Populated knowledge graph

Deliverable ID:	D4.3
Dissemination Level:	PU
Project Acronym:	AISA
Grant:	892618
Call:	H2020-SESAR-2019-2
Topic:	SESAR-ER4-01-2019
Consortium Coordinator:	FTTS
Edition date:	16 July 2021
Edition:	00.01.00
Template Edition:	02.00.02

SESAR JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Tomislav Radišić/FTTS	Assistant Professor	6 May 2021
Mia Bazina/FTTS	Project Associate	17 May 2021
Dorea Antolović/FTTS	Project Associate	17 May 2021

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Tomislav Radišić/FTTS	Assistant Professor	29 June 2021
Bernd Neumayr/JKU	University Assistant	7 July 2021
Michael Schrefl/JKU	University Professor	7 July 2021

Approved for submission to the SJU By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Tomislav Radišić/FTTS	Assistant Professor	16 July 2021

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
00.00.01	17/05/2021	Initial draft	Mia Bazina	New document
00.00.02	08/06/2021	Draft	Dorea Antolović	New sections
00.00.03	29/06/2021	Final draft	Tomislav Radišić	First final draft
00.00.04	09/07/2021	Comments integrated	Tomislav Radišić	Comments integrated to final draft
00.01.00	16/07/2021	First issue	Tomislav Radišić	First issue

Copyright Statement

AISA

AI SITUATIONAL AWARENESS FOUNDATION FOR ADVANCING AUTOMATION

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892618 under European Union’s Horizon 2020 research and innovation programme.

Abstract

A knowledge graph is a collection of descriptions of various data which is put into context and then provides a framework for future work. This populated knowledge graph provides insight into traffic situations and all the data needed to describe it with enough detail so it would be functional both in real life and as a simulation. This deliverable will serve as a valuable input into further development of the conceptual model and will be a basis for the development of the deliverable D4.4 Facts, rules and queries capturing en-route ATC operations.

1 Table of contents

List of Figures

1 Introduction

This deliverable handles task 4.3. which is Populating the Knowledge Graph. Certain Air Traffic Management (ATM) information references and data exchange models already exist in a UML (Unified Modeling Language) form, such as AIXM (Aeronautical Information Exchange Model), FIXM (Flight Information Exchange Model) and WXXM (Weather Information Exchange Model). However, a portion of the knowledge graph however had to be created for this project and is described as *plain data*. Even though AIXM and FIXM contain most of the needed data, it was difficult to connect them all together into a single knowledge graph. This was made possible by creating a higher-level knowledge graph that, for example, can mention both datalink and meteorology for a certain flight.

The main purpose of this deliverable is to identify and combine relevant data into a conceptual schema. A UML-to-RDFS/SHACL (Resource Description Framework Schema/Shapes Constraint Language) mapper has been developed (see Deliverable D4.1) that produces the vocabulary and schema of the Knowledge Graph (KG). Based on that vocabulary and schema, the KG is populated with aeronautical and other relevant data. In addition, ML (Machine Learning) modules have been developed and adapted as inputs into the KG.

The populated knowledge graph will be the basis for the AI SA system as it will provide the necessary knowledge about the traffic situations that will be studied during this project. The KG consists of accumulated data mostly from AIXM and FIXM as these models provide most of the data needed to build a traffic situation. The attributes were carefully selected and are thoroughly explained in Chapter 5 of this deliverable. To validate the written KG and all the coded data, a validation program was used that mapped the KG and ran it through SHACL to make sure everything has been described according to the prescribed rules and limitations. And lastly, to complete the KG, other, more complex, inputs were necessary. This is where the ML modules come into play. Three of them were developed and each of them is valuable as an input into the KG: the 4D trajectory prediction module, the conflict detection module, and an air traffic complexity estimation module.

2 Knowledge graph

A knowledge graph is a knowledge base designed to integrate data. Knowledge graphs are usually used to store various descriptions of a structure, such as objects or even concepts, with semantics. They combine databases, graphs and knowledge bases that can be used to interpret data and deduce new facts. Ontologies are the essence of formal semantics (branch of linguistics and logic concerned with meaning) of a knowledge graph. Ontologies are the data schema of the graph. They ensure that there will be an understanding of the represented data and its meaning. To understand knowledge graphs it is also important to understand their instruments [1].

It is often that a description of a structure contains a classification in regards to a class hierarchy. Class is somewhat of an object-oriented design, where one entity usually only belongs to one class. Relationships between entities usually have types that clarify the nature of the relationship which can also have formal definitions. There are also symmetric relationships and transitive relationships. Categories describe an aspect of its semantics and are described and ordered into a taxonomy. It is also worth mentioning free-text descriptions that is a human-friendly text description which is provided to further describe the intentions of the design for the entity [1].

