Ontology inference using spatial and trajectory domain rules

Rouaa Wannous
L3i Laboratory
University of La Rochelle, France
Email: rouaa.wannous@univ-lr.fr

Jamal Malki
and Alain Bouju
L3i Laboratory
University of La Rochelle, France

Cecile Vincent
LIENSS laboratory and UMR 7372
CNRS/University of La Rochelle, France
Email: cvincent@univ-lr.fr

***Abstract*—Capture devices give rise to a large scale spatio-temporal data describing moving object's trajectories. These devices use different technologies like global navigation satellite system (GNSS), wireless communication, radio-frequency identification (RFID), and sensors techniques. Although capture technologies differ, the captured data share common spatial and temporal features. Thus, relational database management systems (RDBMS) can be used to store and query the captured data. For this, RDBMS define spatial data types and spatial operations. Recent applications show that the solutions based on traditional data models are not sufficient to consider complex use cases that require advanced data models. A complex use case refers to data, but also to domain knowledge, to spatial reasoning or others. This article presents a sample application based on trajectories that require three types of independent data models: a domain data model, a semantic model and a spatial model. We analyze each of them and propose a modeling approach based on ontologies. This work introduces a high-level trajectory ontology and a generic spatial ontology. Also, we present our ontology matching approach for integrating the sample trajectory domain to both defined ontologies. This work has a special focus to the problem of defining ontology inference using business rules combined with spatial rules. We present an implementation framework for declarative and imperative parts of ontology rules using an RDF data store. We discuss various experiments based on spatial inference calculation. We discuss these results and present some solutions to improve the complexity of calculating spatial ontological inferences.**

***Keywords*—Spatial data model, Semantic data model, Trajectory data model, Spatial rules, Bussines rules, Ontology inference**

I. INTRODUCTION

Spatial database management systems are created to manage spatial data in terms of storing, computing relationships and querying. Several applications are based on spatial databases like geographic information systems (GIS), urban planning [5], route optimization [17] and traffic monitoring [14]. On the other hand, advances in information and communication technologies have encouraged collecting spatial, temporal and spatio-temporal data of moving objects [11]. Large databases need to be analyzed and modeled to meet the user's needs. However, to answer these queries we need to take into account the domain knowledge.

This paper deals with marine mammals tracking applications, namely seal trajectories. Trajectory data are captured by sensors included in a tag glued to the fur of the animal behind the head. The captured trajectories consist of spatial, temporal and spatio-temporal data. Trajectories data can also contain some meta-data. These sets of data are organized into sequences. Every sequence, mapped to a temporal interval, characterizes a defined state of the animal. In our application, we consider three main states of the seal: haulout, dive and cruise. Every state is related to the seal's activity. For example, the foraging activity occurs during dives. Although temporal aspects are important in such studies, we mainly focus here on the spatial dimension of these data.

We assume that our trajectory data are stored and managed in a spatial relational database. Then, we consider the query (Example 1) based on a schema (Code 1) of two spatial tables.

Example 1: Which dives are contained within a zone

```
Table Dive (idDive:integer, refSeal:string, maxDepth:real,
     shape:line(startPoint, endPoint));
Table Zone (idZone:integer, name:string, shape:polygon(
     points[]));
```

Code 1. Spatial schema

To answer the query (Example 1), we need a relational database language supporting spatial data. ISO/IEC 13249-3 SQL/MM [1] is the effort to standardize extensions for multi-media and application-specific packages in SQL. The standard is grouped into several parts. The part 3 [2] is the international spatial standard that defines how to store, retrieve and process spatial data using SQL. It defines how spatial data are represented, and the functions available to convert, compare, and process spatial data in various ways. Code 2 gives the SQL/MM expression of the query (Example 1).

```
SELECT D.idDive, D.refSeal
FROM  Dive D, Zone Z
WHERE Z.shape.ST_Contains(D.shape) AND Z.idZone = 5;
```

Code 2. The SQL/MM query of Example 1

The SQL/MM expression (Code 2) is based on a relational model of the trajectory data. This model represents the domain by a set of attributes and their values. Therefore, this model cannot take into account the domain knowledge as given by experts. We describe here for instance the query (Example 2).

