

MASTER: A Multiple Aspect View on Trajectories

Ronaldo dos Santos Mello¹, Vania Bogorny¹, Luis Otavio Alvares¹,
Luiz H. Z. Santana¹, Carlos Andres Ferrero^{1,2}, Angelo Augusto Frozza^{1,3},
Geomar Andre Schreiner¹, Chiara Renso⁴

¹Universidade Federal de Santa Catarina - PPGCC, Florianopolis, Brazil

²Instituto Federal de Santa Catarina, Lages-SC, Brazil

³Instituto Federal Catarinense, Camboriu-SC, Brazil

⁴ISTI-CNR, Pisa, Italy

{r.mello, vania.bogorny, luis.alvares}@ufsc.br
{luiz.santana, schreiner.geomar}@posgrad.ufsc.br,
andres.ferrero@ifsc.edu.br, angelo.frozza@ifc.edu.br,
chiara.renso@isti.cnr.it

Abstract. For many years trajectory data have been treated as sequences of space-time points or stops and moves. However, with the explosion of the Internet of Things (IoT) and the flood of Big Data generated on the Internet, like weather channels and social network interactions that can be used to enrich mobility data, trajectories become more and more complex, with multiple and heterogeneous data dimensions. The main challenge is how to integrate all these information with trajectories. In this paper we introduce a new concept of trajectory, called multiple aspect trajectory, propose a robust conceptual and logical data model that supports a vast range of applications, and differently from state-of-the-art methods we propose an efficient storage solution for efficient multiple aspect trajectory queries. The main strength of the model is the combination of simplicity and expressive power to represent heterogeneous aspects, ranging from simple labels to complex objects. We evaluate the proposed model in a tourism scenario and compare its query performance against *SECONDO* extension for symbolic trajectories.

1. Introduction

For many years, moving object database research has focused on how to represent, store, and query the physical movement of objects, limiting movement data to the spatial position of an object along time. The physical movement is well known as *raw trajectory*, and is generally represented as a sequence of points $T = \langle p_1, p_2, \dots, p_n \rangle$, with $p_i = (x_i, y_i, t_i)$, $p_i \in T$, where x, y is the position of the object in space at time instant t .

With the explosion of the *Internet of Things (IoT)* and the flood of *Big Data* generated on the Internet, like weather channels and social network interactions (e.g., *Flickr*, *Facebook*, *Twitter*, *Foursquare*), it is now possible to collect huge volumes of movement data about people, animals, and objects as cars, buses, drones, etc. Sensors installed either

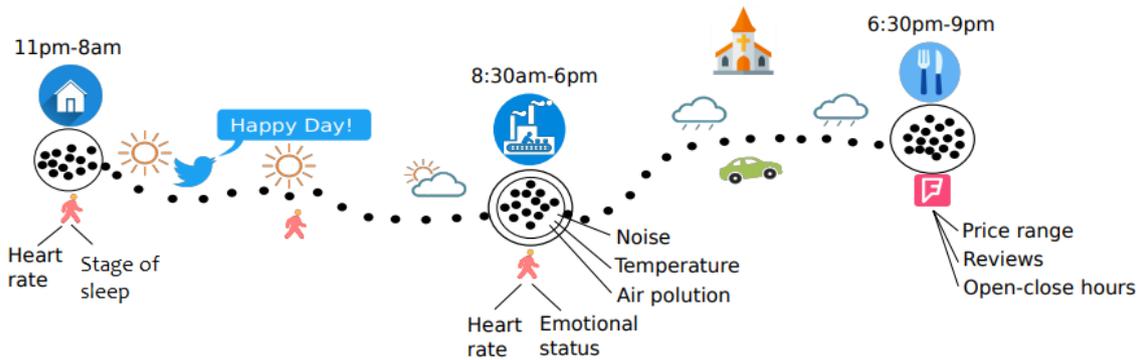


Figure 1. An example of a multiple aspect trajectory

indoor (e.g. *smart homes*) or outdoor allow the collection of data about the *place*, as temperature, air pollution, noise, luminosity, etc, or about the *object* that is moving around or inside this place, as the heart rate (with a smart watch), the emotional status (with a microphone that analyses the voice intonation), blood pressure, sleeping stages, etc. By collecting all these information we have a new type of movement data, i.e., a trajectory enriched with different semantic aspects. As an example, let us consider the trajectory of *Peter*, shown in Figure 1.

Peter stays at *Home* from 11pm to 8am, when he goes to *Work* from 8:30am to 6pm, and finally he goes for dinner at a *Japanese Restaurant*. Peter has a smart watch that constantly collects his heart rate, and his sleeping stages at home during the night. Peter goes to work on foot when the weather is sunny, and he tweets a message that characterizes his humor. His working place is a smart office equipped with numerous sensors that collect information about the place as noise, temperature, air pollution, and humidity, as well the emotional status of the staff using microphones. After work, the weather changes to rainy, and Peter decides to take a taxi to move to a Japanese Restaurant. The restaurant has its own attributes, as open and close hours, average price, spatial location, and reviews.

We can observe from Figure 1 that a trajectory became a complex object with numerous complex data dimensions that are contextual to the movement and heterogeneous in the form, which we define in this paper as *aspects*. The more aspects we have, the more complete is the representation of the real movement of an object, and more useful and interesting information we can infer about objects and places. The challenge is how to integrate all these heterogeneous information in a single trajectory representation, and the main questions we want to answer in this paper are: (i) *Is it possible to define a data model that is simple in structure, but generic enough to represent any aspect related to the movement, and that covers a large number of applications?* (ii) *Is there a way to efficiently query and extract patterns from data represented in this model?*

A few data models were proposed to consider some limited semantic information about movement, as the well known model of stops and moves [Spaccapietra et al. 2008] and the generalized CONSTANT model [Bogorny et al. 2014]. Based on these data models, works as [Fileto et al. 2015] and [Nogueira et al. 2017] propose frameworks and ontologies to enrich trajectories with Linked Open Data. Existing trajectory data models

essentially add semantic labels to trajectories, as the means of transportation, the goal of the trip, and the category of the visited places [Bogorny et al. 2014]. However, these models have some limitations. First, they focused on the conceptual representation only, and do not go deeper in the logical level and storage technologies. Furthermore, they do not consider dynamic and complex aspects that involve movement, not being flexible to represent heterogeneous aspects.

We claim that multiple aspects represent a new view over trajectories, and a new paradigm concerning mobility data. These aspects are not simple semantic labels, but may be complex objects and/or heterogeneous information intrinsically associated to the physical traces of the moving objects.

In this paper we introduce the concept of multiple aspect trajectory and propose a novel approach for modeling this kind of trajectories called *MASTER*. *MASTER* comprises a conceptual and a logical data model for multiple aspect trajectories, as well as a storage solution that is very appropriate for multiple aspect trajectory queries.

The main contributions of this paper are summarized as follows: *(i)* we introduce the concept of multiple aspect trajectory that generalizes the state-of-the-art semantic trajectory definition into a more complex but realistic trajectory based on the notion of aspect. We define different kinds of aspects and we also discuss the semantic meaning of an aspect to distinguish between different possible meanings for an aspect name. We provide a conceptual data model for multiple aspect trajectories with a large expressive power, that combines simplicity with a powerful representation of different types of moving objects and a variety of spatial, temporal and semantic aspects that are relevant to a vast range of applications; *(ii)* a conversion of the conceptual data model to a logical schema in the Resource Description Framework (RDF) standard to be adherent to the *Semantic Web* standards [Berners-Lee et al. 2001]; *(iii)* the usage of a triplestore based on NoSQL databases for maintaining RDF data, since these databases represent a new and efficient technology to maintain and query trajectory data *(iv)* a twofold evaluation, where we first analyze the expressive power of *MASTER* for representing multiple aspect trajectories in a tourism application and discuss different kinds of queries over multiple aspect trajectories with increasing complexity, as well as a query performance comparison with the well know spatio-temporal database *SECONDO* extended for symbolic trajectories [Güting et al. 2015].