Knowledge graphs are often related to open data projects, where the focus is on the connections between concepts and entities [2]. They are also related to search engines and knowledge engines, as well as various social networks. To build a knowledge graph for this project, various data inputs were needed. Those inputs are further explained in the next chapter.

3 Data inputs

3.1 AIXM

The Aeronautical Information Exchange Model (AIXM) is a digital format of aeronautical information in the scope of Aeronautical Information Services (AIS). Considering the growing automation, AIS is doing its best to make a switch from paper products to storing everything in form of digital data. AIXM is enabling the transition to digital aeronautical data throughout the data chain, with regards to the segment connecting AIS with the user. The official AIXM website is maintained by the European Organisation for the Safety of Air Navigation (Eurocontrol) and the Federal Aviation Administration (FAA) [3]. AIXM provided us with a lot of data needed to describe the used airspace for AISA (Swiss sector LSAZM567) and other static data such as navaids and airports. All of this and more is thoroughly and legibly described and shown via diagrams that facilitate the search for necessary data.

3.2 FIXM

The Flight Information Exchange Model (FIXM) is a global exchange standard that deals with flight (and flow) information. It is implemented in UML (Unified Modeling Language) and XML (Extensible Markup Language) and is compatible with the Flight and Flow Information for a Collaborative Environment (FF-ICE) concept. FIXM is one of the components of the Information Exchange Models layer [4].

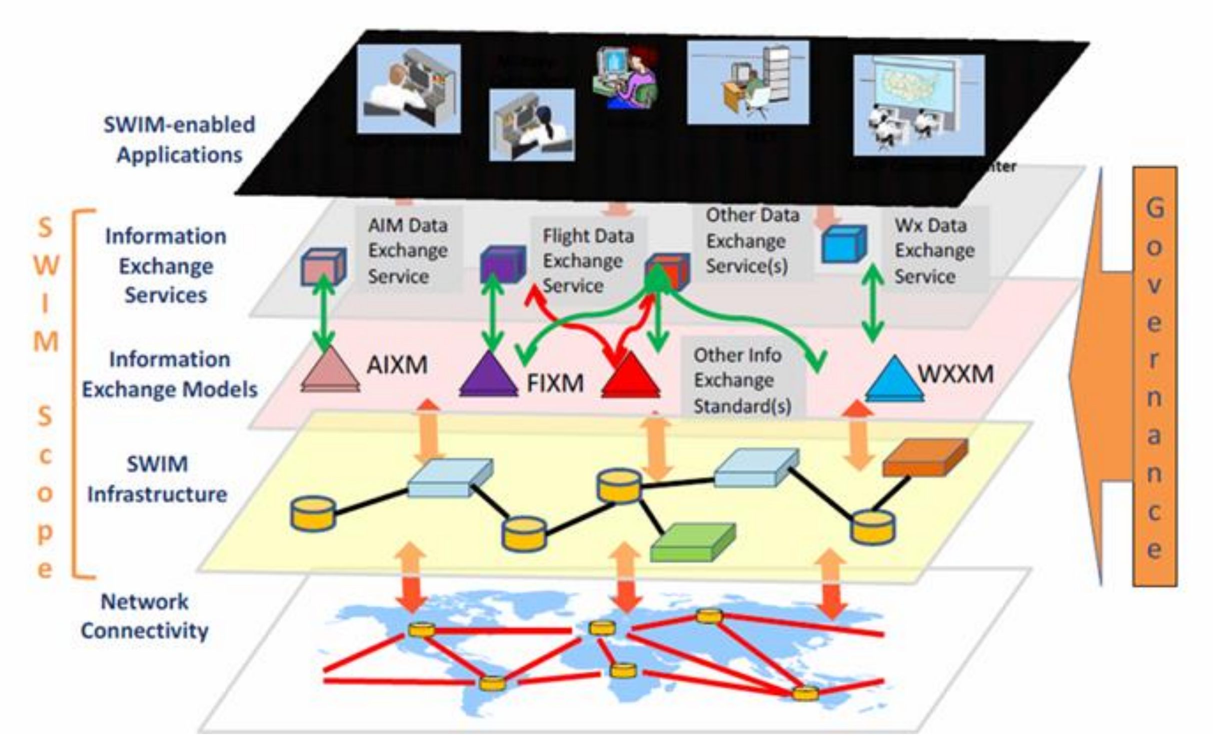

SWIM-enabled Applications
SWIM Scope
Information Exchange Services
Information Exchange Models
SWIM Infrastructure
Network Connectivity
AIM Data Exchange Service
Flight Data Exchange Service
Other Data Exchange Service(s)
Wx Data Exchange Service
AIXM
FIXM
Other Info Exchange Standard(s)
WXXM
Governance

Figure 1. ICAO SWIM global interoperability framework [4]

FIXM only manages the information elements needed by stakeholders to exchange at varying times, according to FF-ICE. It also supports the content of individual Air Traffic Service (ATS) message fields that are described in ICAO Doc 4444 [5] and new flight data identified as a part of the requirements for FF-ICE [4]. Considering FIXM focuses on flight data, which is of great importance to this project, FIXM data has been used to describe traffic situations and aircraft positions in space and time. The most important named graph that deals with flights and uses FIXM data is AC_ROUTE_INFO.ttl which is described in Chapter 5.

