Enhancement of Trajectory Ontology Inference Over Domain and Temporal Rules

Rouaa Wannous, Jamal Malki and Alain Bouju
L3i laboratory
University of La Rochelle
La Rochelle, France
Emails: {rwanno01, jmalki, abouju}@univ-lr.fr

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

***Abstract*—Capture devices rise large scale trajectory data from moving objects. These devices use different technologies like global navigation satellite system (GNSS), wireless communication, radio-frequency identification (RFID), and other sensors. Huge trajectory data are available today. In this paper, we use an ontological data modeling approach to build a trajectory ontology from such large data. This ontology contains temporal concepts, so we map it to a temporal ontology. We present an implementation framework for declarative and imperative parts of ontology rules in a semantic data store. An inference mechanism is computed over these semantic data. The computational time and memory of the inference increases very rapidly as a function of the data size. For this reason, we propose a two-tier inference filters on data. The primary filter analyzes the trajectory data considering all the possible domain constraints. The analyzed data are considered as the first knowledge base. The secondary filter then computes the inference over the filtered trajectory data and yields to the final knowledge base, that the user can query.**

Keywords–Trajectory ontology modeling; Ontology inference; Domain rules; Temporal rules; Data filter algorithm.

I. INTRODUCTION

Advances in information and communication technologies have encouraged collecting spatial, temporal and spatio-temporal data of moving objects [1]. The raw data captured, commonly called trajectories, traces moving objects from a departure point to a destination point as sequences of data (sample points captured, time of the capture). Raw trajectories do not contain goals of traveling nor activities accomplished by the moving object. Large datasets need to be analyzed and modeled to tackle user's requirements. To answer user's queries we also 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 datasets 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 a seal: *hauling out, diving* and *cruising*. Every state is related to a seal's activity. For example, a foraging activity of seal occurs during the state diving.

Our goal is to enrich trajectory data with semantics to extract more knowledge. In our previous work [2], we tackled trajectory data connected to other temporal and spatial sources of information. We directly computed the inference over these data. The experimental results addressed the running time and memory problems over the ontology inference computation. Furthermore, we tried to solve these problems by defining some domain constraints, time restrictions in [3] and inference refinements in [4]. The proposed refinements enhanced the inference computation, however, they did not fully solve the problems.

In the present work, we introduce two-tier inference filters on trajectory data. In other words, two distinct operations are performed to enhance the inference: primary and secondary filter operations. The primary filter is applied to the captured data with the consideration of domain constraints. The primary filter allows fast selection of the analyzed data to pass along to the secondary filter. The latter computes the inference over the data output of the primary filter. The global view of this work is detailed as the following steps:

- Semantic trajectory data is an RDF dataset based on an ontology trajectory;
- For analyzing the data, filtering or indexing could be applied. In our case, we carry out a place-of-interest process to analyze data. The analyzed data are stored in a knowledge repository;
- The secondary filter computes inferences over the data with the consideration of domain knowledge;
- The semantic trajectory data and the new data inferred are stored in the knowledge repository.

This paper is organized as follows. Section II summarizes recent work related to trajectory data modeling using ontology approach and some introduced solutions to tackle the problem of the inference complexity using data filtering. Section III illustrates an overview of the ontological modeling approach used. This trajectory ontology contains temporal concepts mapped to W3C OWL-Time ontology [5] in Section IV. Section V details the implementation of the trajectory ontology, the domain ontology rules and the temporal rules. Section VI addresses the complexity of the ontology inference over the domain and temporal rules. Section VII introduces the primary filter over trajectory data based on a place-of-interest process. Section VIII evaluates the ontology inference over the filtered data. Finally, Section IX concludes this paper and presents some prospects.

