Trajectory ontology inference over domain and temporal rules. Inference complexity analyzing

Rouaa Wannous[1] Jamal Malki[1] Alain Bouju[1] Cecile Vincent[2]

[1] L3i laboratory , LIENSs laboratory, University of La Rochelle, France
[2] LIENSs laboratory, University of La Rochelle, France

{rwannous, jmalki, abouju, cvincent}@univ-lr.fr

Abstract

Capture devices rise a large scale trajectory data about moving object's movements. 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. In this article, we are interested in these data, so we use an ontological data modeling approach to build a trajectory ontology. 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 running 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 data analyzed are considered as the first knowledge base. These data is passed to the secondary filter. Then the latter computes the inference over the filtered trajectory data. The secondary filter results yield the final knowledge base where the user can query.

Keywords

Trajectory ontology modeling, Ontology inference, Temporal rules, Data filter algorithm.

Résumé

Les dispositifs de capture de trajectoires d'objets en mouvement produisent généralement des volumes de données très important. Ils se développent grâce á différentes technologies, telles que les systèmes de navigation globale par satellite (GNSS), les communications sans fil, l'identification par radio-fréquence (RFID) et d'autres types de capteurs. Nous nous intéressons dans cet article á ces larges volumes de données que nous organisons au sein de modèles ontologiques dans le but de construire une ontologie des trajectoires. Cette ontologie est interfacée avec une ontologie temporelle afin de gérer les concepts relatifs au temps. Nous présentons l'implémentation d'un framework pour les phases de déclaration et de mise en œuvre des règles au sein d'une base de données sémantiques. Un mécanisme d'inférence est lancé sur ces données sémantiques. Son temps d'exécution ainsi que sa charge en mémoire augmentent rapidement en fonction du volume de données. Afin de limiter ce problème, nous proposons un double filtre d'inférence sur les données. Le premier filtre analyse les données de trajectoire en prenant en compte toutes les contraintes de domaine possibles. Les données de sortie forment alors la première base de connaissances et sont transférées au deuxième filtre. Celui-ci effectue l'inférence sur ces données filtrées. Les données en sortie de ce second filtre forment la base de connaissances finales que l'utilisateur va pouvoir interroger.

Mots Clef

Modélisation de trajectoire, Règles temporelles, Inférence, filtrage de données.

1 Introduction

Advances in information and communication technologies have encouraged collecting spatial, temporal and spatio-temporal data of moving objects [5]. 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 don't contain goals of traveling nor activities accomplished by the moving object. Large datasets need to be analyzed and modeled to tackle the user's requirements. To answer these queries we need also 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 the seal : haulout, dive and cruise. Every state is related to seal's activity. For example, a foraging activity occurs during dives.

Our goal is to enrich trajectory data with semantics to extract more knowledge. In our previous work [15], we tackled trajectory data connected to other temporal and spatial source 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 try to solve these problems by defining some domain constraints, time restrictions [9] and inference refinements [14]. The proposed refinements enhanced the inference computation, however, they do not definitely solve the inference problems.
In the present work, we introduce a 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 applies all the possible domain constraints over the captured data. So, the primary filter permits 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 the ontology trajectory ;
- For analyzing 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 ;
- Secondary filter computes inferences over the data with the consideration of the domain knowledge ;
- Semantic trajectory data and the new data inferred are stored in the knowledge repository ;

This paper is organized as follows. Section 2 illustrates an overview of the ontological modeling approach used. This trajectory ontology contains temporal concepts, so Section 3 presents W3C OWL-Time ontology [7] which is mapped to our ontology. Section 4 details the implementation of the trajectory ontology, the domain ontology rules and the temporal rules. Section 5 addresses the complexity of the ontology inference over domain and temporal rules. Section 6 introduces the primary filter over trajectory data based on place-of-interest process. Section 7 evaluates the ontology inference over the filtered data. Section 8 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. Finally, Section 9 concludes this paper.

2 Trajectory ontology modeling

2.1 Trajectory domain ontology

This paper considers trajectories of seals. The data comes from the LIENSs [1] (CNRS/University of La Rochelle) 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 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 [11]. Figure 1 shows an extract of the seal trajectory ontology, called `owlSealTrajectory`. Table 1 gives a dictionary of its concepts.

