Specification of Semantic Trajectories Supporting Data Transformations for Analytics: The datAcron Ontology

Georgios M. Santipantakis, George A. Vouros, Christos Doulkeridis, Akrivi Vlachou
University of Piraeus
Piraeus, Greece
{gsant|georgev|cdoulk}@unipi.gr, avlachou@aueb.gr

Gennady Andrienko, Natalia Andrienko, Georg Fuchs
Fraunhofer Institute IAIS
Sankt-Augustin, Germany
{{natalia|gennady}.andrienko|georg.fuchs}@iais.fraunhofer.de

Jose Manuel Cordero Garcia, Miguel Garcia Martinez
CRIDA
Madrid, Spain
{jmcordero|mgmartinez}@e-crida.enaire.es

ABSTRACT

Motivated by real-life emerging needs in critical domains, this paper proposes a coherent and generic ontology for the representation of semantic trajectories, in association to related events and contextual information, to support analytics. *The main contribution of the proposed ontology is twofold: (a) The representation of semantic trajectories at varying, interlinked levels of spatio-temporal analysis, (b) enabling data transformations that can support analytics tasks.* The paper presents the ontology in detail, in connection to other well-known ontologies, and demonstrates how data is represented at varying levels of analysis, enabling the required data transformations. The benefits of the representation are shown in the context of supporting visual analytics tasks in the air-traffic management domain.

CCS CONCEPTS

• **Information systems → Web Ontology Language (OWL)**;

KEYWORDS

semantic trajectory, moving object ontology, knowledge integration, data transformation, analytics tasks

ACM Reference Format:
Georgios M. Santipantakis, George A. Vouros, Christos Doulkeridis, Akrivi Vlachou, Gennady Andrienko, Natalia Andrienko, Georg Fuchs, and Jose Manuel Cordero Garcia, Miguel Garcia Martinez. 2017. Specification of Semantic Trajectories Supporting Data Transformations for Analytics: The datAcron Ontology. In *Proceedings of Semantics2017, Amsterdam, Netherlands, September 11–14, 2017,* 8 pages.
https://doi.org/10.1145/3132218.3132225

Semantics2017, September 11–14, 2017, Amsterdam, Netherlands

ACM ISBN 978-1-4503-5296-3/17/09…$15.00
https://doi.org/10.1145/3132218.3132225

1 INTRODUCTION

Many critical domains w.r.t. economy and safety, such as the Maritime and the Aviation domains, where Maritime Situation Awareness (MSA) and Air Traffic Management (ATM), respectively, impose emergent and challenging problems, require analysis of moving objects' behaviour over time: Challenges concern effective detection and forecasting of moving entities' trajectories, as well as recognition and prediction of important events by exploiting information about objects' behaviour and contextual data. Due to these needs, semantic trajectories are turned into "first-class citizens", forming a paradigm shift towards operations that are built and revolve around the notion of trajectory.

Motivated by these needs, our work focuses on trajectories and solutions towards managing data that are connected via, and contribute to enriched views of trajectories. We revisit the notion of semantic trajectory and extend it towards *representing, storing and transforming the wealth of information available in disparate and heterogeneous data sources. Information integrated in a representation where trajectories are the main entities, allows the computation of meaningful moving patterns, the recognition and prediction of the behaviour and states of moving objects.* Data transformations cannot be precomputed at production level, since it would be necessary that all the possible transformations are stored. The proposed approach stores only the fundamentals needed, and provably enables the transformation of data at consumption level. Therefore, motivated by real-life emerging needs in MSA and ATM domains, this paper proposes a coherent and generic ontology for the representation of semantic trajectories, in association with related events and contextual information, and demonstrates data transformations at consumption level for visual analytics tasks.

This work makes the following contributions:

(a) proposes an ontology for the representation of semantic trajectories at varying levels of spatio-temporal analysis. Trajectories can be seen as (i) sequences of positions of moving objects, derived from raw data, (ii) as aggregations of raw data, signifying meaningful events (generalizing on the stops-moves model [12]), providing a synoptic view of raw trajectories [9], (iii) as temporal sequences of meaningful trajectories segments (each revealing specific behaviour, event, goal, activity etc), (iv) as mere geometries. Representations at any such level of analysis are linked to each other, as well as to contextual information and co-occurring events.

(b) demonstrates the data transformations support, via enhanced SPARQL queries. Such transformations can adapt available data to

the analysis goals or to specific requirements of the methods that the analyst wants to apply. Without loss of generality, transformations are exemplified in the ATM domain via concrete examples to effectively support visual analytics in important real-world cases.

The paper is organised as follows: Section 2 motivates the need for the representation of semantic trajectories at varying levels of detail and the data transformations for supporting analytics tasks. Section 3 presents the datAcron ontology, and section 4 presents the data transformations enabled via the specification of enhanced SPARQL queries, supporting visual analytics tasks. The paper briefly presents the related works in section 5 and concludes with section 6.

2 MOTIVATION, TERMINOLOGY, OBJECTIVES AND SCOPE

The proposed ontology is a generic conceptual framework for the representation of semantic information relevant to the movement of objects, revolving around the notion of trajectory. To make the objectives of this ontology clear and provide concrete examples for its use, we elaborate in a scenario from the air traffic management (ATM) domain, concerning flow management (FM). Before that, we explain the basic notions of trajectory, event and contextual information.

