Future Generation Computer Systems 68 (2017) 491–499

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Trajectory ontology inference considering domain and temporal dimensions—Application to marine mammals

Rouaa Wannous [a,*], Jamal Malki [a], Alain Bouju [a], Cécile Vincent [b]

[a] Laboratoire L3i, Pôle Sciences & Technologie, Avenue Michel Crépeau, 17042 La Rochelle, France
[b] Centre d'Etudes Biologiques de Chizé, CNRS UMR 7372, 79360 Villiers-en-Bois, France

HIGHLIGHTS

- Modeling approach: from raw data to semantic trajectory modeling based on an ontological approach.
- Trajectory ontology inference: how domain and temporal rules are modeled and computed.
- Application domain trajectory model: how the approach can be applied to a specific domain.
- Design and implementation with the focus on inference system.
- Experiments and results.

ARTICLE INFO

Article history:
Received 30 April 2015
Received in revised form
11 January 2016
Accepted 21 January 2016
Available online 3 March 2016

Keywords:
Trajectory ontology modeling
Ontology inference
Domain rules
Temporal rules
Data filter algorithm

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. To accomplish reasoning over trajectories, the ontology must consider mobile object, domain and other knowledge. In our approach, we suggest expressing this knowledge in the form of rules. To annotate data with these rules, we need an inference mechanism over trajectory ontology. Experiments over our model using domain and temporal rules address an inference computation complexity. This complexity has two important factors: time computations and space storage. In order to reduce the inference complexity, we proposed optimizations, such as domain constraints and temporal neighbor refinements. In this paper, we define a refinement specifically for the application domain. Then, we evaluate our contribution over real trajectory data. Finally, the results show the positive impact of the last refinement on reducing the complexity of the inference mechanism. This refinement reduces half of the time computation and then allows considering larger data sets.

1. 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 data sets 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 data sets 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 occurs during the state diving.

* Corresponding author.
E-mail addresses: rwannous@univ-lr.fr (R. Wannous), jmalki@univ-lr.fr (J. Malki), abouju@univ-lr.fr (A. Bouju), cvincent@univ-lr.fr (C. Vincent).

http://dx.doi.org/10.1016/j.future.2016.01.012
0167-739X/

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 time computation and space storage problems of the ontology inference. Then, we proposed some solutions to reduce the inference complexity by defining time restrictions in [3] and inference passes refinements in [4]. These later studies focus mainly on the term of time computation.

In the present paper, we continue studying the ontological inference complexity, specially in terms of inference space storage complexity. We propose 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.

This paper is organized as follows. Section 2 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 3 illustrates an overview of the used domain data model. This trajectory ontology contains temporal concepts mapped to W3C OWL-Time ontology [5] in Section 4. Sections 5 and 6 details the trajectory ontology inference and the integrated knowledge. In Section 7, we implement the trajectory ontology, the domain ontology rules and the temporal rules. Section 8 addresses the complexity of the ontology inference over the domain and temporal rules. Section 9 introduces an application domain inference refinement. Section 10 evaluates the ontology inference over the proposed refinement. Finally, Section 11 concludes this paper and presents some prospects.

2. Related work

Data management techniques including modeling, indexing, inferencing and querying large data have been actively investigated during the last decade [6–8]. Most of these techniques are only interested in representing and querying moving object trajectories [9,2,10]. A conceptual view on trajectories is proposed by Spaccapietra et al. [11] in which trajectories are a set of stops and moves. Each part contains a set of semantic data. Based on this conceptual model, several studies have been proposed such as [12,9]. Alvares et al. [12] 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.

Yan et al. [9] proposed a trajectory computing platform which exploits a spatio-semantic trajectory model. A difference is made between the semantic and spatial dimensions in order to provide a data model representation that supports different levels of abstraction. Authors present a solution for extracting semantic trajectories from raw ones. One of the layers of the presented 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. However, this work did not discuss the computation complexity of the platform going from the raw data step processing to knowledge extraction and finally decision-making.

Based on a space–time ontology and events approach, Boulmakoul et al. [13] proposed a trajectory patterns of moving objects. Important packages of the trajectory patterns are "Space Time Path Domain", "Activity Domain", "Observation and Measure Domains" and "Region Of Interest" packages. These packages are then transformed to a unified moving object trajectory queries expressed in SQL-like relational database language. Queries operations on space and time are performed using simple relational entities and functions. So they seem to rely on a pure SQL-based approach not on semantic queries. This work also did not discuss the evaluation of the proposed approach on real data sets.

In [14], authors gave a brief outline of a scalable data collection framework for the unified moving object trajectories meta model. They gather different kinds of geographical data based on the unified moving object trajectories' meta-model. The collection framework offers components to collect spatio-temporal data. They test the scalability of the proposed system by a vehicle tracking simulator which generates and simulates spatio-temporal data of different moving objects. Recently, Boulmakoul et al. [15] proposed a trajectory's data model which has advantages of both conceptual and ontological space–time. So they extend the model with new patterns as the space–time path to describe activities of the moving object and the composite region of interest. The case study is presented for tracking travelers at the airport.

3. Modeling approach

3.1. Design and methodology

Our work is based on moving objects trajectories. This requires a trajectory data model and a moving object model. Moreover, to enrich data with knowledge, a semantic model should be taken into consideration. Therefore, we need a generic model to consider the trajectory, moving object and semantic models simultaneously as shown in Fig. 1. The semantic trajectory model can consume captured data of trajectories and other external data as shown in Fig. 1 link (1). These data are related to an application domain. This requires an application domain trajectory model which consists of domain model, as shown in Fig. 1 link (2). The latter will support semantics related to users' needs. In the domain model, we also find the necessary semantics related to the real moving object, its trajectories, its activities and others. This semantics is often designed by a domain expert. In general, considering various facets of data involves that the semantic trajectory model must be extended by other models: application domain, temporal and spatial models. Then, the main issue is to build and design the semantic trajectory model with its required components.