Example 2: In which zones is the seal foraging

Even if the SQL/MM language provides spatial operations to solve the query (Example 1), it is not designed to resolve the query (Example 2). Indeed, the later query combines spatial data (`zone`), spatial operation (`contains`) and the semantic domain knowledge related to the seal's activity (`foraging`). We notice that there is a semantic gap between the considered relational model of the trajectory data and the business process related to the domain knowledge as shown by the query (Example 2). Therefore, this paper addresses three main issues:

1) Trajectory domain model: The relational data model, in our trajectories data, is not suitable. Indeed, if for example we are interested to consider a generalization like (`Dive is a kind of Sequence`), all that the relational model can supply as a natural mechanism to express this constraint is a foreign key which concerns only the data and not the structuring links, as the generalization. In our work, an effective way to take into account the domain knowledge can be made through the user's needs. These needs are generally studied by the domain knowledge experts to formulate requirements or rules. As an example, the activity `foraging` is not a value or a set of values.
2) Spatial model: In the considered examples, the relations `dive` and `zone` can be assimilated to general spatial classes, respectively, `line` and `polygon`. Although, these are certainly not the only objects of the data model which can have spatial properties. Therefore, we believe that all spatial classes and properties must be considered regardless of the data model. The independent spatial model must be endowed with all the spatial reasoning.
3) Links between models: We based our approach on the definition of various separated models. Accordingly, we need to look at the problem of establishing links between these models. In the data engineering field, this problem is also known as data integration or mapping. Indeed, the study of this question is not recent and it arose from the need of reusing models.

This paper is organized as follows. Sections II illustrates the domain and semantic data models. Section III discusses OGC spatial data model implemented in different database systems and OGC spatial ontology model. Section IV details the implementation of the ontologies, the domain and spatial rules. Section V evaluates the spatial ontology inference over semantic trajectories. Section VI proposes enhancements over spatial ontology inference and experimental results evaluate the impact of them. Section VII summarizes some recent related work on spatial model and spatial semantics. Finally, Section VIII concludes this paper and presents some future prospects.

II. TRAJECTORY ONTOLOGY MODELING

A. Trajectory domain model

We consider trajectories of seals. The data are provided by LIENSs[1] laboratory in collaboration with SMRU[2]. These laboratories work on marine mammals' ecology. Trajectory data of seals between their haulout sites along the coasts of the English Channel or in the Celtic and Irish seas are captured using GNSS systems.

From the analysis of the captured data, we define a seal trajectory ontology that we connect to the trajectory domain ontology. The trajectory domain ontology is our model used in many moving object applications. Details of the modeling approach is discussed in [16]. Figure 1 shows an extract of the seal trajectory ontology, called `owlSealTrajectory`. Table I gives a dictionary of its concepts their relationships.

B. Seal trajectory model

In this work, we propose a Semantic Domain Ontology (Figure 2) based on activities organized as general ones linked to trajectory, and a hierarchy of basic activities linked to sequences of the trajectory domain ontology. The Seal Domain Ontology (Figure 2) is dealing with seal's activities. According to the domain expert, four activities (*resting, traveling, foraging* and *traveling-foraging*) are related to the three states of a seal. The seal trajectory ontology sequences are associated with these main activities.

