



City Research Online

City, University of London Institutional Repository

Citation: Vouros, G. A., Santipantakis, G. M., Doulkeridis, C., Vlachou, A., Andrienko, G., Andrienko, N., Fuchs, G., Cordero Garcia, J. M. & Martinez, M. G. (2019). The datAcron Ontology for the Specification of Semantic Trajectories. *Journal on Data Semantics*, 8(4), pp. 235-262. doi: 10.1007/s13740-019-00108-0

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/23210/>

Link to published version: <https://doi.org/10.1007/s13740-019-00108-0>

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

The datAcron Ontology for the Specification of Semantic Trajectories

Specification of Semantic Trajectories for Data Transformations Supporting Visual Analytics

George A. Vouros, Georgios M. Santipantakis,
Christos Doulkeridis , Akrivi Vlachou ·
Gennady Andrienko, Natalia Andrienko, Georg
Fuchs · Jose Manuel Cordero Garcia, Miguel
Garcia Martinez

Received: date / Accepted: date

Abstract As the number of moving objects increases, the challenges for achieving operational goals w.r.t. the mobility in many domains that are critical to economy and safety emerge dramatically. This, in domains such as the Air Traffic Management domain, dictates a shift of operations' paradigm from *location-based*, as it is today, to *trajectory-based*, where trajectories are turned into “first-class citizens”. Additionally, the increasing amount of data from heterogenous and disparate data sources, imply the need for advanced analysis methods that require exploiting spatio-temporal mobility data in appropriate forms and at varying levels of abstraction. All these call for an in-principle way for organising integrated views of mobility data, with trajectories playing the main role. In this paper, based on a comprehensive framework identifying fundamental spatio-temporal data types and specific conversions among these types, we propose an ontology for modelling semantic trajectories, integrating spatio-temporal information regarding mobility of objects, at multiple, interlinked levels of abstraction, as needed by analysis methods. We validate the ontological specifications towards satisfying the needs of visual analysis tasks in the complex Air Traffic Management domain, using real-world data.

1 Introduction

Many tasks in critical domains w.r.t. economy and safety, such as Flow and Traffic Management in the aviation domain, impose emergent and challenging problems, re-

Department of Digital Systems
University of Piraeus, Greece
E-mail: georgev@unipi.gr

Institute IAIS
Fraunhofer, St. Augustin, Germany

CRIDA
Spain

quiring analysis of mobility data. This data comprises surveillance data in association with data about the moving objects (e.g. geometric information, objects' physical and operational characteristics), and contextual information (e.g. regions and points of interest, weather information, traffic, etc.), coming from disparate and heterogeneous data sources.

Challenging problems include effective information provision for situation awareness, identification of recurrent patterns of behaviour and decision making at different scales and levels of abstraction, as well as, the prediction of moving objects' behaviour under specific circumstances. These challenges are significant, given that their achievement aims to reduce factors of uncertainty regarding operations, enhance punctuality of activities, advance planning efficiency, and reduce operational costs.

To address the complexity of these challenges, which increases significantly as the number of moving objects increases, a shift of operations' paradigm from *location-based*, as it is today, to a *trajectory-based* has been proposed: Trajectories are turned into "first-class citizens", and mobility of objects, decision making, assessment of situations and planning of operations revolve around the notion of trajectory.

Following a data-driven approach towards addressing these challenges, we need to consider how we do represent trajectories to satisfy the data needs and requirements of analysis tasks. Our approach is based on two principles, towards supporting exploratory tasks requiring the synergy between humans and computational tasks: First, trajectories should reveal objects' behaviour in explicit terms, at different levels of abstraction considering their geometric, contextual, and analysis-specific features. In doing so, analysis tasks can retrieve data about trajectories at any level of abstraction that is appropriate for their purposes, switching between abstraction levels, delving into the details of mobility phenomena and providing overviews in generic terms. Second, data transformations (or conversions) require trajectories to integrate spatial events into temporal sequences, while, on the other hand, these events need to be aggregated into spatial time series, associated to geographic contexts. This supports identifying re-occurring patterns of behaviour. Combining these abilities, allows identifying re-occurring patterns of behaviour at varying levels of abstraction, enhancing our understanding of mobility phenomena and thus, decision making.

At this point we need to clarify that, as far as abstraction is concerned, aggregation and generalisation provide the basic means for abstracting data/information. In this paper when we refer to "abstraction" we consider any possible combination of aggregation and generalization,. When it is necessary, we specify explicitly which kind of abstraction is required.

Our objective is to specify an ontology for modelling semantic trajectories, integrating spatio-temporal information regarding mobility of objects at multiple, inter-linked levels of abstraction, supporting appropriate data transformations, as needed by visual analysis tasks. Visual analytics impose specific requirements to support the combination of human and computational data processing through interactive visual interfaces, enabling analysis of spatio-temporal [8] and mobility data [5], sophisticated data analysis, and informed decision making [9], at varying levels of abstraction.

Existing models and ontologies for the representation of semantic trajectories do not provide the flexibility needed to represent semantic trajectories and associated

data and events at multiple levels of abstraction. They usually specify models for representing trajectories at different levels (from raw to semantic), where each level associates trajectories with a different kind of information. In cases where abstractions (mainly aggregations of geometric information) are supported, these are limited to specific types of abstraction and to a restricted number of levels, and in many cases present limitations in associating abstract specifications among themselves. Consequently, switching between levels of abstraction as needed by exploratory analysis tasks is limited. This, implies weaknesses to the exploration of data for the purposes of analysis task, and in particular for the purposes of visual analytics. These issues are further discussed in detail with explicit references to existing trajectory models, in section 3.

Specific contributions that this work makes are as follows:

(a) We revisit fundamental data types for visual analysis tasks revolving around the notion of semantic trajectory, specifying conversions among these types of data: These types and conversions provide an in-principle framework for identifying trajectories' constituents, and provide a comprehensive framework for validating ontological specifications towards the provision of appropriately transformed data, satisfying data requirements of visual analysis methods.

(b) We revisit the notion of "semantic trajectory", as a meaningful sequence of trajectory parts at any level of abstraction. By being meaningful, a semantic trajectory is associated with human-interpretable and machine-processable information, revealing objects behaviour in explicit terms. Dealing with multiple levels of abstraction, we support analysis of moving objects' behaviour at any scale and/or level of abstraction that is appropriate for analysis tasks.

(c) We validate the ontology by means of enhanced SPARQL queries, using real-world data from the Air Traffic Management domain, in concrete cases of importance for flow management.

The paper is organised as follows: Section 2 motivates the need for an ontology for the representation of semantic trajectories, specifying requirements for such a representation. It further revisits fundamental data types and outlines the data transformations for supporting analytics tasks w.r.t. trajectories. Section 3 reviews weaknesses and limitations of existing proposals for representing semantic trajectories. Section 4 provides background information about the Flow Management cases of the Air Traffic Management domain for validating ontological specifications. Section 5 presents the datAcron ontology for the representation of semantic trajectories. Section 6 presents how data transformations are supported by the ontological specifications, supporting visual analytics tasks for the purposes of Flow Management cases. The paper concludes with discussion remarks and plans for future work in Section 7.

2 Semantic Trajectories: Requirements and Fundamental Types of Information

Our main goal is to specify an ontology that provides a comprehensive semantic model for the representation of trajectories, integrating spatio-temporal information regarding mobility of objects at multiple, interlinked levels of abstraction, supporting appropriate data transformations, as needed by visual analysis tasks.

Towards this objective, first we specify the requirements for the representation of semantic trajectories defining important terms, and then we recall the fundamental types of spatio-temporal mobility data and data transformations/conversions appropriate for supporting mobility analytics, with a clear focus on visual analytics: Fundamental data types and conversions provide a comprehensive framework for validating ontological specifications.

2.1 Requirements for the representation of semantic trajectories

Aiming to a comprehensive semantic model of trajectories that integrates mobility data, we aim at representing all these features that are necessary to the representation of semantic trajectories, including geometric, geographic and application specific information [33].

As reported in [33], geometric information concerns the evolution of location of a moving object during a given time interval. The temporal sequence of raw (surveillance) data specifying the moving object spatio-temporal positions reported from sensing devices defines a *raw trajectory* [22]. Using geometric information we may answer queries like “Return objects which were located at x, y, z at time t ”. However, geometric information may be specified at varying levels of aggregation, revealing representations and knowledge regarding the behaviour of a moving object at different spatio-temporal scales, which can be useful for various tasks: For instance, a trajectory may be represented as a line rather than as a temporal sequence of positions. Such a representation may ease computations that reveal patterns of movement, or computations regarding spatial/topological relations with other geometries. Alternatively, a trajectory can be represented as a temporal sequence of lines representing sub-trajectories, each one of special interest on its own (e.g. each one crossing a specific region of interest, or corresponding to a specific phase of movement), or as a sequence of aggregated raw positions with high concentration in spatio-temporal regions or points of interest.

The reader may have noticed that in explaining the significance of specifying a trajectory at multiple levels of geometric abstraction, we “associated” geometric information with geographical (e.g. special areas or points of interest) and application-specific (e.g. phases of movement) information. This further supports the usefulness of having multiple levels of geometric abstractions, each one serving different purposes towards representing and analysing the behaviour of moving objects. Having these geometric abstractions, we may answer queries like “Return objects that crossed the spatial region X during the time interval $[t_begin, t_end]$ ”, “Return objects whose trajectories crossed spatial regions that properly include region X during the time interval $[t_begin, t_end]$ ”, and “Return objects whose trajectories include an aggregation of positions close to a specific point of interest”.

Different levels of geometric abstraction provide alternative constituents for structuring trajectories. According to [22] a *structured trajectory* consists of a sequence of *trajectory parts* that can be either *raw positions* reported from any sensing device, aggregations of raw positions referred as *nodes*, or *trajectory segments*.

A trajectory *segment* is a trajectory itself, which may be part of a whole trajectory. A *node* provides an aggregation of raw positions. Segments and nodes aggregate information that may instantiate a behaviour pattern. For example, a sequence of raw positions may instantiate a “turn” event, while a set of positions may instantiate a “stop” event. These aggregations can be represented by a single node or segment, associated to an event type (e.g. “turn” or “stop”, respectively), and to the corresponding set of raw positions.

Segments of trajectories and nodes can be defined with different objectives depending on the application and target analysis, and are thus associated with application-specific information. As defined in [22], a maximal sequence of raw data that comply with a given pattern defines an *episode*. In this work we consider *events* as a generalisation of episodes. *Events* represent specific or abstract happenings and are associated to trajectory parts, providing application specific information that is relevant to the trajectory. As a consequence, queries such as “Return objects whose trajectories contributed to congestion events in a specific spatial-temporal region”, or “Return objects whose trajectories comprise a segment that is associated with a high-speed event” can be answered.

Geographical features allow turning the geometric information representing the spatial path into a geographical trace [33] which is meaningful for humans and computational processing tasks. This requires associating trajectory parts to (types of) geographic regions: Shops/spots/buildings of different kinds, regions of special interest (e.g. touristic, commercial or industrial), etc. Generalising geographical features, we can draw semantic associations between trajectory parts, supporting further the abstraction of trajectories (e.g. any trajectory crossing many shops can be a “shopping trajectory”, irrespectively the kind of shops crossed. Specific types of shopping trajectories may indicate specific types of shops crossed). In this work, we generalise geographical features to *contextual*. This comprises features of the moving objects, as well as features of moving objects’ environment, considering that these features are associated to objects’ movement. These may include weather attributes, space configuration features, as well as aggregated data about co-occurring trajectories – i.e., traffic. This enables answering not only queries such as “Return trajectories that crossed region X”, but also queries such as “Return trajectories that crossed any region with specific weather conditions [specified as conditions in weather attributes]”.

It must be noted that events aggregate different types of features. An event pattern may comprise contextual features (e.g. crossing a spatial region, or a region with a specific weather condition), features of moving objects (e.g. reaching highest possible altitude), geometric and geographical features, and/or other events regarding the mobility of the object (e.g. moving in low-speed or descending). Events may be *low – level* - associated to basic behaviour - or *complex* - associated to complex patterns of behaviour.

A trajectory part may be associated with any event that co-occurs with it spatially and/or temporally: E.g. Bad weather conditions, or traffic regulations associated with a spatial 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. By being meaningful, a semantic trajectory is associated with contextual information and re-

lated events, towards revealing objects' deliberative or accidental behaviour in explicit terms, thus contributing to understanding the rationale for that behaviour and providing comprehensive information about the occurring behaviour.

Given the above definition of a semantic trajectory, a semantic trajectory can be specified at different levels of abstraction, depending on the geometric features, contextual features, and events considered. Abstraction may happen by means of aggregation, generalisation, or both. In doing so, we may retrieve semantically associated trajectories, based on the semantic features they aggregate and information to which they are associated. For instance we may retrieve "trajectories crossing sensitive areas and associated to suspicious events". Such trajectories may be represented at varying aggregation levels. They may cross areas with different types of sensitivity and they may be associated with different types of suspicious events.

We conjecture that abstractions of a single trajectory should be interlinked, so as any application to be able to get any relevant information that is necessary for its purposes, being able to move in a continuum between specialised / basic information and generalised / aggregated information, through querying and applying data transformations. This supports, for instance, delving into the details regarding a trajectory part associated to a complex event of type "suspicious behaviour", by inspecting geometrical, contextual and application-specific features at the appropriate level of detail.

2.2 Fundamental Data Types and Data Transformations for Visual Analytics

Given our aim to represent trajectories towards supporting data-driven approaches to challenging problems in critical domains, this section presents generic spatio-temporal data transformations to serve analysis goals on mobility data.

As mentioned in [7], there are three fundamental types of spatio-temporal data associated with mobility: trajectories of moving objects, spatial event data, and spatial time series.