The semantic trajectory modeling approach is tightly related to the problem of a semantic gap between this model and raw data. Link (1) in Fig. 1 presents this gap. Moreover, our approach involves multiple models and then must establish semantic mappings among them, to ensure interoperability. In Fig. 1, links (2) and (3) match the domain, temporal and spatial models with the semantic trajectory model. This matching extends the capabilities of our approach. For more efficient semantic capabilities, we want to annotate the data with domain, temporal and spatial knowledge. These knowledge are defined by experts representing users' needs. Annotating data with these knowledge could be done automatically or manually. We cannot use a manual annotation over huge data. Therefore, we choose an automatic annotation which can be accomplished by an ontology inference mechanism. This inference mechanism derives new semantics from existing information using additional knowledge. Later in this paper, we will present this inference mechanism as sets of rules.

3.2. Semantic trajectory ontology

In [8,16], we proposed a methodology for modeling trajectory data. This methodology focused on several real cases. For each case,

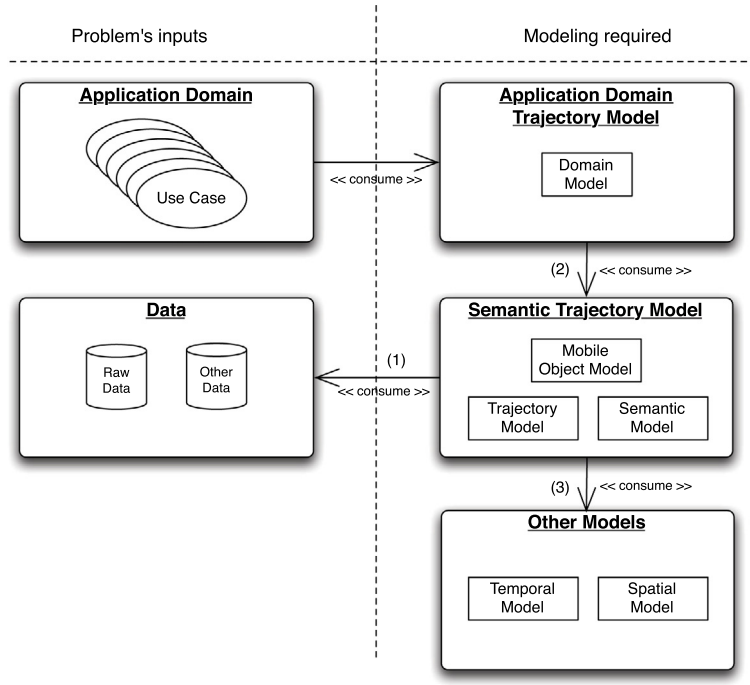


Fig. 1. Problem and its modeling required.

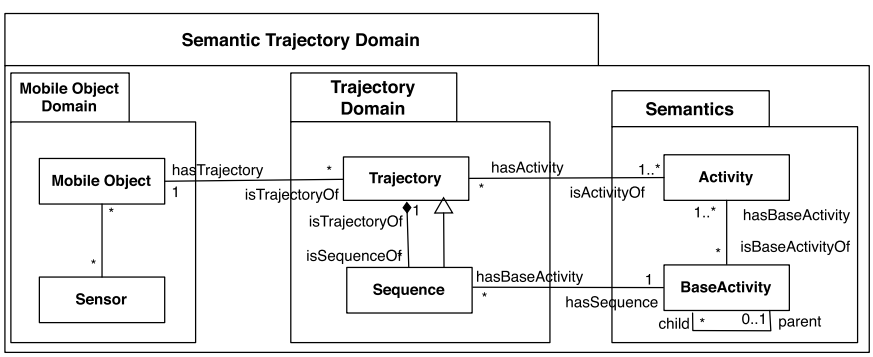


Fig. 2. Semantic trajectory modeling approach.

we define a context, data capture, an analysis process of these data, and a domain model. From these models, we define a trajectory pattern also called generic trajectory model, Fig. 2. A trajectory is a set of sequences of spatio-temporal path covered by a moving object and has an activity.

To build the trajectory ontology, we use model transformation techniques introduced by the Model Driven Engineering (MDE) community. For this, we choose an automatic transformation from UML model into a formal OWL ontology. We use transformer tool called uml2owl[1] [17]. This transformer, based on the meta-model eCore Eclipse, takes as input a UML model and turns it into OWL-DL ontology. So, we transform the trajectory data model (Fig. 2) to an OWL ontology, named `owlSemanticTrajectory`. Fig. 3 presents the declarative part of this ontology. It contains three parts: mobile object, trajectory and semantic ontologies. By definition, a trajectory is a set of spatio-temporal concepts. Spatial and temporal models can be reused to enrich description of the concepts in the trajectory ontology to represent their spatial and temporal properties. Table 1 gives a dictionary of the main concepts of the trajectory ontology.

Table 1
Dictionary classes of the trajectory domain.

Classes	Description
Trajectory	Logical form to represent sets of sequences
Sequence	Spatio-temporal interval representing a capture
GeoSequence	Spatial part of a sequence
Specific sequence	Metadata part of a sequence

4. 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[2] 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 Fig. 4 described in detail in [5]. We are mainly interested in the `ProperInterval` concept and its two properties `hasBeginning` and `hasEnd`.