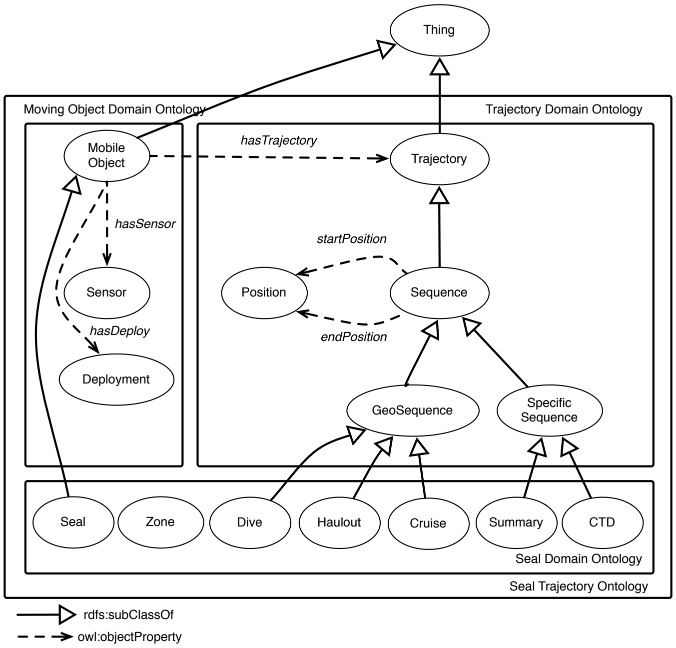

Figure 1. Overview of Seal Trajectory Ontology

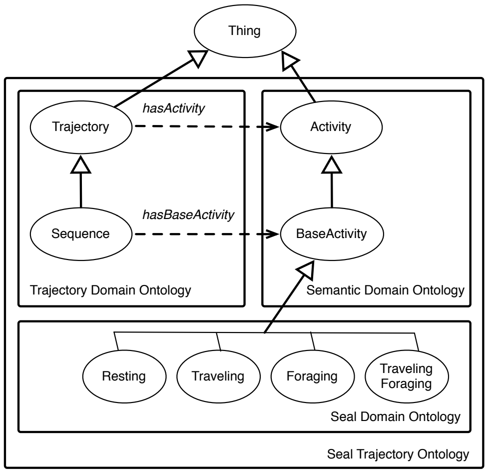

Figure 2. Overview of Seal Trajectory Ontology

---

[1] http://lienss.univ-larochelle.fr - CNRS/University of La Rochelle

[2] SMRU: Sea Mammal Research Unit- http://www.smru.st-and.ac.uk

Table I. SEAL TRAJECTORY ONTOLOGY DICTIONARY

Trajectory domain ontology	
Concept	**Description**
Trajectory	logical form to represent sets of sequences
Sequence	spatio-temporal interval representing a capture
GeoSequence	spatial part of sequence
Specific Sequence	metadata associated of a capture
startPosition, endPosition	object properties to represent the end and the beginning of a sequence
Seal domain ontology	
Concept	**Description**
haulout	a state of a seal when it is out of the water (on land) for at least 10 minutes
cruise	a state of a seal where it is in the water and shallower than 1.5 meter
dive	a state of a seal where it is in the water and deeper than 1.5 m for 8 seconds
Summary, CTD	metadata about deployment's conditions of the sensor, marine environment
dive_dur, sur_dur, max_depth	data properties: dive duration, surface duration and maximum depth of a dive, respectively
TAD	Time Allocation at Depth: data properties to define the shape of a seal's dive [5]

III. OGC SPATIAL DATA MODEL

We choose OGC to support spatial data, thanks for providing an OpenGIS simple feature specification for SQL [4]. This specification describes a standard set of SQL geometry types based on OpenGIS geometry model. Each spatial data is associated with a well-defined spatial reference system (SRID). SRID is a Spatial Reference IDentifier which supports coordinate system to uniquely identify any position on the earth. Latitudes and longitudes can be traced back to arbitrarily exact locations on the surface of the earth.

A. *OGC geometry object model*

The OGC geometry object model is based on extending the Geometry Model specified in the OpenGIS Abstract Specification. It is distributed computing platform neutral and uses OMT (Object Modeling Technique) notation. Figure 3 shows the object model for geometry. The base Geometry class has subclasses for Point, Curve, Surface and Geometry Collection. Each geometry object is associated with a Spatial Reference System (SRS) and has a Well-Known Text (WKT) presentation. Figure 3 shows aggregation lines between the leaf collection classes and their element classes. The OGC geometry object model defines relational operators on geometries. These are boolean methods that are used to test for the existence of a specified topological spatial relationship between two geometries. The specification is based on the Dimensionally Extended Nine-Intersection Model (DE-9IM) which describes the following kinds of spatial relationships: {`Touches, Crosses, Equals, Disjoint, Contains, Overlaps, Within, Intersects`}. The DE-9IM representation was developed by Clementini and others [7], [8] based on the seminal works of Egenhofer and others [9], [10].

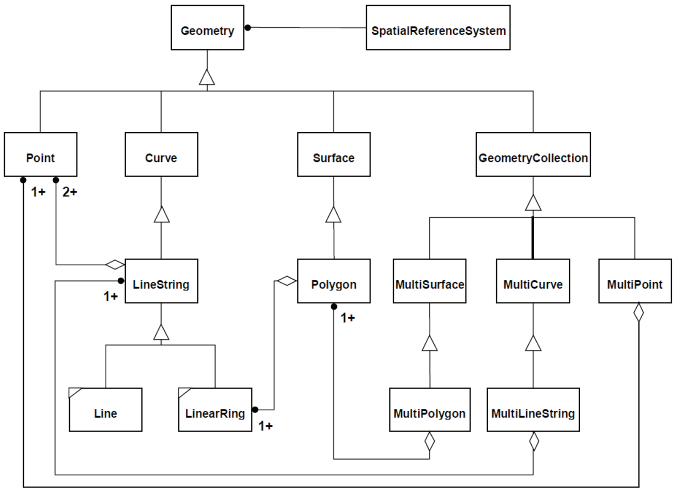

Figure 3. The OGC geometry object model hierarchy

B. *OGC model in Oracle Spatial*

Spatial supports the object-relational model for representing geometries. This model corresponds to an "SQL with Geometry Types" implementation of OpenGIS simple feature specification for SQL [4]. Spatial stores a geometry in Oracle native spatial data type for vector data, `SDO_GEOMETRY`.

The spatial relationship is based on geometry locations. The common spatial relationships are based on topology and distance. To determine spatial relationships between entities in the database, spatial has several secondary filter methods: `SDO_RELATE` operator evaluates topological criteria; `SDO_WITHIN_DISTANCE` operator determines if two spatial objects are within a specified distance of each other; `SDO_NN` operator identifies the nearest neighbors for a spatial object.