Individual trajectories provide information on the movement of individual objects. Aggregated traffic data are *spatial time series* describing how many moving objects were present in different spatial locations and/or how many objects moved from one location to another during different time intervals. The time series may also include aggregate characteristics of the movement, such as the average speed and travel time. Time series describing the presence of objects are associated with distinct locations, and time series describing aggregated moves (often called fluxes or flows) are associated with directed links between pairs of locations. In both cases *spatial time series* are represented as chronologically ordered sequences of values of time-variant thematic attributes associated with spatial locations or spatial entities (for example, regions of special interest).

Spatial events emerge at spatial locations and exist for a period of time. Spatial events are described by their spatial regions, existence times, and contextual features. Events may occur irrespectively of trajectories, but somehow be related to trajectories (e.g. weather events, regulations imposed in a spatio-temporal region), or may be derived from trajectories (e.g. a turn of a moving object, short distance between a pair of objects, or large number of moving objects in a spatio-temporal region).

Based on these types of spatio-temporal data and following the approach of [23], the fundamental types of queries can be seen as transformations combining three basic components: (a) space (*where*), (b) time (*when*), (c) object or event (*what*). These components can be used in three basic types of queries:

- Retrieve the trajectories/events in a region for a time period (*when&where* → *what*).
- Retrieve the region occupied by a trajectory/event or set of trajectories/events, at a given time instant or period (*when&what* → *where*).
- Retrieve the time periods that a non-empty set of trajectories/events appear in a specific location or area (i.e., *where&what* → *when*).

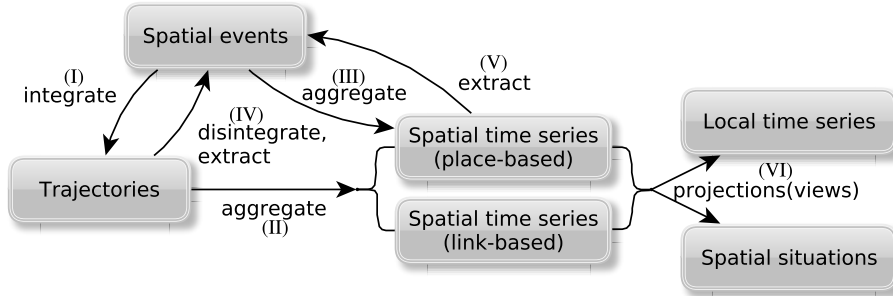


Fig. 1 Conversions between different representations

Exploiting these fundamental data types and queries, we aim to support the generic transformations depicted in Figure 1 [7], in support of visual analytics tasks. Briefly, as Figure 1 shows, trajectories integrate spatial events (transformation I), while these events, similarly to trajectories, may be aggregated to spatial time series. These may be either place-based, i.e. associated to a specific spatial region (transformation III), or link-based, such as flows of trajectories between pairs of spatial regions (transformation II). Projections of spatial time series may result to spatially-referenced time series or to spatial situations (transformations VI). These transformations impose specific requirements to representations, so as to answer queries regarding trajectories, aggregations of features and events.

More specifically, the left part of the diagram in Figure 1 shows the tight relationships between spatial events and trajectories. In fact, trajectories comprise parts that are associated to spatial events. Even in raw trajectories, each record represents the presence of an object at a specific location at some instant in time. As it is further shown in Figure 1, trajectories are obtained by integrating spatial events. In the simplest case, for each moving object, all (raw) position records are linked in a chronological sequence. Reciprocally, trajectories can be transformed to spatial events either by full disintegration back into the constituent events or by extraction of particular events of interest such as sharp turns, entering/exiting a region, crossing a waypoint, etc. Spatial events that are close in space and time can be united into more complex spatial events. For example, a spatio-temporal concentration of many moving objects

entering/crossing a spatial region during a small time window may be treated as a single event of traffic congestion.

Spatial time series can be obtained from spatial events or trajectories through spatio-temporal aggregation. For instance, spatial regions specify *spatial compartments*, and time can be divided into intervals called *time windows*. For each spatial compartment and time window, the spatial events or moving objects that appear in the compartment during the associated time window are binned together and counted. The result is a place-based time series in which temporal sequences of aggregate values are associated with the spatial compartments. From such spatial time series, in turn, it is possible to extract more complex spatial events; for example, events of high traffic density and high demand for a specific spatial region and for specific temporal intervals.

Trajectories can also be aggregated into link-based time series: for each pair of spatial compartments and for a specific time window, the objects that moved from the first to the second compartment during this time interval (specifying a *link* between compartments during that period) are counted. Aggregated characteristics of their movement may be calculated.

Discrete place-based and link-based spatial time series can be viewed in two complementary ways. On the one hand, they consist of temporally ordered sequences of (aggregated) values associated with individual places or links, i.e., *local time series*. On the other hand, a spatial time series is a temporally ordered sequence of the distribution of spatial events, moving objects, or collective moves (flows) of objects over the whole space of interest, together with the spatial variation of various aggregate characteristics. These distributions are called *spatial situations* [5].

Based on the requirements for the representation of semantic trajectories specified in the first part of this section, and the framework of fundamental types of mobility data and conversions between them, presented in the second part of this section, we proceed to propose a model for the representation of semantic trajectories, which aims at (a) supporting the representation of semantic trajectories at multiple, inter-linked levels of abstraction, (b) structuring trajectories by means of different types of trajectory parts, (c) associating events at varying levels of abstraction with trajectory parts, (d) supporting the transformations needed for visual analysis tasks.

3 Related work

Existing approaches for the representation of trajectories, either (a) use plain textual annotations instead of semantic associations to features of interest [3], [12], [13], having limitations towards machine-processable information for the purposes of mobility analysis tasks; (b) constrain the types of events that can be used for structuring a trajectory [33], [3], [12] [29]; or (c) make specific assumptions about the constituents of trajectories [32], [29], [13], [20] [15], [17], thus providing limitations to the specification of trajectories at varying levels of abstraction according to needs.

To a greater extent than previous proposals, we aim to support the representation of trajectories at multiple, interlinked levels of detail.

More specifically, although authors in [15] provide a rich set of constructs for the representation of semantic trajectories, these are specified as sequences of episodes, each associated with raw trajectory data, and optionally, with a spatio-temporal model of movement. Beyond representing trajectories only as sequences of episodes, there is no fine association between abstract models of movement and raw data, providing limitations to analysis tasks that need both of them in association. On the other hand, [13], [29] and [32] provide a two-levels analysis where semantic trajectories are lists of semantic sub-trajectories, and each sub-trajectory is a list of spatial points. Authors in [17], based on the two-levels analysis of trajectory models, introduce an ontological pattern for the specification of trajectories.

Regarding events and episodes, most of the proposed models are based on the “stop-move” model [34] [30], or they are connected to features at specific levels of abstraction: In [13] events -mostly related to the environment rather than to the trajectory itself- are connected to points. This may lead to ambiguities as far as the association of events to trajectories crossing the same points is concerned, especially for the events concerning the trajectory itself rather than the environment. In [15] episodes concern things happening in the trajectory itself, and may be associated to specific models of movement. However, 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. Contextual information in [15] is related to movement models, episodes or semantic trajectories, which is quite generic as a model, while in [32], [29] and [17] fixes and states represent basic behavioural features of the moving object. These may also represent contextual features, and are associated to trajectory points, or in [29] they specify domain-specific features. Finally, in [13] environment attributes are associated to points only, and can only be assigned specific values.

As noted in the previous section, the specification of trajectories at various layers, from raw to semantic, depending on the information associated to trajectories (as it is done in [33]) is orthogonal to the goal of providing specifications of trajectories at multiple levels of abstraction. A different approach to that is proposed in [21], where trajectories are associated with qualitative descriptions of movement, at different aggregation levels, much like the distinction between low-level and complex events made above. However, trajectories are specified as sequences of segments associated to at least two key points providing quantitative information on movement, with no association to any type of events or activities.

This lack of flexibility to specify semantic trajectories at multiple levels of abstraction regarding geometric and contextual information, as well as events, and the lack of the capability to link these specifications so as to be able to switch between abstractions flexibly, is a common feature among previous efforts. In addition to that, to the best of our knowledge, there is no work that considers the requirements of analysis tasks in structuring trajectories, so as to support fundamental types of data and transformations between them.

Specifically, considering data transformations for analysis tasks, apart from the structural transformations between or within the different types of spatio-temporal data specified in section 2, there exist transformations that change the scale, or level of detail, which may be beneficial for particular tasks. For example, Chu et al. [14] transform trajectories into sequences of traversed map regions (e.g.streets) and apply

text mining methods for discovery of "topics", i.e., combinations of regions that have a high probability of co-occurrence in one trip. The extraction of "topics" is done for different time intervals. By investigating the temporal evolution of the topics, it is possible to understand where objects travel in different times of the day and days of the week. Al-Dohuki et al. [1] transform trajectories into texts consisting of region names and text labels denoting speeds (low, medium, and high). Furthermore, a discrete representation of aggregated movements between places can be treated as a graph, to which graph analysis methods can be applied [16][19]. As such, these various transformations enable the comprehensive analysis of traffic data from multiple complementary perspectives [10].

To the best of our knowledge, the ontology presented in this paper for the specification of semantic trajectories, namely the *datAcron ontology*, is the first one to provide the flexibility needed to represent trajectories at multiple, interlinked levels of abstractions. Furthermore, and to a greater extent than other models and ontologies proposed, it is validated in the context of data transformations needed by analysis tasks, in highly complex problem cases in the aviation domain.

The *datAcron ontology* has been succinctly presented in [28,25]. Here we delve into the details of the specifications, while, also to a greater extent and detail than all previous publications, we show in detail how the *datAcron ontology* supports a wide range of generic data transformations that are required by analysis tasks, supporting the provision of information at various levels of analysis and form.

4 The Flow Management domain.

To further contextualize our work and show concrete examples of specifications, in this section we provide basic background information and specify the types of entities and data required in data analysis scenarios from the Air Traffic Management (ATM) domain, concerning Flow Management (FM).

Flow management (FM) provides some of the most explorative scenarios in the aviation domain. Actually, 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. In brief, its objectives include (a) detecting cases where air traffic demand at times exceeds the available capacity of the ATC system and (b) imposing flight regulations to resolve these demand-capacity imbalances.

In demand-capacity imbalance situations the expected number of flights in some part of the airspace, called sector, exceeds the prescribed sector capacity, i.e., the capability of the air traffic controllers responsible for this sector to handle flights safely: This results to the occurrence of *hotspots*. Regulations change the departure times of some flights in order to prevent sector overload. This results in flight delays, and thus, increased costs of operations. These delays can also result in creating new hotspots at other times and/or in other sectors, with cascading effects to the whole system. The capability to predict the emergence of hotspots well in advance could be used to improve flight planning, decisions on active sector configurations used (specified below), and it can improve assessment of regulations that should be imposed to re-

duce the occurrences of hotspots, resulting in fewer and smaller delays. However, it is currently not clear how to achieve these goals. This is mainly due to the existing factors of uncertainty, and thus to the low predictability of the actual operations (e.g. of actual trajectories and events) taking place at specific time instants. Hence, it is necessary to analyse available historical data to identify patterns of human experts' decision making, revealing expert knowledge (e.g. features, rules and criteria) likely to be used in different circumstances.

As the general mission of visual analytics is to provide techniques and tools supporting human understanding of data, comprehension of the phenomena reflected in the data and analytical reasoning, the exploration of the complex historical data relevant to the FM cases is an appropriate task for visual analytics.

4.1 The Flow Management entities

In the following, we provide a comprehensive list of the FM entities, along with details.

- *Flight plans* provide specifications of planned or intended trajectories consisting of spatiotemporal events, such as of flying over specific waypoints (i.e. fixed coordinates among which airways are set). Flight plans also specify information concerning estimated take-off time, and, in case of delay caused by a regulation, the calculated take-off time of the flight.
- *Air blocks*, are static airspace volumes defined by geometries specifying spatial 2D projections of the airspace volume, and lower and upper flight levels.
- *Sectors* are static spatial 3D objects comprising other sectors or airspace volumes that are defined by air blocks. Each sector is managed by a specific number of air traffic controllers (typically two, Executive and Planning Controllers).
- *Sector configurations* are alternative divisions of airspace into sectors. Sectors constitute the minimum unit that an Air Traffic Controller operates. The number of sectors dividing the airspace may vary in different times, allowing to operate the airspace with the appropriate number of controllers according to demand conditions, ensuring safety of operations in low cost.
- *Opening schemes* or *active configurations* are the sector configurations actually deployed in a given airspace, associated with time intervals of their validity. The schedule of active sector configurations is continuously refined as getting closer to operation time, when the available information about flight plans (and thus, demand) is progressively refined. This introduces an uncertainty factor to the planning of operations.
- *Capacities* are referring to sectors: for each sector at a specific time instant, the capacity value of that sector may either be undefined (if the sector is not active at that time instant) or specify the upper limit of the number of flights crossing that sector in a time period with pre-specified duration (typically one hour). The capacity of a sector is the same at any time instant at which it is active.
- *Predicted weather* is a spatial time series of multiple predicted weather attributes referring to 3D locations (longitude, latitude, altitude).

Figure 2 illustrates two alternative sector configurations in the Spanish airspace.

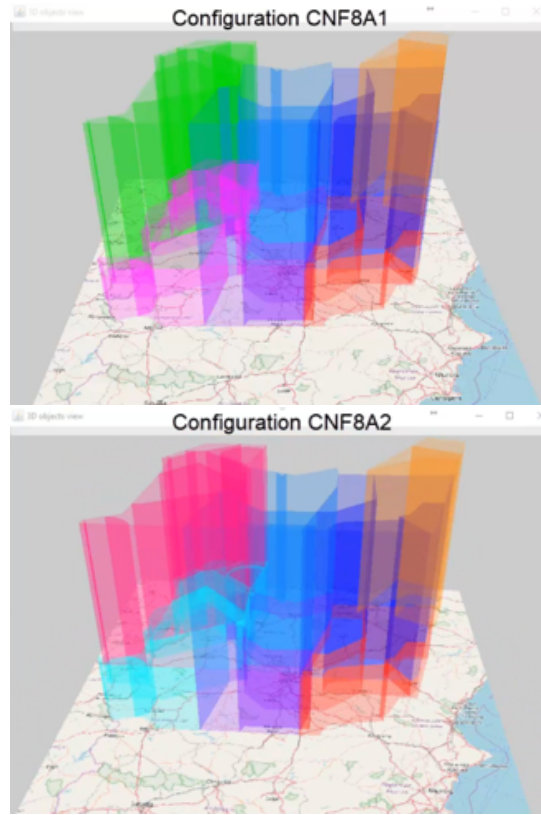


Fig. 2 Configurations of sectors in the Spanish airspace. Colours are for distinguishing between sectors. Illustrations have been created using the V-Analytics platform [6].