[1] http://perso.univ-lr.fr/ghillair/projects.html.

[2] http://www.w3.org/2006/time.

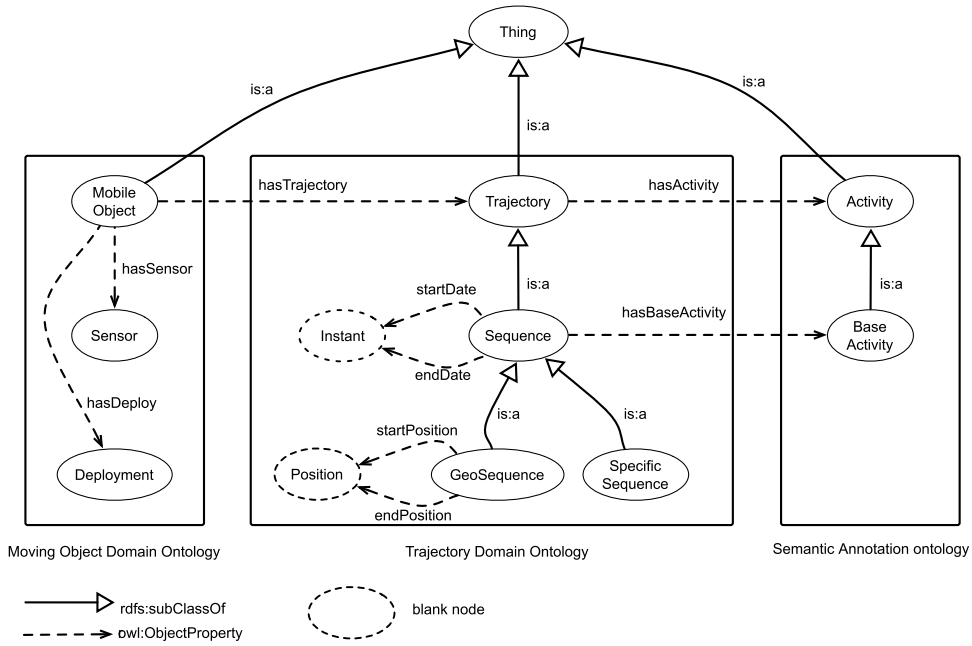


Fig. 3. A view of the semantic trajectory ontology `owlSemanticTrajectory`.

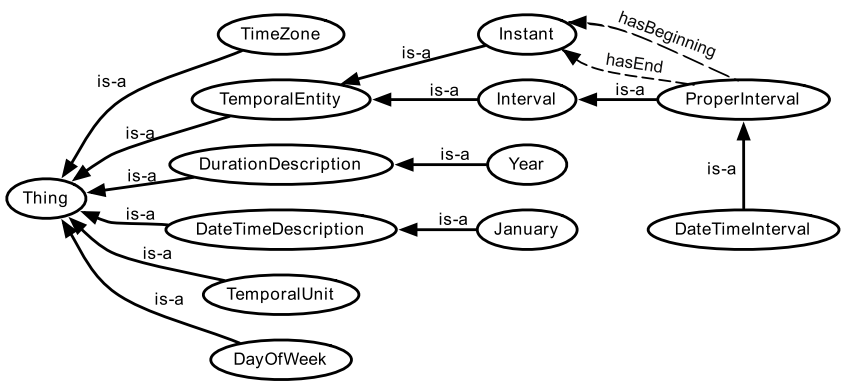


Fig. 4. A view of the OWL-Time ontology.

5. Trajectory ontology inference

Inference is the ability to make logical deductions based on ontologies, and optionally individuals. It derives new knowledge based on rules. A rule's definition, Fig. 5, has an antecedent, filters and a consequent. If knowledge are represented using RDF triples, then the antecedent is a set of triples, filters apply restrictions, and finally consequent is a new derived triple.

In the present work, we consider two kinds of inference:

1. Inference using standard rules: Our semantic trajectory ontology is based on RDF, RDFS and OWL constructs. Inference mechanism associates with each construct a rule. The results sets are called standard rules. An example of standard rules is OWL-Prime in Oracle RDF triple store [18].
2. Inference using temporal rules: Our semantic trajectory ontology uses temporal relationships as defined by Allen's algebra [19]. Each relationship is defined as a rule such as: `intervalAfter`, `intervalBefore`, `intervalDuring`, etc.

6. Trajectory ontology inference using domain rules

Our application domain is seals' trajectories, where a seal is considered as a mobile object. The captured data comes from the LIENSs laboratory[3] in collaboration with SMRU.[4] We consider three main states of a seal: `Dive`, `Haulout` and `Cruise`. Every state is related to a seal's activity, like `Resting`, `Traveling` and `Foraging`.

The captured data can also contain some meta-data called `CTD` (Conductivity-Temperature-Depth) about the marine environment such as water conductivity, temperature and pressure. Starting from our semantic trajectory ontology `owlSemanticTrajectory` we define the seal trajectory ontology, named `owlSealTrajectory`, Fig. 6. Formally, each activity is declared in the ontology and associated to a domain rule.

7. Implementation

Our implementation framework uses Oracle RDF triple store [18]. Based on a graph data model, RDF triples are persisted, indexed and queried, like other object-relational data. In this framework, we create the following models and rulebases (a set of rules):

[3] Lab. CNRS/University of La Rochelle—http://lienss.univ-larochelle.fr.
[4] SMRU: Sea Mammal Research Unit—http://www.smru.st-and.ac.uk.