For example, the `SDO_RELATE` operator implements a nine intersection model for categorizing binary topological relationships between points, lines, and polygons. This yields to the set of spatial relationships:

```
SDO_covers, SDO_coveredby, SDO_contains, SDO_equal,
SDO_touch, SDO_inside, SDO_anyinteract, SDO_overlaps
```

Code 3. Topological relationships in Oracle Spatial

C. *OGC spatial ontology*

In our approach, we rewrite the OGC OMT class diagram (Figure 3) in UML class diagram. Then, we use model transformation techniques introduced by the Model Driven Engineering (MDE) community. For this, we choose an automatic transformation from UML class diagram into a formal ontology in OWL. We use transformer tool called *uml2owl Eclipse* [12]. This transformer, based on the meta-model *eCore Eclipse*, takes as input a UML class diagram and turns it into OWL-DL ontology. So, we transform the UML class diagram (Figure 3) to an OWL ontology, called `owlOGCSpatial`, Figure 4 presents an extract of it.

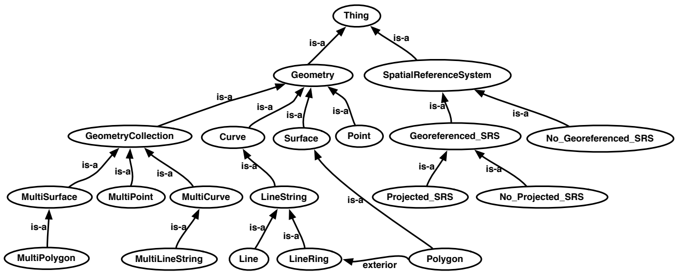

Figure 4. A view of `owlOGCSpatial` ontology

IV. IMPLEMENTATION OF ONTOLOGIES

A. *Seal trajectory ontology rules*

The seal trajectory ontology (Figure 1) is dealing with the seal's activities. Each seal activity has both a declarative part and an imperative part. The imperative parts of the activities are defined as rules in the ontology. A rule is an object that can be used by an inference process to query semantic data.

Oracle Semantic Technologies is a rule-based system where rules are based on `IF-THEN` patterns and new assertions are placed into working memory. Thus, the rule-based system is said to be a deduction system. In deduction systems, the convention is to refer to each `IF` pattern an antecedent and to each `THEN` pattern a consequent. `SEM_APIS.CREATE_RULEBASE` procedure defines user-defined rules in a `rulebase`. Our rulebase is called `sealActivities_rb`. The system automatically associates a view called `MDSYS.SEMR_rulebase-name` to insert, delete or modify rules in a `rulebase`. Code 4 gives `foraging_rule` definition based on domain expert's conditions. From line 4 to 10, we construct a subgraph and necessary variables needed by the `IF` part of the `foraging_rule`. Line 11 gives the `THEN` part of the rule. Line 12 defines the namespace of ontology.

```
 1 EXECUTE SEM_APIS.CREATE_RULEBASE('sealActivities_rb')
 2 INSERT INTO mdsys.semr_sealActivities_rb
 3 VALUES( 'foraging_rule',
 4 '(?diveObject rdf:type        s:Dive             )
 5  (?diveObject s:max_depth     ?maxDepth          )
 6  (?diveObject s:tad           ?diveTAD           )
 7  (?diveObject s:dive_dur      ?diveDur           )
 8  (?diveObject s:surf_dur      ?surfaceDur        )
 9  (?diveObject s:seqHasActivity ?activityProberty )',
10 '(maxDepth > 3) and (diveTAD > 0.9) and            (
       surfaceDur/diveDur < 0.5)',
11 '(?activityProberty rdf:type  s:Foraging         )',
12 SEM_ALIASES(SEM_ALIAS('s','owlSealTrajectory#')));
```

Code 4. Implementation of the `foraging rule`

B. *Spatial ontology rules*

Open GIS specification considers two kinds of spatial relationships:

- Topological relationships based on the DE-9IM operators defined as methods on `Geometry` class: `Equals, Within, Touches, Disjoint, Intersects, Crosses, Contains, Overlaps, Relate`;
- Functions for Distance Relationships: `Distance`.

In this work, we consider topological relationships. Each relationship has a declarative part as an RDF, and an imperative part, formally, an associated rule as `IF-THEN` pattern. We create a rulebase named `owlOGCSpatial_rb` to hold spatial relationships rules. For example, the rule (Code 5) presents the imperative part of the spatial relationship `Contains`. Lines 4 to 10 in Code 5 represent the `IF` side of the rule. We construct a subgraph and necessary variables, namely, the two spatial objects `sObj1` and `sObj2`, respectivity, their strings coordinates `wktSObj1` and `wktSObj2`, and the `srid` which is the Spatial Reference System Identifier. The `IF` side of the rule evaluates the spatial relationship between the two spatial objects using a function called `evalSpatialRelationship`. This function builds a bridge between the ontology spatial rules and spatial operators in Oracle DBMS. Line 11 in Code 5 is the consequent or the `THEN` part of the rule.

```
 1 EXECUTE SEM_APIS.CREATE_RULEBASE('owlOGCSpatial_rb');
 2 INSERT INTO mdsys.semr_owlOGCSpatial_rb
 3 VALUES('Contains_rule',
 4 '(?sObj1    rdf:type    os:Geometry)
 5  (?sObj2    rdf:type    os:Geometry          )
 6  (?sObj1    os:srid     ?srid                )
 7  (?sObj2    os:srid     ?srid                )
 8  (?sObj1    os:wkt      ?wktSObj1            )
 9  (?sObj2    os:wkt      ?wktSObj2            )',
10 '(evalSpatialRelationship(sObj1, wktSObj1, sObj2, wktSObj2
       , srid, ''Contains'' ) = 1)',
11 '(?sObj1    os:Contains   ?sObj2)',
12 SEM_ALIASES(SEM_ALIAS('os','owlOGCSpatial#')));
```