2.1 Trajectories, events and contextual information

Starting from the definitions of *raw*, *structured* and *semantic trajectories* provided in [9], a *raw trajectory* is a temporal sequence of raw data specifying the moving object's spatio-temporal positions. Raw data can be aggregated, analyzed and semantically annotated, providing multiple abstractions of a trajectory. A maximal sequence of raw data that comply with a given pattern defines an *episode* [9]. In this work we focus on *events* as a generalisation of episodes, taking also into consideration –in conjunction to movement data– *contextual information* (i.e. any information -mostly about the environment of an object- that affects its movement, including other trajectories).

A *structured trajectory* (simply, trajectory) consists of a sequence of *trajectory parts* that can be either *raw positions* reported from sensing devises, aggregations of raw positions referred as *semantic nodes* or simply *nodes*, or *trajectory segments.*

A *semantic node* provides a meaningful abstraction or aggregation of raw positions, e.g. a set of raw positions may signify a "turn" event, represented as a single semantic node associated to the resource representing the "turn" event. A *trajectory segment* is a trajectory itself, part of a whole trajectory. Segmentation of trajectories can be done with different objectives depending on the application and target analysis. Any trajectory part may be associated with a co-occurring event. For example, a bad weather region may co-occur with a trajectory crossing-it (thus, related spatially) during a time period (related temporally).

A *semantic trajectory* is a meaningful sequence of trajectory parts, signifying events, activities, goals, etc. of moving entities.

2.2 The flow management domain

Mobility analysis tasks require a wealth of information from disparate and heterogeneous sources. As a running example for the representation of entities and data transformations to spatial and time series of events, we elaborate in scenarios from the air traffic management (ATM) domain, concerning flow management (FM). FM is an extremely important service for airlines to operate in a safe and efficient way, complementary to Air Traffic Control (ATC). The objective of FM is to ensure an optimum flow of air traffic to or through areas within which traffic demand at times exceeds the available capacity of the ATC system. The scenarios have been specified by domain experts in the datAcron research project[1].

The entities of particular interest for the FM domain are:

- *Air blocks*, specified by geometries, which are static spatial 2D projections of airspace volumes.
- *Sectors*, which are static spatial 3D objects comprising airspace volumes that are defined by air blocks, with lower and upper flight levels.
- *Flight information regions* (FIR) that are static spatial 3D objects. Each of them is the responsibility of a certain control unit. For Europe, there are usually 2 divisions for the lower and upper air spaces. FIRs are quite large: some FIRs cover entire counties (Belgium and Luxembourg are joined in one FIR), and some countries are divided into two or more FIRs. Spain, for instance, has the same 3 FIRs regardless upper or lower air space.
- *Sector configurations* are alternative divisions of airspace into sectors. These constitute the minimum unit that an Air Traffic Controller operates. The number of sectors dividing the FIR space may vary, hence allowing to operate the FIR with the appropriate number of controllers according to demand conditions, ensuring safety of operations.
- *Opening schemes* or *active configurations* are the sector configurations actually deployed in a given airspace with time intervals of their validity.
- *Capacities* refer to sectors (a.k.a. traffic volumes): for each sector and time unit, the capacity value of that sector may be either undefined (if the sector is not active at that time) or specify the upper limit of the number of flights in any time period with pre-specified duration (typically one hour). The capacities consider controllers workload, and are fixed values for the same sectors every time they are active.
- *Flight plans* are specifications of trajectories consisting of spatial events of flights crossing air blocks and sectors, and flying over specific waypoints (fixed coordinates among which airways are set). Each event specifies the entry (resp. exit) coordinates, flight level and time to (resp. from) a sector, or the time that the flight will fly over a waypoint. Flight plans specify other information such as estimated take-off time, and, in case of delay caused by a regulation, the calculated take-off time of the flight.
- *Predicted weather* is a spatial time series of multiple weather attributes referring to 3D locations (longitude, latitude, altitude).

[1]Detailed description of the scenarios is available online: http://ai-group.ds.unipi.gr/datacron/system/files/datACRON_D6.1.pdf

On each operation day, the flow management monitoring process analyses periodically (typically every 20 minutes) the demand for each sector, by counting the expected number of flights in the sector during the next period (typically one hour, to match the definition of capacity). If a potential demand versus capacity imbalance is detected (a *hotspot*) a regulation may be applied to adjust the demand values to the available capacity.

Although the reason to apply a regulation may vary to bad weather conditions, strikes, etc., for the purposes of this article we focus on regulations applied due to hotspots. Therefore, a *regulation* is a special type of event that occurs as a measure that a flow manager takes to solve an excess of demand. The attributes of any regulation include the location (sector), start and end times, and reason codes (e.g. "C" for delays).

Regulations usually result in delays in the departure time of flights crossing that area, which introduces yet another factor of unpredictability to airlines' operations. Therefore, airlines need to predict the occurrence of regulations well in advance, so as to reduce unpredictability. Identifying patterns of regulations is important towards this goal.

To make the objectives of the proposed ontology clear we show how data for the above entities in the FM domain can be provided to support identifying regulations' patterns, interdependencies between affected flights and areas, and supporting the choice of sector configurations based on expected demands. These cases are explained in detail in section 4.

2.3 Objectives

We aim to provide a coherent and generic ontology for integrating all data from disparate sources, representing fully-fledged semantic trajectories at varying levels of spatio-temporal analysis.