The rest of this paper is organized as follows. Section 2 introduces multiple aspect trajectories and details the *MASTER* approach. Section 3 is dedicated to the *MASTER* evaluations. Section 4 presents the related work and Section 5 concludes the paper.

2. The *MASTER* Approach

This section presents our approach for modeling multiple aspect trajectory data called *MASTER*. We define a conceptual and a logical model for multiple aspect trajectories, and we also detail the adopted storage solution.

The main strength of our conceptual model is the combination of simplicity and the expressive power for representing aspects. An aspect may be related to a moving object, to the entire trajectory or a trajectory part, and may hold any type of data, ranging from simple labels to complex objects. It is important to emphasize that the focus of

this paper is not on how to obtain these heterogeneous aspects, which may be provided by specific extraction and transformation tools, but on how trajectory data are related to these aspects. Our focus is on how to represent any aspect data, independent of the application domain, in a single and simple model.

For the logical model, we consider a graph-based representation (the RDF standard [Pan 2009]) that is generic enough to model trajectories and aspects extracted from heterogeneous data sources, like geolocated structured record files and geolocated social media posts (e.g., tweets). Finally, we consider NoSQL databases for efficient storage and retrieval of large amounts of trajectory data. Our inspiration comes from the *polyglot persistence* approach [Sadalage and Fowler 2013], which states that a conceptual data model can be split and mapped to several database models for maximizing query performance.

The following subsections detail the MASTER approach.

2.1. The Conceptual Model

In this section we introduce a conceptual data model for multiple aspect trajectories, which is shown in Figure 2. We start the description of the model with the new concept introduced in this paper: an *aspect*.

An aspect is a real world fact that is relevant for trajectory data analysis, and it is characterized by an *aspect type*. For instance, the aspect *train* belongs to an aspect type *transportation mode*, and an aspect *rain* belongs to an aspect type *weather condition*. An aspect type has a set of attributes and it may also be a subtype of a more general aspect type, allowing the modeling of an aspect type *subtypeOf* hierarchy, like *POI ← accommodation ← hotel*. More formally, an aspect type is defined as follows.

Definition 1 An Aspect Type $asp_{type} = (desc, ATT, asp_{supertype})$ is a categorization of a real-world fact with a description *desc*, a set of attributes $ATT = \{a_1, a_2, \dots, a_k\}$ that hold its properties, and a (possibly empty) supertype aspect $asp_{supertype}$.

An aspect type and its attributes act as a metadata definition for an aspect. As a consequence, an aspect is always related to an aspect type and its attributes. For example, given an aspect type *weather condition*, some of its attributes could be *temperature*, *wind speed* and *climate*. In the following, we define an aspect.

Definition 2 An Aspect $asp = (desc, SAT)$ is a relevant real-world fact, where *desc* is the aspect description, and $SAT = \{sat\}$ is a set of aspect types that the aspect may hold, being $sat_i = (asp_{type_k}, ATV_k)$, $sat_i \in SAT$, a tuple with an aspect type asp_{type_k} and a non-empty set $ATV_k = \{a_1 : v_1, a_2 : v_2, \dots, a_n : v_n\}$ of attribute-value pairs so that each pair $(a_i : v_i) \in ATV_k$ is an instantiation of a property a_i of asp_{type_k} with a (atomic or multivalued) value v_i .

An aspect definition supports numbers, ranges, text, geometries (when an aspect describes, for example, the shape of a hurricane at a specific time instant), or any type of complex object. For sake of understanding, we give some examples of aspects, aspect types and their attributes. An aspect type *hotel* may have the following attributes: *geographic coordinates*, *address*, *stars*, *types of rooms* (multivalued), and *facilities* (multivalued). An aspect related to this type could be *Il Campanario Resort* with the following attribute-values: *geographic coordinates*: -27.439771, -48.500802; *address*: Buzios Ave.,

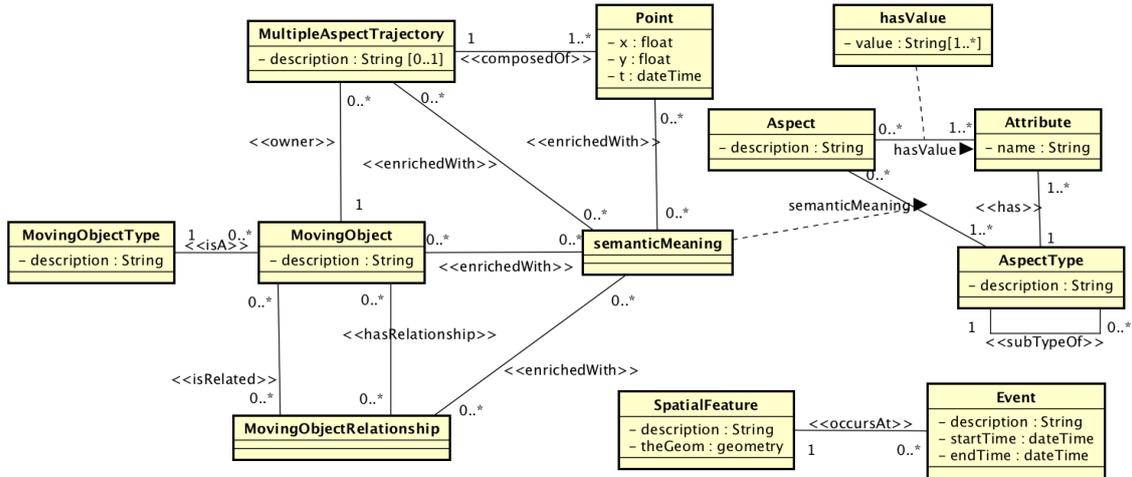


Figure 2. The Conceptual Model for Multiple Aspect Trajectories

Florianopolis; *stars*: 5; *types of rooms*: {suite, suite junior}; *facilities*: {gym, swimming pool, restaurant, bar, beach service}. Another aspect type *mood* may have the attributes *emoticon* and *intensity*. An aspect of this type could be *happy* with the attribute-values: *emoticon*: :-D; *intensity*: high.

One aspect may hold several meanings in the real world. For instance, an aspect *Sao Paulo* may have the meanings of *town*, *state*, *soccer team* or the *holy Sao Paulo*. These meanings are categorized by a specific aspect type. As the same aspect may hold different meanings, if we relate the aspect directly to a trajectory or a trajectory point, we lose this meaning. In order to solve this problem, we introduce the concept of *semantic meaning*. The *semantic meaning* represents the relationship between an aspect and an aspect type, as shown in Figure 2. It provides the exact meaning of the aspect related to a trajectory or some of its points. In the following we define a *semantic meaning* more formally, which could hold (*Sao Paulo, Town*) as an instance example.

Definition 3 A *Semantic Meaning* $SM = (asp, asp_{type})$ is an association between an aspect *asp* and an aspect type asp_{type} that gives the context of the aspect, so that asp_{type} belongs to the aspect types of the aspect *asp*.

An aspect with its semantic meaning can be associated to a multiple aspect trajectory, a trajectory point, a moving object, or a relationship between moving objects in the conceptual model (see Figure 2). When an aspect varies frequently during the object movement, the aspect with its semantic meaning is associated to each trajectory point and it is called *volatile aspect* (VA). Some examples are the *visited places* (or stops) and the heart rate. An aspect is also associated to a point when it represents a sparse and instant happening, like a social media post or check-in. When an aspect does not change during an entire trajectory, it is called a *long term aspect* (LTA), and is associated to the multiple aspect trajectory. Examples of this kind of aspect are the *town* on which the trajectory occurs or the person *occupation*. When an aspect holds during the entire life of an object, it is called a *permanent aspect* (PA) and is associated to the object and not to the trajectory. One example is the person *birthplace*. These aspect categories are directly related to the query performance. Queries on volatile aspects, *i.e.*, queries related to trajectory points,

will be more time consuming, while long term and permanent aspects will be retrieved more quickly.