3.3 Plain input

Even though AIXM and FIXM contain a great majority of needed data and connections between said data, some of it still needs to be added. The used validation program later explained in Chapter 6 supports a plain input as well which allows for a brand-new UML class diagram/model to be made with all the missing data and connections. Based on it a named graph can be written that matches the diagram and the constraints that are set in place. An example of data that could not have been described by other means is FLAS (Flight Level Allocation Scheme) which is only contained in LoA (Letters of Agreement).

4 Machine Learning modules

The objective of Work Package 3 (WP 3) was to develop machine learning modules that will be used as inputs into the KG system. A 4D trajectory prediction module with TUBS (Technische Universität Braunschweig) as task lead, conflict detection module developed by UPM (Universidad Politécnica de Madrid) and an air traffic complexity estimation module by project coordinators FTTS (Sveučilište u Zagrebu Fakultet prometnih znanosti). An example of how an ML module will be used as input in KG is shown in Appendix B.

4.1 4D trajectory prediction module

A two-step trajectory prediction method for the module has been developed during WP 3. The first step is a prediction of a granular static track of flights using the neural network. This step does not include temporal information about aircraft track positions and only uses the filed flight plans as input. Since this is a low requirement approach this step can be executed as soon there is knowledge about the initial flight plan and waypoints used during training. Flight plans with trajectories similar to the training dataset will naturally have better prediction results than those that have not been trained [6].

For the second step of this method, a combination of the actual aircraft state and an aircraft-fixed pattern is used to predict aircraft position in 100 NM with an input of the previously predicted aircraft track. This is the dynamic part of the trajectory prediction since it requires as input the actual aircraft position and track-angle and needs to be managed during the flight itself. The actual aircraft state during operation might be obtained through ADS-B data. The predicted future position can be calculated if the actual ground speed is known, as well as the timestamp of passing. Considering the noisy ADS-B data regarding ground speed, the prediction timestamps will also turn out to be noisy so the distance parameter of 100 NM was used for the final trajectory prediction [6].

The prediction reliability-metric is indicated by the validation in the second part of the trajectory prediction. Some position predictions could have both a high-reliability value and high deviation in terms of the true future position as well as the other way around. A lot of predictions have less than 6 NM position deviations, but a lot of them have deviations of more than 11 NM, especially if the reliability metric is lower than 0.1. These big deviations that can span up to 140 NM are probably caused by prediction errors that arise from the second part of trajectory prediction regarding the borders of the waypoints grid. These prediction errors could be recognised if an assessment is made of the actual distance from the aircraft state to the predicted position [6].

4.2 Conflict detection module

With the aim of situational awareness, the module uses the concept of Situation of Interest (SI) which would be the expectation of an aircraft pair to intersect with a horizontal separation that is lower than a predetermined one which infringes the vertical separation minima. The determined separation for this module is chosen to be 10 NM horizontally and 1000 ft vertically. Two approaches have been developed, a static mode and a dynamic one. The static mode focuses on the planner ATCO and predicts the SI and its safety metrics when an aircraft shows up in the airspace in regards to the aircraft already located within the airspace. This kind of prediction is only performed once and is not updated. The 4D trajectory prediction of different aircraft is not obtained as input. The dynamic mode focuses on the tactical ATCO and predicts the SI and its safety metrics throughout the aircraft evolution within the airspace. Unlike the static mode, the predictions can be performed dynamically more than once. The dynamic mode also receives the 4D trajectory prediction of other aircraft within the airspace in the same period and takes into account the SI prediction which means its safety metrics evolve according to the trajectory evolution [7].

The entire work on the module has been developed considering only ADS-B data which has been extracted from the OpenSky Network [8] and provided a basis for the ML database. Even though the sector LSAZM567 was used for this work, this methodology can be applied to different airspaces. Since current ADS-B trajectories do not provide SI or conflicts, a database with SI samples has been built. In order to force situations in which SI would appear the trajectories have been modified to enter the airspace within the same time period. There have been three experiments which tackled the imbalance of the database regarding SI samples, which is 5% in the static mode and 18% in the dynamic one. The reference model gives the best theoretical result in balancing the SI without SI samples regarding classification problems, the pure model examines the entire dataset and the hybrid has an operational filter that focuses on aircraft pairs with a distance lower than 20 NM and 1000 ft [7].