II. RELATED WORK

Data management techniques including modeling, indexing, inferencing and querying large data have been actively investigated during the last decade [4][6][7]. Most of these techniques are only interested in representing and querying moving object trajectories [2][4][8]. A conceptual view on trajectories is proposed by Spaccapietra et al. [9] in which trajectories are a set of stops, moves. Each part contains a set of

 ISBN: 978-1-61208-358-2

semantic data. Based on this conceptual model, several studies have been proposed, such as [8][10]. Yan et al. [8] proposed a trajectory computing platform that exploits a spatio-semantic trajectory model. One of the layers of this platform is a data preprocessing layer which cleanses the raw GPS feed, in terms of preliminary tasks such as outliers removal and regression-based smoothing. Alvares et al. [10] proposed a trajectory data preprocessing method to integrate trajectories with the space. Their application concerned daily trips of employees from home to work and back. However, the scope of their paper is limited to the formal definition of semantic trajectories with the space and time without any implementation and evaluation. Trajectory filtering visualises a subset of available trajectories [11]. This is useful to view interesting trajectories and discard uninteresting ones. Trajectory filtering can be run in two modes: soft, hard filtering.

Based on a space-time ontology and events approach, Boulmakoul et al. [12] proposed a generic meta-model for trajectories to allow independent applications. They processed trajectories data benefit from a high level of interoperability, information sharing. Their approach is inspired by ontologies, however the proposed resulting system is a pure database approach. Boulmakoul et al. have elaborated a meta-model to represent moving objects using a mapping ontology for locations. In extracting information from the instantiated model during the evaluation phase, they seem to rely on a pure SQL-based approach not on semantic queries. Taking these limitations into account, we defined and implemented two tier inference filters over trajectory data to clean and analyze the data and solve the inference computation problem. Baglioni et al. [13][14] are based on the conceptual model on trajectories [9]. They represent annotated trajectories in an ontology encompassing geographical and application domain knowledge. They consider different kinds of stops and temporal knowledge to discriminate among them. Afterwards, they use ontology axioms to infer behavior of patterns using Oracle and OWLPrime to test the axioms. Moreover, Perry et al. in [15] apply an inference mechanism over their ontology. This inference is based on several domain specific table functions and only on RDFS rules indexes. They use a military application domain and apply complex queries require sophisticated inference methods. In their implementation, they use Oracle DBMS and demonstrate the scalability of their approach with a performance study using both synthetic and real-world RDF datasets.

III. Trajectory Ontology Modeling

A. *Trajectory Domain Ontology*

This paper considers trajectories of seals. The data are provided by LIENSs [16] laboratory in collaboration with SMRU [17]. 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 by Mefteh [18]. Figure 1 shows an extract of the seal trajectory ontology, called `owlSealTrajectory`. Table I gives a dictionary of its concepts and their relationships.

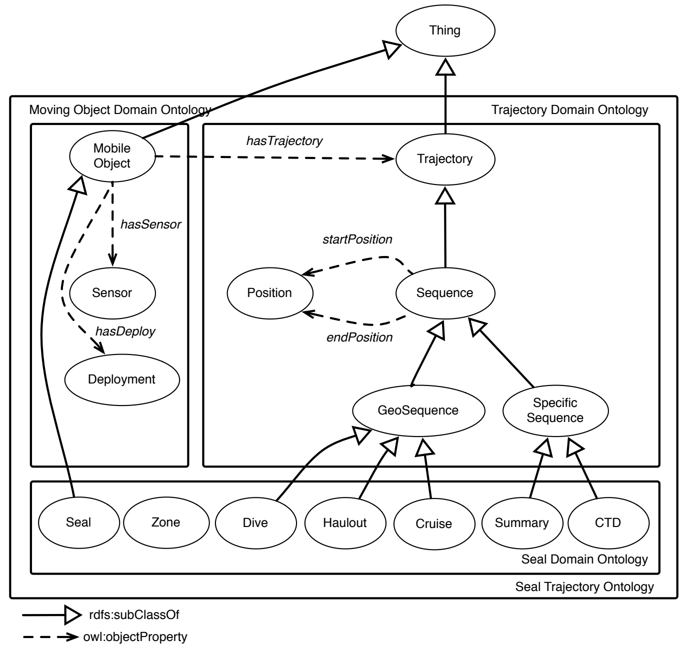

Figure 1. Overview of the seal trajectory ontology

B. *Seal Trajectory Ontology*

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.