Based on these foundations, we now define a multiple aspect trajectory.

Definition 4 A Multiple Aspect Trajectory $mat = (P, S_LTA, mo, desc)$ is a sequence of points $P = \langle p_1, p_2, \dots, p_n \rangle$ of a moving object mo , a (possible empty) set of long-term aspects S_LTA , being $S_LTA = \{SM_{lta}\}$ a set of semantic meanings, and a description $desc$, with $p_i = (x_i, y_i, t_i, S_VA)$, $p_i \in P$, being x and y the spatial position of mo at the time instant t , and S_VA the set of volatile aspects related to p_i , where $S_VA = \{SM_{va}\}$ is a set of (possible empty) semantic meanings.

A multiple aspect trajectory belongs to a moving object. A *moving object* is any entity that moves along space and time. This object is always associated to a type, which can be a person, a drone, an animal, a car, or even a natural phenomenon, like a hurricane. We formally define it in the following.

Definition 5 A Moving Object $mo = (mo_{type}, desc, S_PA)$ is an entity that can physically move in space and time, having a description $desc$, a set of (possible empty) permanent aspects S_PA , being $S_PA = \{SM_{pa}\}$ a set of semantic meanings, and a type mo_{type} that categorizes it.

A new feature in MASTER when compared to state-of-the-art data models for trajectories is the *moving object relationship*. A moving object may hold any type of relationship with other objects, and these relationships may also be characterized by different aspects such as the type of relationship (e.g. *friendship, professional, family*). We define a moving object relationship in Definition 6.

Definition 6 A Moving Object Relationship $mor = (mo1, mo2, S_RA)$ is a relevant association between two moving objects $mo1$ and $mo2$ that holds a (possible empty) set of relationship aspects S_RA , being $S_RA = \{SM_{ra}\}$ a set of semantic meanings.

Finally, we model *spatial features* and *events*. The first one denotes any relevant POI that is not spatially related to trajectory points, so it is not an aspect. Instead, it means any POI located in the trajectory neighborhood, like a nearby restaurant. In Figure 1, an example of spatial feature is the church located between the POIs *work* and *restaurant*. So, when a trajectory point intersects a relevant POI, it is modeled as an aspect, and as spatial feature otherwise. Spatial features are useful for answering spatial queries like *which are the restaurants located at a distance less than α from the trajectory of object A?* or *which trajectories are located inside a town B?* Similarly, an event denotes a happening that does not have a relationship with trajectories, but it is relevant for queries that investigate events in the trajectory neighborhood. An event occurs at a spatial feature and is valid for the period that it happened.

It is worth mentioning that in MASTER we do not define the concept of *subtrajectory*, as in other data models like CONSTANT [Bogorny et al. 2014]. Instead, our purpose is to model multiple aspect trajectories at the lowest granularity level for all aspects, allowing any possible generalization later during an analysis phase, *i.e.*, we consider the segmentation process as an analytical step. Another important point is that our model is expressive enough to describe both sparse and dense trajectories.

The idea behind MASTER is to represent trajectories for any type of application, not being limited to a subset of semantic information as stops, i.e, the POIS, or information inferred from the raw trajectory as speed, acceleration or transportation mode. The variety and the amount of aspects will depend on what is important for the application. In tourism applications, for instance, one can model all aspects related to the tourists such as: the visited POIS, their categories, prices, and reviews, the transportation means, the social status of the tourist (that can be inferred from the types of visited POIS), the weather conditions in order to find POI visits influenced by weather conditions, the general mood or opinions of the tourist in relation to the town or local services, etc. In a smart city application one may want to add the aspects of home and work, working hours, transportation mode, social status, weather condition, etc. In recommendation applications one may want to represent the moving objects and their physical relationships (when objects have encounters detected in GPS trajectories), or virtual relationships in social media, the opinions about things and places, etc. In a bird migration application one can model the bird trajectories with aspects being the regions where they fly, regions where they eat, relationships with other species where they eat or rest, the temperature of the place, the types of vegetation where they eat or rest, etc.

2.2. The Logical Model

In this section we present the MASTER logical model for the conceptual model introduced in the previous section. As stated before, we adopt the *Resource Description Framework (RDF)* [W3C 2018a] as our logical data model because RDF data can be modeled as a graph, which is a flexible data structure to represent the high heterogeneity of possible aspects, as well as the great number of aspect relationships with trajectories, points and moving objects. Besides, on using RDF we are consonant with the *Semantic Web* standards of WWW Consortium (W3C) for publishing and manipulating data on the Web [W3C 2018b]. Our intention is to further let publicly available trajectory data as *Linked Open Data (LOD)* in RDF format to be accessed through the *Semantic Web* existing tools.

RDF is expressed by triples that define a relationship between two resources, where the resources, called *subject* and *object*, are nodes, and the relationship, called *predicate*, is a directed edge from the subject to the object. For instance, we can define a predicate *:has* between two resources: *:aspectType* (subject) and *:attribute* (object).

Figure 3 shows the proposed logical model, where dotted arrows represent an entity-attribute relationship, continuous arrows represent relationships between entities, and the ellipsis represent entities or attributes. A predicate label followed by a cardinality pair denotes a multivalued relationship. One example is a *point* that may be enriched with zero to several *semantic meanings*. An RDF triple schema in such a modeling is represented by two ellipsis connected by an arrow. One example is a *moving object* (subject) that is the *owner* (predicate) of a *multiple aspect trajectory* (object).

The conversion of the conceptual model to a logical schema in RDF was inspired by the works of [Choi et al. 2013, Bagui and Bouressa 2014, Daniel et al. 2016], which propose the following mapping rules:

- an *entity* is converted to a *node*;

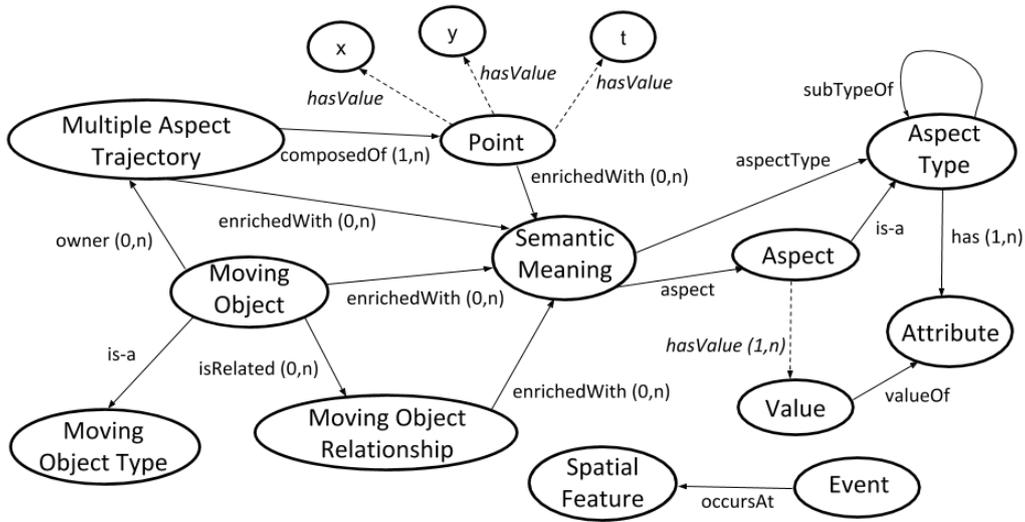


Figure 3. The Logical Model for Multiple Aspect Trajectories

- an *attribute* of an entity (or relationship) ER is converted to a *node* A , and an *edge* is defined from ER to A in order to connect them¹;
- a *relationship* between entities is converted to an *edge* that connects the entities, or an *intermediate node* between the entities.

There is only one rule for the mapping of entities and attributes, so their conversion is straightforward. Even entities without attributes, like *MovingObjectRelationship* (see Figure 2)), became nodes in the logical model because they have relationships with other entities.