The FM monitoring process computes periodically (typically every 20 minutes) the foreseen 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 for a specific sector a regulation may be applied to adjust the demand values to the available capacity for that sector.

A *regulation* is a special type of event that occurs as a measure that a network manager takes to solve an excess of capacity. The attributes of any regulation include the location (sector), start and end times, and reason codes (e.g. "C" for regulations due to demand capacity imbalances, or "W" for regulations due to weather conditions).

Regulations imposed to sectors usually result in delays imposed to flights crossing that area. Delayed flights may cause hotspots (and thus, new regulations) to other sectors in the airspace, etc. This introduces an additional factor of unpredictability to

operations. Therefore, we need to understand such cascading effects and provide the ability to stakeholders to plan the occurrence of regulations well in advance, reducing uncertainty.

On the other hand, ideally, sector configurations should be chosen so that the demand in each sector does not exceed the sector capacity, thus, not imposing the need to regulate flights, while making efficient use of resources. In reality, demand-capacity imbalances happen quite often for a set of reasons (deviations of actual flights from flight plans, weather conditions, strikes etc.), causing flight regulations and delays, contributing to unpredictability of operations. In search for models that might support enhanced pre-tactical planning, we need to understand how configuration choices are made by airspace managers, and how trajectories are planned by airspace users, allowing better management of demand-capacity imbalances and assessment of regulations at the pre-tactical stage of operations.

Towards supporting mobility analytics to address the FM specific challenges and to achieve the operational goals, it is clear that we need to exploit information about trajectories at multiple levels of detail: Raw trajectories should be represented, and these should be linked to segments of trajectories that are spatially included into airspace compartments: Such compartments of interest are sectors, which however may be active (i.e., be part of an active configuration) or not. Given that different sectors are "constructed" from the same air blocks, we can specify trajectories as series of segments crossing air blocks, which are then aggregated - depending on the aggregation of air blocks into sectors - on series of segments crossing active sectors. As an "intermediate" level of representation between trajectories as series of raw positional data and as series of trajectory segments, we can specify trajectory nodes associated to events of importance, and thus with spatial and temporal contextual features (e.g. entering/exiting an air block). Trajectory parts may be associated to events and features regarding weather conditions, regulations, traffic. etc. Contextual features and events can be specified at varying levels of generalisation, supporting semantic associations between trajectories and their parts (e.g. trajectory segments crossing air blocks regulated due to any reason, or trajectory segments crossing air blocks regulated due to weather conditions or traffic).

Specific FM cases, specifying analysis targets, data sets and data transformations needed are detailed in Section 6.

5 The datAcron Ontology

The datAcron ontology¹ was developed by group consensus over a period of 12 months following a data-driven approach according to the HCOME methodology [18]. It is a $\mathcal{SIN}(\mathcal{D})$ ontology, according to the Description Language notation for the expressiveness of ontologies, and has been designed to be used as a core ontology towards integrating data from heterogeneous data sources of surveillance and contextual data, in association to recognised (low-level and high-level) events, towards supporting analysis tasks exploiting semantic trajectories.

¹ http://ai-group.ds.unipi.gr/datacron_ontology/

Following the HCOME methodology, the following specific phases of engineering have been followed:

Specification of aim, scope, requirements and identification of collaborators: In this initial phase, we had to be acquainted with terminology regarding semantic trajectories and with analysis goals related to mobility data in several scenarios in two critical domains: Air Traffic Management and Maritime Situation Awareness. Thus, we had to identify the data requirements of analysis tasks, and specify the queries to be answered from the ontology. The fundamental data types specified in section 2.2 provide the basic framework for representing and exploiting mobility data through transformations.

Knowledge acquisition, development and ontology maintenance: The development of the datAcron ontology has been driven by ontologies related to our objectives: DUL, SimpleFeature, NASA Sweet and SSN, as well as schemes and specifications regarding data from different data sources. These ontologies served as top ontologies, whose specifications are further refined to the specification of datAcron and domain-specific classes/properties. Standard ontology development and maintenance tasks (e.g. improvisation, versioning, documentation), together with consultation from experts on data analysis and domain-specific tasks took place. It must be pointed out, that following a data driven approach, the major goal was to provide “interfaces” with computational and analysis tasks that either provide data to populate the ontology, or fetch data to be exploited for analysis purposes. Thus, ontological specifications should support ontology population and querying in adequate and lossless ways. I.e. annotating, representing and associating data using the appropriate terms, adequately, and without losing any valuable bit of information that would affect analysis results.

Exploitation and Validation: During this phase, the ontological specifications have been validated in (a) populating the ontology by means of RDF generators, and in (b) providing data in appropriate forms for data analysis tasks. Refinements of ontological specifications proposed during this phase, or changes in the required features to be exploited, had to be incorporated in the ontology.

It must be pointed out that these phases happened iteratively: E.g. the specification of a new data source providing any kind of features in different forms, trigger the first phase, with potential consequent activities in the other phases.

5.1 Core vocabulary and overall structure

As explained in section 2.1 and illustrated in Figure 3, a trajectory (Trajectory) can be segmented to several trajectory parts (TrajectoryParts). Each trajectory part can be a trajectory segment, a trajectory node, or a position provided by a raw surveillance data source. Segments and nodes can be further analysed iteratively to other, less abstract trajectory parts.

The generic pattern of specifying structured trajectories is presented in section 5.2.

Trajectories and trajectory parts can be associated with geometric and contextual information, as well as with events represented by the class `dul:Event`. As already pointed out, events are important happenings associated to the mobility of objects. These may occur in the environment of moving objects and affect their mobility, or may be derived from trajectories. Ontology patterns for associating contextual information and events to trajectory parts are presented in section 5.2.

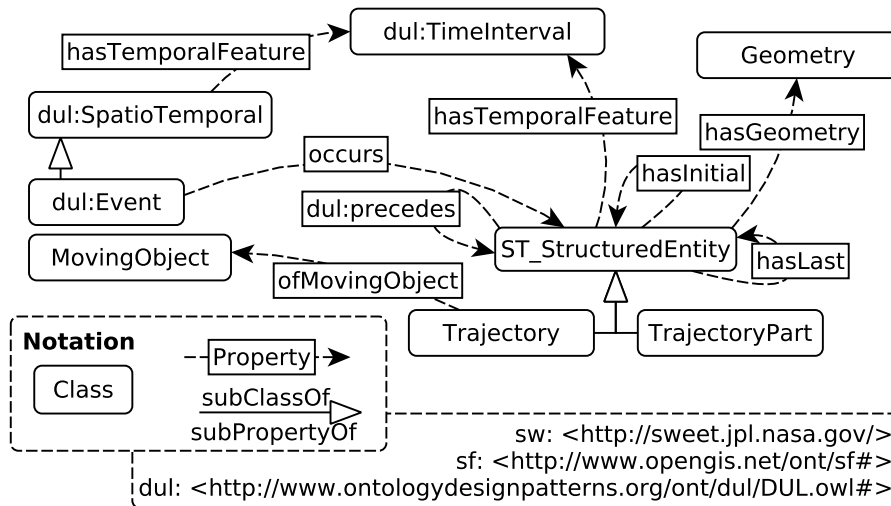


Fig. 3 The main concepts and relations of the proposed ontology.

5.2 Patterns for the representation of semantic trajectories

Figure 4 illustrates the generic pattern of structured trajectories. The main concept in this pattern is the `Trajectory`, which is a subclass of `Spatio-Temporal Structured Entities` represented by the class `ST_StructuredEntity`. 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 (e.g. a storm covering an area), while it also represents (structured) trajectories and their parts. A structured trajectory, as well as any of its parts of type `TrajectoryPart`, can be a temporal sequence of `TrajectoryPart` entities.

Direct subclasses of `Trajectory` are the

- `IntendedTrajectory`: planned trajectories specified by an `dul:InformationEntity` such as a `FlightPlan`,
- `ActualTrajectory`: trajectories constructed from actual positioning data, after compression of the raw positional data, preserving positional data and associated with low-level events representing important trajectory changes (e.g. turns, increase/decrease of speed, change of altitude etc),
- `RegulatedTrajectory`: trajectories that have been modified by an operational event, such as a regulation,

- **RawTrajectory**: trajectories constructed by the raw unprocessed sequence of positional data of moving objects.

An **ActualTrajectory** can be further distinguished to a **ClosedTrajectory** (i.e., a trajectory that has reached its destination) and to an **OpenTrajectory** (i.e., a trajectory in progress).

The **TrajectoryPart** class is further refined to 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 interval. The latter holds in case the node aggregates several raw positions. A **Node** can be the result of a data processing component computing aggregations and abstractions of raw positional data.
- **RawPosition**: represents the raw (unprocessed) positional data. Each raw position instance is associated to a point in space and a time instant.

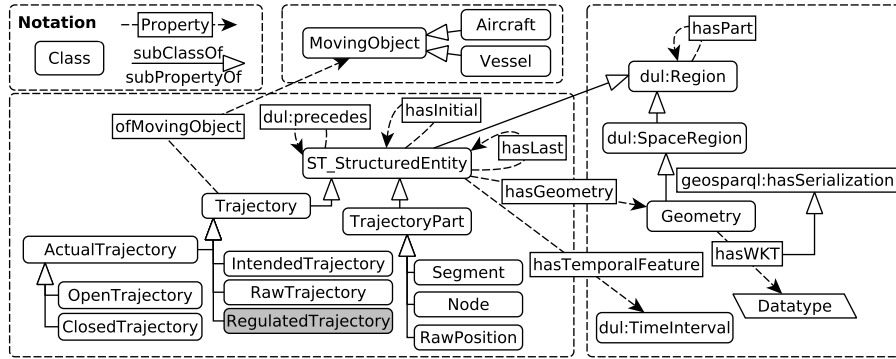


Fig. 4 The pattern of structured trajectories. Domain specific concepts in gray

A specific trajectory, as well as any of its trajectory parts, being instances of `dul:Region` can be associated to its parts via the `dul:hasPart` property or via the subproperties `hasInitial`, `hasLast` which indicate the first and last part of the `ST_StructuredEntity`, respectively. For instance, a trajectory may comprise a sequence of trajectory segments (e.g. segments within sectors), who on their own turn comprise other segments (e.g. segments within air blocks), nodes (e.g. entering or exiting any airspace compartment), 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, we can keep a long history of its movement. It must be noted that this combination of properties supports sharing trajectory parts between trajectories even of the same object with no ambiguity: For instance, a trajectory node or segment can be shared between the actual and the intended trajectory of an aircraft, without mixing the trajectories.

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 data property `hasWKT`, which is sub-property of `geosparql:hasSerialization`.

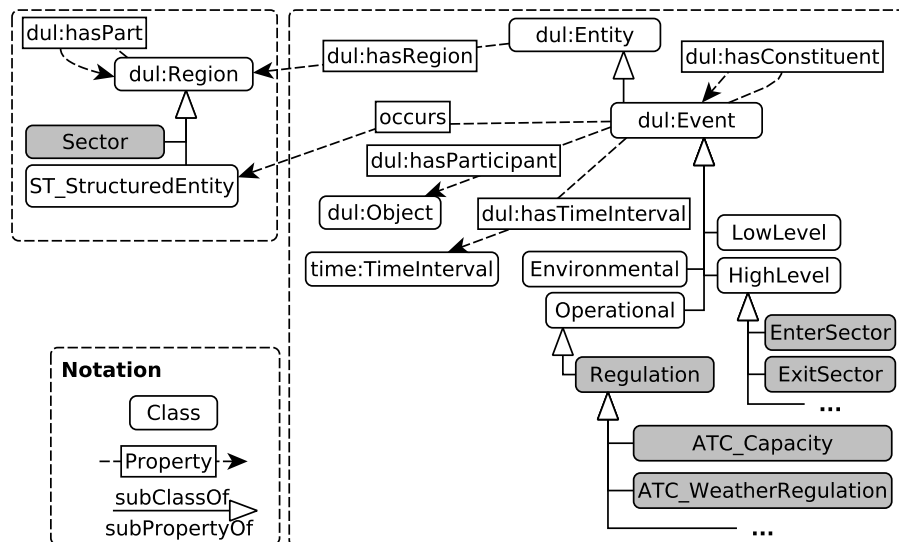


Fig. 5 The pattern of trajectories linked with events. Domain specific concepts in gray

Trajectories and trajectory parts can be associated with events and contextual features of importance. Specifically, events can be associated with any `ST_StructuredEntity` (i.e. with any trajectory and trajectory part), via the property `occurs`. This is illustrated in Figure 5. An event can be associated with other events via the properties `dul:hasConstituent` or `dul:hasPart`. This is the case for high-level (complex) events (e.g. hotspot occurrence) associated with other high-level (e.g. regulation imposed to a sector and events signifying individual flights entering a sector) or low-level events. An event may involve participants (associated via the property `dul:hasParticipant`) and it holds for a specific `TimeInterval` specified by the property `dul:hasTimeInterval`. An event can be a:

- *LowLevel* event, in case its detection requires data from a single trajectory: For instance a `TopOfClimb` is such an event.
- *HighLevel* event, in case its detection requires contextual data and maybe, data from multiple trajectories. For example, events of type `EnterSector` involve information about sectors crossed by a trajectory. As another example, the occurrence of hotspots requires data about sectors and multiple trajectories.