The static mode proved to have lower accuracy since it can only predict the safety metrics once an aircraft enters the airspace considering other aircraft within the airspace. Hypothetically, the operational limit is about 95% and the best classification metrics were achieved for the hybrid model (75 %) and the safety metric named MinDis (Minimum Distance) that gave the RMSE (Root Mean Squared Error) of 3.6 NM. The dynamic mode surprisingly increased the performances of the classification and regression models from the static mode. The classification algorithms yielded almost 99% of reliability and the 4D trajectory predictions that were provided as inputs were at 90%. The hybrid model here also gave the best values for the MinDis safety metric and it improved the RMSE from 3.8 to 1.5 NM [7].

The work on this module confirmed that ML techniques can be applied for conflict detection purposes and used in isolation of 4D trajectory predictions, or with them. Used ML techniques have improved the prediction ability from the 4D trajectory predictions that were considered as data input. This means that ML techniques improve the initial 4D trajectory predictions considering that the ML model “learns” from the initial predictions and improves them [7].

4.3 Air traffic complexity estimation module

The created model for air traffic complexity estimation is proof that air traffic complexity can be determined based on ATCO tasks, which means it does not depend on a person or the airspace. One of its advantages is that it is an LOAA (Learn Once Apply Anywhere) model which has been confirmed through validation in realistic airspace. The model was able to calculate the air traffic complexity with errors that were below ATCO assessment levels [9].

The contribution of individual ATCO tasks to the overall complexity is determinable from the coefficients of the linear regression model. Since automation is constantly rising in ATC (Air Traffic Control), it will become increasingly more plausible to determine how the automation of a specific task can lower the workload of an ATCO and the air traffic situation complexity. This model also identifies the most complex aircraft, those being the ones that cause the highest rise in complexity, in a designated traffic situation. After the identification of such aircraft, the model checks how certain parameter changes regarding that aircraft affect the overall complexity [9].

ATCO tasks have been coded based on ten basic tasks, four of which have been further categorized according to their parameters. This means that each classification adds a one-digit code to the final task designation. This elaborate classification is thoroughly explained in a dedicated deliverable D3.1 Air traffic complexity module [9].

5 Populated knowledge graph

Populating the knowledge graph using automated data translators has proven to be an inadequate approach considering the complexity of the sources. They lack standardization and data must be extensively pre-processed to make sure the quality of the information in the knowledge graph remains high. Even though there are many points of overlap between AIXM and FIXM concepts, those overlaps are not especially well-articulated. Some common concepts are named differently and vary in detail which hinders the task at hand. Also, as already mentioned in Chapter 3, there was a need to create KGs "from scratch" considering there were no alternatives for data that needed to be put together. For those reasons, automatic data translation has been repurposed to manual data translation.

Data described later in this chapter can be divided into static and dynamic data. Static data is the one that does not change over time. This would be data about the sector (horizontalProjection, lowerLimit, upperLimit), military areas (timeInterval, status), entry and exit points (type, relativeLocation), designated points (designator, type, name etc.), navaids (name, elevation, frequency etc.), airport data (fieldElevation, operationalStatus, aixm:warning etc.) and static aircraft data (aircraft address, departureAerodrome, aerodromeOfDestination, wakeTurbulence etc.). When talking about dynamic data we would consider data that changes over time, such as 4D flight-related data (track, actualSpeed, flightLevel, position, verticalRate) and for the ML modules time-related data (timeOfPrediction and sequenceNumber). The dynamic integration of the KG will be more thoroughly discussed in deliverable D4.4 Facts, rules and queries capturing en-route ATC operations. All of the RDF instances described in the text below show only one of the examples for each instance that contains the aforementioned data.

5.1 AC_ROUTE_INFO.ttl

This RDF instance describes some of the most important data regarding a flight. It uses data from FIXM that enables us to provide enough information about the flight for the system to be able to generate a traffic situation. It is divided into three separate sections for clarity and to simplify the repetitive process of populating the knowledge graph. The first section contains unchanged data in each iteration of the instance, whereas the data stored in the second and third part will have to be modified in each time step. The example used to demonstrate the RDF instance will be AC_ROUTE_INFO.ttl.