Code 5. Implementation of the `Contains_rule`

V. SPATIAL ONTOLOGY INFERENCE ON SEMANTIC TRAJECTORIES

A. *Ontology inference*

Inferencing is the ability to make logical deductions based on rules defined in the ontology. Inferencing involves the use of rules, either supplied by the reasoner or defined by the user. At data level, inference is a process of discovering new relationships, in our case, new triples. Inferencing, or computing entailment, is a major contribution of semantic technologies that differentiates them from other technologies. In Oracle Semantic Technologies, inference process is based on entailments. We distinguish two entailments regimes [18]:

1) Standard entailment: there are several standard entailment regimes: semantics of RDF, RDFS and OWL. Support for RDF and RDFS is simplified by the availability of axioms and rules that represent their semantics. Support for major subsets of OWL-Lite and OWL-DL vocabularies have been provided. In this work, we use the subset `OWLPRIME` [22];
2) Custom entailment: since the standard vocabularies cannot handle all varieties of semantic application data, it becomes important to provide support for entailment based on arbitrary user-defined rules. In this work, we defined a rulebase for trajectory semantics `sealActivities_rb` and a rulebase for spatial relationships `owlOGCSpatial_rb`.

In Oracle Semantic Technologies, an entailment contains precomputed data inferred from applying a specified set of rulebases to a specified set of semantic models. Code 6 creates an entailment using seal trajectory and spatial models.

Other options are also required like number of rounds that the inference engine should run. In case of applying user-defined rules `USER_RULES=T`, the number of rounds should be assigned as default to `REACH_CLOSURE`.

```
SEM_APIS.CREATE_ENTAILMENT('owlSealTrajectory_idx',
SEM_MODELS('owlSealTrajectory','owlOGCSpatial'),
SEM_RULEBASES('OWLPrime','sealActivities_rb','
     owlOGCSpatial_rb'),
SEM_APIS.REACH_CLOSURE, NULL, 'USER_RULES=T');
```

Code 6. Entailment over the models and rullbases

B. *Spatial ontology inference*

The spatial ontology inference is the process of applying spatial ontology rules to compute topological relationships between spatial objects. Query 2, where we are looking for zones where the seal is foraging, combines the seal trajectory semantic `Foraging` and the spatial relationship `Contains`. To resolve it, the system needs an entailment over seal trajectory and spatial rules. The system must know the spatial relationships between zones and dives, considered as spatial polygons and lines, respectively. Figure 5 illustrates the computation algorithm of the inference over spatial objects. For every two spatial objects, the inference procedure calls spatial rules. The function `evalSpatialRelationship` calls the corresponding Oracle spatial operator for the current running spatial rule. The result of this function is returned to the spatial rule for checking the relationship between the two considered spatial objects. Computing a new relationship generates and saves a new inference triple.

VI. ENHANCE SPATIAL INFERENCE

A. *Restrictions and constraints refinement*

The spatial ontology rules are computed redundantly to calculate the spatial relationships between geometries during the inference mechanism. To enhance the inference process, the user can define, for example, domain constraints limit the computation in a useful way for their work's objective. These limitations can be directional considering objects in the same direction or can be distance constraints considering a specific distance between objects or restrictions related to the type of the considered objects.

In our case, we define a refinement called *area of interest*. This refinement limits the computation of the inference to the objects located in a specified area. The area of interest refinement is given by Algorithm 1, considering two geosequences (S_a, S_r) and a given *area*. This algorithm gives the spatial relationship between the two considered geosequences S_a, S_r as output. This algorithm checks if these two geosequences belong to the interested area to compute the spatial relationship between them. If they do not belong to this area, the algorithm goes for another spatial candidate.