Orthogonal to the classification between low-level and high-level events, we also have the following classes of events:

- *Operational* event, if it is issued by operators, affecting regions or groups of entities for a specific time interval. For example, a regulation (`Regulation`) is ap-

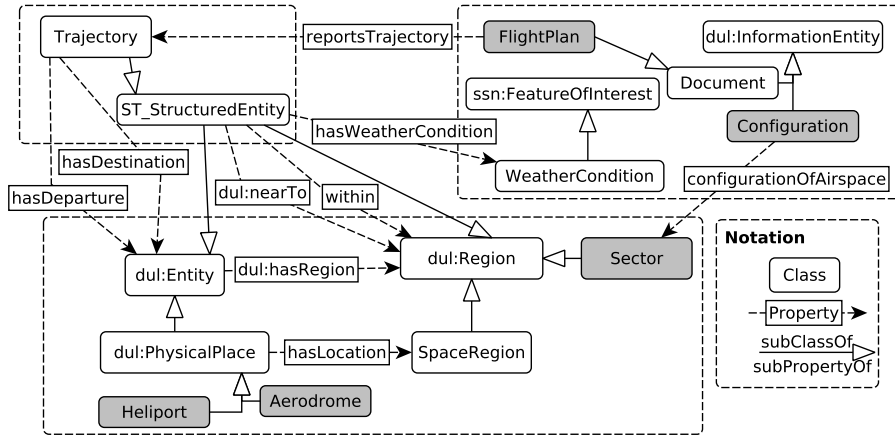


Fig. 6 The pattern of trajectories linked with contextual information. Domain specific concepts in gray

plied on a sector and remains active for a time interval, and indirectly affects all the trajectories crossing the sector.

- *Environmental event*, if it happens in the environment and affect the mobility of moving objects. Extreme weather conditions are such events.

It must be noted that associating events to trajectory parts satisfies the requirement to associate events at varying levels of trajectory aggregation. For instance, a low-level “turn” event associated with a node, is associated to any trajectory part (e.g. trajectory segment) that comprises that node. Also, each trajectory part may be associated to multiple events and thus, provide rich information about objects’ behaviour. For instance, a low-level “turn” event associated with a node may co-occur with a low-level “descend” event associated with a trajectory segment comprising that node. In addition to that, the trajectory segment can be further associated with events of type “CrossingSector”.

In addition to events, trajectory parts can be linked to contextual information. Such information may concern static aspects of the environment (e.g. airports, airspaces, etc), dynamic aspects (e.g. changing sector configurations, opening schemes, forecasts of weather conditions). The pattern for linking trajectory parts with contextual information is illustrated in Figure 6. Without loss of generality, subsequent paragraphs and Figure 6 provide examples of associating contextual entities of interest for the FM cases to trajectories.

Weather conditions are very important to trajectories in the FM domain: Each *TrajectoryPart* can be associated with entities of type *WeatherCondition*, which is defined as a subclass of *ssn:FeatureOfInterest*. This represents any entity whose properties are being estimated or calculated in the course of an observation.

Of particular interest to the FM domain 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 airport of a trajectory can be considered as contextual information, linked via the properties `hasDeparture` and `hasDestination`, respectively. The properties range to the class `dul:Physical-Place`, which can be further refined to domain specific classes such as `Airport`, or `Helipoint`.

Finally, a `FlightPlan` is also a domain specific entity of type `dul:InformationEntity` which is associated to an `IntendedTrajectory` via the property `reportsTrajectory`. Flight plans provide information on the intended trajectory and, in case a regulation has affected the trajectory, report the regulated intended trajectory.

5.3 Examples

As a concrete and simple example of a trajectory specified at multiple levels of abstraction, Fig. 7 shows the representation of a trajectory crossing an airspace compartment: The trajectory is represented both as a geometry projected in two dimensions, and as a temporal sequence of trajectory segments, which are indicated in different colour, depending on whether each segment occurs within the compartment or not. This structure results through a topological link discovery process where the trajectory geometry is used as a first indication of the potential fact that the trajectory crosses the air compartment (filtering step). This is further verified by exploiting the raw trajectory positional data and identifying the trajectory segments that spatially occur within the compartment. Additional information to trajectory segments is provided by associated events that are not shown in the figure, to keep it simple. Hence, beyond the representation of the trajectory as a sequence of trajectory segments, at a second level of abstraction, the trajectory is represented as a temporal sequence of semantic nodes, each one signifying an important event occurring across the trajectory. For instance, trajectory nodes H, L, M, and K are associated to entry/exit events, representing the relation of raw positions with the airspace compartment. Trajectory segments and nodes are further associated to positional raw data.

As a further more elaborated example, Fig. 8 shows an example of associating trajectories with information about events and contextual information. The two maps in the upper part show the trajectories of the flights performed between Paris Orly and Lisbon (left) and between London Heathrow and Madrid (right) during April 2016. Information about crossing sectors in which various types of regulations were applied has been attached to the points of the trajectories, denoting the regulation reason codes. In the map, the trajectories are represented by segmented lines; the segments are coloured according to the regulation reasons of their starting points. For the segments that were not in regulated sectors, the regulation reason code is empty. These segments are represented by thin dashed lines. The two images in the middle represent the segments of the trajectories between London and Madrid that were going through regulated sectors. On the left is a space-time cube with a geographic map lying in the base and the vertical dimension representing time. The time axis is oriented upward. The segments are positioned in the cube according to their geographic coordinates and time stamps. We can see that there were many flights between London and Madrid that crossed sectors with regulations in action (precisely, 196 out of

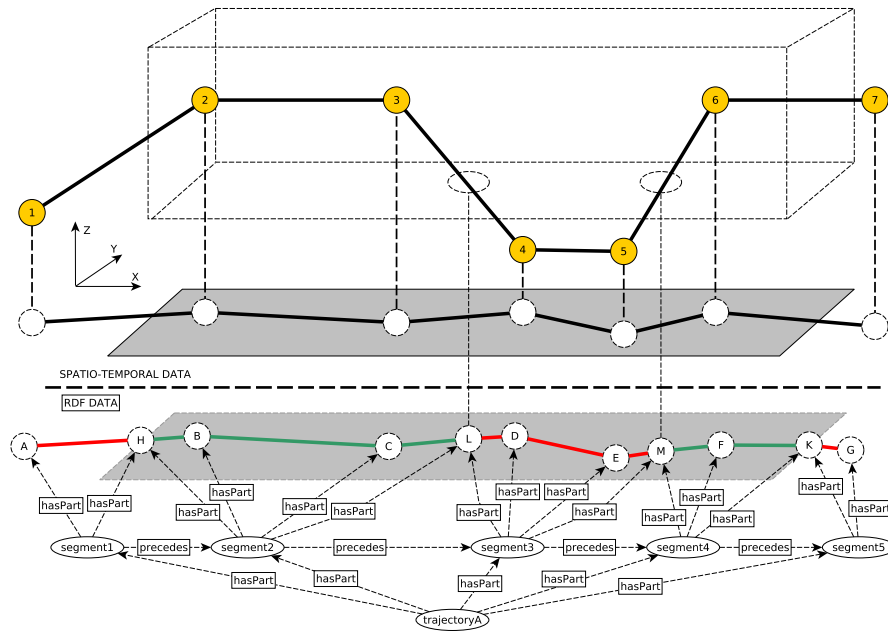


Fig. 7 A simple example of representing a trajectory crossing an airspace compartment.

228), and this mostly happened over the Bay of Biscay an the northwest of France. In the first three days, the regulation reason code for these parts of the flights was mostly “R” (ATC Routing), and the most frequent reason in the remaining days was “C” (ATC Capacity due to hotspots). On the right, the same segments are shown in a 3D view where the vertical dimension represents the flight altitude. The two images at the bottom represent the same information as above, together with the remaining segments of the trajectories.

6 Data Transformations for Visual Analytics

In this section we show how data transformations can be supported by the proposed ontology, according to the needs of visual analysis tasks in two major FM cases: Case FM01, aiming to the discovery of patterns of regulations, and case FM02, aiming to the analysis of hotspots occurrences. Cases specify scenarios with specific analysis objectives and data needs. Appropriate visualizations show data-driven exploratory analysis results towards identifying patterns of behaviour and supporting decision-making.

6.1 Data sets

To explore the capacities of the ontology to support visual analysis tasks we exploit the following data sets:

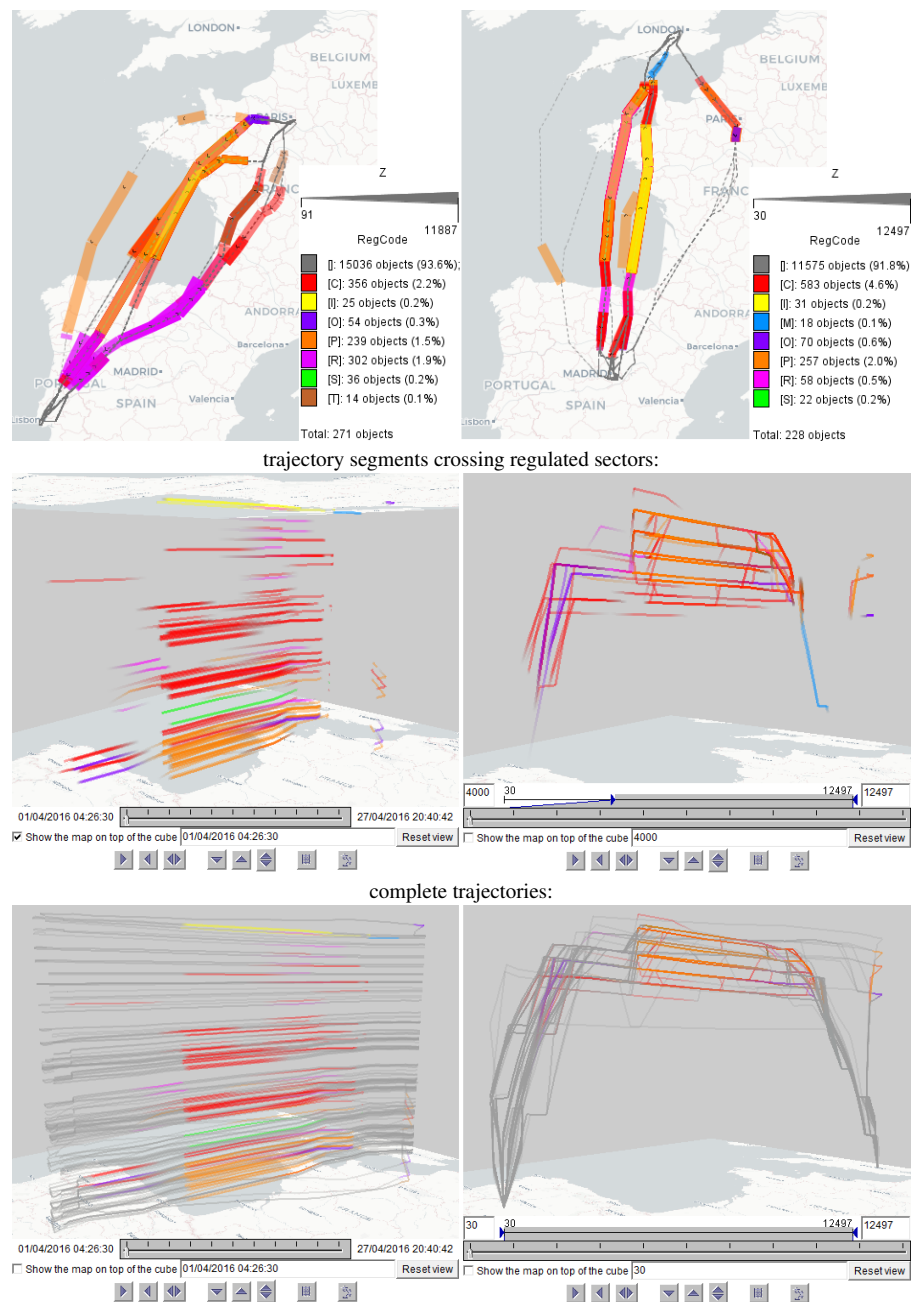


Fig. 8 Examples of trajectories enriched with information about crossing sectors in which regulations were applied.

- CFMU Regulations: This data set provides historical data of regulations applied by the control flow management unit (CFMU) on sectors in the European airspace, during April 2016.
- Sector Configuration: This data set describes the structure of sector configurations for specific periods of time within April 2016.
- Flight Plans: This data set contains the submitted flight plans prior to the take off for the flights operated during April 2016, to/from airports worldwide. However, only a few flights have destination/origin a non-European airport.
- Entry/Exit points: This data set is derived from the combination of sector configurations and flight plans. A spatiotemporal link discovery task [24] interpolates the altitude, latitude, longitude and time an aircraft enters/exits each air block (and sector). Having these entry/exit points we can specify trajectories as sequences of trajectory segments, each one topologically being “within” a crossed airspace compartment (shown in Fig. 7).
- NOAA grib binary files: This data set is a collection of 96 binary files reporting 3-hour weather forecasts, starting from April 1st, to April 24th, 2016.

These data sets are provided by heterogeneous (and often voluminous) data sources. We have introduced the RDF-Gen [27,26] method which converts data into triples with low latency, w.r.t. a given ontology (in our case, the datAcron ontology). The main idea of RDF-Gen is to use a SPARQL-like triple template for each data source, to convert raw data from the source to RDF triples. RDF-Gen templates allow the use of custom functions for cleaning and converting data values, generating URIs, and generating triples populating the ontology. This ontology population task by means of the appropriate RDF generator templates, as already pointed in the introductory part of Section 5, is an ontology validation task performed during ontology development. However, we do not delve into the details of this process here.

Among the data sets listed, the flight plans data set is the most voluminous. Specifically, this data set reports 958,288 flight plans (please recall that flight plan updates are possible, and flight plans can report at most three trajectory types), which are converted to 1,548,628,183 triples. The link discovery task for interpolating entry/exit positions for air blocks and constructing the corresponding trajectory segments for each trajectory, generates 283,906,720 additional triples, resulting to a total of 1,832,534,903 triples.