By means of this ontology we aim at supporting (a) services for answering spatio-temporal queries concerning vessels' trajectories along with aspects that affect and are affected by the mobility of moving objects, thus providing all the necessary information that analytics tasks require, (b) transformations between different representations of data required for analytics tasks.

As already mentioned, transformations can adapt available data to the analysis goals or to specific requirements of the methods that the analyst wants to apply. Transformations aim to extract relevant parts of the data or reduce irrelevant details, converting movement data from one form to another, to support different task foci: movers, spatial, events, space, and time. The role of data transformation is to prepare data for analysis, that is, to convert data to a form fitting a task or required by an analytical tool, maybe changing the structure of the data [2].

Following the approach of [10], the queries that this ontology must satisfy in the first place can be seen as combinations of three basic components: (a) space (*where*), (b) time (*when*), (c) object (*what*). These components can be used in three basic types of queries:

- Retrieve the objects (e.g. flights) in a region (e.g. a sector) for a time period (*when*&*where* → *what*).
- Retrieve the location (or geometry) occupied (resp. covered) by an object (e.g. a flight plan) or set of objects (e.g. flights), at a given time instant or period (*when*&*what* → *where*).

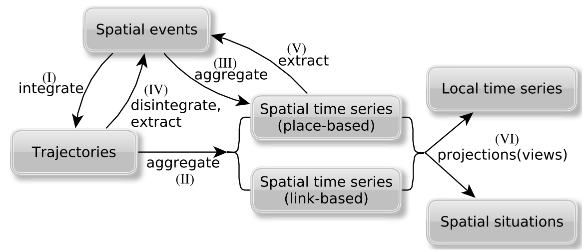


Figure 1: Conversions between different representations

- Retrieve the time periods that a non-empty set of objects (e.g a set of flights) appears in a specific location or area (e.g. cross a specific sector or FIR) (i.e. *where*&*what* → *when*).

Exploiting these fundamental types of queries, and to a greater extent than other representations of trajectories, we aim to show how the proposed ontology supports transformations between different representations of data, for the benefit of analysis tasks' effectiveness. Such generic transformations are depicted in Figure 1 [3] and will be demonstrated using data from the above-mentioned FM entities. Briefly, as Figure 1 shows, trajectories integrate spatial events (e.g. entering or exiting a sector) (transformation I), while these events, similarly to trajectories, may be aggregated to spatial time series: Place-based, such as hotspots detected in sectors (transformation III), or link-based, such as flows of flights between pairs of sectors (transformation II). Projections of these time series may result to spatially-referenced time series or to spatial situations (transformations VI). These transformations impose specific requirements to answering queries, regarding aggregations, extraction of events and projections of data, demonstrated in subsequent sections of this article.

3 THE DATACRON ONTOLOGY

The datAcron ontology[2] was developed by group consensus of ATM and MSA domain experts, data and visual analysts and knowledge engineers, over a period of 12 months following a data-driven approach according to the HCOME methodology [7]. It has been designed to be used as a core ontology for the MSA and ATM domains, towards supporting analysis tasks. Its development has been driven by ontologies related to our objectives (e.g. DUL[3], SimpleFeature[4], NASA Sweet[5] and SSN[6]), as well as schemas and specifications regarding data sources from the different domains.

3.1 Core vocabulary and overall structure

According to the above specifications, and illustrated in Figure 2, a trajectory (`Trajectory`) can be segmented to trajectory parts (`TrajectoryParts`), each including other segments and/or more semantic nodes. Each semantic node may be associated with a specific raw position or a temporally ordered sequence of raw positions of a moving object.

[2]Documentation is available online at http://ai-group.ds.unipi.gr/datacron_ontology
[3]http://www.ontologydesignpatterns.org/ont/dul/DUL.owl
[4]http://www.opengis.net/ont/sf
[5]https://sweet.jpl.nasa.gov/
[6]https://www.w3.org/2005/Incubator/ssn/ssnx/ssn

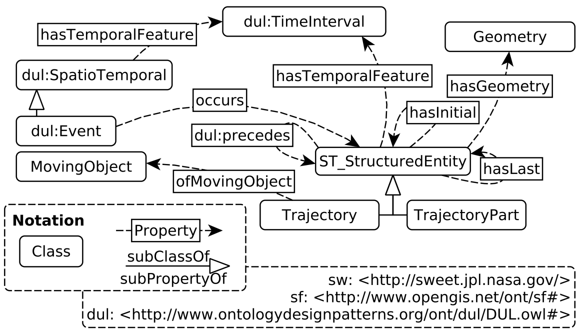


Figure 2: The main concepts and relations of the proposed ontology.

The generic pattern of specifying structured trajectories is presented in section 3.2.

Trajectories and trajectory parts can be associated with contextual information, as well as with events (`dul:Event`). Although events may happen independently from the trajectory but co-occur with the trajectory, we focus on happenings on the trajectory itself (e.g. a "turn" or a "gap of communication") and to moving object's information (e.g. vessel in a protected or in a bad-weather area). Patterns for the specification of events and their associations to trajectory parts are presented in section 3.2, together with types of contextual information represented and respective associations to trajectories.

3.2 Patterns of semantic trajectories

As already said, the proposed ontology enables the specification of trajectories at various level of abstraction. Figure 3 illustrates the generic pattern of raw and structured trajectories.