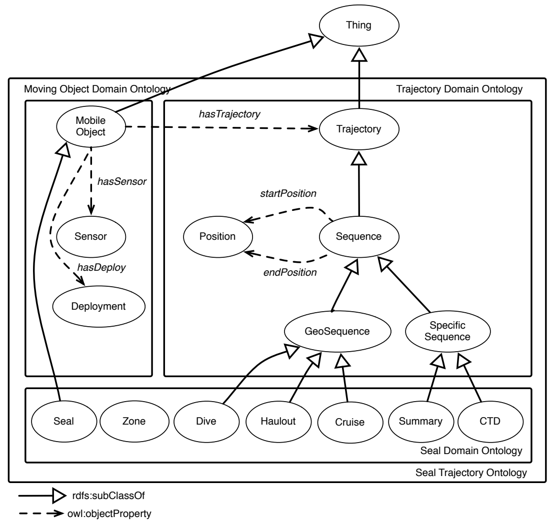

FIGURE 1 – Overview of the seal trajectory ontology

2.2 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. The Seal Domain Ontology (Figure 2) considers seal's activities. According to the domain expert, the seal trajectory ontology sequences are associated with four main activities : resting, traveling, foraging and traveling-foraging.

3 Time ontology

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 seal trajectory ontology, we choose a mapping approach between our ontology and the `OWL-Time` [3] ontology [7] developed by the World Wide Web Consortium (W3C). This mapping is detailed in our previous work [15]. An extract of the declarative part of this ontology is shown in figure 3 described in detail in [7]. We are mainly interested in the `ProperInterval` concept and its two properties `hasBeginning` and `hasEnd`.

1. http ://lienss.univ-larochelle.fr
2. SMRU : Sea Mammal Research Unit- http ://www.smru.st-and.ac.uk
3. `http://www.w3.org/2006/time`

TABLE 1 – Seal trajectory ontology dictionary

Concept	Description
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 seal where it is not in the water and wet for 40 seconds
cruise	a state seal where it is in the water and shallower than 1.5 meter
dive	a state seal where it is in the water and deeper than 1.5 m for 8 seconds
CTD	Conductivity-Temperature-Depth: metadata about marine environment
Summary	metadata about deployment's conditions of the sensor
dive_dur, sur_dur, max_depth	data properties: dive duration, surface duration and maximum depth of a dive
TAD	Time Allocation at Depth: data properties to define the shape of a seal's dive [7].

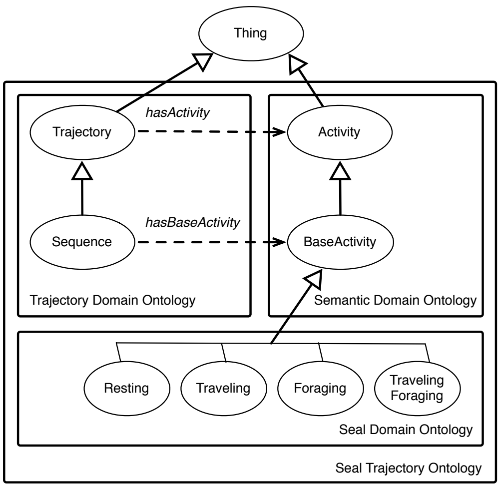

FIGURE 2 – Overview of Seal Trajectory Ontology

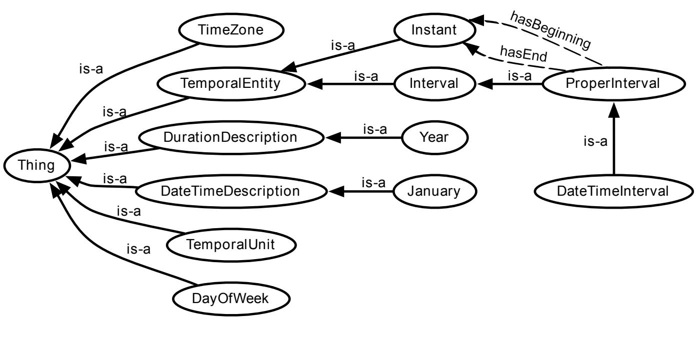

FIGURE 3 – A view of `OWL-Time` ontology

4 Ontology rules

4.1 Seal trajectory ontology rules

Seal trajectory ontology (Figure 2) considers seal's activities. Each seal activity has both a declarative part and an imperative corresponding part. The imperative parts of 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` pattern 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 `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`. Code 1 gives the `foraging_rule` definition based on the domain expert's conditions. From line 4 to 10 of Code 1, we construct a subgraph and necessary variables needed by the `IF` part of `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 1 – Implementation of `foraging rule`

4.2 Time ontology rules

The `OWL-Time` ontology declares the 13 temporal interval relationships based on Allen algebra [1]. We implement the rule base `owlTime_rb` to hold the interval temporal relationships. For example, the code 2 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 temporal schema [6]. In code 2, the line 10 expresses the condition that the beginning of the reference interval is bigger than the end of the argument interval. Line 11 is the consequent of rule.