Different from the mapping of entities and attributes, there is no consensus for the mapping of relationships. In this case, we decided to consider the conversion to an edge if the relationship has no properties related to it, and the conversion to a node otherwise. On doing so, we avoid the generation of too many nodes in the RDF schema, and only the relationships that hold *semantic meaning* and *hasValue* as properties (see Figure 2) were mapped to nodes, and the second one was renamed to *Value* for sake of understanding. We also decide to maintain only the connections of *Aspect* and *Attribute* with the *Value* node to avoid a redundant edge between *Aspect* and *Attribute*.

Another logical design decision is related to the edge direction in the mapping of binary relationships. As a relationship is not directed at the conceptual level, we may think of two mapping alternatives: (i) the definition of two predicates; or (ii) the definition of one predicate to represent one of the two possible relationship directions. Alternative (i) provides a better understanding of the relationship semantics in both directions, but the logical modeling becomes more complex and requires an integrity constraint control to maintain cross references consistent. On the other hand, alternative (ii) may reduce the readability, but provides a more simple logical modeling and avoids consistency checking. We adopt alternative (ii) giving preference to the relationship direction that is more

¹We define a label *hasValue* for this predicate to identify it as an entity-attribute or relationship-attribute connection.

probable to be traversed according to the expected workload for queries over multiple aspect trajectories besides reducing the size of the logical schema. Nevertheless, SPARQL queries are able to traverse RDF graphs in both edge directions, being not limited to the predicate direction.

2.3. The Storage Solution

In this section we present the adopted storage solution for maintaining data represented in the MASTER logical model. This solution is called *Rendezvous* [Santana and dos Santos Mello 2017]. *Rendezvous* is a triplestore based on NoSQL databases for querying large RDF datasets. NoSQL databases have been proposed for managing big data efficiently [Sadalage and Fowler 2013]. Therefore, as multiple aspect trajectories are highly heterogeneous and multidimensional, NoSQL databases are a suitable storage resource for this new type of trajectory. Compared to related work, *Rendezvous* was chosen due to its multimodel NoSQL support for storing RDF data and its efficient processing of typical SPARQL queries.

Rendezvous manages RDF data in a distributed database architecture that is flexible enough to store RDF data on multiple NoSQL databases, with different data models, according to the application workload. *Rendezvous* adopts SPARQL and its extension to query geographic data called GeoSPARQL [OGC 2018]. SPARQL queries can be categorized into *star*, *chain* and *complex* queries. These query categories depend on the location of the variables in a query condition. The *star* category is characterized by queries that retrieve a node based on filters over nodes very close to it (we consider a traversal of at most two predicates). *Chain* queries are composed by a long chain of joins (distance of three predicates or more) between nodes. *Complex* queries are combinations of the two previous types with (optional) additional conditions with simple filters.

As a workload-aware triplestore, *Rendezvous* stores RDF triples that occur typically in star-shaped queries in NoSQL document databases. This kind of NoSQL database model is suitable to store and retrieve complex data with all related properties, like a trajectory and its related aspects. Indeed, RDF triples occurring typically in chain-shaped queries are stored in NoSQL graph databases, which benefits from keeping the subgraph that composes this chain besides the high performance for graph traversal operations.

On considering multiple aspect trajectories, *Rendezvous* is able to store the moving object, the trajectory and the aspect data together in one or more NoSQL databases if they are frequently filtered and/or retrieved in SPARQL queries. For example, Figure 4 (a) shows part of an RDF graph that maintains data about a person trajectory (MAT_x) and its related semantic meanings (the mean of transportation, city of location and person mood aspects). This RDF graph is in accordance to our logical model defined in the previous subsection, and it has a star-shape whose center is MAT_x . Supposing that this shape is commonly requested in queries that intend to retrieve city trajectories made by bus where the person is happy, this star-shaped RDF fragment is stored in a NoSQL document as a typical JSON document (see Figure 4 (b)). The document stores together the trajectory data as well as the data that are relevant to the semantic meanings, providing a fast retrieval for this kind of query as only one document must be accessed.

Another example is shown in Figure 5 (a). It presents a chain-shaped RDF fragment denoting that *John* has a trajectory that passed by the *Eiffel tower*. Supposing again that this shape is frequently requested in queries that need to retrieve people that visit this

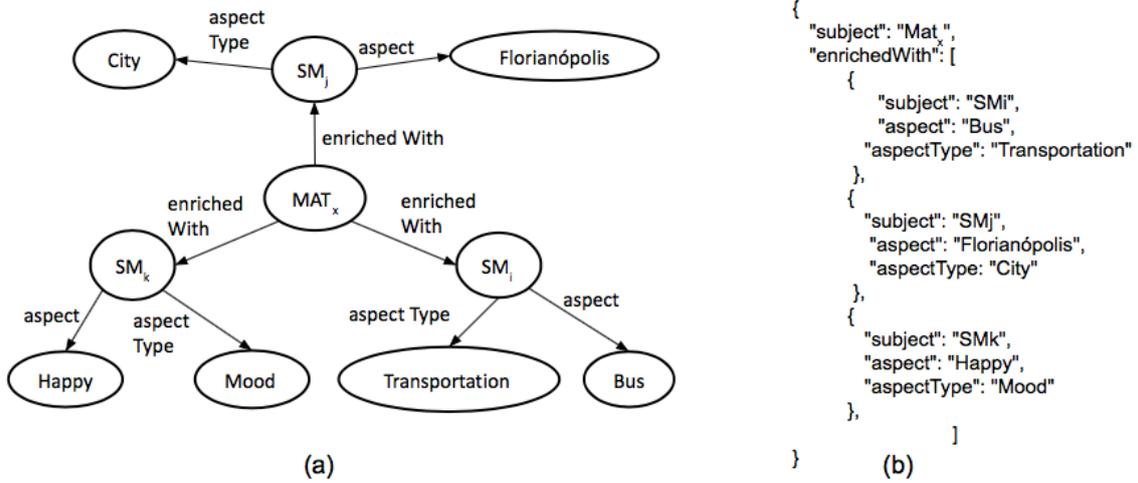


Figure 4. A star-shaped RDF instance (a) and its storage in a NoSQL document database (b)

touristic point, this chain-shaped RDF fragment is stored in a NoSQL graph document as a summarized graph composed by the first and the last RDF fragment nodes and an edge with a property that holds the ordered list of covered nodes and edges in the chain (see Figure ??). This summarized stored graph also provides a fast data access and retrieval for this kind of query.

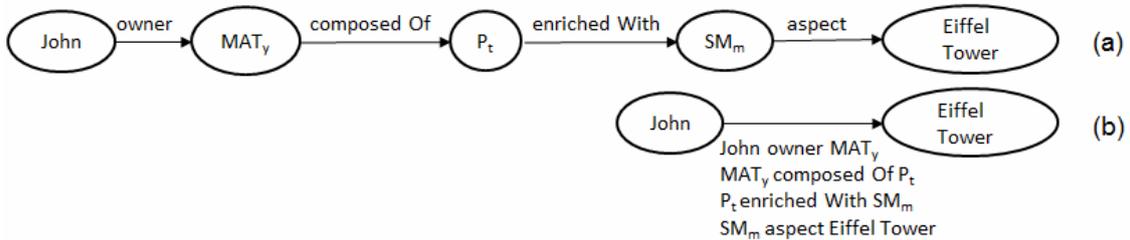


Figure 5. A chain-shaped RDF instance (a) and its storage in a NoSQL graph database (b)

For both previous examples, *Rendezvous* manages in-memory indexes that maintain the identification of the nodes present in star-shaped RDF fragments, as well as the identification of the nodes and edges occurring in chain-shaped RDF fragments. These indexes, as well as indexes that indicate the RDF partitions in one or more NoSQL document or graph databases where these RDF fragments are stored, allowing an efficient processing of the queries that compose the typical application workload.

After detailing our strategy for storing trajectories modeled according to the MASTER data model, in the next section we present a MASTER evaluation.

