgriculturalAreas to ForestSemiNaturalAreas	-	-	2	16255

area between a time *t*2 and a time *t*3, the hypothesis of an isolated event becomes very questionable. In this work, we called *trend patterns* the study of phenomena based on the recurrence of occasional patterns. Detecting *trend patterns* can be summarized as seeking paths in a graph.

As our continuum model is a graph-based model using semantic web technologies, we would introduce SPARQL Query as the easiest way to detect *trend pattern*. Specifically, we use a particular tool from SPARQL Query called *Property Path*. A *Property Path* is a possible route through a graph between two graph nodes. A trivial case is a property path of length exactly 1, which is a triple pattern. *Property Paths* allow for more concise expressions for some SPARQL basic graph patterns and they add also the ability to match connectivity of two resources by an arbitrary length path. Here, we present some examples to introduce *Property Path* capabilities:

Alternatives. Match one or both possibilities

```
{
ck:TSUrban1990 ck:hasConversion|ck:PartOfSeparation ?TS2000
}
```

The query gives all timeslices for which identity was Urban landcover and have undergone a phenomenon of a Conversion or a Separation. As a result, we can easily know to which covers the timeslices tend to evolve after a Conversion or a Separation.

Sequence. Find the landcovers of any timeslices for which the landcover type was *Urban Discontinuous* before undergoing a Conversion.
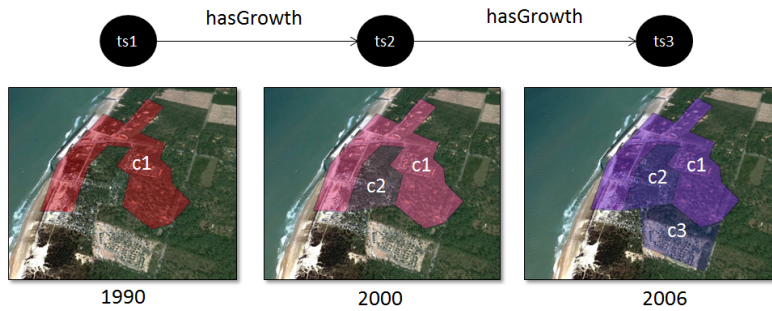
24

Figure 15: Example of a Trend pattern

```
{
  ?TS1990 rdf:type ck:UrbanDiscontinuous .
  ?TS1990 ck:hasConversion/rdf:type ?LandCover2000 .
}
```

Sequence. Find all timeslice landcovers that are neighbors of neighbors of the timeslice *timeslice_urban1*.

```
{
  ck:timeslice_urban1 ck:hasAdjacency/ck:hasAdjacency [...
  ...]/rdf:type ?LandCover .
}
```

Filtering duplicates. Because checksem:hasAdjacency is a symmetric property, the example above may include the timeslice *timeslice_urban1* itself as a result. This could be avoided with:

```
{
  ck:timeslice_urban1 ck:hasAdjacency/ck:hasAdjacency ?ts.
  FILTER ( ck:timeslice_urban1 != ?ts )
  ?ts rdf:type ?LandCover
}
```

Inverse Property Paths. These two query are equivalent: the second reverses the query direction which swaps the roles of subject and object.

```
{ ?ts rdf:type ?LandCover }
{ ?LandCover ^rdf:type ?ts }
```

Inverse Path Sequence. Find all the timeslices which have been annexed.

```
{
 ?x ck:hasAnnexation/^ck:PartOfAnnexation ?y .
 FILTER(?x != ?y)
}
```

which is equivalent to (?gen1 is a system generated variable):

```
{
 ?x ck:hasAnnexation ?gen1 .
 ?y ck:PartOfAnnexation ?gen1 .
 FILTER(?x != ?y)
}
```

Arbitrary length match. Find the timeslices that can be reached from *checksem:timeslice_urban1* by *checksem:hasGrowth*:

```
{
ck:timeslice_urban1 rdf:type ck:UrbanContinuous .
ck:timeslice_urban1 ck:hasGrowth+ ?ts .
}
```

Alternatives in an arbitrary length path. Find all timeslices that can be reached from *checksem:timeslice_urban1* by *checksem:hasGrowth* or *checksem:hasAnnexation*:

```
{
ck:timeslice_urban1 (ck:hasGrowth|ck:hasAnnexation)+ ?ts
}
```

Arbitrary length path match. Some forms of limited inference are possible as well. For example, for RDFS, all types and supertypes of a resource:

```
{ <http://example/thing> rdf:type/rdfs:subClassOf* ?type }
```

Subproperty. All subProperty from the dataset

```
{
?x ?p ?v .
?p rdfs:subPropertyOf* :property .
}
```