The first section of the instance defines the unique flight and its permanent information, such as aircraft description, flight operator, aerodrome of departure and destination airport. This data is meant to remain unchanged with each time-step in the knowledge graph but will be included in each iteration to help connect the following, changeable, information to the correct flight.

```
s1:ID_002
       a fixm:EfplFlight ;
       fixm:flightType [ rdf:value "SCHEDULED" ] ;
       fixm:aircraftDescription s1:ID_003;
       fixm:flightIdentification s1:ID_004 ;
       fixm:operator s1:ID_005 ;
       fixm:departure s1:ID_006 ;
       fixm:aerodromesOfDestination s1:ID_007 .
s1:ID_003
       a fixm:Aircraft ;
       fixm:aircraftAddress [ rdf:value "4242DC" ] ;
       fixm:aircraftPerformance [ rdf:value "D" ] ;
       fixm:aircraftQuantity [ rdf:value 1 ] ;
       fixm:wakeTurbulence [ rdf:value "M" ] ;
       fixm:aircraftType s1:ID_011  .
s1:ID_011
       a fixm:AircraftType ;
       fixm:icaoModelIdentifier [ rdf:value "B738" ] .
s1:ID_004
       a fixm:FlightIdentification ;
       fixm:aircraftIdentification [ rdf:value "AFL2529" ] .
s1:ID_005
       a fixm:AircraftOperator ;
       fixm:operatorCategory [ rdf:value "AIR_CARRIER" ] ;
       fixm:operatingOrganization s1:ID_017 .
s1:ID_002
       a fixm:EfplFlight ;
       fixm:flightType [ rdf:value "SCHEDULED" ] ;
       fixm:aircraftDescription s1:ID_003;
       fixm:flightIdentification s1:ID_004 ;
       fixm:operator s1:ID_005 ;
       fixm:departure s1:ID_006 ;
       fixm:aerodromesOfDestination s1:ID_007 .
s1:ID_003
       a fixm:Aircraft ;
       fixm:aircraftAddress [ rdf:value "4242DC" ] ;
       fixm:aircraftPerformance [ rdf:value "D" ] ;
       fixm:aircraftQuantity [ rdf:value 1 ] ;
       fixm:wakeTurbulence [ rdf:value "M" ] ;
       fixm:aircraftType s1:ID_011  .
s1:ID_011
       a fixm:AircraftType ;
       fixm:icaoModelIdentifier [ rdf:value "B738" ] .
s1:ID_004
       a fixm:FlightIdentification ;
       fixm:aircraftIdentification [ rdf:value "AFL2529" ] .
s1:ID_005
       a fixm:AircraftOperator ;
       fixm:operatorCategory [ rdf:value "AIR_CARRIER" ] ;
       fixm:operatingOrganization s1:ID_017 .
s1:ID_017
       a fixm:Organization ;
       fixm:name [ rdf:value "AFL" ] .
s1:ID_006
       a fixm:EfplFlightDeparture ;
       fixm:estimatedOffBlockTime [ rdf:value "2019-07-
03T22:09:00.000Z"^^xsd:dateTime ] ;
       fixm:departureAerodrome s1:ID_016.
s1:ID_016
       a fixm:IcaoAerodromeReference ;
       fixm:code [ rdf:value "LEMG" ] .
s1:ID_007
       a fixm:AerodromesOfDestination ;
       fixm:aerodromeOfDestination s1:ID_018 .
s1:ID_018
       a fixm:IcaoAerodromeReference ;
       fixm:code [ rdf:value "UUEE" ] .
s1:ID_017
       a fixm:Organization ;
       fixm:name [ rdf:value "AFL" ] .
s1:ID_006
       a fixm:EfplFlightDeparture ;
       fixm:estimatedOffBlockTime [ rdf:value "2019-07-
03T22:09:00.000Z"^^xsd:dateTime ] ;
       fixm:departureAerodrome s1:ID_016.
s1:ID_016
       a fixm:IcaoAerodromeReference ;
       fixm:code [ rdf:value "LEMG" ] .
s1:ID_007
       a fixm:AerodromesOfDestination ;
       fixm:aerodromeOfDestination s1:ID_018 .
s1:ID_018
       a fixm:IcaoAerodromeReference ;
       fixm:code [ rdf:value "UUEE" ] .
```

The next distinct section details the current aircraft position, as well as speed, track and vertical rate. The position is defined using position time, flight level and coordinates. The position data changes with each time step, while the speed, track and vertical rate may remain unchanged from one time step to the next, depending on the actual state of the aircraft.

```
s1:FDER11
       a fixm:AircraftPosition;
       fixm:positionTime [rdf:value "2019-07-04T00:00:01.000Z"^^xsd:dateTime];
       fixm:actualSpeed s1:FDER111;
       fixm:flightLevel [rdf:value 360];
       fixm:track s1:FDER113;
       fixm:srsName [ rdf:value "urn:ogc:def:crs:EPSG::4326" ];
       fixm:position s1:FDER114.
s1:FDER111
       a fixm:ActualSpeed;
       fixm:uom "KNOTS"; rdf:value "254.2233177703353" .
s1:FDER113
       a fixm:Direction;
       fixm:uom "DEGREES";
       rdf:value 54.63753811293095 .
s1:FDER114
       a fixm:SignificantPoint;
       rdf:value "45.82722473144531 5.814246200933689" .
s1:ROCD
       a fixm:VerticalRate;
       fixm:uom "FEET_PER_MINUTE" ;
       rdf:value "300" .
s1:COORD
a fixm:CoordinationStatus ;
       fixm:coordinationStatus [ rdf:value "ASSUMED" ] .
```