3. A MASTER Evaluation

The MASTER objective is to propose an expressive data model for multiple aspect trajectories, as well as an efficient solution for storing and querying multiple aspect trajectories. With this objective in mind, we evaluate MASTER over two perspectives: (i) a qualitative analysis at the conceptual level by modeling a tourism application, similarly to the

approach used in [Bogorny et al. 2014] to evaluate the CONSTANT model, as well as an evaluation at the logical level to attest that an RDF-based storage strategy is suitable to answer the main types of multiple aspect trajectory queries; and (ii) a quantitative evaluation at the storage level by comparing the query running time performance of our storage solution with a baseline. With this experiment we show the feasibility of MASTER as a data model that can be efficiently stored and accessed. These evaluations are presented in the following section.

3.1. Qualitative Evaluation

In this section we model a tourism application with MASTER and present the main categories of queries related to multiple aspect trajectories.

3.1.1. Instantiating MASTER in a Tourism Application Scenario

In order to show how to use MASTER for modeling a tourism application, let us assume that tourists track their visiting experience in Paris with smartphones, and that these data are later enriched with different information collected from many data sources including social media interactions like Foursquare, Facebook, Flickr, and Twitter. These trajectories have details about the visited POIs (obtained from Foursquare checkins as the hotel name, stars and price; restaurant name, price, and if accept credit cards; museum name and type; etc.), photos and comments posted on Flickr and Facebook, and the messages posted on Twitter by the tourists.

Figure 6 shows an example of part of a trajectory generated by John Smith, who starts his tour in Paris at Ritz Paris hotel. From the hotel he goes to the Louvre museum where he tweets and posts a photo on Flickr, just before entering the museum, where he turns his smartphone off. After leaving the museum, he took a picture of the Arc de Triomphe du Carrousel, that was posted on Facebook, and crossed the Jardin des Tuileries going to the Le Baudelaire restaurant, where he had lunch.

Figure 7 shows an example of the MASTER instantiation in RDF, presenting a very small part of two trajectories, John and his sister Mary. We consider two trajectories just for an example, because different from state-of-the-art, MASTER supports the definition of relationships between moving objects. In this Figure, colors highlight permanent aspects (gray nodes), long term aspects (pink nodes) and volatile aspects (blue nodes), respectively. The model shows trajectories of two moving objects, (*John* and *Mary*), of type *person*, which have a relationship that is enriched with an aspect *brother* of type *kinship*. This is represented by the semantic meaning *SM4*. *Mary* has a multiple aspect trajectory *MAT1* with two volatile aspect types connected to its points *p11* and *p12* (*Mood and Transportation*). They denote that part of *Mary* movement was performed by *train* and that she was *happy* at point *p11*. *MAT1* also holds a *long term* aspect type *occupation*, which means that *Mary* was *retired* for the entire trajectory.

On the other hand, *John* was born in *Florianopolis* (*SM5*) and his *gender* is *male* (*SM6*). His trajectory *MAT2* has two volatile aspects: *SM7* indicating that at points *p121* and *p122* he was at the *Ritz Paris hotel*, that is a 5 stars and expensive hotel, and *SM3* indicates that he was *happy* at point *p923* when he posted on Facebook a photo and the message "I love Paris!". Besides, it is important to notice that the aspect *Florianopolis* has

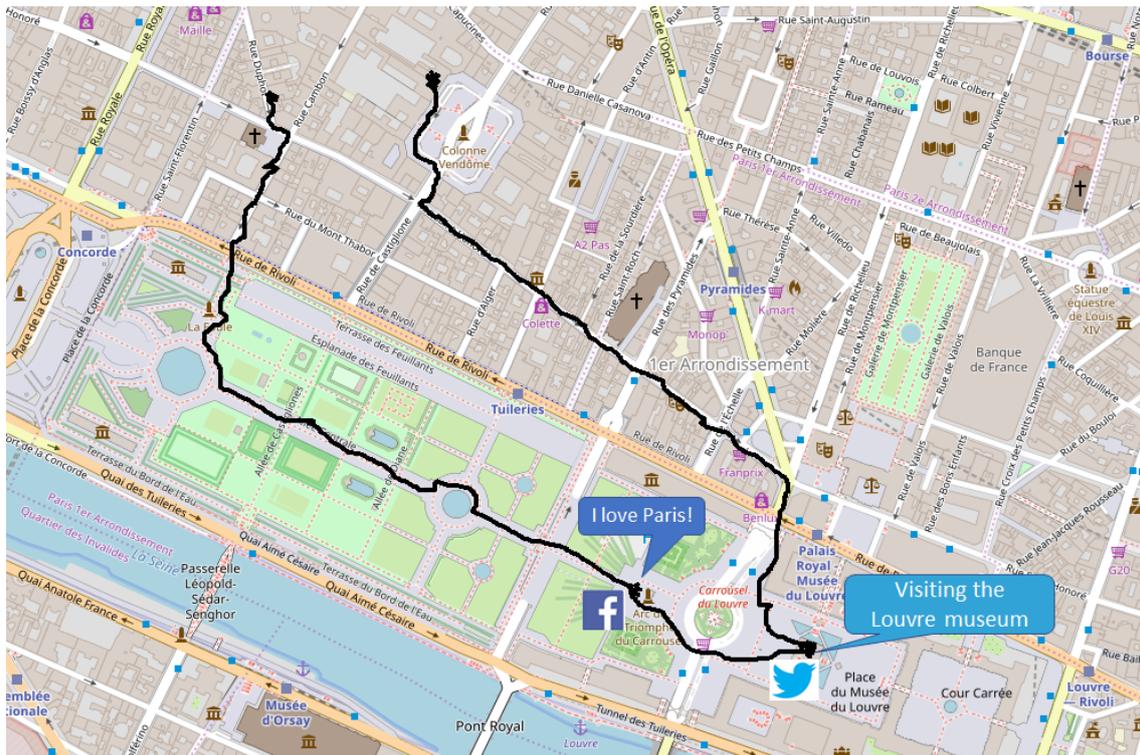


Figure 6. Example of a tourist trajectory in Paris

two semantic meanings: *town* and *birthplace*. For the context of *John*, *Florianopolis* is his *birthplace*. This example highlights the relevance of the semantic meaning modeling in MASTER.

3.1.2. Querying Multiple Aspect Trajectories

We claim that three main types of queries can be posed to multiple aspect trajectories: (i) queries that return *moving objects* (e.g., *which are the moving objects that were born in Florianopolis and are male?*); (ii) queries that return *trajectories* (e.g., *which are the trajectories that stayed at an accommodation place?*); and (iii) queries that return *aspects* (e.g., *which accommodations were visited in Paris by persons that were born in Florianopolis and are male?*). These three queries are examples of *star-shaped*, *chain-shaped* and *complex* queries, respectively. Table 1 shows these queries written in SPARQL or GeoSPARQL with an arbitrary complexity depending on the number of entities that must be considered to generate the query result, which allows an efficient processing by our RDF storage solution (see section 2.3).

The query examples shown in Table 1 are formulated over the MASTER instantiation presented in Figure 7. The first query (Table 1(a)) exemplifies a star-shaped query. In this case, the star center is a *moving object* entity and the related filters refer to aspects and the moving object type entities. As stated before, this kind of query can be efficiently processed by NoSQL document databases that may maintain moving object documents where the related entities are the document attributes that can be filtered. This query returns *John*.