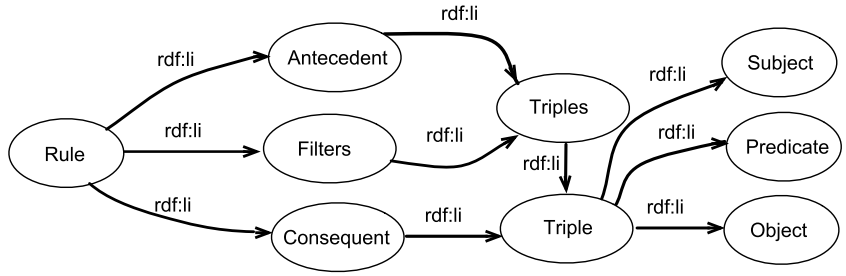


Fig. 5. Rule's definition.

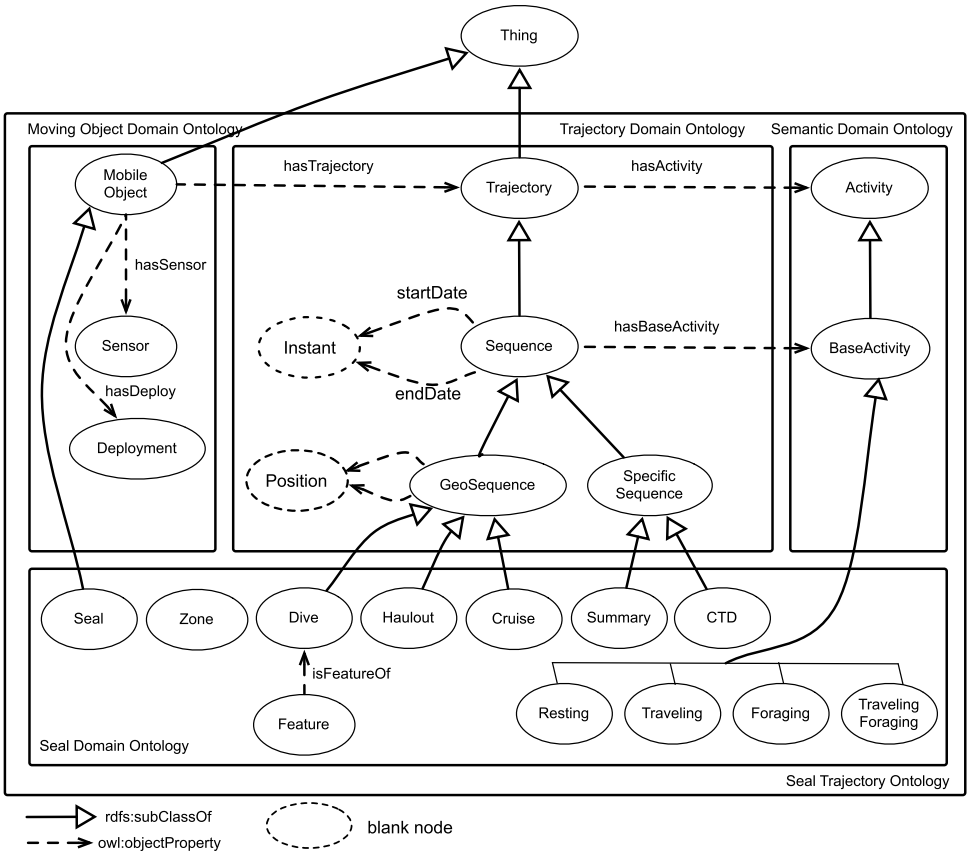


Fig. 6. Overview of the seal trajectory ontology with their activities.

- `owlTrajectory`, `owlTime` and `owlSealTrajectory`: declarative part of the trajectory, time and seal ontologies;
- `OWLPrime`: rulebase of the standard rules;
- `Time_Rules`: a rulebase of the temporal holding the interval temporal relationships. The declarative part of the `intervalAfter_rule` is presented in Fig. 7 based on operations defined in the `TM_RelativePosition` table of the ISO/TC 211 specification about the temporal schema [20].
- `Seal_Rules`: a rulebase of the seal rules. According to the domain expert, there is a correlation between the geometrical shape of dives and activities. To classify geometric shapes of dives, the `TAD` index is computed over a set of data. For this classification, we can distinguish three patterns:
  - dive shaped V: if $0 \leq TAD < 0.7$
  - dive shaped U + V: if $0.7 \leq TAD < 0.9$
  - dive shaped U: if $0.9 \leq TAD < 1$.

  In addition to the geometric shape of dives, according to the domain expert, we take into account the maximum dive depth and surface ratio which is the ratio between surface duration and dive duration. The decision Table 2 summarizes conditions of the IF parts of rules associated with activities. Based on this table, Fig. 8 gives an example of rule definition, `foraging_rule`, in the system.

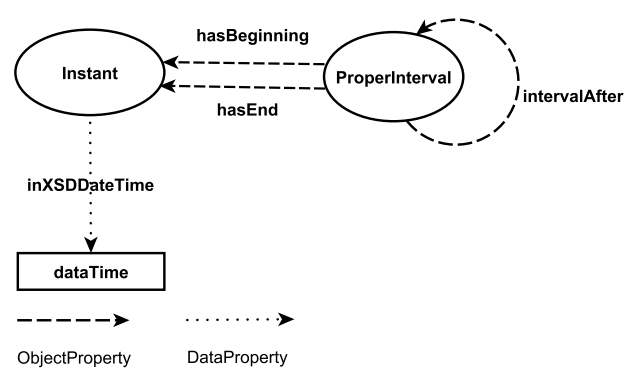


Fig. 7. Declarative part of `intervalAfter` rule.

Table 2
Decision table of IF parts of seal activities.