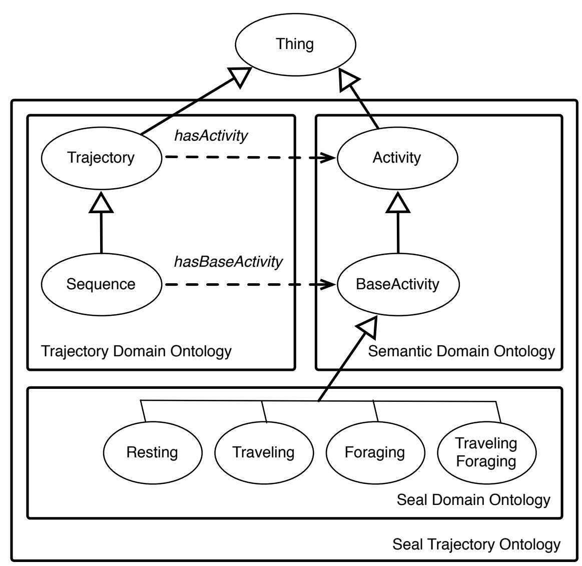

Figure 2. Overview of Seal Trajectory Ontology

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.

 ISBN: 978-1-61208-358-2

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]

IV. Time Ontology

The seal trajectory ontology includes concepts that can be considered as temporal. For example, the concept `Sequence` is a temporal interval. To integrate temporal concepts and relationships in the seal trajectory ontology, we choose a mapping approach between our ontology and the `OWL-Time` ontology [5] developed by the World Wide Web Consortium (W3C). This mapping is detailed in our previous work [2]. An extract of the declarative part of this ontology is shown in Figure 3 described in detail by Jerry and Feng [5].

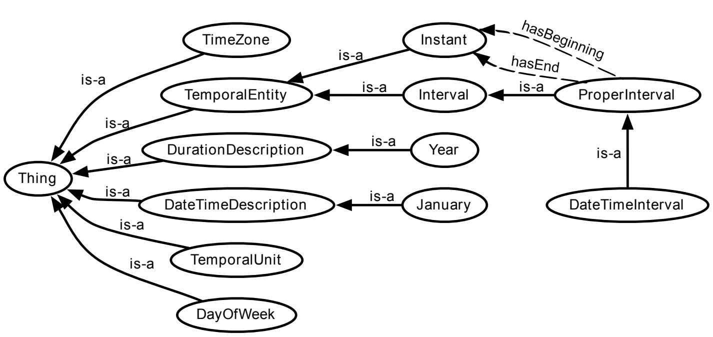

Figure 3. A view of the `OWL-Time` ontology

We are mainly interested in the `ProperInterval` concept and its two properties `hasBeginning` and `hasEnd`.

V. Implementation of Ontologies

A. General Framework Implementation

For the implementation of the ontologies, we use Oracle Semantic Technologies. These technologies have evolved since Oracle DBMS version 10g, 11g and take the name of "Oracle Spatial and Graph - RDF Semantic Graph" in Oracle DBMS version 12c. This system provides support for persistence, inference and querying ontologies through implementation of RDF, RDFS and a large part of OWL standards. The DBMS defines a core in its metabase to support technologies related to ontological data. It stores the ontology declaration with data as RDF triples in the system under the scheme `MDSYS`. Each triple {subject, predicate, object} is handled as a basic data object. Detailed description of this technology can be found in Oracle Semantic Technologies Developer's Guide [19]. To create declarative and imperative parts of the seal trajectory and time ontologies, we:

1) Create the declarative parts of the ontologies;
2) Create instances and population of the ontologies;
3) Consistency checking of the ontological instances;
4) Create the imperative parts of the ontology (seal trajectory ontology rules and temporal rules).