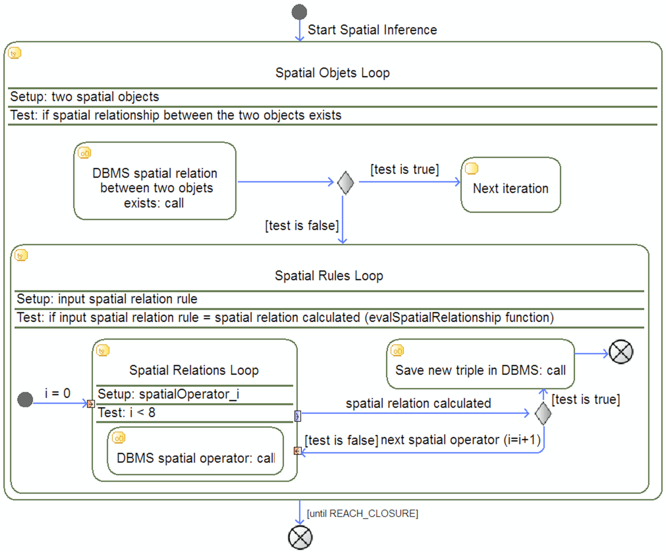

Figure 5. Activity diagram for spatial inference process

```
input : Two geosequences: a referent $S_r$ and an
        argument $S_a$
input : An interested area area
output: Spatial relationship between $S_r$ and $S_a$
initialization;
if ($S_a$, $S_r$) $\in$ area then
    calculate spatial relationship between $S_r$ and $S_a$;
end
go the next geosequence $S_{a+1}$;
```

Algorithm 1: Area of interest refinement algorithm

B. *Passes refinement*

We mention the `REACH_CLOSURE` problem in the case of applying user-defined rules. To control the number of the cycles done by the engine of Oracle during computing the inference, we define a refinement called *Passes refinement*. The passes refinement is illustrated by Algorithm 2 to effect the cycles of the engine. This algorithm takes the two considered geosequences (S_a, S_r) as input and provides the spatial relationship (*Res*) between (S_a, S_r) as output. This algorithm checks the computation of the inference between these geosequences if it exists. In the case of the inference passes for the first time, the inference process will be computed normally and its results will be given as output for this algorithm. In the other case where the inference is performed once before, the algorithm reads the saved result from the database and assigns it as a result to this pass. The function `evalSpatialRules` considers this refinement to optimize the passes of the engine during the computation of the inference.

```
input : Two geosequences: a referent $S_r$ and an
        argument $S_a$
output: Spatial rule between $S_r$ and $S_a$ in Res
initialization;
if (INFERENCE ($S_a$, $S_r$)) $\in$ database then
    Res := result of the spatial rule from the database;
else
    Res := calculate inference between $S_r$ and $S_a$;
    Save Res in the database;
end
go the next geosequence $S_{a+1}$;
```

Algorithm 2: Passes refinement algorithm

C. Experimental results

In this section we evaluate the two spatial refinements we introduced in this work. In this evaluation, we consider sets of real seal trajectory data. The inference uses the eight spatial rules and the trajectory domain foraging rule, while the evaluation curves is given by the number of dives.

Firstly, we evaluate the area of interest refinement. Related to the seal trajectory domain and to our domain knowledge, we limit the area of interest restraint to 500 meters. We pass this candidate to Algorithm 1. The experimental results of this proposed refinement are shown in Figure 6. The results show its impact by the three following experiments:

1) *Spatial ontology rule calls - constraints refinement* presents the executions of the spatial ontology rules using the constraints refinement;
2) *Spatial ontology rule calls not executed* gives the reduced executions of the rules after the refinement;
3) *DMBMS spatial operator calls - constraints refinement* provides Oracle spatial operator calls during the inference process with the refinement.

We observe a decrease in both of the spatial ontology rules computation and DBMS spatial operator calls. For example, considering 250 dives, in the normal case of inference, the executions of the spatial ontology rules is 1000 000 and DBMS spatial operator calls is 125 000. However in the refinement case Figure 6, the executions of the rules is 130 000 and DBMS operator calls is 16 000.

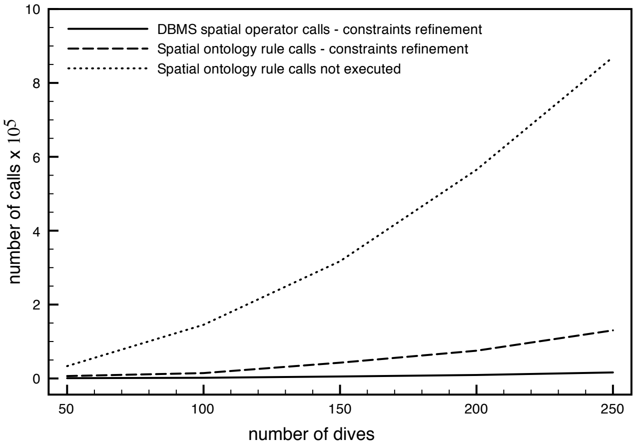

Figure 6. Enhancement of the spatial ontology inference with constraints refinement

Secondly, we evaluate the proposed passes refinement. The experimental results are shown in Figure. 7. The impact results are shown by the three following experiments:

1) *Spatial ontology rule calls* presents the spatial ontology rule calls during the inference process;
2) *Spatial ontology rule calls - passes refinement* displays the spatial ontology rule calls with the passes refinement;
3) *Spatial ontology rule calls - passes and constraints refinement* provides the spatial ontology rule calls with both refinements.