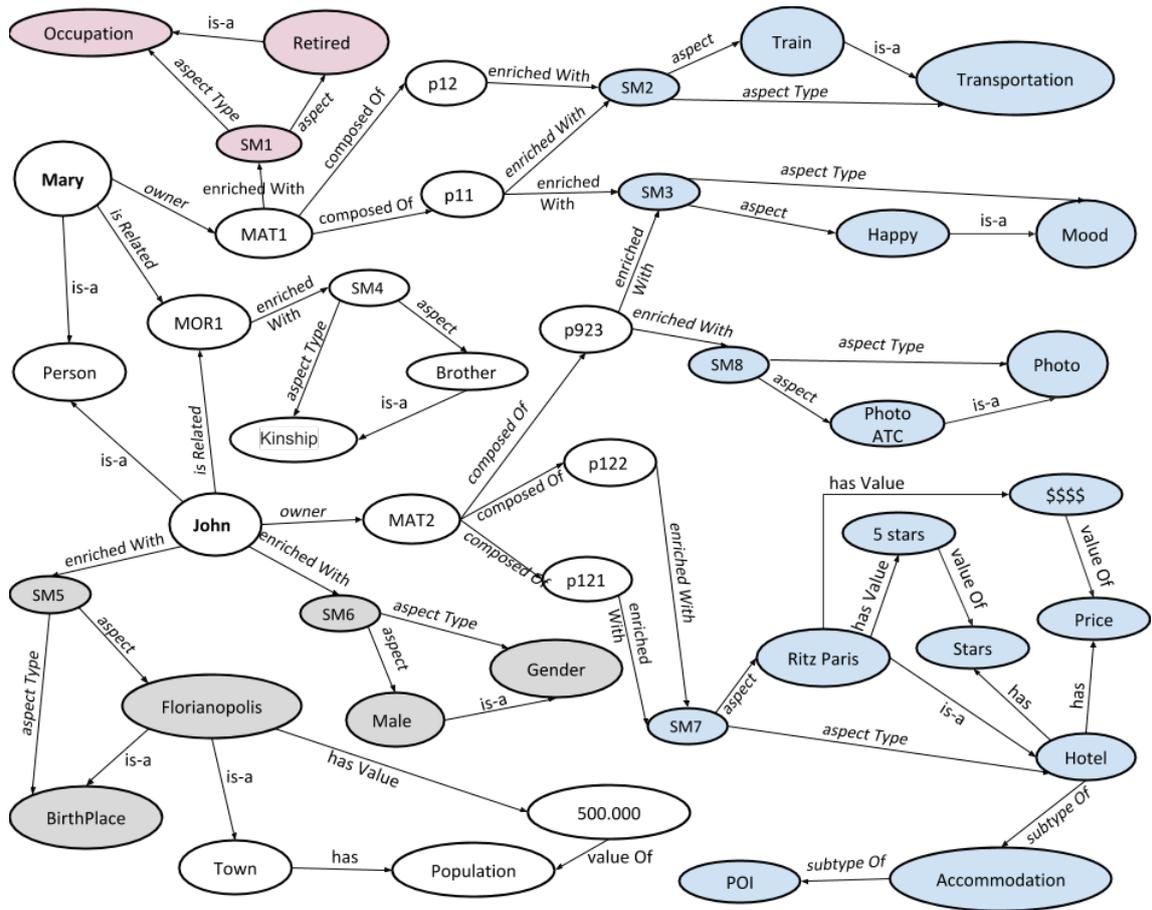


Figure 7. Example of MASTER Instantiation of the tourist scenario in RDF

The second query (Table 1(b)) exemplifies a chain-shaped query, where the chain is defined by a traversal that starts at a *trajectory* node and follows through a *point* of it that is enriched with a *semantic meaning* whose *aspect type* is, in turn, a subtype of an *Accommodation* aspect type node. This kind of query can be efficiently processed by NoSQL graph databases through a depth-first traversal. This query returns the trajectory *MAT2*.

The third query (Table 1(c)) exemplifies a complex query. In this case, the star component is represented by the filters over a moving object node *mo*, and the chain component is represented by the traversal from an aspect node *a* to a moving object node *mo*. Additionally, we have a simple GeoSPARQL filter (*geof:sfIntersects*) that limits the search to trajectories that visited the city of *Paris*. This kind of query can be processed through a join of data retrieved from NoSQL graph and document databases. It returns *Ritz Paris* if this aspect belongs to a trajectory that visited *Paris*.

These examples show the feasibility of the logical model in RDF proposed by MASTER to support the main types of queries over multiple aspect trajectories by using SPARQL and an efficient triplestore like *Rendezvous*.

Table 1. Typical SPARQL Queries over Multiple Aspect Trajectories

<pre>SELECT mo? WHERE { mo? is-a 'Person' mo? enrichedWith sm? sm? aspect 'Florianopolis' sm? aspectType 'birthplace' mo? enrichedWith sm2? sm2? aspect 'Male' sm2? aspectType 'Gender' }</pre> <p style="text-align: center;">(a) Star</p>	<pre>SELECT t? WHERE { t? composedOf p? p? enrichedWith sm? sm? aspectType at? at? subtypeOf 'Accommodation' }</pre> <p style="text-align: center;">(b) Chain</p>	<pre>SELECT a? WHERE { a? rdfs:type 'Aspect' a? is-a at? at? subtypeOf 'Accommodation' sm? aspectType a? p? enrichedWith sm? FILTER(geof:sfIntersects(?p, Paris)) t? composedOf p? mo? owner t? mo? is-a 'Person' mo? enrichedWith sm? sm? aspect 'Florianopolis' sm? aspectType 'birthplace' mo? enrichedWith sm2? sm2? aspect 'Male' sm2? aspectType 'Gender' }</pre> <p style="text-align: center;">(c) Complex</p>
---	---	---

3.2. Quantitative Evaluation

This section presents a quantitative evaluation in terms of query performance by comparing the adopted MASTER storage solution (*Rendezvous*) with a state-of-art spatio-temporal database system for *symbolic trajectories* proposed by Güting et al. in [Güting et al. 2015], called *Secondo* [Valdés et al. 2013]. The intention here is to compare two databases that are able to maintain semantic trajectories as there is no other solution for multiple aspect trajectory storage and querying apart from *Rendezvous*.

Secondo was initially developed for raw trajectories, and the definition of symbolic trajectory intends to associate different semantic labels to trajectory segments, which could be used to represent a limited set of aspects. Formally, a symbolic trajectory st is a sequence of tuples with the form $(st_{begin}, st_{end}, label)$, where st_{begin} and st_{end} are the points that delimit a subtrajectory $subt \in st$, and $label$ is a semantic information (e.g., a POI name, a transportation mean name). This tuple denotes a maximal subtrajectory where the label holds.

A first limitation of a symbolic trajectory and, as a consequence, of the *Secondo* storage strategy is the representation of the semantic information as a label. Instead, we model semantic information in terms of aspects that may have an arbitrary schema defined by an aspect type and its attributes. Another limitation is that a symbolic trajectory associates only one semantic information at a time for a trajectory or subtrajectory. For example, if we consider *transportation* and *streets* as semantic information, two symbolic trajectories must be created, one for the *transportation* and another one for the *streets*, each one represented as a relational table indexed by time. For a query that involves multiple aspects in *Secondo*, all time intervals must be searched to check where all aspects

hold together, while *Rendezvous* is able to directly access any aspect node stored in an RDF graph. These limitations of the *Secondo* data model had generated, as expected, worse performance results than *Rendezvous* for querying a large volume of trajectory data enriched with semantic information, as detailed in the following.

In order to compare the query performance and scalability of *Rendezvous* against *Secondo*, we consider a dataset called *BerlinMOD*². BerlinMOD is a benchmark for spatiotemporal relational database management systems, created by the *Secondo* team, which is able to generate semantic trajectories of vehicles. The benchmark provides more than 25 types of spatial and/or temporal and/or semantic queries. Some examples of BerlinMOD query types are: (i) *What are the pairs of vehicles of type "truck" whose trajectories have ever been as close as "10m" or less to each other?* (a spatio-semantic query); (ii) *What vehicle of type "passenger" has a trajectory that reached a point "(X,Y)" before all the trajectories of vehicles of the same type during the time interval "[t_s, t_e]"?* (a spatio-temporal semantic query). On using the *BerlinMOD* generator, we created a 19.45 GB dataset with around 53 million trajectories.