B. Seal Trajectory Ontology Rules

The seal trajectory ontology (Figure 2) is dealing with the seal's activities. Each seal activity has both a declarative part and an imperative corresponding 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. User-defined rules are defined using the `SEM_APIS.CREATE_RULEBASE` procedure 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`. Figure 4 gives the `foraging_rule` definition based on domain expert's conditions. From line 4 to 10 of Figure 4, 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#')));
```

Figure 4. Implementation of the `foraging rule`

C. Time Ontology Rules

The `OWL-Time` ontology declares the 13 temporal interval relationships based on Allen algebra [20]. We implement the rule base `owlTime_rb` to hold the interval temporal relationships. For example, Figure 5 presents the implementation of the imperative part of the `intervalAfter_rule` based on operations defined in the table `TM_RelativePosition` of the ISO/TC 211 specification about the temporal schema [21].

```
1  EXECUTE SEM_APIS.CREATE_RULEBASE('owlTime_rb')
2  INSERT INTO mdsys.semr_owltime_rb
3  VALUES('intervalAfter_rule',
4  '(?tObj1     rdf:type ot:ProperInterval          )
5   (?tObj2     rdf:type owltime:ProperInterval     )
6   (?tObj1     ot:hasEnd ?end1                     )
7   (?end1      :inXSDDateTime ?endTime1            )
8   (?tObj2     ot:hasBeginning ?begin2             )
9   (?begin2    ot:inXSDDateTime ?beginTime2        )',
10 '(beginTime2  > endTime1                         )',
11 '(?tObj2     owltime:intervalAfter ?tObj1        )',
12 SEM_ALIASES(SEM_ALIAS('ot','http://www.w3.org/2006/time#')
       ));
```

Figure 5. Implementation of the `intervalAfter rule`

In Figure 5, line 10 expresses the condition that the beginning of the reference interval is bigger than the end of the argument interval, as explained in the following condition. Line 11 is the consequent of the rule.

$$self.begin.position > other.end.position$$
$$\text{where}$$
$$\begin{cases} self & = & tObj2 \\ other & = & tObj1 \\ self.begin.position & = & beginTime2 \\ other.end.position & = & endTime1 \end{cases}$$

VI. Trajectory 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 the 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, an entailment contains precomputed data inferred from applying a specified set of rulebases to a specified set of semantic models. Figure 6 creates an entailment over the seal trajectory and time models. This entailment uses a subset of OWL rules called `OWLPrime` [19], the seal trajectory and time ontologies rules. Other options are also required like the number of rounds that the inference engine should run. When applying user-defined rules `USER_RULES=T`, the number of rounds should be assigned as default to `REACH_CLOSURE`.

In our experiment, we measure the time needed to compute the entailment (Figure 6) for different sets of real trajectory data for one seal. Its movements are captured from 16 June until 18 July 2011 and we have got 10 000 captured data. In this experiment, the seal activity rulebase contains only the foraging rule. The input data for this entailment are only dives. Figure 7 shows the experiment results for the computation time in seconds needed by the entailment. For example, for

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

Figure 6. Entailment over the `owlSealTrajectory` and `owlTime` ontologies

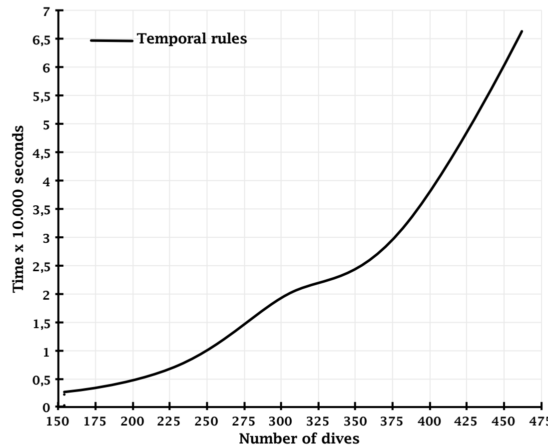

Figure 7. Entailment computation time with all temporal rules and the foraging activity

450 dives, the inference takes around 60 000 seconds ($\simeq$ 16.6 hours).

We notice the huge time taken from the inference mechanism over a small data.

VII. Place Of Interest over Trajectory Data