Negated Property Paths. Find nodes connected but not by rdf:type (either way round):

```
{ ?x !(rdf:type|^rdf:type) ?y }
```

5.5. The neighborhood effects on landparcel development

So far, our understanding about evolutions only deals with landparcel independently of each other. As a result, we are able to extract knowledge about landparcel life-cycle answering the questions: *when, what, where and how is a change*? However, some interactions may occur between different landparcels especially when studying neighboring landparcels. Generally speaking,

the study of the environment of a spatio-temporal entity opens the way towards a new answer which is: *why the change occurred*?. In this work, we are limited to landparcel data so the analysis of the causes of the change is restricted to the study of neighboring landparcel as a proof of concept. Figure 16 depicts an example of interaction between neighboring landparcel. In this case study, we focus on three landparcel each having a different identity of each other which are *Road Rails*, *Industrial or Commercial Units*, and *Agricultural and Natural Vegetation* according to Corine Land Cover dataset. Obviously, the case study is represented at the three time points of our dataset: 1990, 2000, 2006. Both the graph stored in our triplestore and the screenshot we obtained from our system are displayed in the Figure 16.

Focusing on the graph, we have three timeslices *R*1990, *Ind*1990 and *Ag*1990 belonging respectively to classes *Road Rails*, *Industrial or Commercial Units*, and *Agricultural and Natural Vegetation* at the begining of the process. We notice that *Ind*1990 and *Ag*1990 are connected by an adjacency relationships while *R*1990 is disjoint from them. Between 1990 and 2000, timeslices *R*1990, *Ind*1990 and *Ag*1990 are connected respectively to *R*2000, *Ind*2000 and *Ag*2000 by a continuation relationship. More specifically, *R*1990 is connected to *R*2000 through a *hasGrowth* relationship while *Ind*1990 and *Ag*1990 are connected respectively to *Ind*2000 and *Ag*2000 by a *hasEquality* relationship. If the adjacency relationship remains logically between *Ind*2000 and *Ag*2000, we notice that a new one appeared between *Ind*2000 and *R*2000. Then, between 2000 and 2006, timeslices *R*2000, *Ind*2000 and *Ag*2000 are connected respectively to *R*2006, *Ind*2006 and *Ag*2006 by a continuation relationship. This time, *R*2000 is connected to *R*2006 through a *hasEquality* relationship while *Ind*2000 and *Ag*2000 are connected respectively to *Ind*2006 and *Ag*2006 by a *hasGrowth* a *hasReduction* relationship.

Based on this knowledge graph, it is possible to give an interpretation to explain the situation. According to images below the graph, we notice that between 1990 and 2000 a road has been extended to connect the industrial or commercial landparcel. As the industrial or commercial landparcel is more accessible to consumers, it then developed by expanding its infrastructure giving rise to an urbanization process. However, because it was connected to an Agricultural and natural vegetation landparcel, this latter suffers the consequences of the urbanization process and sees its territory reduced. Meanwhile, the lost territory is replaced by Urban Continuous landparcel. The following query helps to find neighbor-hood effects involving growth or reduction (other patterns can be defined) after the expansion of a road:

Code 14: A SPARQL Query to detect growing and reducing neighborhood effects

```
SELECT ?tsNeighbour ?LandCover
WHERE
{
    ?ts rdf:type checksem:Road_Rails .
    ?ts checksem:hasGrowth ?ts2 .
    ?ts2 checksem:hasAdjacency/(checksem:hasGrowth|
    checksem:hasReduction) ?tsNeighbour.
    ?tsNeighbour rdf:type ?LandCover .
}
```

6. Conclusions

We have presented in this paper a pragmatic model and method to analyze Land Cover Change using a semantic and perdurantism approach. The Land Cover Change Continuum (L3C) model allows the modeling of spatial datasets in a triplestore. The knowledge about the spatial entities and their environments are also modeled in the ontology using a class hierarchy. In addition, the spatio-temporal transitions model the filiation relationships in order to track the identity of landparcels. This model is applied to a LULCC case study using CORINE Land Cover datasets. Unlike most of the spatio-temporal domain models using a relational database, our model is based on Semantic Web technologies which gives it the advantage of going farther in the modeling and analysis of the geospatial environment context. As a result, the machine is able to move closer to the expert knowledge giving an interpretation of informations resulting from the analysis. The understanding about the dynamics of the phenomena is thereby improved compared with other models in the literature which are confined to only give meaning to the data. The LC3 model defines spatio-temporal layers and spatio-temporal transitions to represent spatio-temporal dynamics. Each transition correspond to a filiation relationship and links two consecutives layers. In other words, it allows to follow the different evolution and the life cycle of an entity. Moreover, it carries the knowledge related to the phenomenon that made an entity changing from one state to another state. Most of the time, filiation relationships are missing in datasets. Thus, some pre-treatments have to be perform in order to populate the model and check consistency of system.