Rules	Max dive depth (m)	Dive shape or TAD	Surface ratio = surface dur/dive dur
Resting	<10	>0.9	>0.5
Traveling	>3	<0.7	<all
Foraging	>3	>0.9	<0.5

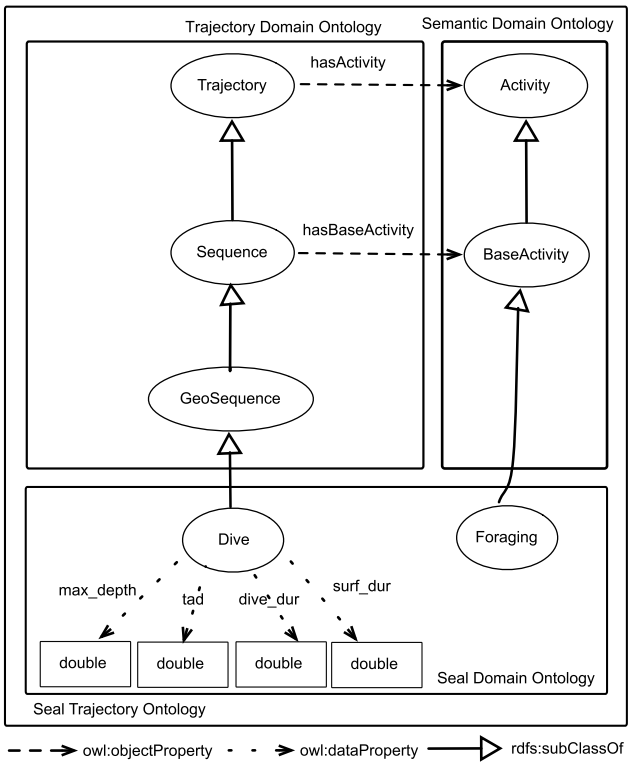


Fig. 8. Declarative part of `Foraging rule`.

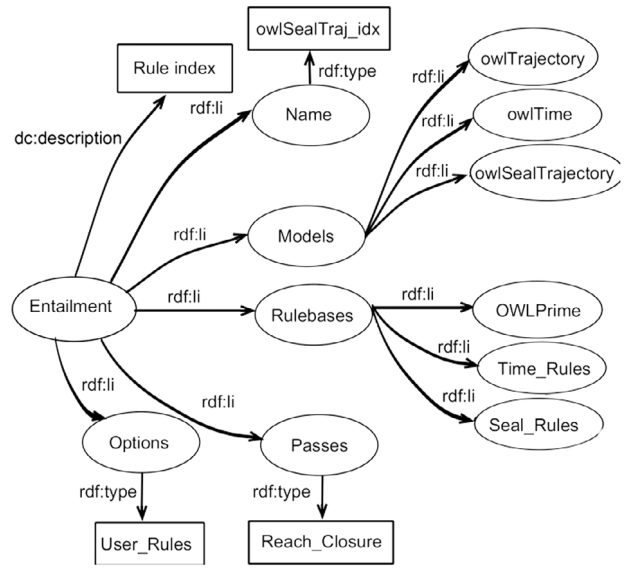


Fig. 9. Seal ontology inference: entailment.

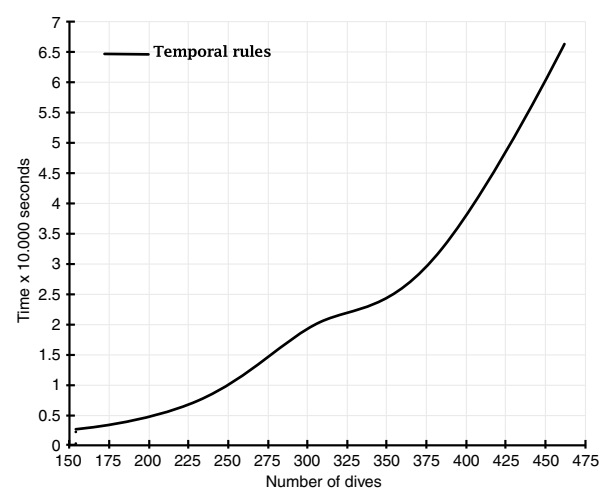


Fig. 10. Entailment computation time over the temporal rules.

In our framework, inference mechanism creates a rule index, Fig. 9. A rule index (entailment) is an object containing pre-computed triples from applying a specified set of rulebases to a specified set of models. If a graph query refers to any rulebases, a rule index must exist for each rulebase-model combination in the query. The `USER_RULES=T` option is required while applying user-defined rules. The default number of rounds that the inference engine should run is `SEM_APIS.REACH_CLOSURE`.

8. Experiments

In our experiment, we measure the time needed to compute the entailment (Fig. 9) for different sets of real trajectory data. We consider one seal's trajectory data captured from 16 June until 18 July 2011. We have 10 000 captured data. In this experiment, the seal rulebase contains only the foraging rule. The input data for this entailment are type of dives. Fig. 10 shows the experiment results for the computation time in seconds needed by the entailment. For example, for 450 dives, the inference takes around 60 000 s (≃16.6 h). Fig. 11 shows the experiment results for the number of the triples inferred by the inference mechanism. For example, for 450 dives, the inference generates around 2 200 000 triples.

9. Application domain inference refinement

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. Fig. 12 shows the two-tier inference filter refinement. The primary filter is applied to the captured data to classify them into a set of interesting places. 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 places. Then, instead of annotating each sequence in the model, we annotate each place with the expert knowledge. 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 [11] named the former `moves` (Definition 1) and the latter `stops` (Definition 2). For this reason, a trajectory is seen as a sequence of moves going from one stop to the next one.