5 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 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.

```
EXECUTE SEM_APIS.CREATE_RULEBASE('owlTime_rb')
INSERT INTO mdsys.semr_owltime_rb
VALUES('intervalAfter_rule',
'(?tObj1     rdf:type ot:ProperInterval         )
 (?tObj2     rdf:type owltime:ProperInterval    )
 (?tObj1     ot:hasEnd ?end1                    )
 (?end1      :inXSDDateTime ?endTime1           )
 (?tObj2     ot:hasBeginning ?begin2            )
 (?begin2    ot:inXSDDateTime ?beginTime2       )',
'(beginTime2  >  endTime1                       )',
'(?tObj2     owltime:intervalAfter ?tObj1       )',
SEM_ALIASES(SEM_ALIAS('ot','http://www.w3.org/2006/
     time#')));
```

Code 2 – Implementation of `intervalAfter rule`

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 3 creates an entailment over the seal trajectory and time models. This entailment uses a subset of OWL rules called `OWLPrime` [12], the seal trajectory and time ontologies rules. 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`.
In our experiment, we measure the time needed to compute the entailment (Code 3) 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. So, the input data for the entailment are only dives. Figure 4 shows experiment results for the computation time in seconds needed by the entailment. For example, for 450 dives, the inference takes around 60 000 seconds ($\simeq$ 16.6 hours).

```
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');
```

Code 3 – Entailment over `owlSealTrajectory` and `owlTime` ontologies

6 Place Of Interest over trajectory data

Our proposal is to analyze the captured data before computing the ontology inference. This analyzing is achieved by our primary filter. This filter considers trajectories which are segmented by the object positions. These positions changes and stays fixed. Spaccapietra [13] 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.*

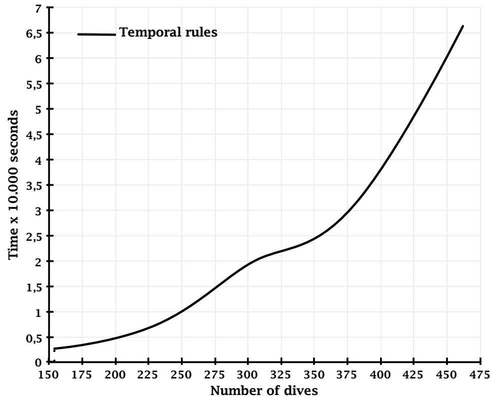

FIGURE 4 – Entailment computation time with all temporal rules and the foraging activity

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

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. This filter is explained in Algorithm 1. 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 the 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, Algorithm 1 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 it is taken as input for this algorithm. The output of the first step is *Points_Neighbors*. The second step of this filter, lines 10-16, defines the interesting places starting from the *Points_Neighbors*. In general, we could take all the members of the *Points_Neighbors* or we could apply a condition over the (*Peak*). For example, the application view

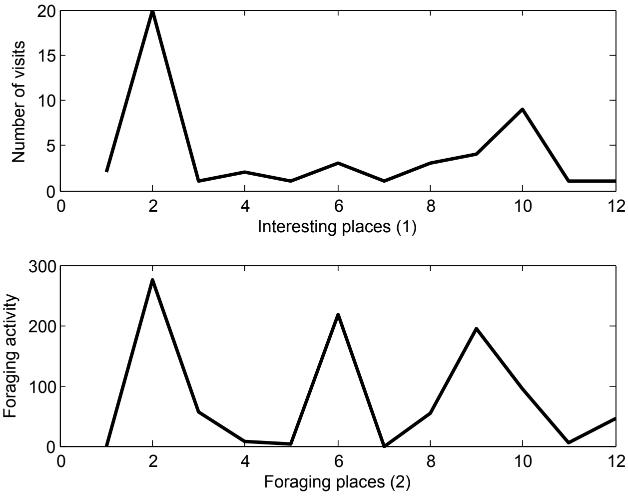

FIGURE 5 – Interesting and foraging places

could be interesting in places which having 60 points and over, or could be interesting in any place having at least a point. For defining places, the coordinate of the group neighbors could be an interesting place with two condition. Every point belongs to a place should be far from the stop data more than the fixed radius. Any place should not have any neighbor place within the radius distance, otherwise we merge the two coordinates and increase the visits number. The result of this step $(Places)$ is the output of this algorithm.