After the populating of the ontology, we were able to query and infer knowledge about the CORINE datasets over three time stamps. Results demonstrated that the model does perform what it was conceived for and increase our understanding of spatio-temporal dynamics.
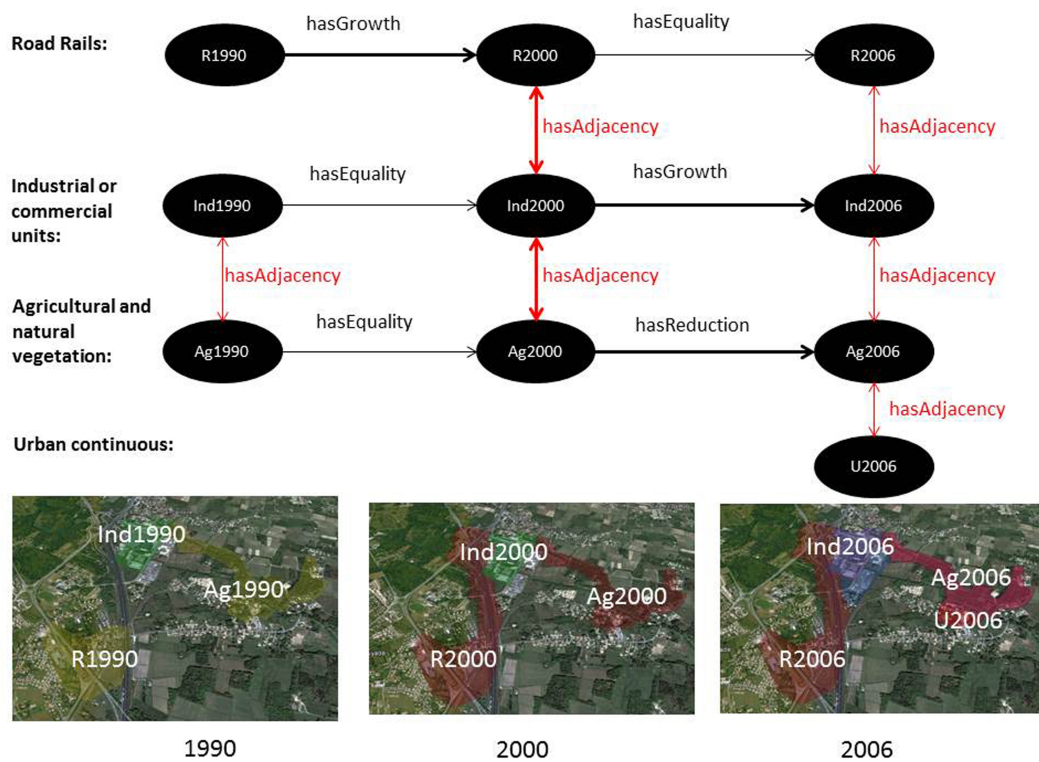
Figure 16: Detection neighborhood effects on a landparcel development

As an initial objective, our model permit to obtain parcel centered statistics. Future work should focus on the integration of complementary entities like cities, coasts, infrastructures, roads. The proximity of these items should impact the spatial-temporal transitions and the filiation relationships. For instance, the model should be able to infer the impact of a new highway between two cities, or the influence of a city in neighbor rural areas. Moreover, we would like to go further in the analysis of the influence of neighbors in the land cover change process by taking into consideration the adjacency relations between different kind of entities, not only landparcel.

In this work, uncertainty is address only to deal with the filiation calculation process as Corine Land Cover dataset do not allow to retrieve continuous transitions between two consecutives instants. The method detailed in the article define threshold that allow treating this uncertainty. But uncertainty can also exist while inferring phenomena on the basis of rules. In fact, the interpretation can be considered true but with a certain degree of truth [45]. In this case, we talk about vagueness. Currently, uncertainty and vagueness are not the concerns of our work but are planned for future development using fuzzy logics.

The nature of the ontology eases the integration of various spatial datasets. In future work we plan to integrate heterogeneous data sources into our analysis.

7. Acknowledgments

8. Bibliography

[1] M. Bhatt, J. O. Wallgrün, Geospatial narratives and their spatio-temporal dynamics: Commonsense reasoning for high-level analyses in geographic information systems, ISPRS International Journal of Geo-Information 3 (1) (2014) 166–205.