The main concept in this pattern is the `Trajectory`, which is a subclass of Spatio-Temporal Structured Entity (`ST_Struc- turedEntity`). This, being a subclass of `dul:Region` represents a region in a dimensional space and time, used as a value for a quality of an Entity, while it also represents (structured) trajectories and their parts. A structured trajectory, as well as any of its parts, can be a temporal sequence of `TrajectoryPart` entities.

Direct subclasses of `Trajectory` are the

- `IntendedTrajectory`: planned trajectories build by an `dul:InformationEntity` such as a `FlightPlan`,
- `ActualTrajectory`: trajectories constructed from actual positioning data, after some processing of the raw positional data, `RegulatedTrajectory`: intended trajectories that have been modified by an operational event such as a regulation,
- `SyntheticTrajectory`: trajectories constructed by a simulation process, and
- `RawTrajectory`: trajectories constructed by the raw (unprocessed) sequence of positions of the moving object.

An `ActualTrajectory` can be further distinguished to a `Closed-Trajectory` (i.e. a trajectory that has reached its destination) and to an `OpenTrajectory` (i.e. a trajectory in progress).

The `TrajectoryPart` can be further distinguished to one of the following subclasses:

- `Segment`: associated to a spatial region and a time proper interval.
- `Node`: associated to a point in space and a time instant or time period. The latter holds in case the node aggregates several raw positions. A `Node` can be the result of a data processing component computing compressions or aggregations of the raw positioning data.
- `RawPosition`: represents the raw (unprocessed) positioning data. Each raw position instance is associated to a point in space and a time instant.

A specific trajectory, as well as any of its trajectory parts, being instances of `dul:Region` can be associated to their parts via the `dul:hasPart` property or via the subproperties `hasInitial`, `hasLast` which indicate the first and last part of the `ST_Structured- Entity`, respectively. For instance, a trajectory may comprise a sequence of trajectory segments, who on their own turn comprise other segments, nodes, or raw positions, and so on. The temporal sequence of structured entities is specified by means of the property `dul:precedes`. Trajectories related via the property `dul:precedes` represent subsequent trajectories of a specific object, and thus keep a long history of its movement. It must be noticed that this combination of properties allow to sharing trajectory parts between trajectories with no ambiguity: For instance, a trajectory node can be shared between the actual and the intended trajectory of an aircraft.

Each structured entity (i.e. trajectory or trajectory part) can be associated to a specific *geometry* (`sf:Geometry`), representing a point or region of occurrence, and a *temporal entity* (`dul:TimeInterval`) specifying a time interval of occurrence. The Geometries of structured entities can be serialized into Well-Known-Text (WKT) and asserted as values to the property `hasWKT`, which is sub-property of `geosparql:hasSerialization`.

Finally, trajectories can be members of `TrajectoryCluster` entities, via the `dul:hasMember` property.

Towards the specification of semantic trajectories, trajectories are associated with events and contextual information. Specifically, each trajectory and trajectory part, being instances of `ST_Structured-Entity`, can be associated via the property `occurs` with events, as illustrated in Figure 4. An event can be associated with other events via the properties `dul:hasConstituent` or `dul:hasPart`: This is the case for high-level events associated with other high-level or low-level events. An event involves at least one participant (associated via the property `dul:hasParticipant`) and it holds for a specific `TimeInterval` specified by the property `dul:hasTimeInterval`. An event can be:

- `LowLevel`, in case its detection requires data from a single trajectory,
- `HighLevel`, in case its detection requires contextual data and maybe, data from multiple trajectories. For example, events of type `EnterSector` involve information about active sectors along with trajectories. As another example, hotspots require data about active sectors and multiple trajectories.
- `Operational`, if they are issued by domain specific operators, affecting regions or groups of entities for a specific time interval. For example, a regulation (`Regulation`) is applied on a sector and remains active for a time interval, and indirectly

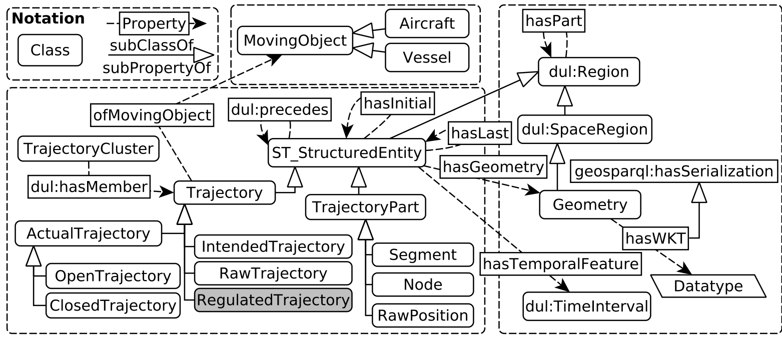


Figure 3: The pattern of structured trajectories. Domain specific concepts in gray

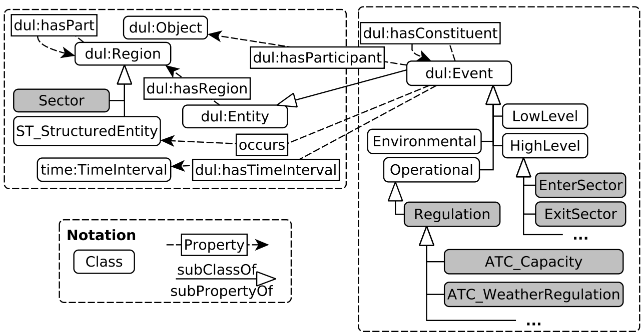


Figure 4: The pattern of trajectories linked with events. Domain specific concepts in gray

affects all the intended trajectories crossing the sector. A trajectory is linked through the property `affectedBy`, with corresponding regulation.