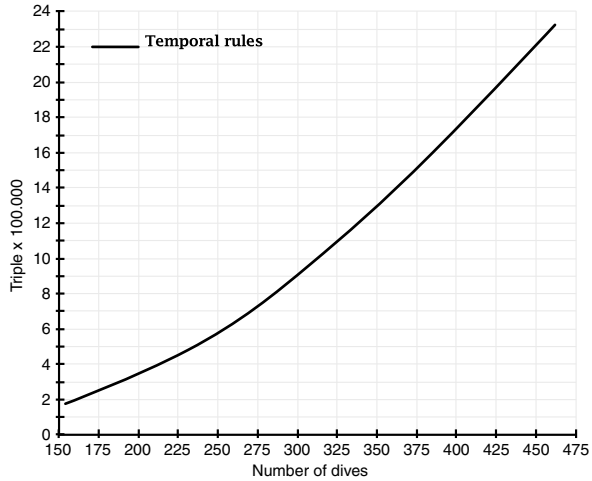


Fig. 11. Triples inferred over the temporal rules.

Definition 1 (*Move*)**.** A move is a part of a trajectory represented as a spatio-temporal line.

Definition 2 (*Stop*)**.** A stop is a part of a trajectory having a time interval and represented as a single point.

The interesting places are related to where the moving object stays more and visits more often. The definition of a place of interest is based on a neighbor of points notion.

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*)**.** For a given point (p), a *peak* is the cardinality of the list $Neighbor(p)$.

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 at least a $Neighbor_i$ and the number of the moving object's visits *nVisits*: $Places = \{(Neighbors_i, nVisits_i)_{i=1}^{n} := Neighbors_i \in Move, nVisits_i \in number\}$.

The place of interest process is given by the algorithm 1. Algorithm 1 lines 5–9 gather move data into groups of neighbors, named *Points_Neighbors*. These groups are defined with respect to a *radius*. This radius is a fixed distance between two points to calculate the neighbors. It is application dependent.

Lines 10–24 build the interesting places, named *Places*. In line 10, we consider each neighbor in the preceding calculated *Points_Neighbors*. In line 11, we consider a threshold related to the cardinality of the current neighbor. This threshold is application dependent. In lines 12–14, every point of the current neighbor that belongs to a place should be far from the stop data more than the fixed radius. According to the value of the distance between the current point and the current place, we distinguish two cases. Lines 15 and 16, the point is far from the place, so we create a new place holding the current neighbor. Else, lines 18 and 19 add the current neighbor to the current place and increase the number of visits of this place.

```
   input  : Move
   input  : Stop
   input  : radius
   output: Places
 1 initialization;
 2 Neighbor ← ϕ;
 3 Points_Neighbors ← ϕ;
 4 Places ← ϕ;
 5 for each p_i ∈ Move do
 6 |   calculate Neighbor(p_i);
 7 |   Points_Neighbors ← (p_i, Neighbors(p_i));
 8 |   Move ← Move − Neighbor(p_i);
 9 end
10 for each pn ∈ Points_Neighbors AND
11      condition(peaks_pn) do
12 |    for each p_i ∈ pn do
13 |    |  for each place ∈ Places AND
14 |    |       condition(distance(p_i, Stop) > radius) do
15 |    |    |  if distance(p_i, place) > radius then
16 |    |    |  |  new_place = new Place(pn, 1);
17 |    |    |  else
18 |    |    |  |  place.Neighbors + = pn;
19 |    |    |  |  place.nVisits     + = 1;
20 |    |    |  end
21 |    |    |  break;
22 |    |  end
23 |  end
24 end
```

Algorithm 1: The Place Of Interest algorithm

10. Research results

We consider trajectories of one seal captured from 16 June until 18 July 2011. This data set contains about 10 000 dives. To analyze our data, we pass them to the Place Of Interest algorithm. This algorithm analyzes the data and gives as output interesting places, as shown in Fig. 13.

Fig. 14 shows the evaluation of the two-tier inference refinement over real data. We evaluate the space storage consumed by the inference. For that purpose, this experiment gives number of triples generated by the temporal inference on different sets of dives. The results show its impact by the following experiments:

1. Temporal rules: this experiment analyzes the inference on real data taking into account standard temporal rules;
2. Temporal rules refined − Real data: this experiment analyzes the inference on real data considering optimized temporal rules;
3. Temporal rules refined − Generated data: this experiment analyzes the inference on generated data as in the previous experiment;
4. Temporal rules refined + 2tiers refinement − Real data: this experiment analyzes the inference on real data using optimized temporal rules and the two tier refinement algorithm.

This experiment, Fig. 14, shows that the inference generates far less triples with the two tier refinement, results 4, than the conventional inference results 1–3. The reason is that the result 4 is achieved over a filtered data. However, the filtered data should maintain the main properties of the data. So, we conclude that it is not interesting to consider all the trajectory data in the inference mechanism. In this work, the filtered data is obtained thanks to the Place-Of-Interest algorithm.

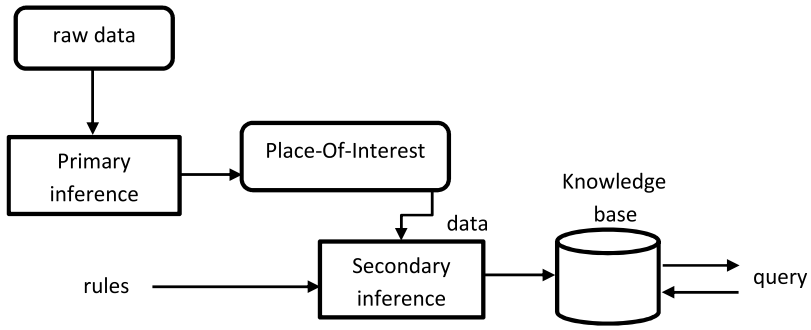


Fig. 12. A two-tier inference filter refinement.