We introduce a two-tier inference refinement on trajectory data. In other words, two distinct operations are performed to enhance the inference: primary and secondary inference operations. Figure 8 shows the two-tier inference filter refinement. The primary filter is applied to the captured data to classify them into a set of interested places, called Area-Restricted Search (ARSs). The primary filter allows fast selection of the classified data to pass along to the secondary inference. The latter computes the inference mechanism considering the ARS. Then, instead of annotating each sequence in the model, we annotate the ARSs with the expert knowledge activity model. The inference process is computed for each ARS. The secondary inference yields the final knowledge data that the user can query.

Our proposal is to analyze the captured data before computing the ontology inference. This analysis is achieved thanks to our primary filter. This filter considers trajectories that are segmented by the object positions. These positions change and remain fixed. Spaccapietra [9] named the former `moves` and the latter `stops`. For this reason, a trajectory is seen as a sequence of moves going from one stop to the next one.

Definition 1 (Stop): A stop is a part of a trajectory having a time interval and represented as a single point.

Definition 2 (Move): A move is a part of a trajectory represented as a spatio-temporal line.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-358-2

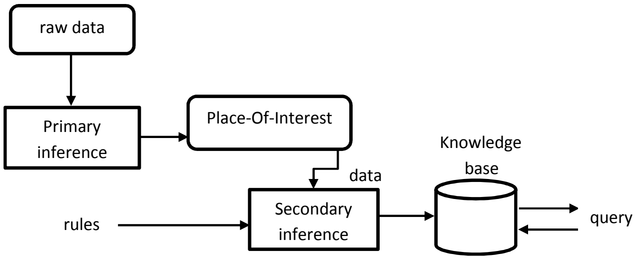

Figure 8. Two-tier inference filter refinement

input : *Move*
input : *Stop*
input : *radius*
output: *Places*
1 initialization;
2 $Neighbor \leftarrow \phi$;
3 $Points\_Neighbors \leftarrow \phi$;
4 $Places \leftarrow \phi$;
5 **for** *each* $p_i \in Move$ **do**
6 calculate $Neighbor(p_i)$;
7 $Points\_Neighbors \leftarrow (p_i, Neighbors(p_i))$;
8 $Move \leftarrow Move - Neighbor(p_i)$;
9 **end**
10 **for** *each* $p_i \in Points\_Neighbors$ *AND condition*$(p_i, peaks_i)$ *AND condition*$(distance(p_i, Stop) > radius)$ **do**
11 **if** $distance(p_i, Places[j]) > radius$ **then**
12 $Places[k] \leftarrow (Neighbors_i, 1)$;
13 **else**
14 $Places(Neighbors_j, nVisits_j) = ([Neighbors_j, Neighbors_i], nVisits_j + 1)$;
15 **end**
16 **end**

Figure 9. The Place Of Interest algorithm

The primary filter defines interesting places for a moving object. The interesting places are related to where the moving object stays more and visits more often. This filter is explained in Figure 9. This algorithm takes the two parts of a trajectory (move and stop) data as input and gives as output interesting places. The following definitions are used by this algorithm:

Definition 3 (Neighbors): Neighbors for a point (p_i) are a list of points from the Move data where the distance between p_i and any neighbor point is smaller than a fixed radius. $Neighbor(p_i) = \{(p_j)_{j=1}^{n} : p_i, p_j \in Move,\ distance(p_i, p_j) < radius\}$.

Definition 4 (Peak): A peak$_i$ is a cardinality of the list $Neighbor(p_i)$. $(peaks_i)_{i=1}^{n} = \#(Neighbor(p_i))_{i=1}^{n}$.

Definition 5 (Points_Neighbors): Points_Neighbors are a list of points and their neighbors. $Points\_Neighbors = \{(p_i, Neighbors_i)_{i=1}^{n} : p_i, Neighbors_i \in Move\}$.

Definition 6 (Places): Place$_i$ is an interesting place which contains the $Neighbor(p_i)$ and number of its visits $(nVisits)$ by the moving object. $Places = \{(Neighbors_i, nVisits_i)_{i=1}^{n} : Neighbors_i \in Move,\ nVisits_i \in number\}$.