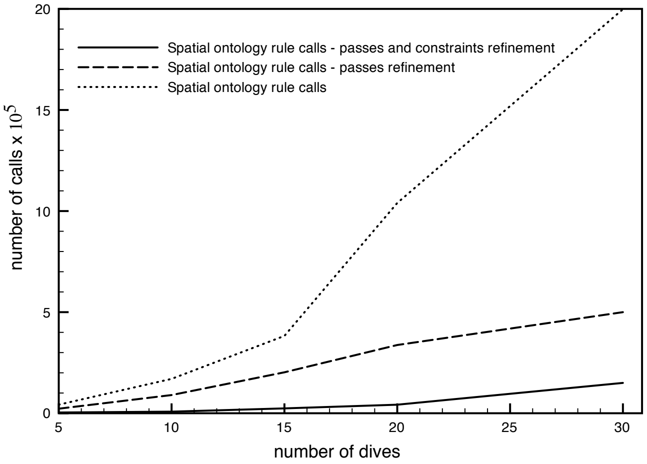

Figure 7. Evaluation of the spatial ontology inference over the proposed refinement

We observe a decrease in the spatial ontology rule calls with both refinements together. For example, considering 300 dives, in the normal case of inference, the spatial ontology rule calls is 2 000 000. However in the passes refinement case, the spatial ontology rule calls is 500 000. Finally in the both refinements case, the spatial ontology rule calls is 150 000. The final results are therefore considered as good impacts over the complexity and the size of the inference process.

VII. RELATED WORK

Several studies work on spatial domain, however so far there is no standard spatial ontology model. In 2007, the Geospatial Incubator Group (GeoXG), a W3C working group, tried to provide an overview of geospatial foundation ontology to represent geospatial concepts [15]. Moreover, GeoSPARQL ontology [13] represents and queries geospatial data on the semantic web. GeoSPARQL is based on OGC simple features model [4], with some adaptations for RDF. GeoSPARQL is a common query language for the Geospatial semantic web that can handle and index linked spatio-temporal data.

Enriching trajectory application domain with spatial model leads to manage semantics on trajectories. A conceptual view on trajectories is proposed by Spaccapietra et al. [19] in which the trajectories are a set of stops and moves. Stops are the important places of the trajectory where the object has stayed for a minimal amount of time. Vandecasteele et al. [20] adopted the trajectory data model proposed in [19] to detect abnormal ship behaviour by an enhanced spatial reasoning ontology. They integrated the spatial dimension into their ontology, and defined with their experts domain rules in Semantic Web Rule Language (SWRL). Nevertheless, the integration of the spatial dimension cannot yet be fully implemented in the ontology due to the lack of appropriate structures.

Moreover, by Battle et al. [3], a Geo-Ontology is an ontology design patterns for semantic trajectories. The authors used their semantic trajectory pattern to annotate two kinds of databases: trajectories generated by human travelers and by animals. This work lacks semantics and mainly do not support inference over domain rules to enhance the semantic trajectories. The computational time taken by the inference mechanism including OWL-Time ontology is addressed by [21]. Based on a space-time ontology and events approach, Boulmakoul et al. [6] proposed a generic meta-model for trajectories to allow independent applications processing trajectories data benefit from a high level of interoperability, information sharing. Their approach is inspired by ontologies, however the proposed resulting system is pure database approach and a pure SQL-based approach not on semantic queries. Related to all those limitations, we design and implement an ontological trajectory framework integrated with the spatial dimension. The computation of domain and spatial rules as user-defined rules in our framework are the scope of this paper. We also propose some enhancements for the inference mechanism.

VIII. CONCLUSION AND FUTURE WORK

In this work, we propose a modeling approach based on ontologies applied to the problem of thematic and spatial reasoning over trajectories. Our approach considers three separated ontology models: a general trajectory domain model, a domain knowledge or semantic model and a spatial domain model. We discuss the trajectory domain ontology and the semantic domain ontology. The considered semantic trajectory domain ontology connects the two previous models while considering moving object domain ontology. The spatial ontology model is based on the OGC standard. Therefore, we detail the specification of OGC Consortium and its different implementation in DBMS. To implement imperative part of the ontologies, we consider the framework of Oracle Semantic Data Store. To define the thematic and spatial reasoning, we implement rules related to the considered models. Thematic rules are based on domain trajectory activities and the spatial rules are based on spatial relationships. We compute the spatial ontology inference over semantic trajectories. Spatial rules directly influence the ontological inference process. This inference can be enhanced, so we address its main problems. For this reason, we propose some domain constraints and an inference refinement to enhance the spatial ontology inference. We evaluate our proposal on real trajectory data. The experimental results show the positive impact of the proposed approach. Finally, the objective of this paper is to extract in further details spatial characteristics revealed by and associated with moving object trajectory domain. So far, we used some domain application constraints over the ontological rules to effect positively the computation of the inference mechanism. For the future work, we would like to use a two-tier inference filters. In other words, two distinct operations are performed to enhance the inference: primary and secondary filter operations. The primary filter applies all the domain constraints over the captured data. In this paper, we consider a few of the domain interests, however for the future work, we will try to collect all the possible refinements, for example, analyzing data, classification or indexing. Then the primary filter permits fast selection of the filtered data or the analyzed data to pass along to the secondary filter. The latter computes the inference mechanism and yields the final knowledge data.