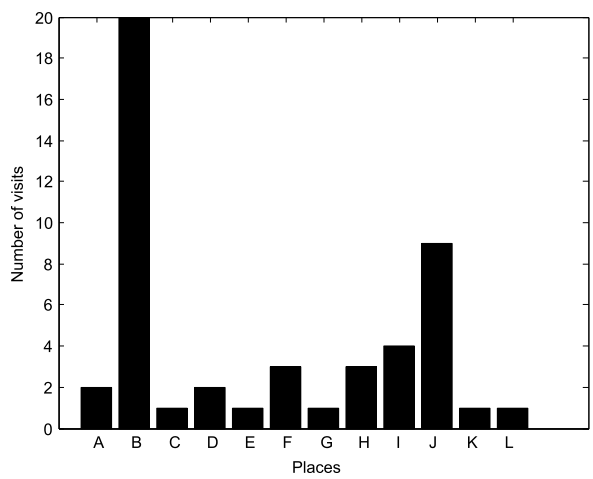


Fig. 13. Data to interesting places.

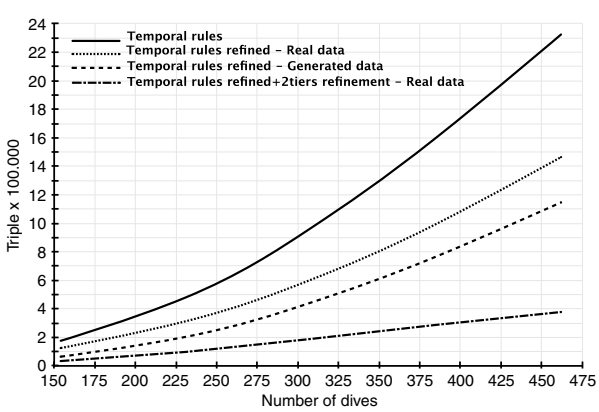


Fig. 14. Enhancement over the two-tier inference refinement.

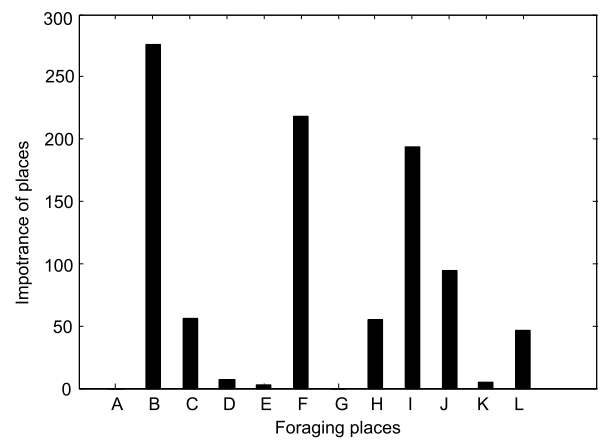


Fig. 15. Interesting places to foraging places.

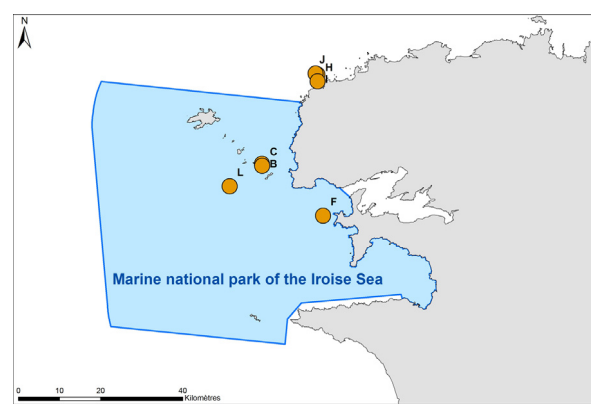


Fig. 16. Marine park in western Brittany with the evaluated foraging places.

This solution entails two main challenges:

1. The inference process on temporal intervals must be adapted for the case of the places of interest which are temporal regions in our work;
2. The definition of the places of interest is often domain dependent. It means that it is necessary to take into account domain's requirements to build the places of interest without altering data quality.

To confirm the quality of the interesting places, we query the semantic resulting data to find all the foraging places, as shown in Fig. 15. Fig. 16 is a map showing these foraging places. These obtained foraging places overlap with the majority of the fishing zones known by the biologists in this geographical zone.

11. 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. To implement the declarative and imperative parts of the ontologies, we consider the framework of Oracle Semantic Triples Store. To define the domain and temporal reasoning, we implement rules related to the considered models. The domain rules consider seals 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 inference filters to consider all the captured data.
The main contributions of this work are:

- How to use an ontological approach modeling for semantic trajectories;
- How to define inferences on semantic trajectories to answer user queries;
- What is the complexity of these inferences;
- What can we do to face and reduce the inferences costs of inference complexity.

Some directions for our future work include:

- extend the inference mechanism to take into account domain, time and spatial rules;
- study an ontology assisted users queries using OWL-DL ontologies to help end-users to formulate complex queries on trajectory data;
- study the inference complexity using different inferences engine.