The first step of the primary filter, Figure 9 lines 5-9, gathers the move data into groups of neighbors. These groups are defined with respect to a *radius*. This radius is a fixed distance between two points to calculate the neighbors. The candidate of the radius is related to the application view of a trajectory, and is an input for this algorithm. The output of the first step is *Points_Neighbors*, from which the second step starts.

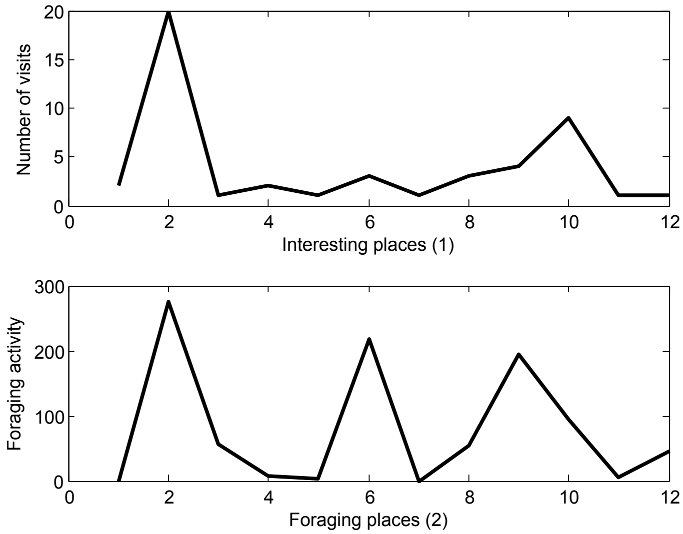

Figure 10. Interesting and foraging places

Lines 10-16, the second step, defines the interesting places. In general, we can consider all the members of *Points_Neighbors* or we can apply a condition over the *Peaks*. For example, the application view could be interesting in places that have 60 points and over, or could be interesting in any place having at least a point. For defining a place, the coordinates of the neighbors could be an interesting place after applying two conditions. Every point that belongs to a place should be far from the stop data more than the fixed radius. Any place should not have any neighbor within the radius distance, otherwise we merge the two coordinates and increase the visits number. The result of this step is the output *Places* of this algorithm.

VIII. Experimental Results

To analyze our data, we consider the same datasets in Section VI. We pass these data to the Place Of Interest algorithm. This algorithm analyzes the data and gives as output the places and their visits, as shown in Figure 10 interesting places (1). However, the main goal is to define foraging places among the captured data from 16 June until 18 July 2011. We look forward to analyse all the 10 000 captured data.

Defining foraging places is the objective of the secondary filter. The secondary filter computes the entailment over the interesting places. This filter specifies foraging places among 10 000 captured data. It determines the number of foraging activity for each place, as shown in Figure 10 foraging places (2). We can notice that the places 1, 4, 5, 7 and 11 are not considered as foraging places. Places 2, 6 , 9 and 10 are the significant foraging places. Finally, the results of the primary filter are decreased the captured data from 10 000 into 6 170 interesting raw trajectories organized in places.

By the normal inference ontology computation results, we could not be able to consider all the captured data. We computed the inference just for 500 raw data. However, using the primary filter and defining the interesting places helped us to define foraging places over all the captured data. These inferred data are considered as the final knowledge data that the user can query.

IX. Conclusion

In this work, we proposed a modeling approach based on ontologies to build a trajectory ontology. Our approach considers three separated ontology models: a general trajectory domain model, a domain knowledge or semantic model and a temporal domain model. We map the spatial concepts in the trajectory ontology to the spatial ontology. To implement the declarative and imperative parts of the ontologies, we consider the framework of Oracle Semantic Data Store. To define the thematic and temporal reasoning, we implement rules related to the considered models. The thematic rules are based on the domain trajectory activities and the temporal rules are based on Allen relationships. Then, we define and apply two-tier inference filters. In other words, two distinct operations are performed to enhance the inference: primary and secondary filter operations. The primary filter analyzes the trajectory data into places of interest. The secondary filter computes the ontology inference over the semantic trajectories using the ontology domain and temporal rules. The latter filters the interesting places into domain activity places. The experimental results show that we are able with the two-tier filters to consider all the captured data, whereas we could not even compute the ontology inference. For the evaluation, we use a PC with Linux system over a processor i5-250M, 2.5GHz and 8G memory.