6.2 datAcron namespaces for functions

Data transformations cannot be fully supported by standard SPARQL 1.1 queries, since most of the queries involve spatial and temporal functions. We have extended standard SPARQL 1.1 with the following namespaces regarding functions:

- `SPARQL_functions.converters`: These include functions for converting given values to a specific format, e.g. the conversion of latitude, longitude, altitude and time values into a single string representation for each 4D point. An important function in this namespace is the `getWeatherAVG()`, which given the name of a weather variable, a geometry, an altitude range and a timestamp, retrieves the average

value for the weather variable within the airspace volume defined from the geometry and the altitude range.

- SPARQL.functions.distance: These are various distance functions between geometries. For cases where high performance is preferred over accuracy, the GeoElasticDistance() function (based on *Vincenty's formulae* [31]) can be used in the computations. For all the cases where accuracy is important, this namespace provides the function geodesicDistance() which is implemented on top of geographicLib². This function computes the distance between the centroids of given geometries in meters, and provides accuracy up to 10^{-9} m.
- SPARQL.functions.spatial: These are functions implementing all the OGC topological relations between pairs of geometries. Each function accepts WKT representations of geometries as arguments, and returns boolean true if the topological relation holds or false otherwise.
- SPARQL.functions.temporal: These are functions implementing all the temporal relations described in Allen's interval algebra [2]. Each function returns true if the corresponding temporal relation holds, or false otherwise. For example, the function *during_sf()* returns true if the temporal interval defined by the first two arguments (start and end time instants), is during, starts or finishes within the interval specified by the third and fourth arguments.

6.3 Validation setup

We have implemented a SPARQL 1.1 endpoint, on top of which we have implemented procedures for producing the required time series spanning within specific time periods. These procedures take as input a shifting time window duration and a time step for shifting the time window, instantiate query parameters (e.g. parameters concerning the time window) and pose the queries. Subsequently, parameters of parameterised queries are identified by “\$”.

For instance, in cases where we need to generate time series of counts of entities, the corresponding procedure uses a parameterized SPARQL query, where the time period of interest, the time window and the time step for shifting the window are parameters to be instantiated. The procedure builds a sequence of queries for subsequent time windows of a given duration. The starting points of subsequent windows differ by a number of minutes equal to the time step specified.

Specifically, given a time step Δt , a time window duration wd and a period $[TimeStart, TimeEnd]$, the i -th query of n iterations, where $n = \frac{(TimeEnd - wd - TimeStart)}{\Delta t}$, concerns the time interval $[TimeStart + i * \Delta t, TimeStart + wd + (i * \Delta t)]$.

6.4 Pre-processing steps and auxiliary structures

To increase the efficiency of query answering, we pre-compute intermediate results and store these in auxiliary structures. This method is by analogy to the spatial data bases which rely on specialised indices (i.e. spatial indices such as R-Tree) to improve

² publicly available online at <https://geographiclib.sourceforge.io/>

query answering performance. This is an additional way of exploiting data fetched via the SPARQL endpoint. Further more, the auxiliary structures (in addition to custom made functions) overcome limitations of SPARQL (i.e. such as iterative queries), and in the same time simplify the SPARQL queries used in the end (e.g. to increase the computational efficiency of query answering, no nested queries are used for the use cases) without affecting the validation of ontological specifications.

As already specified above, the link discovery process segments a trajectory to those parts that are within air blocks, by computing the spatio-temporal entry/exit points per trajectory and air block. Given that sectors comprise air blocks we can represent trajectories at different aggregation levels, depending on whether we are focusing on air blocks or sectors, according to the ontology specifications. The additional triples computed by the link discovery process are of the form $(?x \text{ :within } ?y.)$ representing trajectory segments $?x$ that occur spatially in air blocks $?y$.

To further increase efficiency we use an in-memory HashMap relating sectors with sets of airblocks. For the cases where a sector comprises another sector, we associate the former with the set of airblocks composing the latter. The HashMap is constructed using the query:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX sp: <java:SPARQL_functions.spatial.>
SELECT ?s ?airblock_wkt (str(?lower) as ?lowerLevel)
      (str(?upper) as ?upperLevel) WHERE {
  ?s dul:hasPart+ ?airblock .
  ?airblock :hasGeometry ?g ;
    :hasLowerLevel ?lower ;
    :hasUpperLevel ?upper .
  BIND(sp:getGeom(?g) as ?airblock_wkt) .
}
```

where $(?s \text{ dul:hasPart+ } ?airblock.)$ traverses the property path built from one or more occurrences of `dul:hasPart`, specifying the structure of sectors in terms of constituent air blocks and sectors. The above query reports the URIs of sectors, as well as the air block projection geometry in WKT and the lower/upper flight levels for each air block that a sector comprises.

Furthermore, the ontology is populated with triples stating regulations imposed on sectors (i.e. regulation events) for specific time intervals, with a potential cancellation time per regulation. The duration of a regulation is the time interval between the starting time and the earliest time instant between regulation cancellation (if it is specified) and ending time.

As we will see in subsequent sections, we need to associate sets of regulations and affected sectors to temporal intervals. The *temporal interval of any set of regulations* is the union of individual regulation's intervals $I_1 \cup \dots \cup I_n$. In some cases we need pairs of sectors (S_1, S_2) that are affected by temporally overlapping regulations R_1, \dots, R_n . We say that two regulations R_κ, R_λ are temporally overlapping if $I_\kappa \cap I_\lambda \neq \emptyset$.

Being interested in pairs of sectors affected by temporally overlapping regulations, as a pre-processing step, we retrieve the necessary data regarding sectors and regulations imposed on sectors from the ontology and pre-compute the pairs of sectors affected by temporally overlapping regulations, together with the respective tem-

poral intervals of all regulations per pair of sectors. This results to triples of the form (`?sectorX associatedByOverlappingRegulationWith ?sectorY.`). To further increase the query-answering performance in many cases, this relation among sectors is also stored in an in-memory `IntervalTree`, s.t. given a time interval Q_κ , we can effectively retrieve the pairs of sectors affected by regulations whose temporal interval overlaps with Q_κ .

The query behind this process is as follows:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ontology/dul.owl#>
PREFIX sp: <java:datAcron.unipi.gr.sparql_functions.>
CONSTRUCT { ?Sector0 :associatedByOverlappingRegulationWith ?Sector1 }
WHERE {
  ?Regulation0 a :FM_Regulation .
  ?Regulation0 dul:hasRegion ?Sector0 ;
    dul:hasTimeInterval ?TimeInterval0 .
  ?TimeInterval0 :TimeStart ?TimeStart0 ;
    :TimeEnd ?TimeEnd0 .
  ?Regulation1 a :FM_Regulation .
  ?Regulation1 dul:hasRegion ?Sector1 ;
    dul:hasTimeInterval ?TimeInterval1 .
  ?TimeInterval1 :TimeStart ?TimeStart1 ;
    :TimeEnd ?TimeEnd1 .
  FILTER(sp:overlaps(?TimeStart0,TimeEnd0,?TimeStart1,TimeEnd1 )
}
```

6.5 Visual analytics enhanced via data transformations

Subsequently, we show how ontology specifications support the full range of data transformations needed for visual analysis tasks in the context of Flow Management cases. Representative visualisations obtained during the visual analysis process are presented and discussed in detail.

6.5.1 FM01: Discovering patterns of regulations

This case, as described by its title, aims at providing an understanding of the occurrence of regulations. Recall that a regulation is a particular type of event that applies to airspace sectors and affect the trajectories crossing these sectors. Imposing regulations to trajectories (resulting to regulated trajectories), however, may necessitate the applications of regulations to other sectors of the airspace.

In this case we validate the ontology specifications in three exploratory cases towards (a) discovering daily or weekly patterns of regulations with a particular reason code imposed on individual sectors, (b) understanding how trajectories crossing pairs of sectors, thus providing links between sectors, affect regulations imposed on these sectors, and (c) understanding how weather conditions affect the occurrence of regulations and their impact on trajectories.

FM01 requires the following transformations that are presented in detail in subsequent paragraphs: (a) at first a spatial events to spatial time series transformation,

and then, (b) transforming trajectories to spatial time series (place-based and link-based), and transforming link-based spatial time series to place-based spatial time series. Finally, it requires (c) transforming trajectories into time series of spatial events, enriched with additional information (e.g. weather attributes).

(a) Discover regular temporal patterns of regulations:

Although this case does not involve trajectories, it is important as a first step towards the FM01 objective: We need to generate spatial time series of counts of regulations of a particular type (i.e. with a particular reason code; e.g. code "C" for regulations due to the occurrence of hotspots) per sector and time windows of a chosen duration. Among these time series, we aim to find time series with high periodicity with regard to the daily and weekly time cycles. The transformation demonstrated in this case is the aggregation of spatial events (in this case, regulations) into spatial time series (aggregation III in Figure 1).

To find out whether there are sectors where regulations occur regularly in time, we compute time series of regulation counts by sectors and days. The parameterized query is as follows:

```
PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX tmp: <java:SPARQL_functions.temporal.>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT (count(distinct ?regulation) as ?count)
WHERE {
  ?regulation a $regulation$ ;
  dul:hasRegion $sector$ ;
  dul:hasTimeInterval ?t .
  ?t :TimeStart ?s ; :TimeEnd ?e.
  FILTER(tmp:overlaps(?s,?e,
    $StartDate$^^xsd:DateTime,
    $EndDate$^^xsd:DateTime))
}
```

Here, parameters that must be instantiated are: `$regulation$` (regulation type), `$sector$` (sector name), `$StartDate$` (time period start), `$EndDate$` (time period end). The time series of regulation counts within the specified period are produced by executing the query for each consecutive time window.

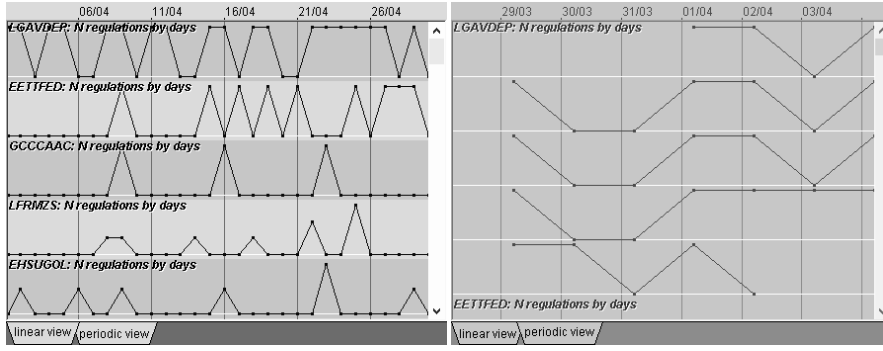


Fig. 9 Time series of regulation counts by sectors and days; left: linear view (per sector); right: periodic view (sector, per week)

In Fig. 9 the resulting time series are represented in a line plot. The image on the left presents a linear view of time. The horizontal axis represents the time span of the data, i.e., 1 month. The time series per sector are shown one below another. We can observe time series with frequent occurrences of regulations and time series with less frequent but quite regular occurrences. On the right, a periodic view of the time series is illustrated where each time series is divided into weekly pieces shown one below another. Only the time series for a single sector is visible in the current view port. This time series has high periodicity: regulations occurred on all Fridays and Mondays and all Saturdays, except the last one.

(b) Discover interdependencies between sectors:

As a first step towards discovering interdependencies between sectors, we need to find “patterns between regulated sectors”: Such patterns concern regulations in some sectors that often lead to regulations in other sectors. Therefore, given any pair of sectors S_1 and S_2 affected by temporal overlapping regulations, we need to find whether a regulation applied in S_1 (or S_2) affect the time where trajectories cross S_2 (resp. S_1), causing a new regulation in S_2 (resp. S_1). Therefore, as a first step, we need to count the number of flights’ trajectories in both directions.

In the first intermediate step we exploit the in-memory pre-computed Interval-Tree: for the time interval each trajectory lasts, let that be \mathcal{d} , we query the Interval-Tree and retrieve the pairs of sectors affected by temporally overlapping regulations whose temporal interval overlaps with \mathcal{d} . After verifying that each sector in a pair is crossed by the trajectory (i.e., a link between these sectors exists), we increase an integer counting the trajectories crossing the pair.

In more detail, the process first retrieves the geometries of regulated intended trajectories, as they have been reported by flight plans, using the query:

```

PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
SELECT DISTINCT ?ti ?wkt WHERE {
  ?fp :reportsTrajectory ?ti ;
      :reportsTrajectory [ a :RegulatedTrajectory ].
  ?ti a :IntendedTrajectory ;
      dul:hasGeometry/:hasWKT ?wkt .
}

```

Then, for each regulated intended trajectory, we retrieve the constituent spatio-temporal positions. We compute the time interval d where the trajectory occurs (i.e., the difference between the timestamps of the first and last spatio-temporal trajectory position), and we use d to query the intervalTree for pairs of regulated sectors whose regulations' temporal interval overlaps with the trajectory interval. We filter out pairs with at least one sector not crossed by the trajectory (i.e. no trajectory segment is within that sector), and for the remaining pairs we verify that each sector is crossed by the trajectory (i.e. by checking whether there is a trajectory segment within each sector of the pair), and the corresponding counter of the pair is increased by one.