[2] E. F. Lambin, B. Turner, H. J. Geist, S. B. Agbola, A. Angelsen, J. W. Bruce, O. T. Coomes, R. Dirzo, G. Fischer, C. Folke, P. George, K. Homewood, J. Imbernon, R. Leemans, X. Li, E. F. Moran, M. Mortimore, P. Ramakrishnan, J. F. Richards, H. Skånes, W. Steffen, G. D. Stone, U. Svedin, T. A. Veldkamp, C. Vogel, J. Xu, The Causes of Land-use and Land-cover Change: Moving Beyond the Myths, Global Environmental Change 11 (4) (2001) 261 – 269. doi:http://dx.doi.org/10.1016/S0959-3780(01)00007-3.

[3] R. Mahmood, A. Quintanar, G. Conner, R. Leeper, S. Dobler, R. A. Pielke, A. Beltran-Przekurat, K. G. Hubbard, D. Niyogi, G. Bonan, P. Lawrence, T. Chase, R. McNider, C. Wu, Yuling andMcAlpine, R. Deo, A. Etter, A. Carleton, J. Adegoke, S. Vezhapparambu, S. Asefi, U. Nair, E. Sertel, D. R. Legates,

R. Hale, O. W. Frauenfeld, A. Watts, M. Shepperd, C. Mitra, V. G. Anantharaj, S. Fall, H.-I. Chang, R. Lund, A. Trevino, P. Blanken, J. Du, J. Syktus, Impacts of Land Use/Land Cover Change on Climate and Future Research Priorities, Bulletin of the American Meteorological Society 91 (1) (2010) 37–46.

[4] B. Harbelot, H. Arenas, C. Cruz, Continuum: A spatiotemporal data model to represent and qualify filiation relationships, in: 4th ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS) 2013, 2013.

[5] E. Meskovic, Z. Galic, M. Baranovic, Managing Moving Objects in Spatio-temporal Data Streams, in: Proceedings of the 2011 IEEE 12th International Conference on Mobile Data Management - Volume 02, MDM '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 15–18. doi:10.1109/MDM.2011.44.
URL http://dx.doi.org/10.1109/MDM.2011.44

[6] G. Del Mondo, M. A. Rodríguez, C. Claramunt, L. Bravo, R. Thibaud, Modeling Consistency of Spatio-temporal Graphs, Data & Knowledge Engineering 84 (2013) 59–80. doi:10.1016/j.datak.2012.12.007.
URL http://dx.doi.org/10.1016/j.datak.2012.12.007

[7] G. Del Mondo, J. G. Stell, C. Claramunt, R. Thibaud, A Graph Model for Spatio-temporal Evolution, Journal of Universal Computer Science 16 (11) (2010) 1452–1477.

[8] P. Muller, Topological spatio–temporal reasoning and representation, Computational Intelligence 18 (3) (2002) 420–450.

[9] T. Nyerges, M. Roderick, S. Prager, D. Bennett, N. Lam, Foundations of sustainability information representation theory: spatial–temporal dynamics of sustainable systems, International Journal of Geographical Information Science 28 (5) (2014) 1165–1185.

[10] B. Harbelot, H. Arenas, C. Cruz, A semantic model to query spatialtemporal data, in: V. Popovich, C. Claramunt, M. Schrenk, K. Korolenko (Eds.), Information Fusion and Geographic Information Systems (IF and GIS 2013), Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, 2014, pp. 75–89. doi:10.1007/978-3-642-31833-7_5.
URL http://dx.doi.org/10.1007/978-3-642-31833-7_5

[11] J. Stell, G. Del Mondo, R. Thibaud, C. Claramunt, Spatio-temporal evolution as bigraph dynamics, Spatial Information Theory (2011) 148–167.

[12] K. Hornsby, M. J. Egenhofer, Identity-based change: a foundation for spatio-temporal knowledge representation, International Journal of Geographical Information Science 14 (3) (2000) 207–224.

[13] M. P. Armstrong, Temporality in spatial databases, in: Proceedings: GIS/LIS, Vol. 88, 1988, pp. 880–889.

[14] J. Chen, H. Wu, S. Li, A. Liao, C. He, S. Peng, Temporal Logic and Operation Relations Based Knowledge Representation for Land Cover Change Web Services, ISPRS Journal of Photogrammetry and Remote Sensing 83 (0) (2013) 140 – 150. doi:http://dx.doi.org/10.1016/j.isprsjprs.2013.02.005.

[15] G. Langran, N. R. Chrisman, A framework for temporal geographic information, Cartographica: The International Journal for Geographic Information and Geovisualization 25 (3) (1988) 1–14.