Finally, the last section of the instance contains information from the flight plan – the planned route and trajectory data, part of which can be seen below. For the AISA project, the information for those attributes is taken from the Eurocontrol NEST (Network strategic tool) software. The planned route data, once defined, will remain unchanged with each time step. The trajectory data will change by removing the previously next-planned position if it has already been reached in the following iteration. That way, the planned trajectory points will always be just those that are planned to come after the current position.

```
s1:RFL
       a fixm:Route;
       fixm:requestedAltitude [ fixm:uom "FL" ; rdf:value "370" ] .
s1:ID_008
       a fixm:EfplTrajectoryRoutePair ;
       fixm:route s1:ID_020 ;
       fixm:trajectory s1:ID_021 .
s1:ID_020
       a fixm:EfplRoute ;
       fixm:efplFlightRules [ rdf:value "IFR" ] ;
       fixm:routeText [ rdf:value "*MLG3 *MLG4 MLG MGA BLN ANZAN NASOS INTAX CJN
BRITO LARDA RONNY TOPTU GONUP TOU GAI MAKIL DITEV MEN NINUN MEZIN LATAM OTROT ETREK
LUXAN ARKOX LTP GIPNO NAVLA SOPLO OMASI GAMSA DOLUP ENORU POLON ASLUX ETUBO RUDKA
REWED ORHIM GOVIK MNS ABKOM PASUD MOSON KOMOG RUBAG GUBON FK" ] .
```

```
s1:ID_021
       a fixm:Trajectory ;
       fixm:trajectoryPoint s1:P1, s1:P2, s1:P3, s1:P4, s1:P5, s1:P6, s1:P7, s1:P8,
s1:P9, s1:P10.
s1:P1
       a fixm:EfplTrajectoryPoint ;
       fixm:trajectoryPointType [ rdf:value 0 ] ;
       fixm:point s1:P11 .
s1:P11
       a fixm:EfplPoint4D ;
       fixm:srsName [ rdf:value "urn:ogc:def:crs:EPSG::4326" ] ;
       fixm:pos [ rdf:value 46.64286804199210, 7.661356111852130 ] ;
       fixm:time [rdf:value "2019-07-04T00:00:01.000Z"^^xsd:dateTime];
       fixm:flightLevel s1:P12 .
s1:P12
       a fixm:FlightLevel ;
       fixm:level [ rdf:value 360 ] ;
       fixm:unit [ rdf:value "S" ] .
```

Used attributes:

- fixm:EfplFlight -> fixm:flightType – "Indication of the rule under which an air traffic controller provides categorical handling of a flight" [4].

- fixm:Aircraft -> fixm:AircraftAddress – "A code that enables the exchange of text-based messages between suitably equipped ATS ground systems and aircraft cockpit displays" [4].

- fixm:Aircraft -> fixm:aircraftPerformance – "A coded category assigned to the aircraft based on a speed directly proportional to its stall speed, which functions as a standardized basis for relating aircraft maneuverability to specific instrument approach procedures" [4].

- fixm:Count -> aircraftQuantity – "Number of aircraft flying in a formation in which the aircraft are governed by one flight plan" [4].

- fixm:Aircraft -> fixm:wakeTurbulence – "International Civil Aviation Organization (ICAO) classification of the aircraft wake turbulence based on the maximum certified takeoff mass" [4].

- fixm:Aircraft -> fixm:aircraftType – "The manufacturer and model of the airframe expressed either as an ICAO-approved designator or a text description" [4].

- fixm:Aircraft -> fixm:registration – "A unique, alphanumeric string that identifies a civil aircraft and consists of the Aircraft Nationality or Common Mark and an additional alphanumeric string assigned by the state of registry or common mark registering authority" [4].

- fixm:FlightIdentification -> fixm:aircraftIdentification – "Name used by ATS units to identify and communicate with an aircraft" [4].

- fixm:AircraftOperator -> fixm:operatorCategory – "Specifies the category of the flight operator operating the flight" [4].

- fixm:AircraftOperator -> fixm:operatingOrganization – "The full official name of the State, Organisation, Authority, aircraft operating agency, handling agency etc" [4].

- EfplFlightDeparture -> fixm:estimatedOffBlockTime – "The estimated time at which the aircraft will commence movement associated with departure" [10].