To analyze our data, we consider the same datasets in Sect. 5. 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 Fig 5 interesting places (1). Finally, the results of the primary filter are decreased the captured data from 10 000 into 6 170 interesting raw trajectories.

input : *Move*
input : *Stop*
input : *radius*
output: *Places*
initialization;
$Neighbor \leftarrow \phi$;
$Points\_Neighbors \leftarrow \phi$;
$Places \leftarrow \phi$;
for *each* $p_i \in Move$ **do**
    calculate $Neighbor(p_i)$;
    $Points\_Neighbors \leftarrow (p_i, Neighbors(p_i))$;
    $Move \leftarrow Move - Neighbor(p_i)$;
end
for *each* $p_i \in Points\_Neighbors$ *AND condition*$(p_i, peaks_i)$ *AND condition(distance*$(p_i, Stop) > radius$*)* **do**
    if $distance(p_i, Places[j]) > radius$ **then**
        $Places[k] \leftarrow (Neighbors_i, 1)$;
    else
        $Places(Neighbors_j, nVisits_j) = ([Neighbors_j, Neighbors_i], nVisits_j + 1)$;
    end
end

Algorithm 1: The Place Of Interest algorithm

7 Experimental results

We analyze the trajectory data and define the interesting places. However, the main goal is to define foraging places among these them. This is the goal of the secondary filter. The secondary filter computes the entailment over the interesting places. This filter specifies foraging places and determines the number of foraging activity for each place, as shown in Figure 5 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.

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

8 Related work

Data management techniques including modelling, indexing, inferencing and querying large data have been actively investigated during the last decade [16, 10, 8]. Most of these techniques are only interested in representing and querying moving object raw trajectories [17, 15, 3]. A conceptual view on trajectories is proposed by Spaccapietra et al. [13] in which trajectories are a set of stops, moves. Each part contains a set of semantic data. Based on this conceptual model, several studies have been proposed such as [2, 17]. Alvares et al. [2] proposed a trajectory data preprocessing method to integrate trajectories with the spatial. 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. Yan et al. [17] proposed a trajectory computing platform which exploits spatio-semantic trajectory model. One of the layers of this platform is data preprocessing layer which cleanses the raw GPS feed, in terms of preliminary tasks such as outliers removal and regression-based smoothing. Based on a space-time ontology and events approach, Boulmakoul et al. [4] proposed a generic metamodel 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. Boulmakoul et al. have elaborated a meta-model to represent moving objects using a mapping ontology for locations. Actually, in extracting information from the instantiated model during the evaluation phase, they seem rely on a pure SQL-based approach not on semantic queries. Related to all those limitations in the state of the art, we define and implement two tier filters over trajectory data to clean and analyze the data and solve the computation problem.

9 Conclusion and future work

In this work, we propose 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. Thematic rules are based on domain trajectory activities and the temporal rules are based on Allen relationships. Then, we consider 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 experimental results shows that we are able with the two- tier filters to answer user query over all the captured data, whereas we could not without it 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. For the future work, we look for a server PowerVault NX400 with processor E5-2420 at 1.90GHz 6 cores and 16Gb ram with 4 drives TB.

Références

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, 1983.

[2] L. O. Alvares, V. Bogorny, B. Kuijpers, A. F. Macedo, B. Moelans, and A. Vaisman. A model for enriching trajectories with semantic geographical information. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 22 :1–22 :8. ACM, 2007.

[3] 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*, volume 5232, pages 344–353. Springer Berlin/-Heidelberg, 2008.

[4] 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.

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

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

[7] R. H. Jerry and P. Feng. An ontology of time for the semantic web. In *ACM Transactions on Asian Language Information Processing*, volume 3, pages 66–85, 2004.

[8] 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*, volume 31/1-2012, pages 71–96, 2012.

[9] J. Malki, R. Wannous, A. Bouju, and C. Vincent. Temporal reasoning in trajectories using an ontological modelling approach. *Control and Cybernetics*, pages 1–16, 2012.

[10] P. Matthew. *A framework to support spatial, temporal and thematic analytics over semantic web data*. PhD thesis, Wright State University, 2008.

[11] 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.

[12] Oracle. Oracle Database Semantic Technologies Developer's guide 11g release 2, 2012.

[13] 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.

[14] 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*, volume 488 of *Studies in Computational Intelligence*, pages 249–258. Springer International Publishing, 2013.

[15] 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.

[16] 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*, pages 259–270. ACM, 2011.

[17] 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*, pages 60–75. Springer Berlin/Heidelberg, 2010.