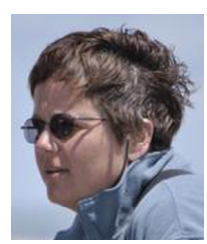
References

[1] R. Güting, M. Schneider, Moving Objects Databases, Morgan Kaufmann, 2005.
[2] R. Wannous, J. Malki, A. Bouju, C. Vincent, Time integration in semantic trajectories using an ontological modelling approach, in: New Trends in Databases and Information Systems, in: Advances in Intelligent Systems and Computing, vol. 185, Springer, Berlin, Heidelberg, 2013, pp. 187–198.
[3] J. Malki, R. Wannous, A. Bouju, C. Vincent, Temporal reasoning in trajectories using an ontological modelling approach, Control Cybernet. (2012) 1–16.
[4] R. Wannous, J. Malki, A. Bouju, C. Vincent, Modelling mobile object activities based on trajectory ontology rules considering spatial relationship rules, in: Modeling Approaches and Algorithms for Advanced Computer Applications, in: Studies in Computational Intelligence, vol. 488, Springer International Publishing, 2013, pp. 249–258.
[5] R.H. Jerry, P. Feng, An ontology of time for the semantic web, in: ACM Transactions on Asian Language Information Processing, 2004, pp. 66–85.
[6] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, 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] P. Matthew, A framework to support spatial, temporal and thematic analytics over semantic web data (Ph.D. thesis), Wright State University, 2008.
[8] J. Malki, A. Bouju, W. Mefteh, An ontological approach modeling and reasoning on trajectories. taking into account thematic, temporal and spatial rules, in: TSI. Technique et Science Informatiques, Vol. 31/1-2012, 2012, pp. 71–96.
[9] Z. Yan, C. Parent, S. Spaccapietra, 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.
[10] M. Baglioni, J. Macedo, C. Renso, M. Wachowicz, An ontology-based approach for the semantic modelling and reasoning on trajectories, in: Advances in Conceptual Modeling—Challenges and Opportunities, Vol. 5232, Springer, Berlin, Heidelberg, 2008, pp. 344–353.
[11] S. Spaccapietra, C. Parent, M. Damiani, J. Demacedo, F. Porto, C. Vangenot, A conceptual view on trajectories, Data Knowl. Eng. (2008) 126–146.
[12] L.O. Alvares, V. Bogorny, B. Kuijpers, A.F. Macedo, B. Moelans, 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, ACM, 2007, pp. 22:1–22:8.
[13] A. Boulmakoul, L. Karim, A. Lbath, Moving object trajectories meta-model and spatio-temporal queries, Int. J. Database Manage. Syst. (IJDMS) (2012) 35–54.
[14] A. Boulmakoul, L. Karim, A. Lbath, Article: A high performance scalable data collection system for moving objects, Int. J. Comput. Appl. 67 (9) (2013) 36–43.
[15] A. Boulmakoul, L. Karim, A.E. Bouziri, A. Lbath, A system architecture for heterogeneous moving-object trajectory metamodel using generic sensors: Tracking airport security case study, IEEE Syst. J. 9 (1) (2015) 283–291.
[16] R. Wannous, Computational inference of conceptual trajectory model: considering domain temporal and spatial dimensions (Ph.D. Thesis), La Rochelle university, 2014.
[17] G. Hillairet, ATL use case—ODM implementation (Bridging UML and OWL), Technical Report, 2007.
[18] Oracle, Oracle Database Semantic Technologies, 2009.
[19] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM 26 (11) (1983) 832–843.
[20] ISO/TC_211, Geographic information—Temporal schema, ISO 19108, 2002.

Rouaa Wannous is a post doctoral at the University of La Rochelle, France. She was an assistant Professor at the University Institute of Technology, La Rochelle in 2015. In 2014, she achieved her Ph.D. in Computer Science from the L3i laboratory at the University of La Rochelle, France. Main research interests include: Trajectory ontology modeling, spatio-temporal data modeling, environmental applications, ontology inference using user-defined rules.

Jamal Malki received his Post-Graduate degree in applied mathematics from the University of Paris Dauphine in 1997. In 1998, He joined the research group IMEDIA of the INRIA-Rocquencourt institute to work in the field of images retrieval by content. In 2001, he obtained his Ph.D. for research in objects spatio-temporal relationships modelling and reasoning from the University of La Rochelle, L3i laboratory. Since 2001, he has been with the computer science department of the University Institute of Technology, University of La Rochelle, where he was an Assistant Professor, became an Associate Professor in 2003. Since this date, he was a permanent member of the L3i Laboratory and his research interests include masses data analysis, modelling and querying. His works consider different cases when data can be trajectories, e-learning traces and enterprise data applications.

Alain Bouju received an engineering degree in computer science from the "École nationale supérieure d'électronique, d'électrotechnique, d'informatique, d'hydraulique et des télécommunications" (ENSEEIHT), Toulouse, France in 1990, a Toulouse M.S. degree in computer science, a Ph.D. degree in artificial intelligence from SUPAERO "École nationale supérieure de l'aéronautique et de l'espace" (as "National Higher School of Aeronautics and Space"), France, and an habilitation to direct research (HDR) from La Rochelle University, France in 2004. He is currently Associate Professor in Laboratory of Informatics, Image and Interactions (L3i). His current research interests lays in the area of the Mobility, Trajectory, Geographic Information System, Intelligent Transportation Systems and Smart Cities.



Cécile Vincent is a research lecturer at the University of la Rochelle (France) since 2002. She achieved her Ph.D. in marine biology at the University of Brest in 2001. Her research mainly focuses on the ecology of marine mammals, especially seals, as assessed from telemetry. The outcomes of this research help the management and conservation of the species in France and in Europe. For several years she collaborates with colleagues in computer science at the University of La Rochelle in order to develop new methods in the analysis of trajectories