The SPARQL queries supporting the above process are as follows:

1. Select the intended trajectories of regulated flight plans and the time interval in which they occur:

```

SELECT DISTINCT ?ti ?wkt ?timeStart ?timeEnd WHERE {
  ?fp :reportsTrajectory ?ti ;
      :reportsTrajectory [ a :RegulatedTrajectory ] .
  ?ti a :IntendedTrajectory ;
      dul:hasStart ?s ;
      dul:hasEnd ?e .
  ?ti dul:hasGeometry/:hasWKT ?wkt .
  ?s :hasTemporalFeature/:TimeStart ?timeStart .
  ?e :hasTemporalFeature/:TimeStart ?timeEnd .
}

```

2. Select the spatio-temporal positions of the trajectories:

```

SELECT DISTINCT ?ti ?position ?time WHERE {
  ?fp :reportsTrajectory ?ti ;
      :reportsTrajectory
        [ a :RegulatedTrajectory ] .
  ?ti a :IntendedTrajectory .
  ?ti dul:hasPart+ ?p .
  ?p a :RawPosition .
  ?p :hasGeometry/:hasWKT ?position .
  ?p :hasTemporalFeature/:TimeStart ?time .
}

```

The complete query would be:

```

SELECT DISTINCT ?ti ?timeStart ?timeEnd WHERE {
  ?fp :reportsTrajectory ?ti ;
      :reportsTrajectory
      [ a :RegulatedTrajectory ] .
  ?ti a :IntendedTrajectory .
  ?ti dul:hasStart ?Start ;
      dul:hasEnd ?End ;
      dul:hasPart+ ?Part1 ;
      dul:hasPart+ ?Part2 .
  ?Part1 :within ?Airblock1 .
  ?Part2 :within ?Airblock2 .
  ?Sector1 dul:hasPart+ ?Airblock1 .
  ?Sector2 dul:hasPart+ ?Airblock2 .
  ?Sector1 :associatedByOverlappingRegulationWith ?Sector2 .
  ?Start :hasTemporalFeature/:TimeStart ?timeStart .
  ?End :hasTemporalFeature/:TimeStart ?timeEnd .
  ?Regulation1 dul:hasRegion ?Sector1 ;
      dul:hasTimeInterval ?t1 .
  ?Regulation2 dul:hasRegion ?Sector2 ;
      dul:hasTimeInterval ?t2 .
  ?t1 :TimeStart ?start1 ;
      :TimeEnd ?end1 .
  ?t2 :TimeStart ?start2 ;
      :TimeEnd ?end2 .
  FILTER(
    tmp:overlaps(?timeStart, ?timeEnd,
      tmp:minDate(?start1,?start2),tmp:maxDate(?end1,?end2)) &&
    (?Sector1!=?Sector2))
}

```

The results are shown visually in Fig. 10. During April 2016, a total of 8,254 links emerged between regulated sectors. The largest number of flights that moved between two sectors was 2,716. The geographic map (Fig. 10, top left) shows that there were quite many (precisely, 140) local links between sectors that differed in the vertical positions but had the same or overlapping horizontal positions. These links are represented on the maps by circles drawn around the sector centroid positions. The remaining links are represented by curved lines connecting the sector centroids. The line widths and the opacity levels are proportional to the flight counts. Specifically, Fig. 10 shows the following results. Top left: The links between regulated sectors that occurred during the month are represented by curved lines with the widths proportional to the counts of the flights that moved from one regulated sector to another. The circles represent the links connecting sectors that differ in the vertical positions but have the same or overlapping horizontal positions. Top right: Histograms of the link duration in minutes (the maximum is 960 minutes) and the dates when the links occurred (the largest number was on April 2). The bars are divided into segments coloured according to the counts of the trajectories involved in the links; the colour legend is shown above the histograms. Bottom left: the links are represented by points in a radial coordinate system where the angle and the distance from the centre represent the time of the day and the link duration, respectively [4, 11]. The circular grid lines are drawn with the interval of 60 minutes. Bottom right: A density map shows the distribution of the points in the radial coordinate system. The densest areas correspond to the link start times around 5-6 o'clock in the morning and

duration from 1.5 to 4 hours and to the start times around 8-9 o'clock and duration 2-3 hours.

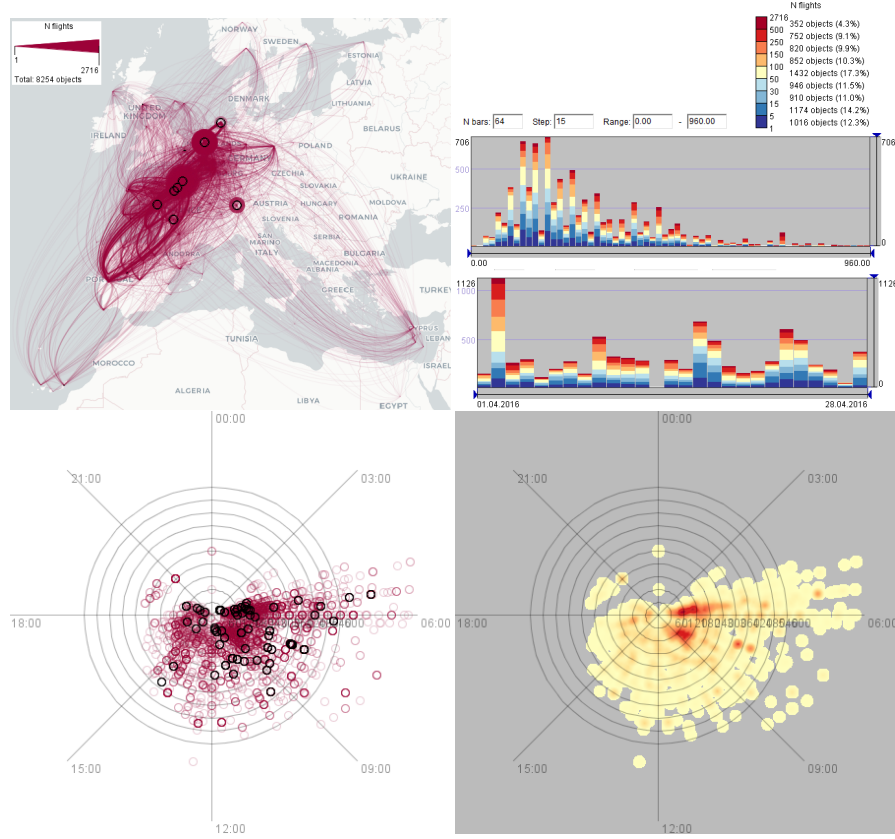


Fig. 10 Interdependencies between sectors considering trajectories providing links between sectors.

Towards getting re-occurring links between sectors affected by temporally overlapping regulations, according to the flows of flights from one sector to the other, we compute time series of links existence: for each pair of sectors (S_1, S_2) with temporally overlapping regulations and for which links exist, we need to compute time series with the number of trajectories crossing S_1 and S_2 for each time window. Time series with multiple peaks would signify interrelationships between sectors that we want to discover. Here trajectories are aggregated into flows between places (sectors) resulting in linked-based spatial-time series (aggregation II in Figure 1).

The temporal window, as specified above, shifts with a pre-specified time step Δt :


```

PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX tmp: <java://SPARQL_functions.temporal.>
SELECT (count (DISTINCT ?tr) as ?cntr) WHERE {
  ?tr a :IntendedTrajectory ;
    dul:hasPart+ ?segment1 ;
    dul:hasPart+ ?segment2 .
  ?segment1 :within ?airbl1 ;
    :hasTemporalFeature ?t1 .
  ?segment2 :within ?airbl2 ;
    :hasTemporalFeature ?t2 .
  ?t1 :TimeStart ?time1 .
  ?t2 :TimeStart ?time2 .
  $Sector1$ dul:hasPart+ ?airbl1 .
  $Sector2$ dul:hasPart+ ?airbl2 .
  $Sector1$ :associatedByOverlappingRegulationWith $Sector2$ .
  FILTER(tmp:during_sf(?time1,?time1,$t+k*Δt,$t+wd+k*Δt) &&
    tmp:during_sf(?time2,?time2,$t+k*Δt,$t+wd+k*Δt) )
}

```

This query concerns a particular pair of sectors associated with temporally overlapping regulations, instantiating the query parameters $\$Sector1\$, \$Sector2\$$. Also, $[t+k*\Delta t, t+wd+k*\Delta t]$ is the k -th sliding time window of duration wd within the specified time period.

Fig. 11 demonstrates visual exploration of link re-occurrences based on the query results. For each pair of linked sectors, all links have been aggregated in a single link, for which a time series of link occurrences was computed. Here, link re-occurrence is explored at the time scale of days. The aggregated links are represented on a map by curves with the line widths proportional to the number of days in which the links re-occurred, ranging from 1 to 18. On the right, only the links that re-occurred in 9 or more days are shown; the links with the maximal re-occurrence are highlighted in black. There were 4,664 sector pairs in total, of which 3,156 (67.7%) re-occurred only once and further 745 pairs (16%) re-occurred twice. The maximal number of different days in which links re-occurred was 18, which was attained by 2 links; and 57 links (1.2%) re-occurred in 9 or more days.

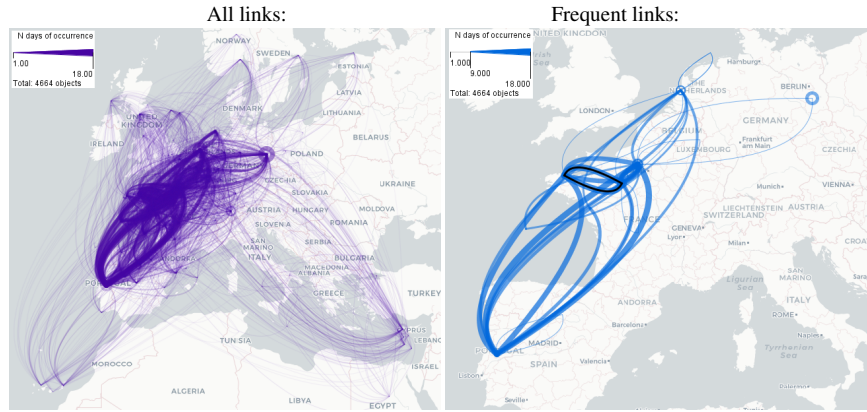


Fig. 11 Links re-occurrence at the time scale of days.

Finally, we can aggregate links by sector pairs and time windows into spatial time series. This is a transformation from spatial time series (place based) to spatial time series (link based) not shown as a direct transformation in Figure 1.

To obtain a spatial time series of links, the following query aggregates for each pair of sectors associated by temporally overlapping regulations, the trajectories intended - according to the flight plans specified - to cross both of them. The number of such trajectories are computed per time window of a given duration.

```

PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX tmp: <java://SPARQL_functions.temporal.>
SELECT (COUNT(DISTINCT ?t) as ?count) WHERE {
  ?t dul:hasPart ?segment1 ;
    dul:hasPart ?segment2 .
  ?segment1 :within ?airblock1 .
  ?sector1 dul:hasPart+ ?airblock1 .
  ?segment2 :within ?airblock2 .
  ?sector2 dul:hasPart+ ?airblock2 .
  ?r dul:hasRegion ?sector1 .
  ?r dul:hasTimeInterval ?time .
  ?time :TimeStart ?ts ; :TimeEnd ?te .
  ?sector1 :associatedByOverlappingRegulationWith ?sector2 .
  FILTER(?sector1!=?sector2) .
  FILTER(tmp:during_sf(?ts,?te,$t+k*Δt$,$t+wd+k*Δt$))
}

```

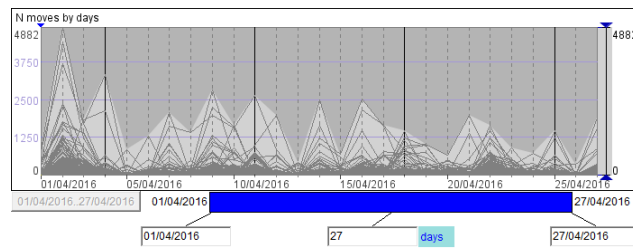


Fig. 12 A time graph shows the daily aggregates of the counts of the flights that took place during the existence of the links between regulated sectors. The aggregates have been initially obtained for the pairs of linked sectors and then further aggregated by areas, as in Fig. ??.

In Figs. 12 and 13, we aggregated the number of trajectories by the links and daily time intervals. The time graph in Fig. 12 shows the time series of the number of trajectories between regulated sectors. The black vertical lines mark Mondays. We see that the Saturday of April 2 was exceptional regarding both the values attained and the number of area pairs with high values. However, the following Saturdays of April 9 and 16 also had quite many area pairs with high values.

To see whether the spatial patterns were similar on different Saturdays and, more generally, whether there was periodic repetition of similar spatial patterns on the same days of the week in different weeks, we have created an animated map display with 4 map panels labelled t , $t + 7$, $t + 14$, and $t + 21$, where t is the currently chosen day in the animation. This visualization enables convenient comparison of the spatial situations on Saturdays, Sundays, Mondays, and so on.

The upper and lower images in Fig. 13 show the situations on Saturdays and Mondays, respectively. It can be noted that, in general, the spatial patterns on the same days of different weeks were not very similar. April 2 and 9 (Saturdays) had similar diagonal patterns with multiple links oriented along the line between the Canary Islands and the northeast of France, whereas the Saturdays of the following two weeks had similar link patterns between the British Islands on the one side and France and Spain on the other side. On April 11, 18, and 25 (Mondays), there were similar star-like patterns, with many links oriented in different directions having starts or ends in the same area. On April 11 and 18, the "stars" appeared around the same area over the Netherlands, whereas on April 25 the "star" moved to Belgium, southwest of the previous location.