We ran experiments considering both *Rendezvous* and *Secondo* as distributed infrastructures in the cloud in order to obtain better performance for large data volumes. *Rendezvous* uses MongoDB 3.4.3 and Neo4J 3.2.5 as the document and graph NoSQL databases, respectively, giving their maturity as representatives of these families of NoSQL data models. All the distributed data nodes are Amazon m3.xlarge spot instances³ with 7.5 GB of memory and 1 x 32 SSD capacity. For all the experiments, we define nodes that represent MongoDB + Neo4J servers, and the *Rendezvous* servers were also installed on each node. In order to provide an equivalent test environment, we installed *Secondo*, following its tutorial⁴, in a cluster with the same size of the *Rendezvous* installation (the same number of nodes of Amazon m3.xlarge spot instances). In both cases, all the queries were issued from a server in the same network, so the latency between the client and the clusters was inexpressive.

We divide the experiments into several rounds (*Round 1* to *Round 4*). We first insert 10% of the dataset and ran all the queries in order to initiate the workload awareness (we call it *Rendezvous warm up*). Then, in the following, now with an existing workload awareness, we ran again all the queries four times over increasing fractions of the dataset (40%, 70% and 100%) and get the average processing times. The results are shown in Figure 8.

As can be seen in this Figure, we obtained an average query processing time that outperforms *Secondo* from *Round 2* on. We present a worse performance only at *Round 1* because *Rendezvous* is not aware of the typical query workload at this time and it spends an extra time to analyse the query shapes as well as to store and index trajectory data usually accessed by these queries in an efficient way for further retrieval. Once aware of the typical workload, *Rendezvous* ran two times faster than *Secondo* in average.

This first set of experiments was carried out over four Amazon nodes. In order to evaluate if the performance is positively affected by an increase in the number of

²<http://dna.fernuni-hagen.de/secondo/BerlinMOD/BerlinMOD.html>

³<https://aws.amazon.com/ec2/instance-types/>

⁴<http://dna.fernuni-hagen.de/secondo/DSecondo/DSECONDO/Website/index.html>

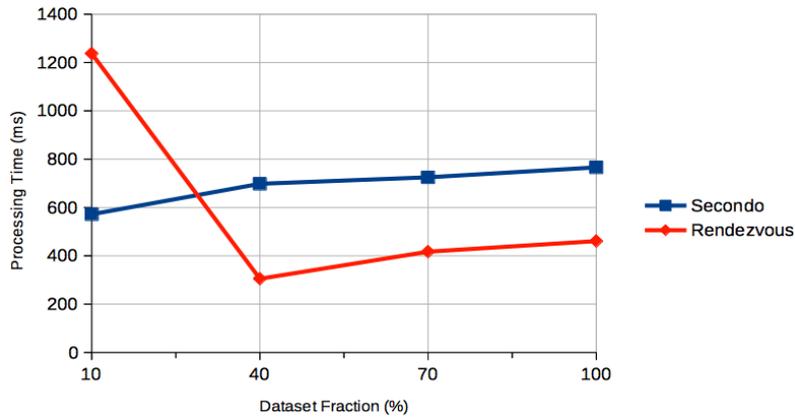


Figure 8. Comparison of Query Performance over BerlinoMOD between *Secondo* and *Rendezvous*

nodes, we duplicate them (eight nodes) maintaining the same setup (memory, storage and servers) on each of them. After that, we executed again all the queries four times and got the average processing time on each round. The results are shown in Figure 9.

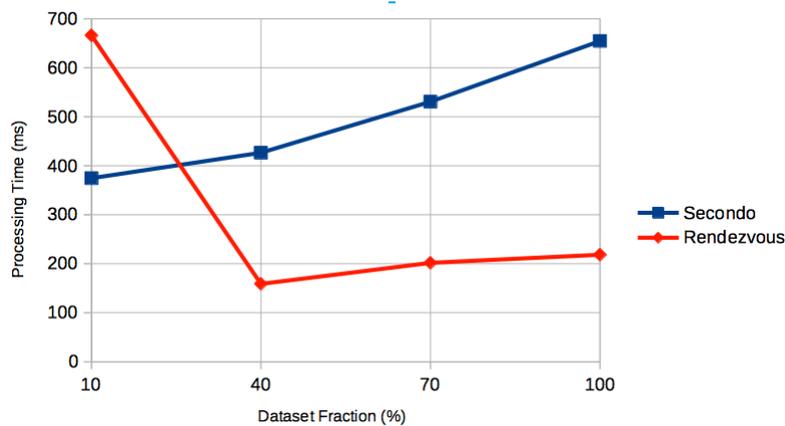


Figure 9. Comparison of Query Performance between *Secondo* and *Rendezvous* over a Larger Number of Nodes

Figure 9 reveals that our storage solution takes more advantage of the increase in the distributed infrastructure than *Secondo*. Both approaches spent less time to process the queries, but *Rendezvous* processing time is faster and does not increase so much with the increase of the dataset if compared to *Secondo*. This is justified mainly by the efficient query execution strategy followed by *Rendezvous*, which benefits from the increase in the number of nodes to parallelize the processing of the parts of a query. The efficient storage into NoSQL databases and workload-aware indexing scheme also contribute to this better performance.

This quantitative evaluation highlights that our solution for storing and querying trajectories with multiple semantic aspects is promising since it demonstrates a good performance for processing queries over increasing fractions of the dataset, and a better scalability when compared to the baseline. In fact, *Rendezvous* is suitable to applications whose data grows exponentially, which may be the case of applications that deal

with multiple aspect trajectories, since the add of new machines to process an increasing number of trajectories improves its query processing time performance.

4. Related Work

In this section we discuss existing initiatives for trajectory data modeling. In 2007, Hornsby in [Hornsby and Cole 2007] proposed to model trajectories as sequences of events. Although this approach gives the notion that movement is based on a sequence of happenings, the main drawback is that events are all homogeneous elements, so not being able to represent the diverse heterogeneous information extracted from different data sources as social media, sensor networks, weather channels, etc, that characterize different aspects related to movement.

In 2008, Spaccapietra in [Spaccapietra et al. 2008] introduced the new concept of semantic trajectories represented as a sequence of stops and moves. The work of Alvares [Alvares et al. 2007] was the first instantiation of the model of stops and moves, and originated several works where stops have been labeled as episodes, POIs, hot spots, etc. The work of [Bogorny et al. 2011] proposed an architecture to store and analyze semantic trajectories, but it is also limited to trajectories represented as stops and moves.

In 2011, Yan [Yan et al. 2011] proposed the SEMITRI framework for labeling stops and moves. Later, Bogorny in [Bogorny et al. 2014] proposed a semantically richer conceptual model, called CONSTANT, where a trajectory can be associated to a limited set of predefined aspects: the activities performed by the object, the means of transportation, the visited POIs, the goal of the trip and some behavior specific patterns. This model is limited to a subset of aspects that are related to subtrajectories or to the entire trajectory, are also presented as labels and only the weather conditions and the POIs can be associated to trajectory points. Additionally, CONSTANT does not support social media data as relationships between moving objects, opinions, and feelings, as well as any aspect type or concept of permanent or temporary aspect.

In 2015, Fileto in [Fileto et al. 2015] proposed Baquara to instantiate the CONSTANT model. The idea is to annotate semantic trajectories with Linked Open Data. It is based on a predefined ontological model of semantic trajectory, where we can find a preliminary notion of "aspect": here an aspect is a semantic label associated to the trajectory points. However, Baquara does not support relationships between moving objects and does not propose a solution for storing and querying huge volumes of trajectories, aspects or moving objects.

More recently, Ruback [Ruback et al. 2016] proposed an alternative framework for enriching movement data with Linked Open Data. While Baquara is based on a predefined ontology representing, in a monolithic way, the enriched trajectory, in the work of Ruback the enrichment is dynamically created through the use of ontology mash-ups. Despite of these innovations, this approach suffers from the same Baquara limitations presented before.

Ferrero in [Ferrero et al. 2016] presented the need for considering multiple and heterogeneous aspects that characterize movement, and the need for integrating all this data together for a more complete and realistic trajectory data analysis. This work, however, only shows the need for integrating multiple aspects, and does neither present a conceptual data model nor a physical strategy for efficient multiple aspect trajectory querying.