- `Environmental`, if they are due to the environment of moving objects. Extreme weather condition are such events.

It must be noticed that associating events to trajectory parts satisfies the requirement to associate multiple events to varying levels of trajectory analysis, according to the information used for the detection of each event: For instance, a low-level "turn" event may co-occur with a low-level "descend" event and thus, both events may be associated to the same semantic node. In addition to that, this semantic node may be associated to a trajectory segment which in its own turn is associated with events of type "DescendingPhase" and "CrossingSector".

In addition to events, trajectory parts of semantic trajectories can be linked to contextual information, i.e. information about entities in the environment that affect the moving object, including other trajectories. Such information may be archival information concerning static aspects of the environment (e.g. airports, airspaces, etc), dynamic (e.g. changing sector configurations), or streaming (e.g. weather forecasts). The pattern for linking trajectory parts with contextual information is illustrated in Figure 5. Without loss of generality, subsequent paragraphs and Figure 5 show associations to contextual entities related to FM scenarios, although other cases may be also specified.

Weather conditions are very important to trajectories in the MSA and ATM domains: Each `TrajectoryPart` can be associated with entities of type `WeatherCondition`, which is defined as a subclass of `ssn:FeatureOfInterest`, (i.e. the entity whose properties are being estimated or calculated in the course of an observation).

Of particular interest to the FM and MSA domains are airspace regions. Structured entities can be linked to spatial regions (instances of `dul:Region`) of particular interest through the properties `within` and `dul:nearTo`.

Also, the departure and destination of a trajectory can be considered as contextual information, linked via the properties `hasDeparture` and `hasDestination`, respectively. The properties range to the class `dul:PhysicalPlace`, which can be further refined to domain specific classes such as `Airport`, `Heliport`, or `Port`.

A `FlightPlan` is also a domain specific entity `dul:InformationEntity` which is associated to an `IntendedTrajectory` or a `RegulatedTrajectory`, via the property `reportsTrajectory`. A simplified example of the triples describing a flight plan for a flight from Capadichino airport (Napoli) to Gatwick airport (London) is as follows:

```
:flight_plan_AA51147955 a :FM_FTFM ;
    :hasIdentification "AA51147955" ;
    :departureAirport :Place_Napoli_Capodichino_Airport ;
    :destinationAirport :Place_London_Gatwick_Airport ;
    :reportsTrajectory :tr_20160401_703202_m1 .
:tr_20160401_703202_m1 a :IntendedTrajectory ;
    :ofMovingObject :Aircraft_4007E4 .
:tr_20160401_703202_m1 :hasStart :n_m1_20160401_703202_1 .
:n_m1_20160401_703202_1 :hasAltitude "91.44"^^unit:meters ;
    :hasLongitude 14.290833 ; :hasLatitude 40.884444 .
                    ...
:tr_20160401_703202_m3_geom :hasWKT
   "LINESTRING(14.290833␣40.884444,␣...␣-0.190278␣51.148056)".
```

4 DATA TRANSFORMATIONS FOR ANALYTICS

In this section, we show how data transformations are supported that are necessary for more advanced analysis tasks, such as discovery of patterns of regulations (denoted FM01), and prediction of sector configurations (denoted FM02).

4.1 Flow management use-cases

Following the analytics requirements in FM, we describe two use-cases that require advanced data transformations.

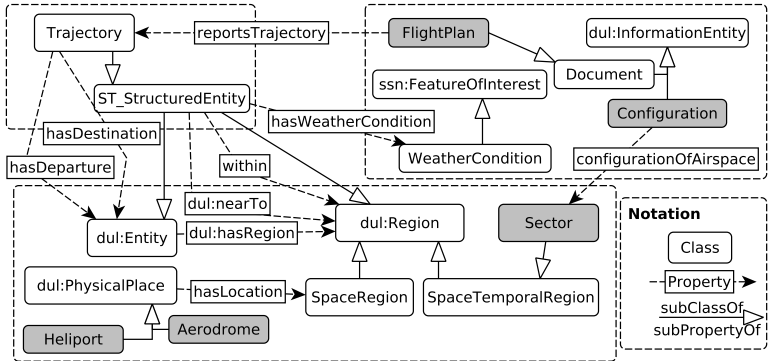


Figure 5: The pattern of trajectories linked with contextual information. Domain specific concepts in gray

FM01. Towards discovering patterns of regulations, we need to *(a)* discover regular temporal patterns of regulations applied to sectors, and *(b)* discover interdependencies between sectors based on regulations.

For FM01(a) we generate time series of counts of regulations per area of interest (e.g. sectors or FIR) by time periods of a chosen length (e.g., 1 hour). Among these time series, we aim to find time series with high periodicity (daily and weekly time cycles). For FM01(b), we need to find "patterns between sectors", where regulations in some sectors or groups of sectors often lead to regulations in other sectors. These regulations can be found as follows: Regulations in sectors S_1 and S_2 may be related if the regulation applied in S_1 affects the times where flights enter S_2 resulting to a new hotspot. This task can be further supported by discovering re-occurring links between sectors. Two sectors are linked via a "link event" if regulated flights cross both sectors during a particular time interval. This can be achieved by computing time series of link existence: for each pair of sectors (S_1, S_2) for which link events exist, we need to compute time series with values 1 for the time intervals when links between S_1 and S_2 existed and 0 for the remaining time intervals. Time series with multiple peaks would signify interrelationships between sectors, possibly caused by the airspace design (density of airways connecting waypoints in certain sectors) or by the density of traffic flows between certain points in space in specific periods (e.g. holidays).