References

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

[2] 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*, ser. Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, 2013, pp. 187–198.

[3] J. Malki, R. Wannous, A. Bouju, and C. Vincent, "Temporal reasoning in trajectories using an ontological modelling approach." Control and Cybernetics, 2012, pp. 761–777.

[4] R. Wannous, J. Malki, A. Bouju, and C. Vincent, "Modelling mobile object activities based on trajectory ontology rules considering spatial relationship rules," in *Modeling Approaches and Algorithms for Advanced Computer Applications*, ser. Studies in Computational Intelligence. Springer International Publishing, 2013, pp. 249–258.

[5] R. H. Jerry and P. Feng, "An ontology of time for the semantic web," in *ACM Transactions on Asian Language Information Processing*, 2004, pp. 66–85, http://www.w3.org/2006/time.

[6] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "SeMiTri: A framework for semantic annotation of heterogeneous trajectories," in *Proceedings of the 14th International Conference on Extending Database Technology*. ACM, 2011, pp. 259–270.

[7] J. Malki, A. Bouju, and W. Mefteh, "An ontological approach modeling and reasoning on trajectories. taking into account thematic, temporal and spatial rules," in *TSI. Technique et Science Informatiques*, 2012, pp. 71–96.

[8] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty, "A hybrid model and computing platform for spatio-semantic trajectories," in *The Semantic Web: Research and Applications*. Springer Berlin/Heidelberg, 2010, pp. 60–75.

[9] S. Spaccapietra, C. Parent, M. Damiani, J. Demacedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," in *Including Special Section: Privacy Aspects of Data Mining Workshop*, 2008, pp. 126–146.

[10] L. O. Alvares and et al., "A model for enriching trajectories with semantic geographical information," in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. ACM, 2007, pp. 1–22.

[11] Geant4, *Geant4 User's Guide for Application Developers*, 2011, ch. Visualization/Trajectory Filtering.

[12] A. Boulmakoul, L. Karim, and A. Lbath, "Moving object trajectories meta-model and spatio-temporal queries," in *International Journal of Database Management Systems (IJDMS)*, 2012, pp. 35–54.

[13] M. Baglioni, J. Macedo, C. Renso, and M. Wachowicz, "An ontology-based approach for the semantic modelling and reasoning on trajectories," in *Advances in Conceptual Modeling - Challenges and Opportunities*. Springer Berlin/Heidelberg, 2008, pp. 344–353.

[14] M. Baglioni, J. A. Fernandes de Macedo, C. Renso, R. Trasarti, and M. Wachowicz, "Towards semantic interpretation of movement behavior," in *Advances in GIScience*, ser. Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, 2009, pp. 271–288.

[15] P. Matthew, "A framework to support spatial, temporal and thematic analytics over semantic web data," Ph.D. dissertation, Wright State University, 2008.

[16] "LIENSs: Cnrs/university of la rochelle," http://lienss.univ-larochelle.fr/.

[17] "SMRU; sea mammal research unit," http://www.smru.st-and.ac.uk/.

[18] W. Mefteh, "Ontology modeling approach on trajectory over thematic, temporal and spatial domains," Ph.D. dissertation, La Rochelle university, 2013.

[19] Oracle, "Oracle Database Semantic Technologies Developer's guide 11g release 2," 2012. [Online]. Available: http://www.oracle.com/technology/tech/semantic-technologies

[20] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, pp. 832–843, 1983.

[21] ISO/TC_211, "Geographic information – temporal schema, ISO 19108," 2002.