In terms of spatio-temporal database systems, HERMES [Pelekis et al. 2006] and Secondo [Gütting et al. 2005] are two well known software prototypes, but they were developed for raw trajectories. As a trajectory in Secondo is represented as a single data type moving object, it does not support the semantic enrichment of individual trajectory points. To overcome this limitation, another representation of trajectories is used to store semantic trajectories in Secondo, which is called symbolic trajectories [Valdés et al. 2013]. These trajectories are represented as subtrajectories annotated with single flat labels indexed by time intervals. To represent for instance the visited POIS and the transportation mode, the same trajectory must be segmented by POIS and by transportation mode.

In summary, apart from the work of Pelekis and Guting that focused on the physical level for representing and querying mobility data, the other previous works have focused on the conceptual representation only.

5. Conclusion

This paper presents an innovative perspective about movement data introducing the concept of multiple aspect trajectories. An aspect is any kind of semantic information that is related to the movement and should be managed in a jointly way with the spatio-temporal facet. We distinguish volatile, long term and permanent aspects, as well as the semantic meaning of an aspect. We provide a complete solution, called MASTER, for data modeling these new type of rich data, from the conceptual model to data storage and querying. The conceptual model combines intuitive formulation with very robust expressiveness of different types of trajectory aspects when compared to related work. A peculiar characteristic of this model is that it supports trajectories with any type of aspect annotation, from sets of attributes characterizing a POI to complex texts from social media posts, to name a few. Moreover, we show that the representation of multiple aspect trajectories is feasible in the RDF format, which can be efficiently processed by multimodel NoSQL databases. As the model represents fine-grained trajectories, *i.e.*, trajectories at a very low granularity level, queries may roll up and down over multiple aspect trajectories. We performed a twofold evaluation of MASTER: a qualitative one, with a tourism application domain, and a quantitative one, where MASTER outperforms, in terms of efficiency in query processing, a state-of-the-art competitor.

Future works include a performance evaluation over larger data sets of enriched trajectories, as well as the evaluation of other Big Data storage technologies, such as *NewSQL* databases, for maintaining multiple aspect trajectories. We also intend to extend MASTER to model data analytics information over multiple aspect trajectories, considering, for instance, dependencies among aspects.

Although out of the scope of this paper, it is also very important to consider privacy issues that these kinds of enriched trajectories might pose. When combining the different semantic aspects to the location information, privacy breach might happen. It is therefore crucial to develop privacy preserving methods to guarantee privacy when multiple aspects are involved.

Acknowledgments

This work was financed in part by the Brazilian agencies Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES, Conselho Nacional de Desenvolvimento Científico e Tec-

nológico - CNPq, and FAPESC, and the MASTER Project which received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie agreement N. 777695.

References

- Alvares, L. O., Bogorny, V., Kuijpers, B., de Macedo, J. A. F., Moelans, B., and Vaisman, A. (2007). A model for enriching trajectories with semantic geographical information. In *15th ACM Int. Symp. on Geographic Information Systems, ACM-GIS 2007, 2007, Seattle, USA, Proceedings*, pages 162–169.
- Bagui, S. and Bouressa, J. (2014). Mapping rdf and rdf-schema to the entity relationship model. *Journal of Emerging Trends in Computing and Information Sciences*, 5(12):953–961.
- Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.
- Bogorny, V., Avancini, H., de Paula, B. C., Kuplich, C. R., and Alvares, L. O. (2011). Weka-stpm: a software architecture and prototype for semantic trajectory data mining and visualization. *Trans. GIS*, 15(2):227–248.
- Bogorny, V., Renso, C., de Aquino, A. R., de Lucca Siqueira, F., and Alvares, L. O. (2014). Constant - A conceptual data model for semantic trajectories of moving objects. *Trans. GIS*, 18(1):66–88.
- Choi, M., Moon, C., and Baik, D. (2013). Transformation of a relational database to RDF/RDFS with er²-idm. *IEICE Transactions*, 96-D(7):1478–1488.
- Daniel, G., Sunyé, G., and Cabot, J. (2016). Umltographdb: Mapping conceptual schemas to graph databases. In *Conceptual Modeling - 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings*, pages 430–444.
- Ferrero, C. A., Alvares, L. O., and Bogorny, V. (2016). Multiple aspect trajectory data analysis: research challenges and opportunities. In Campelo, C. E. C. and Namikawa, L. M., editors, *XVII Brazilian Symposium on Geoinformatics - GeoInfo 2016, Campos do Jordão, SP, Brazil, November 27-30, 2016.*, pages 56–67. MCTIC/INPE.
- Fileto, R., May, C., Renso, C., Pelekis, N., Klein, D., and Theodoridis, Y. (2015). The baquara 2 knowledge-based framework for semantic enrichment and analysis of movement data. *Data & Knowledge Engineering*, 98:104–122.
- Güting, R. H., de Almeida, V. T., Ansorge, D., Behr, T., Ding, Z., Höse, T., Hoffmann, F., Spiekermann, M., and Telle, U. (2005). SECONDO: an extensible DBMS platform for research prototyping and teaching. In Aberer, K., Franklin, M. J., and Nishio, S., editors, *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 1115–1116. IEEE Computer Society.
- Güting, R. H., Valdés, F., and Damiani, M. L. (2015). Symbolic trajectories. *ACM Transactions on Spatial Algorithms and Systems*, 1(2):7.
- Hornsby, K. S. and Cole, S. J. (2007). Modeling moving geospatial objects from an event-based perspective. *Trans. GIS*, 11(4):555–573.

- Nogueira, T. P., Braga, R. B., de Oliveira, C. T., and Martin, H. (2017). Framestep: A framework for annotating semantic trajectories based on episodes. *Expert Systems with Applications*.
- OGC (2018). GeoSPARQL - A Geographic Query Language for RDF Data.
- Pan, J. Z. (2009). Resource description framework. In *Handbook on Ontologies*, pages 71–90. Springer.
- Pelekis, N., Theodoridis, Y., Vosinakis, S., and Panayiotopoulos, T. (2006). Hermes - A framework for location-based data management. In Ioannidis, Y. E., Scholl, M. H., Schmidt, J. W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., and Böhm, C., editors, *Advances in Database Technology - EDBT 2006, 10th International Conference on Extending Database Technology, Munich, Germany, March 26-31, 2006, Proceedings*, volume 3896 of *Lecture Notes in Computer Science*, pages 1130–1134. Springer.
- Ruback, L., Casanova, M. A., Raffaetà, A., Renso, C., and Vidal, V. M. P. (2016). Enriching mobility data with linked open data. In *Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS 2016, Montreal, QC, Canada, July 11-13, 2016*, pages 173–182.
- Sadalage, P. J. and Fowler, M. (2013). *NoSQL distilled : a brief guide to the emerging world of polyglot persistence*. Addison-Wesley.
- Santana, L. H. Z. and dos Santos Mello, R. (2017). Workload-aware rdf partitioning and sparql query caching for massive rdf graphs stored in nosql databases. In *Brazilian Symposium on Databases (SBB D), 2017*, pages 184–195. SBC.
- Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., and Vangenot, C. (2008). A conceptual view on trajectories. *Data and Knowledge Engineering*, 65(1):126–146.
- Valdés, F., Damiani, M. L., and Güting, R. H. (2013). Symbolic trajectories in secondo: Pattern matching and rewriting. In Meng, W., Feng, L., Bressan, S., Winiwarer, W., and Song, W., editors, *Database Systems for Advanced Applications*, pages 450–453, Berlin, Heidelberg. Springer Berlin Heidelberg.
- W3C (2018a). RDF - Semantic Web Standards.
- W3C (2018b). Semantic Web - W3C.
- Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., and Aberer, K. (2011). Semitri: a framework for semantic annotation of heterogeneous trajectories. In *EDBT 2011, Uppsala, Sweden, March 21-24, 2011, Proceedings*, pages 259–270.