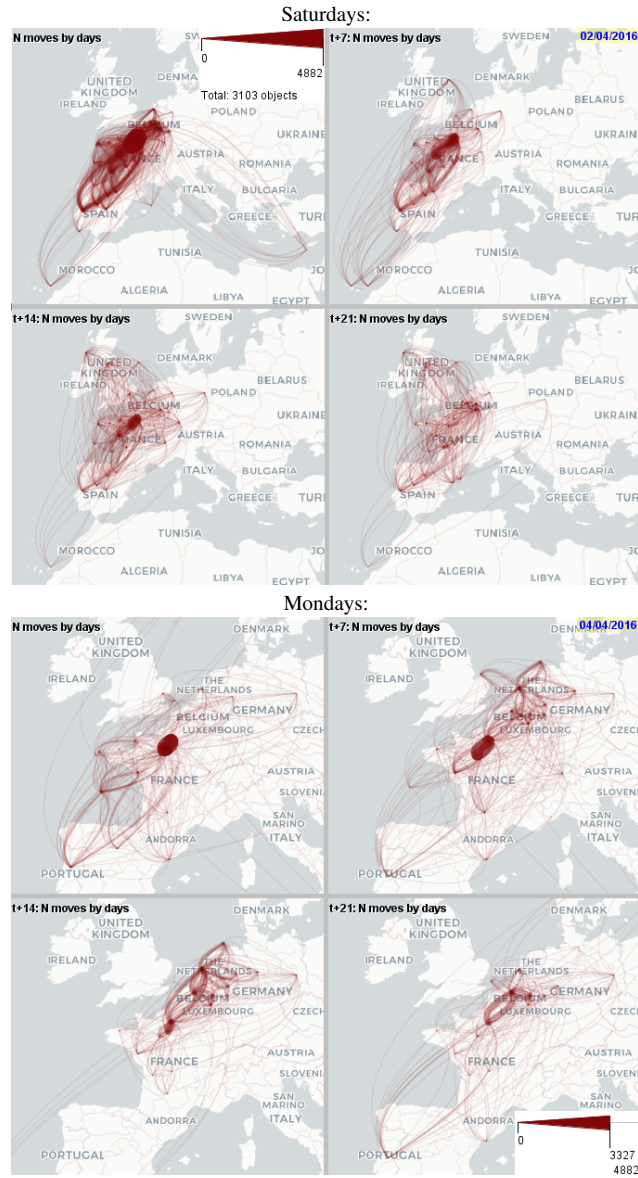


Fig. 13 Comparison of the spatial patterns of the regulations-related flights on Saturdays, starting from April 2 (top), and Mondays, starting from April 4 (bottom).

(c) Discover dependencies between weather conditions and regulations:

In this case we need to find for each regulation of type :ATC.WeatherRegulation (i.e., with reason code “W”) that affects a Sector S , and for each trajectory that intends to cross that sector, the predicted weather conditions at the time the trajectory is going to cross the sector. The objective is to reveal the rationale for the occurrence

of events and understand how trajectories are being affected. As an example, we shall explore the relationships between the flight regulations issued due to windy weather and the wind parameters available in the weather data. The data set contains data about 162 regulations with the reason code “W”. The descriptions of 37 such regulations include the keyword “wind”. We selected two days in which the regulations due to wind were applied not within airports but in sectors crossed on the fly. These days were April 16 and 18, 2016. We extracted the corresponding intended trajectories of the flights and enriched them with wind parameters extracted from the weather data. There are 14 wind attributes with non-null values describing the u- and v-components of the wind, i.e., the west-east and south-north components.

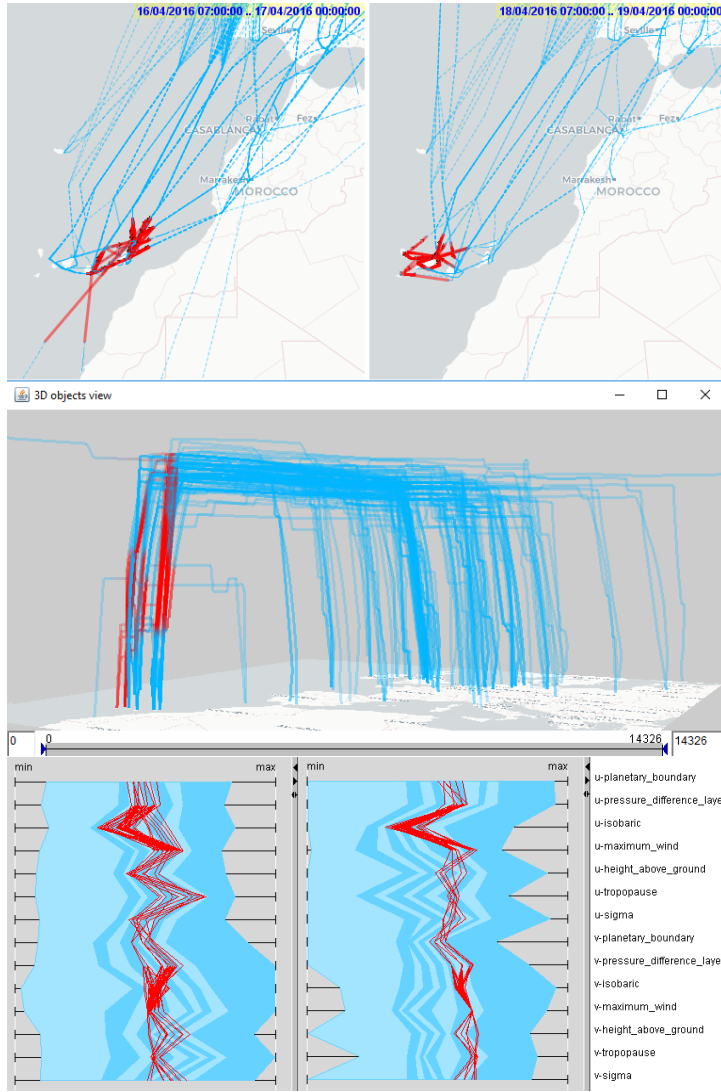


Fig. 14 Exploration of the relationships between the regulations due to winds and the wind parameters extracted from weather data.

Specifically, we first retrieve the intended trajectories that have been regulated, and compute the temporal interval. This query has already specified above.

For each such trajectory and the corresponding temporal interval, we retrieve the sectors affected by regulations identified by reason code "W" (i.e. bad weather conditions), and which temporally overlap with the interval of the trajectory. Each trajectory that is crossing an airtask of a regulated sector by reason code "W", is added in the result set, enriched with values of weather variables. For example, the following query will enrich each position of the trajectory :tr_20160416_125062_m1 with

values of the “u-component_of_wind_isobaric” and “v-component_of_wind_isobaric” weather variables:

```
SELECT ?p ?wkt (str(?alt) as ?altitude) ?unixtime
  (conv:getWeatherAVG(?wkt,?altitude,?unixtime,
    "u-component_of_wind_isobaric") as ?u_comp_wind)
  (conv:getWeatherAVG(?wkt,?altitude,?unixtime,
    "v-component_of_wind_isobaric") as ?v_comp_wind)
WHERE {
  :tr_20160416_125062_m1 dul:hasPart+ ?p .
  ?p a :RawPosition;
    dul:hasTemporalFeature ?t ;
    :hasGeometry ?g .
  ?t :TimeStart ?time .
  ?g :hasWKT ?wkt ;
    :hasAltitude ?alt .
  BIND(tmp:dateTime2unixTime(?time) as ?unixtime) .
} ORDER BY ?unixtime
```

Generalizing, a query for associating trajectories regulated due to weather with weather attributes is as follows:

```
SELECT DISTINCT ?ti
  ?wkt (str(?alt) as ?altitude) (tmp:dateTime2unixTime(?time) as ?
    unixtime)
  (conv:getWeatherAVG(?wkt,?altitude,?unixtime,
    "u-component_of_wind_isobaric") as ?u_comp_wind)
  (conv:getWeatherAVG(?wkt,?altitude,?unixtime,
    "v-component_of_wind_isobaric") as ?v_comp_wind)
WHERE {
  ?fp :reportsTrajectory ?ti ;
    :reportsTrajectory [ a :RegulatedTrajectory] .
  ?ti a :IntendedTrajectory ;
    dul:hasStart ?s ; dul:hasEnd ?e ;
    dul:hasPart+ ?segment ; dul:hasPart+ ?p .
  ?s :hasTemporalFeature/:TimeStart ?timeStart .
  ?e :hasTemporalFeature/:TimeStart ?timeEnd .
  ?p a :RawPosition ;
    :hasGeometry ?pg ;
    :hasTemporalFeature/:TimeStart ?pt .
  ?g :hasWKT ?wkt ;
    :hasAltitude ?alt .
  ?segment a :TrajectorySegment ;
    :within ?airblock .
  ?sector dul:hasPart ?airblock .
  ?regulation dul:hasRegion ?sector ;
    dul:hasTimeInterval ?regulationTimeInterval .
  ?regulationTimeInterval :TimeStart ?start ;
    :TimeEnd ?end .
  ?regulationTimeInterval a :ATC_WeatherRegulation .
  FILTER(tmp:overlaps(?timeStart, ?timeEnd,
    tmp:minDate(?start1,?start2),tmp:maxDate(?end1,?end2)) )
}
```

It must be noted that the execution of this query takes too much time for a typical SPARQL endpoint. On the other hand, breaking down the query to a set of smaller queries, replacing ?ti with each trajectory that satisfies the temporal criteria described above, we can achieve a considerable improvement in the overall performance. In addition to that, the auxiliary structures maintain in memory information

that is frequently requested, thus it can be accessed much faster this way, rather than through the SPARQL endpoint.

In Fig. 14, the upper two images are map fragments showing where the regulations due to the wind conditions were applied on April 16 (left) and 18 (right). The intended flight trajectories are represented by lines. The segments of the trajectories that crossed the regulated sectors are marked in red and the remaining segments are represented by thin light blue lines. We can see that the regulations happened in the region of the Canary Islands, on the east of it on April 16 and on the west on April 18. The 3D perspective view of the trajectories (Fig. 14, middle) shows us that the flights were supposed to cross the affected sectors in their climb or descent phases.

To investigate which of the wind attributes might lead to the decision to issue the regulations, we explore the data using the parallel coordinates plots, as in Fig. 14, bottom. The parallel horizontal axes correspond to the 14 wind attributes, the upper 7 axes to the u-component attributes and the lower 7 axes to the v-component. Each axis has its individual scale from the minimal to the maximal value of the respective attribute. The plots on the left and on the right correspond to April 16 and 18, respectively. The background painting represents the distributions of the attribute values in the trajectory points that were beyond the regulated sectors. The stripes correspond to the deciles of the distributions, i.e., the interval covered by each stripe on each axis includes 10% of values of the respective attributes. Lighter shades correspond to the odd deciles (i.e., the first, third, fifth, seventh, and ninth) and darker shades to the remaining even deciles. We have applied a combination of filters to select only the points located in the vicinity (specifically, within the radius of 350 km) of the flight origins and destinations. The red lines on top of the painting represent the value combinations in the points that were in the regulated sectors.

When we compare the red lines with the background distribution, we see that none of the wind parameters reach especially high or especially low values. The values of the attributes u-maximum-wind and u-tropopause were quite high (corresponding to the upper deciles of the background distributions), whereas the values of the attributes describing the v-component of the wind were relatively low. This indicates that the wind blowing from the west was sometimes quite strong and could be problematic to the ascending and descending aircraft. Since such wind parameters were not exceptional, judging from the background distribution, it can be concluded that they are problematic only in the region of the Canary Islands, possibly, due to the specifics of the local airports. This means that the weather parameters alone may be insufficient for predicting the necessity of applying flight regulations in this or that region.

6.6 FM02: Hotspots occurrence

In this case, we want to support understanding of the rationale for the choice of sector configurations, based on the expected evolution of demand. In doing so, we first retrieve all sectors (active or not) crossed by any trajectory and then, we provide a time series of the number of trajectories intended to cross any sector, providing the evolution of demand per sector.

To compute the evolution of demand we aggregate the trajectories specified by flight plans into spatial time series by sectors and time windows. Two time-dependent attributes may be computed for any sector: *entry count* (how many flights enter the sector during each time interval) or *occupancy* (how many flights are present in that sector during each time interval). These may be counted in overlapping time windows, depending on the step used for shifting the time window. As usually, to produce spatial time series we use a time window of specific duration and a time step, which specifies the time difference between the starting points of two consecutive windows.

These *FM02* cases require in the first place transforming trajectories (as specified by flight plans) into time series of spatial events, and then, transforming trajectories into spatial time series of demands by aggregating them by (active) sectors (aggregation II in Figure 1).

(a) This case first requires for a given intended trajectory specified by a flight plan, to retrieve the series of sectors *S* (active and inactive) crossed by that trajectory, and the trajectory segments crossing each sector in *S*. 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 dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
SELECT ?sector (min(?start) as ?timeEnter) (max(?end) as ?timeExit)
WHERE {
  :flight_plan_AA51147955 :reportsTrajectory ?t .
  ?t a :IntendedTrajectory ;
    dul:hasPart ?segment .
  ?segment a :Segment ;
    :within ?airblock ;
    :hasTemporalFeature ?time .
  ?time :TimeStart ?start ;
    :TimeEnd ?end .
  ?sector dul:hasPart+ ?airblock .
} Group By ?sector
Order By ?timeEnter

```

A more restricted version of the above query, concerns the active sectors during the time period of the flight defined by the first and last node of the trajectory reported by the given flight plan, according to the active sector configurations. The query is as follows:

```

PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
SELECT ?sector (min(?start) as ?timeEnter) (max(?end) as ?timeExit)
WHERE {
  :flight_plan_AA51147955 :reportsTrajectory ?t .
  ?t a :IntendedTrajectory ;
    dul:hasPart ?segment .
  ?segment a :Segment ;
    :within ?airblock ;
    :hasTemporalFeature ?time .
  ?time :TimeStart ?start ;
    :TimeEnd ?end .
  ?sector dul:hasPart+ ?airblock .
  ?f a :FM_Configuration ; :configurationOfAirspace ?airspace ;
    :hasTemporalFeature ?time .
  ?airspace dul:hasPart ?sector .
  ?time :TimeStart ?ts ; :TimeEnd ?te .
  FILTER(tmp:overlap(?start,?end,?ts,?te))
} Group By ?sector
Order By ?timeEnter

```

(b) Finally, we use the following query to compute per sector and time window, the demand for that sector, i.e., the number of trajectories intended to cross that sector during the corresponding period specified by the temporal window. As usually, the time window shifts with a step of Δt minutes.