FM02. In this case the aim is to predict the choice of sector configurations based on expected demands. To this end, we compute the expected demands by aggregating the flight plans into spatial time series by suitable sectors and time intervals. Two time-dependent attributes may be computed for any sector S: *entry hourly count* (how many flights enter S during each time interval) or *occupancy count* (how many flights are present in S during each occupancy period). Occupancy count, used in this paper, concerns overlapping occupancy periods of predefined duration. We may view the occupancy period as a sliding time window, shifted by a number of time units specified by the "time step". That is, two parameters of temporal aggregation are used: occupancy period duration (e.g., 1 hour) and time step, which is smaller than the occupancy period duration (e.g., 15 minutes).

Overview. $FM01(a)$ requires a spatial events to spatial time series (place-based) transformation (Figure 1), $FM01(b)$ requires a transformation from trajectories to spatial time series (link-based), and $FM02$ a transformation from trajectories to spatial time series (place-based).

4.2 Evaluation of data transformations

The datasets involved in these scenarios (regulations, flight plans, etc.) are real data from April 2016, summing up to approximately 1.08 billion triples. For demonstrating purposes we have setup a Jena triple store with 5% of the total number of triples, on a i7-6700HQ CPU at 2.60GHz, with 16GB RAM and Linux OS. The spatio-temporal functions we have implemented (based on the Region Connection Calculus and on Allen's interval algebra) for the enhanced SPARQL queries, extend classes of Jena ARQ engine. These functions are registered to the SPARQL engine by the namespace `<java:datAcron.unipi.gr.sparql_functions.>`, which is accessible to the data transformation queries.

All data transformations presented in this section have been implemented on top of SPARQL queries to the endpoint, which produce data for visual analysis tasks, in particular for pattern detection and analysis. All illustrations have been produced by the V-Analytics tool [2].

Although some of the aggregate computations can be enabled by the `COUNT` function of SPARQL, the time series computation requires iterative SPARQL queries. For this reason, we use an iterative procedure that poses an enhanced, parametrized SPARQL query, whose parameters are updated in each iteration. The iterative process results to a sequence of queries for subsequent time periods. Specifically, given a duration $\Delta t \neq 0$ and a period $[TimeStart, TimeEnd]$, the i-th query of n iterations, concerns the time period $[TimeStart + i * \Delta t, TimeStart + (i+1) * \Delta t]$.

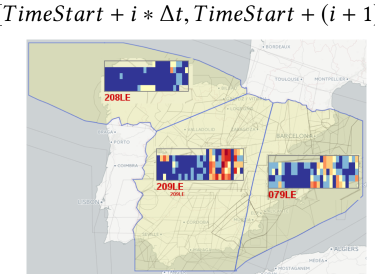


Figure 6: Regulations by FIR

FM01(a). The case of FM01(a) requires data transformation of spatial events to spatial time series aggregate (Figure 1, (III)). In particular, it requires the computation of time series of counts of regulations with a particular reason code, in time intervals of a chosen length. The parametrised query for a given airspace, (e.g. `$airspace$=:Airspace_LBTA_411`), is as follows:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.../DUL.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX myfn: <java:datAcron.unipi.gr.sparql_functions.>
SELECT (COUNT(DISTINCT ?regulation) AS ?count) WHERE {
  ?regulation a $regulation$; dul:hasRegion $airspace$;
  dul:hasTimeInterval ?t . ?t :TimeStart ?s ; :TimeEnd ?e.
  FILTER(myfn:overlaps(?s,?e,
     $t0$^^xsd:DateTime, $t1$^^xsd:DateTime))}
```

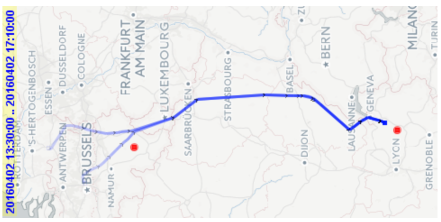
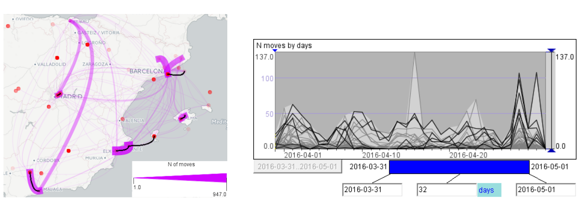


Figure 7: Trajectories associated to regulations

A procedure iterates and poses a query for each regulation type and each subsequent occupancy period in the overall period (e.g. $[2016-04-01, 2016-04-30]$). The variables assigned at each iteration are as follows: `$regulation$` (regulation type), `$t0$` (time window start), `$t1$` (time window end). The function `myfn:overlaps/4` realises the temporal overlap relation in Allen's interval algebra.

Results from these queries can be aggregated at varying levels of airspace partitioning granularity. While aggregations at the level of FIRs provide some useful patterns, as depicted in Figure 6, patterns at lower levels of spatial granularity (e.g. at the level of airlocks - not depicted here) are difficult to be detected. In particular, at FIR level, we clearly observe large numbers of regulations in 209LE (central part of Spain) on Fridays daytime (6:00 - 18:00), Saturdays (6:00-24:00) and, in some weeks, on Sundays. In 079LE (East) we see frequent regulations on Fridays and, in some weeks, on Saturdays. There are only a few regulations in 208LE (West).



(a) Aggregated flights (b) Daily time series

Figure 8: Regulated flights

FM01(b). The data transformation required by the case FM01(b) is finding "patterns between sectors", when regulations in some sectors or groups of sectors often lead to regulations in other sectors. Specifically, for each pair of regulations $R1$ and $R2$ that overlap in time and refer to distinct sectors $S1$ and $S2$, it is required to retrieve the regulated flights (according to intended trajectories specified by flight plans) that were going (before the regulation) to visit both $S1$ and $S2$ during the time period that spans the duration of both regulations $R1$ and $R2$. The returned trajectories are aggregated into flows between sectors. Each such aggregate is considered as a "link event". Thus, we first need to detect the pairs of sectors affected by temporally overlapping regulations and construct the links between them. The SPARQL query that will assert a link (using the property `:associatedByOverlappingRegulationWith`) between pairs of sectors is as follows:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dul: <http://www.ontologydesignpatterns.../DUL.owl#>
PREFIX myfn: <java:datAcron.unipi.gr.sparql_functions.>
CONSTRUCT { ?s0 :associatedByOverlappingRegulationWith ?s1 }
WHERE { ?r0 a ?c . ?c rdfs:subClassOf :FM_Regulation .
  ?r0 dul:hasRegion ?s0 ; dul:hasTimeInterval ?t0 .
  ?t0 :TimeStart ?t00 ; :TimeEnd ?t01 . ?r1 a ?c .
  ?c rdfs:subClassOf :FM_Regulation . ?r1 dul:hasRegion ?s1 ;
  dul:hasTimeInterval ?t1. ?t1 :TimeStart ?t10; :TimeEnd ?t11.
  FILTER(myfn:overlaps(?t00,?t01,?t10,?t11) ) }
```