REFERENCES

[1] ISO/IEC 9075-2:1999. In *Information Technology Database Languages SQL Part 2: Foundation (SQL/Foundation)*, 1999.

[2] ISO/IEC 13249-3:2002 FDIS. In *Information technology Database languages SQL Multimedia and Application Packages Part 3: Spatial*, 2002.

[3] R. Battle and D. Kolas. Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web*, 3(4):355–370, 2012.

[4] D. Beddoe, P. Cotton, R. Uleman, S. Johnson, and J. R. Herring. OpenGIS Simple Features Specification For SQL. Technical report, OGC, 1999.

[5] E. Boeker. *Environmental Science; Physical Principles and Applications*. New York: Wiley, 2001. Ref. GE80.H69 1993.

[6] A. Boulmakoul, L. Karim, and A. Lbath. Moving object trajectories meta-model and spatio-temporal queries. In *International Journal of Database Management Systems (IJDMS)*, volume 4, pages 35–54, 2012.

[7] E. Clementini, P. D. Felice, and P. v. Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *Proceedings of the Third International Symposium on Advances in Spatial Databases*, SSD '93, pages 277–295, London, UK, UK, 1993. Springer-Verlag.

[8] E. Clementini, J. Sharma, and M. J. Egenhofer. Modelling topological spatial relations: Strategies for query processing. *Computers & Graphics*, pages 815–822, 1994.

[9] M. J. Egenhofer and R. Franzosa. Point-set topological spatial relations. Number 5(2), pages 161–174, 1991.

[10] M. J. Egenhofer and J. Herring. A mathematical framework for the definition of topological relationships. pages 803–813, 1990.

[11] R. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.

[12] G. Hillairet, F. Bertrand, and J. Y. Lafaye. MDE for publishing data on the semantic web. In *international workshop on Transformation and Weaving Ontologies and Model Driven Engineering (TWOMDE) at MODELS'08*, 2008.

[13] Y. Hu, K. Janowicz, D. Carral, S. Scheider, W. Kuhn, G. Berg-Cross, P. Hitzler, M. Dean, and D. Kolas. A geo-ontology design pattern for semantic trajectories. In *Spatial Information Theory*, volume 8116 of *Lecture Notes in Computer Science*, pages 438–456. Springer International Publishing, 2013.

[14] F. Korn, S. Muthukrishnan, and Y. Zhu. Checks and balances: monitoring data quality problems in network traffic databases. In *Proceedings of the 29th international conference on Very large data bases - Volume 29*, VLDB '03, pages 536–547. VLDB Endowment, 2003.

[15] J. Lieberman, R. Singh, and C. Goad. W3C Geospatial ontologies - W3C incubator group, 2007.

[16] W. Mefteh. *Approche ontologique pour la modelisation et le raisonnement sur les trajectoires. Prise en compte des aspects thematiques, temporels et saptiaux*. PhD thesis, La Rochelle university, 2013.

[17] S. Sangheon, Pack Xuemin sherman, M. Senior, W. M. Jon, F. Life, and P. Jianping. Adaptive route optimization in hierarchical mobile IPv6 networks.

[18] D. Souripriya and S. Jagannathan. Database technologies for rdf. In *Reasoning Web. Semantic Technologies for Information Systems, 5th International Summer School 2009*, volume 5689 of *Lecture Notes in Computer Science*, pages 205–221. Springer, 2009.

[19] S. Spaccapietra, C. Parent, M. Damiani, J. Demacedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Data and Knowledge Engineering*, pages 126–146, 2008.

[20] A. Vandecasteele and A. Napoli. An enhanced spatial reasoning ontology for maritime anomaly detection. In *International Conference on System Of Systems Engineering - IEEE SOSE*, GIS '06, pages 247–252, 2012.

[21] R. Wannous, J. Malki, A. Bouju, and C. Vincent. Time integration in semantic trajectories using an ontological modelling approach. In *New Trends in Databases and Information Systems*, volume 185 of *Advances in Intelligent Systems and Computing*, pages 187–198. Springer Berlin Heidelberg, 2013.

[22] Z. Wu, G. Eadon, S. Das, E. I. Chong, V. Kolovski, M. Annamalai, and J. Srinivasan. Implementing an inference engine for rdfs/owl constructs and user-defined rules in oracle. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 1239–1248, Washington, DC, USA, 2008. IEEE Computer Society.