```

PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX tmp: <java://SPARQL_functions.temporal.>
SELECT (count(DISTINCT ?tr) as ?demand) WHERE
{
  ?flightPlan :reportsTrajectory ?tr .
  ?tr a :IntendedTrajectory ;
    dul:hasPart ?segment .
  ?segment :within ?airblock ;
    :hasTemporalFeature ?time .
  $$Sector$ dul:hasPart+ ?airblock .
  :entersRegion :occurs ?segment .
  ?time :TimeStart ?s .
  FILTER(tmp:during_sf(?s,?s,$t+k*\Delta t,$t+wd+k*\Delta t))
}

```

As done above, we can restrict this query to the number of trajectories crossing active sectors (i.e., considering the periods in which each sector is active).

```

PREFIX : <http://www.datacron-project.eu/datAcron#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX tmp: <java://SPARQL_functions.temporal.>
SELECT (count(DISTINCT ?t) as ?demand) WHERE
{
  ?f a :FM_Configuration ;
    :configurationOfAirspace ?airspace ;
    :hasCapacity ?capacity ;
    :hasTemporalFeature ?time .
  ?airspace dul:hasPart $Sector$ .
  ?sector dul:hasPart+ ?airblock .
  ?time :TimeStart ?ts ; :TimeEnd ?te.
  ?t a :IntendedTrajectory ;
    dul:hasPart ?segment .
  ?segment :within ?airblock ;
    :hasTemporalFeature ?tn .
  ?tn :TimeStart ?s ;
    :TimeEnd ?e .
  FILTER(myfn:overlaps(?s,?e,?ts,?te) &&
    tmp:during_sf(?s,?s,$t+k*Δt$, $t+wd+k*Δt$))
}

```

7 Concluding remarks

FM Use Case	Transformation	Trajectory Level of Analysis	Interesting features
FM01(a)	Aggregation III (spatial events to spatial time series)	No Trajectories	Transformation of regulations to time series of aggregated regulation events
FM01(b)	Aggregation II (Trajectories to spatial time series (place and link- based))	Trajectories as geometries, as sequences of positions and as sequences of segments crossing air blocks	Trajectories are exploited at different levels of analysis to get time series of link events between sectors
FM01(c)	Transformations IV and VI (Trajectories to spatial events, enriched with weather data)	Trajectories as sequences of segments crossing air blocks and thus, corresponding sectors.	Here weather attributes related to regulations affecting trajectories are transformed into spatial time series along trajectories and through projections to spatial situations.
FM02(a)	Transformations IV (Trajectories to time series of spatial events)	Trajectories as sequences of segments crossing air blocks and thus, corresponding sectors.	
FM02(b)	Aggregation II (Trajectories to spatial time series (place-based))	Trajectories as sequences of segments crossing air blocks and thus, corresponding sectors and the aggregation of these segments per sector and time window.	

Fig. 15 A synoptic view of FM cases and data transformations applied, together with the required levels of trajectory analysis.

This work contributes a generic ontology for the representation of semantic trajectories at varying levels of spatio-temporal analysis to support analysis tasks, supporting our understanding of movement phenomena and of significant events that affect entities' mobility: Trajectories can be seen as temporal sequences of moving objects'

positional data, aggregations of positional data signifying meaningful events, as temporal sequences of trajectories segments, or as geometries.

Delving into these specifications, we show how visual analysis tasks can be supported by the different levels of trajectory specifications, via appropriate data transformations at query time. This happens via the use of SPARQL queries executed in the populated ontology for the purposes of concrete and important real-world cases. Indeed, generic data transformations, shown in the complex and highly exploratory Air Traffic Management domain, adapt available data to the analysis goals, or to specific requirements of the methods that the analyst wants to apply.

Figure 15 shows all the cases considered (1st column), and for each case it specifies the data transformations applied (2nd column), the levels of trajectory analysis required (3rd column) and indications of interesting features as far as the representation is concerned.

As future work we aim to re-use this ontology in different domains where trajectories play important role in analysis of behaviour: Either for traffic analysis in cities, or for human behaviour analysis in crowded places (e.g. buildings, touristic places, festivals etc.), under normal or emergency circumstances, or even in domains where trajectories do not involve spatio-temporal entities, but space-temporal entities, where space is any n-dimensional space where information entities (e.g. images) do exist.

Acknowledgements 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. Al-Dohuki, S., Wu, Y., Kamw, F., Yang, J., Li, X., Zhao, Y., Ye, X., Chen, W., Ma, C., Wang, F.: Semantictraj: A new approach to interacting with massive taxi trajectories. *IEEE Transactions on Visualization and Computer Graphics* **23**(1), 11–20 (2017). DOI 10.1109/TVCG.2016.2598416. URL doi.ieeecomputersociety.org/10.1109/TVCG.2016.2598416
2. Allen, J.F.: Maintaining knowledge about temporal intervals. *Commun. ACM* **26**(11), 832–843 (1983). DOI 10.1145/182.358434. URL <http://doi.acm.org/10.1145/182.358434>
3. Alvares, L.O., Bogorny, V., Kuijpers, B., de Macêdo, J.A.F., Moelans, B., Vaisman, A.A.: A model for enriching trajectories with semantic geographical information. In: GIS, p. 22 (2007)
4. Andrienko, G., Andrienko, N.: Creating maps of artificial spaces to explore trajectories. In: *Proceedings of the 2018 International Conference on Advanced Visual Interfaces, AVI '18*, pp. 53:1–53:3. ACM, New York, NY, USA (2018). DOI 10.1145/3206505.3206557. URL <http://doi.acm.org/10.1145/3206505.3206557>
5. Andrienko, G., Andrienko, N., Bak, P., Keim, D., Wrobel, S.: *Visual Analytics of Movement*. Springer Publishing Company, Incorporated (2013)
6. Andrienko, G., Andrienko, N., Bak, P., Keim, D., Wrobel, S.: *Visual Analytics of Movement*. Berlin, Springer (2013a)
7. Andrienko, G., Andrienko, N., Chen, W., Maciejewski, R., Zhao, Y.: Visual analytics of mobility and transportation: State of the art and further research directions. *IEEE Transactions on Intelligent Transportation Systems* **18**(8), 2232–2249 (2017). DOI 10.1109/TITS.2017.2683539
8. Andrienko, G., Andrienko, N., Demsar, U., Dransch, D., Dykes, J., Fabrikant, S.I., Jern, M., Kraak, M.J., Schumann, H., Tominski, C.: Space, time and visual analytics. *International Journal of Geographical Information Science* **24**(10), 1577–1600 (2010). DOI 10.1080/13658816.2010.508043. URL <https://doi.org/10.1080/13658816.2010.508043>
9. Andrienko, G., Andrienko, N., Jankowski, P., Keim, D., Kraak, M., MacEachren, A., Wrobel, S.: Geovisual analytics for spatial decision support: Setting the research agenda. *International Journal of*

- Geographical Information Science **21**(8), 839–857 (2007). DOI 10.1080/13658810701349011. URL <https://doi.org/10.1080/13658810701349011>
10. Andrienko, N., Andrienko, G.: Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization* **12**(1), 3–24 (2013). DOI 10.1177/1473871612457601. URL <https://doi.org/10.1177/1473871612457601>
 11. Andrienko, N., Andrienko, G., Garcia, J.M.C., Scarlatti, D.: Analysis of flight variability: a systematic approach. *IEEE Transactions on Visualization and Computer Graphics* **25**(1), 54–64 (2019). DOI 10.1109/TVCG.2018.2864811
 12. Baglioni, M., de Macêdo, J.A.F., Renso, C., Trasarti, R., Wachowicz, M.: Towards semantic interpretation of movement behavior. In: *Advances in GIScience*, pp. 271–288. Springer (2009)
 13. Bogorny, V., Renso, C., de Aquino, A.R., de Lucca Siqueira, F., Alvares, L.O.: Constant - A conceptual data model for semantic trajectories of moving objects. *Trans. GIS* **18**(1), 66–88 (2014)
 14. Chu, D., Sheets, D.A., Zhao, Y., Wu, Y., Yang, J., Zheng, M., Chen, G.: Visualizing hidden themes of taxi movement with semantic transformation. In: *Proceedings of the 2014 IEEE Pacific Visualization Symposium, PACIFICVIS '14*, pp. 137–144. IEEE Computer Society, Washington, DC, USA (2014). DOI 10.1109/PacificVis.2014.50. URL <http://dx.doi.org/10.1109/PacificVis.2014.50>
 15. Fileto, R., May, C., Renso, C., Pelekis, N., Klein, D., Theodoridis, Y.: The baquara² knowledge-based framework for semantic enrichment and analysis of movement data. *Data Knowl. Eng.* **98**, 104–122 (2015)
 16. Hamad, K., Quiroga, C.: Geovisualization of archived ITS data-case studies. *IEEE Trans. Intelligent Transportation Systems* **17**(1), 104–112 (2016). DOI 10.1109/TITS.2015.2460995. URL <https://doi.org/10.1109/TITS.2015.2460995>
 17. Hu, Y., Janowicz, K., Carral, D., Scheider, S., Kuhn, W., Berg-Cross, G., Hitzler, P., Dean, M., Kolas, D.: A geo-ontology design pattern for semantic trajectories. In: T. Tenbrink, J. Stell, A. Galton, Z. Wood (eds.) *Spatial Information Theory*, pp. 438–456. Springer International Publishing, Cham (2013)
 18. Kotis, K., Vouros, G.A.: Human-centered ontology engineering: The HCOME methodology. *Knowl. Inf. Syst.* **10**(1), 109–131 (2006)
 19. Kraak, M., Ormeling, F.: *Cartography: Visualization of spatial data*, 3 edn. Guilford Publications, United States (2010)
 20. Nogueira, T.P., Martin, H.: Querying semantic trajectory episodes. In: *Proc. of MobiGIS*, pp. 23–30 (2015)
 21. Paiva Nogueira, T., Bezerra Braga, R., Martin, H.: An Ontology-based Approach to Represent Trajectory Characteristics. In: *Fifth International Conference on Computing for Geospatial Research and Application*. Washington, DC, United States (2014). URL <https://hal.archives-ouvertes.fr/hal-01058269>
 22. Parent, C., Spaccapietra, S., Renso, C., Andrienko, G.L., Andrienko, N.V., Bogorny, V., Damiani, M.L., Gkoulalas-Divanis, A., de Macêdo, J.A.F., Pelekis, N., Theodoridis, Y., Yan, Z.: Semantic trajectories modeling and analysis. *ACM Comput. Surv.* **45**(4), 42 (2013)
 23. Peuquet, D.J.: 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), 441–461 (1994)
 24. Santipantakis, G., Doukeridis, C., Vouros, G.A., Vlachou, A.: Masklink: Efficient link discovery for spatial relations via masking areas (2018)
 25. Santipantakis, G., Vouros, G., Glenis, A., Doukeridis, C., Vlachou, A.: The datAcron ontology for semantic trajectories. In: *ESWC-Poster Session* (2017)
 26. Santipantakis, G.M., Glenis, A., Kalaitzian, N., Vlachou, A., Doukeridis, C., Vouros, G.A.: FAIMUSS: flexible data transformation to RDF from multiple streaming sources. In: *Proceedings of the 21th International Conference on Extending Database Technology, EDBT 2018*, Vienna, Austria, March 26–29, 2018., pp. 662–665 (2018). DOI 10.5441/002/edbt.2018.79. URL <https://doi.org/10.5441/002/edbt.2018.79>
 27. Santipantakis, G.M., Kotis, K.I., Vouros, G.A., Doukeridis, C.: Rdf-gen: Generating rdf from streaming and archival data. In: *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS '18*, pp. 28:1–28:10. ACM, New York, NY, USA (2018). DOI 10.1145/3227609.3227658. URL <http://doi.acm.org/10.1145/3227609.3227658>
 28. Santipantakis, G.M., Vouros, G.A., Doukeridis, C., Vlachou, A., Andrienko, G.L., Andrienko, N.V., Fuchs, G., Garcia, J.M.C., Martinez, M.G.: Specification of semantic trajectories supporting data

- transformations for analytics: The datacron ontology. In: Proceedings of the 13th International Conference on Semantic Systems, SEMANTICS 2017, Amsterdam, The Netherlands, September 11-14, 2017, pp. 17–24 (2017). DOI 10.1145/3132218.3132225. URL <https://doi.org/10.1145/3132218.3132225>
29. Soltan Mohammadi, M., Mougenot, I., Thérèse, L., Christophe, F.: A Semantic Modeling of Moving Objects Data to Detect the Remarkable Behavior. In: AGILE 2017. Wageningen University, Chair group GIS & Remote Sensing (WUR-GRS), Wageningen, Netherlands (2017). URL <https://hal.archives-ouvertes.fr/hal-01577679>
 30. Spaccapietra, S., Parent, C., Damiani, M.L., de Macêdo, J.A.F., Porto, F., Vangenot, C.: A conceptual view on trajectories. *Data Knowl. Eng.* **65**(1), 126–146 (2008)
 31. Vincenty, T.: Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations. In: *Survey Review XXII*, pp. 88–93 (1975). DOI <https://doi.org/10.1179/sre.1975.23.176.88>. URL https://www.ngs.noaa.gov/PUBS_LIB/inverse.pdf
 32. Wen, Y., Zhang, Y., Huang, L., Zhou, C., Xiao, C., Zhang, F., Peng, X., Zhan, W., Sui, Z.: Semantic modelling of ship behavior in harbor based on ontology and dynamic bayesian network. *ISPRS International Journal of Geo-Information* **8**(3) (2019). DOI 10.3390/ijgi8030107. URL <http://www.mdpi.com/2220-9964/8/3/107>
 33. Yan, Z., Macedo, J., Parent, C., Spaccapietra, S.: Trajectory ontologies and queries. *Transactions in GIS* **12**(s1), 75–91 (2008). DOI 10.1111/j.1467-9671.2008.01137.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9671.2008.01137.x>
 34. Yan, Z., Macedo, J., Parent, C., Spaccapietra, S.: Trajectory ontologies and queries. *Transactions in GIS* **12**(s1), 75–91 (2008)