The query will assert triples relating sectors affected from temporally overlapping regulations. Therefore, the time series of trajectories crossing sectors affected by temporally overlapping regulations can be retrieved by the query:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX myfn: <java:datAcron.unipi.gr.sparql_functions.>
SELECT (COUNT(DISTINCT ?t) AS ?count) WHERE {
  { ?s :associatedByOverlappingRegulationWith [] .
    ?t :intendedToCross ?s . ?r0 dul:hasRegion ?s ;
    dul:hasTimeInterval/:TimeStart ?t0 ;
    dul:hasTimeInterval/:TimeEnd ?t1 .
    FILTER(myfn:overlaps(?t0,?t1,$t_start$,$t_end$))
  } UNION { [] :associatedByOverlappingRegulationWith ?s .
    ?t :intendedToCross ?s . ?r0 dul:hasRegion ?s ;
    dul:hasTimeInterval/:TimeStart ?t0 ;
    dul:hasTimeInterval/:TimeEnd ?t1 .
    FILTER(myfn:overlaps(?t0,?t1,$t_start$,$t_end$)) } }
```

where `$t_start$`, `$t_end$` are the time start and time end of the occupancy time period window that slides across the time line. Two of the trajectories crossing sectors affected by regulations are depicted in Figure 7, where the location of regulations is depicted by the (red) dots.

Finally, exploiting the generated links for sectors $\langle S_x, S_y \rangle$ affected by temporally overlapping regulations $R1$, $R2$, we can count the links for a given time interval and a given period. Thus, the time series can be constructed by a sequence of parametrized queries of the form:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dul: <http://www.ontologydesignpatterns.../DUL.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX myfn: <java:datAcron.unipi.gr.sparql_functions.>
SELECT DISTINCT ?s WHERE {
  ?s :associatedByOverlappingRegulationWith [] .
  ?r dul:hasRegion ?s ; dul:hasTimeInterval ?t .
  ?t :TimeStart ?ts ; :TimeEnd ?te .
  FILTER(myfn:during_sf(?ts,?te, $t_start$, $t_end$)) }
```

where `$t_start$,$t_end$` are the start and end times respectively of the sliding time window of the parametrized query.

As depicted in Figure 8, we got (a) total counts of regulated flights between each pair of locations of regulations and (b) their daily time series. Line thickness corresponds to the total counts of regulated flights. Lines have larger curvatures at their ends. Time graphs show the time series for links between pairs of regulation locations. We have observed several peak values in the time series. Six links have peaks on April 28. These links are highlighted on the map.