- EfplFlightDeparture -> fixm:departureAerodrome – An aerodrome from which aircraft departed or is planned to depart, as defined in the flight plan.

- fixm:EpflFlight -> fixm:aerodromesOfDestination – An aerodrome that is planned for the aircraft to arrive, as defined in the flight plan.

- fixm:AircraftPosition -> fixm:positionTime – “The time associated with the Current Position of an active flight, from the radar surveillance report or progress report” [11].

- fixm:AircraftPosition -> fixm:actualSpeed – “Actual flight speed supplied by various sources” [4]

- fixm:AircraftPosition -> fixm:flightLevel – “Current aircraft's altitude at standard air pressure, expressed in hundreds of feet” [12].

- fixm:AircraftPosition -> fixm:track – “The direction the aircraft is flying, over the ground, relative to true north. It is the heading of the aircraft as impacted by the wind” [4].

- fixm:AircraftPosition -> fixm:srsName – “The value of the srsName attribute is a Uniform Resource Identifier (URI). It refers to a definition of the CRS that is used to interpret the coordinates in the geometry” [13].

- fixm:AircraftPosition -> fixm:position – “The actual location of an active flight as reported by surveillance, for a flight tracked by radar, or from the position part of a pilot progress report, for an oceanic flight” [4].

- fixm:VerticalRate – “An expression of an aircraft's vertical rate of change (climb if positive, descent if negative)” [4].

- fixm:CoordinationStatus -> fixm:CoordinationStatusCode – “The status of Coordination and Transfer of Control between the currently Controlling Air Traffic Services Unit (ATSU) to the downstream to be Controlling ATSU” [4].

- fixm:Route -> fixm:requestedAltitude – “The filed altitude (flight level) for the first or the whole cruising portion of the flight” [4].

- fixm:EfplRoute -> fixm:efplFlightRules – “The regulation, or combination of regulations, that governs all aspects of operations under which the pilot plans to fly” [4].

- fixm:EfplRoute -> fixm:routeText – “The ICAO route string as depicted from the flight plan” [4].

- fixm:EfplTrajectoryPoint -> fixm:trajectoryPointType – “A container for information pertinent to a single point in a trajectory. These container elements are ordered according to increasing time/distance from the start point” [11].

- fixm:Trajectory -> fixm:EfplTrajectoryPoint ->fixm:EfplPoint4d – A 4D point in the Extended Flight Plan (EFPL).

- fixm:EfplPoint4D -> fixm:pos – Identifies the location of a trajectory point which includes latitude and longitude.

- fixm:EfplPoint4D -> fixm:Time – “Identifies the time of a trajectory point. The time in the point may be expressed as a choice between an absolute time and a relative time (either relative to the previous 4D point or to the first 4D point)” [11].
- fixm:EfplPoint4D -> fixm:FlightLevel - Identifies the altitude of a trajectory point which includes altitude expressed in Flight Levels (FL).

5.2 LSAZM567.ttl

The LSAZM567 instance contains the horizontal and vertical boundaries of the sector LSAZM567 in the Swiss airspace. The input data was taken from Automatic Dependent Surveillance–Broadcast (ADS-B), NEST and information provided by SkyGuide.

```
s1:ASP_PR1
       a aixm:AirspaceTimeSlice;
       gml:validTime s1:vlt1901;
       aixm:interpretation [rdf:value "BASELINE"];
       aixm:sequenceNumber [rdf:value 2];
       aixm:featureLifetime s1:lt1901;
       aixm:name [rdf:value "LSAZM567"];
       aixm:controlType [rdf:value "JOINT"];
       aixm:designator [rdf:value "nilReason"];
       aixm:localType [rdf:value "maxLength"];
       aixm:type [rdf:value "FIR"];
       aixm:upperLowerSeparation [aixm:uom "FL"; rdf:value "644"];
       aixm:geometryComponent s1:geo_pr1.
s1:vlt1901
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2020-01-19T08:38:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:lt1901
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2010-01-01T00:00:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:geo_pr1
       a aixm:AirspaceGeometryComponent;
       aixm:theAirspaceVolume s1:vol_pr1.
s1:vol_pr1
       a aixm:AirspaceVolume;
       aixm:lowerLimit [aixm:uom "FL"; rdf:value "355"];
       aixm:lowerLimitReference [rdf:value "MSL"];
       aixm:maximumLimit [aixm:uom "FL"; rdf:value "999"];
       aixm:maximumLimitReference [rdf:value "MSL"];
       aixm:minimumLimit [rdf:value "unknown"];
       aixm:minimumLimitReference [rdf:value "unknown"];
       aixm:upperLimit [aixm:uom "FL"; rdf:value "999"];
       aixm:upperLimitReference [rdf:value "MSL"];
       aixm:width [aixm:uom "M"; rdf:value "3000.0"];
       aixm:horizontalProjection s1:crv_pr1.
s1:crv_pr1
       a aixm:Surface;
       gml:patches s1:ID01.
s1:ID01
       a gml:PolygonPatch;
       gml:exterior s1:ID02.
s1:ID02
       a gml:Ring;
       gml:curveMember s1:ID03.
s1:ID03
       a gml:Curve;
       gml:segments s1:ID04.
s1:ID04
       a gml:GeodeticString;
       gml:posList [rdf:value "47.91944444 7.88138889",
"47.83333333 8.29583333",
"47.83333333 8.61666667",
"47.83333333 8.85833333",
"47.89027778 9.13694444",
"47.89 9.55",
"47.88333333 9.55",
"47.8 9.55",
"47.53361111 9.73305556",
"47.33944444 9.92194444",
"47.33666667 9.92472222",
"47.27666667 9.9825" ].
```