[16] M. F. Worboys, A unified model for spatial and temporal information, The Computer Journal 37 (1) (1994) 26–34.

[17] M. J. Egenhofer, K. K. Al-Taha, Reasoning about gradual changes of topological relationships, Theories and methods of spatio-temporal reasoning in geographic space (1992) 196–219.

[18] D. J. Peuquet, N. Duan, An event-based spatiotemporal data model (estdm) for temporal analysis of geographical data, International journal of geographical information systems 9 (1) (1995) 7–24.

[19] C. Claramunt, M. Theriault, Toward semantics for modelling spatio-temporal processes within gis, Advances in GIs Research I (1996) 27–43.

[20] J. Jiang, M. Worboys, Event-based topology for dynamic planar areal objects, International Journal of Geographical Information Science 23 (1) (2009) 33–60.

[21] B. Jiang, C. Claramunt, B. Klarqvist, Integration of space syntax into gis for modelling urban spaces, International Journal of Applied Earth Observation and Geoinformation 2 (3) (2000) 161–171.

[22] B. Jiang, C. Claramunt, A structural approach to the model generalization of an urban street network*, GeoInformatica 8 (2) (2004) 157–171.

[23] B. Jiang, C. Claramunt, Topological analysis of urban street networks, Environment and Planning B 31 (1) (2004) 151–162.

[24] S. M. Hazarika, Qualitative spatial change: space-time histories and continuity, Ph.D. thesis, The University of Leeds (2005).

[25] E. Hyvönen, J. Tuominen, T. Kauppinen, J. Väätäinen, Representing and utilizing changing historical places as an ontology time series, in: Geospatial Semantics and the Semantic Web, Springer, 2011, pp. 1–25.

[26] T. Kauppinen, J. Väätäinen, E. Hyvönen, Creating and using geospatial ontology time series in a semantic cultural heritage portal, Springer, 2008.

[27] C. E. Campelo, B. Bennett, Representing and reasoning about changing spatial extensions of geographic features, Springer, 2013.

[28] R. Reiter, On closed world data bases, Springer, 1978.

[29] U. Hustadt, et al., Do we need the closed world assumption in knowledge representation?, in: KRDB, 1994.

[30] H. J. Levesque, The logic of incomplete knowledge bases, in: On conceptual modelling, Springer, 1984, pp. 165–189.

[31] A. Motro, Integrity= validity+ completeness, ACM Transactions on Database Systems (TODS) 14 (4) (1989) 480–502.

[32] B. Motik, I. Horrocks, U. Sattler, Bridging the gap between owl and relational databases, Web Semantics: Science, Services and Agents on the World Wide Web 7 (2) (2009) 74–89.

[33] E. Sirin, M. Smith, E. Wallace, Opening, closing worlds-on integrity constraints., in: OWLED, 2008.

[34] J. Tao, E. Sirin, J. Bao, D. L. McGuinness, Integrity constraints in owl., in: AAAI, 2010.

[35] E. Sirin, J. Tao, Towards integrity constraints in owl., in: OWLED, Vol. 529, 2009.

[36] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM 26 (11) (1983) 832–843.

[37] F. Baader, W. Nutt, Basic description logics., Description logic handbook (2003) 43–95.

[38] A. Artale, E. Franconi, A Temporal Description Logic for Reasoning About Actions and Plans, Journal of Artificial Intelligence Research 9 (1) (1998) 463–506.

[39] C. Strobl, Dimensionally extended nine-intersection model (de-9im), Encyclopedia of GIS (2008) 240–245.

[40] M. Bossard, J. Feranec, J. Otahel, Corine land cover technical guide: Addendum 2000 (2000).

[41] O. Project, Geotools website, http://www.geotools.org/ (2014).

[42] C. Parsia, Stardog website, http://stardog.com/ (2014).

[43] J.-N. Salomon, L'inondation dans la basse vallée de la garonne et l'estuaire de la gironde lors de la" tempête du siècle"(27-28 décembre 1999)/flooding in the garonne valley and the gironde estuary caused by the" storm of the century"(27-28 december 1999), Géomorphologie: relief, processus, environnement 8 (2) (2002) 127–134.

[44] V. LABOREE12, F. HISSEL, P. SERGENT, Evolution de l'emprise des zones inondables de l'estuaire de la gironde sous l'influence du changement climatique.

[45] T. Lukasiewicz, U. Straccia, Managing uncertainty and vagueness in description logics for the semantic web, Web Semantics: Science, Services and Agents on the World Wide Web 6 (4) (2008) 291–308.