FM02. To support the FM02 case we need to compute expected demands by counting the number of flight plans in each sector for any occupancy period of a specific duration. Considering that each flight plan is an intended trajectory, the basic query is to retrieve the series of sectors (in this case, both active and inactive) crossed by the intended trajectory, and the entry/exit times for each sector. For example, the following query returns the sectors crossed by the trajectory of a given flight plan, e.g. `:flight_plan_AA51147955`:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dul: <http://www.ontologydesignpatterns.../DUL.owl#>
SELECT ?segment ?entry ?exit ?sector WHERE {
 :flight_plan_AA51147955 a :FM_FTFM; :reportsTrajectory ?t.
 ?t dul:hasPart ?segment. ?segment a :Segment; :within ?sector;
 :hasTemporalFeature ?time . ?time :TimeStart ?entry ;
 :TimeEnd ?exit . } ORDER BY ?s
```

Then, given the sectors and the flight plans, we count the demand per sector S for each time period Δt, by considering the flight plans that cross S in a time period $[Entry, Exit]$ that overlaps with Δt.

5 RELATED WORK

Existing approaches for the representation of semantic trajectories either (a) use plain textual annotations instead of semantic links to other entities [1, 4, 5], hindering the provision of semantic links to other data associated with moving objects' behaviour; (b) constrain the types of events that can be used for structuring a trajectory [1, 4]; or (c) make assumptions on the constituents of trajectories [5, 6, 8] (e.g. semantic trajectories in [5] are sequences of sub-trajectories, while in [6] are sequences of episodes). To a greater extent than previous proposals, the proposed ontology *supports the representation of trajectories at multiple, interlinked levels of analysis*: For instance, although [6] provides a rich set of constructs for the representation of semantic trajectories, these are sequences of episodes, each associated with raw trajectory data, and optionally, with a spatio-temporal model of movement. However, there is no fine association between abstract models of movements and raw data. On the other hand, [5] provides a two-levels analysis where semantic trajectories are lists of semantic sub-trajectories, and each sub-trajectory in its own turn is a list of semantic points. Regarding events and episodes, these are connected to specific resources at specific levels of analysis: In [5] events -mostly related to the environment rather than to the trajectory itself- are connected to points only (something that may lead to ambiguities in some cases), while in [6] episodes concern things happening in the trajectory itself, and may be associated to specific models of movement: It is not clear how multiple models of a single trajectory -each at a different level of analysis- connected to a single episode, are associated. Finally, contextual information in [6] is related to movement models, episodes or semantic trajectories, which is quite generic, while in [5] environment attributes are associated to points only, and are assigned specific values. The datAcron ontology has been succinctly presented in [11]. Here we delve into the details of the specifications, while, also to a great extent than all previous proposals, we have shown the datAcron ontology supports data transformations that are required by analytics tasks, providing information of the appropriate form at various levels of analysis.

6 CONCLUDING REMARKS

This work presents the core specifications and usage in data transformation of the datAcron ontology. This ontology describes trajectories of moving objects at various levels of analysis, towards decision support making and event recognition. We have demonstrated data transformation and visual analytics in Flow Management scenarios of ATM using the proposed ontology. We overcame the limitations of SPARQL 1.1 w.r.t. data transformation requirements, by implementing a suite of functions for verifying spatio-temporal relations and parametrized SPARQL queries that can be iteratively processed on our customized SPARQL endpoint.

As a future work, we plan to also demonstrate data transformations for maritime scenarios and extend the implemented suite of functions for our customized SPARQL endpoint.

ACKNOWLEDGMENTS

This work is supported by the datAcron project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687591.

REFERENCES

[1] Luis Otávio Alvares, Vania Bogorny, Bart Kuijpers, José Antônio Fernandes de Macêdo, Bart Moelans, and Alejandro A. Vaisman. 2007. A model for enriching trajectories with semantic geographical information. In *GIS*. 22.

[2] Gennady Andrienko, Natalia Andrienko, Peter Bak, Daniel Keim, and Stefan Wrobel. Visual Analytics of Movement.

[3] Gennady Andrienko, Natalia Andrienko, Wei Chen, Ross Maciejewski, and Ye Zhao. *Visual Analytics of Mobility and Transportation: State of the Art and Further Research Directions*. Vol. PP/99.

[4] Miriam Baglioni, José Antônio Fernandes de Macêdo, Chiara Renso, Roberto Trasarti, and Monica Wachowicz. 2009. Towards semantic interpretation of movement behavior. In *Advances in GIScience*. Springer, 271–288.

[5] Vania Bogorny, Chiara Renso, Artur Ribeiro de Aquino, Fernando de Lucca Siqueira, and Luis Otávio Alvares. 2014. CONSTAnT - A Conceptual Data Model for Semantic Trajectories of Moving Objects. *Trans. GIS* 18, 1 (2014), 66–88.

[6] Renato Fileto, Cleto May, Chiara Renso, Nikos Pelekis, Douglas Klein, and Yannis Theodoridis. 2015. The Baquara^2 knowledge-based framework for semantic enrichment and analysis of movement data. *Data Knowl. Eng.* 98 (2015), 104–122.

[7] Konstantinos Kotis and George A. Vouros. 2006. Human-centered ontology engineering: The HCOME methodology. *Knowl. Inf. Syst.* 10, 1 (2006), 109–131.

[8] Tales P. Nogueira and Hervé Martin. 2015. Querying semantic trajectory episodes. In *Proc. of MobiGIS*. 23–30.

[9] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady L. Andrienko, Natalia V. Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, José Antônio Fernandes de Macêdo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. 2013. Semantic trajectories modeling and analysis. *ACM Comput. Surv.* 45, 4 (2013), 42.

[10] Donna J Peuquet. 1994. It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers* 84, 3 (1994), 441–461.

[11] Georgios Santipantakis, George Vouros, Apostolos Glenis, Christos Doulkeridis, and Akrivi Vlachou. 2017. The datAcron Ontology for Semantic Trajectories. In *ESWC-Poster Session*.

[12] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, José Antônio Fernandes de Macêdo, Fábio Porto, and Christelle Vangenot. 2008. A conceptual view on trajectories. *Data Knowl. Eng.* 65, 1 (2008), 126–146.