Used attributes:

- aixm:Airspace -> aixm:timeSlice -> aixm:interpretation – "Property indicating how the timeslice is to be interpreted" [3].
- aixm:timeSlice -> aixm:sequenceNumber – "Used as unique key for the identification of the Time Slice concerned" [3].
- aixm:timeSlice -> aixm:name – "A complex data type that enables the provision of a NIL reason for any attribute using this type" [3].
- aixm:timeSlice -> aixm:featureLifetime – "The start and end of life of the feature" [14].
- aixm:timeSlice -> aixm:controlType – "A code indicating in terms of civil or military, the type of operations" [3].
- aixm:timeSlice -> aixm:designator – "Allowed types of airspace" [3].
- aixm:timeSlice -> aixm:localType – "A type designator used locally (inside a State or a Region) for a particular airspace sub-category" [3].
- aixm:timeSlice -> aixm:type – "A coded list of values that indicates a type of airspace" [3].
- aixm:timeSlice -> aixm:upperLowerSeparation –"A flight level (e.g., 18000 feet MSL (Mean Sea Level)) that indicates the division of airspace between lower and upper airspace" [3].
- aixm:timeSlice -> aixm:geometryComponent – "The role of the component in the airspace geometry. If the geometry of an airspace is composed of single volume, then the attributes of this association class may be left empty" [3].
- aixm:AirspaceGeometryComponent -> aixm:theAirspaceVolume – "A defined volume in the air, described as horizontal projection with vertical limits" [3].
- aixm:theAirspaceVolume -> aixm:lowerLimit – "The vertical distance of the lower bound (floor) of an airspace volume. The vertical distance is measured relative to a separately specified lower limit reference" [3].
- aixm:theAirspaceVolume -> aixm:loverLimitReference – "The reference surface (Mean Sea Level, Ground, standard pressure, etc.) used for determining the value of the lower limit of an airspace volume" [3].

- aixm:theAirspaceVolume -> aixm:maximumLimit – "The vertical distance of the maximum limit of an airspace volume. The vertical distance is measured relative to a separately specified maximum limit reference" [3].

- aixm:theAirspaceVolume -> aixm:maximumLimitReference – "The reference surface (Mean Sea Level, Ground, standard pressure, etc.) used for determining the value of the maximum limit of an airspace volume" [3].

- aixm:theAirspaceVolume -> aixm:minimumLimit - "The vertical distance of the minimum limit of an airspace volume. The vertical distance is measured relative to a separately specified minimum limit reference" [3].

- aixm:theAirspaceVolume -> aixm:minimumLimitReference – "The reference surface (Mean Sea Level, Ground, standard pressure, etc.) used for the value of the minimum limit of an airspace volume" [3].

- aixm:theAirspaceVolume -> aixm:upperLimit – "The vertical distance of the upper bound (ceiling) of an airspace volume. The vertical distance is measured relative to a separately specified upper limit reference" [3].

- aixm:theAirspaceVolume -> aixm:upperLimitReference – "The reference surface (Mean Sea Level, Ground, standard pressure, etc.) used for the value of the upper limit of an airspace volume" [3].

- aixm:theAirspaceVolume -> aixm:width – "The width of the corridor, in case the AirspaceVolume has been originally defined as having a corridor shape" [3].

- aixm:theAirspaceVolume -> aixm:horizontalProjection – "A 2-dimensional primitive and is composed of one or more surface patches as specified in ISO 19107:2003, 6.3.17.1, which is extended with properties that represent the vertical position (elevation, datum, accuracy)" [3].

- aixm:Surface -> gml:patches – "The patches property element contains the sequence of surface patches. The order of the elements is significant and shall be preserved when processing the array" [15].

- gml:PolygonPatch -> gml:exterior – "Exterior boundary of a surface that consists of several rings" [15].

- gml:Ring -> gml:curveMember – "A curve is a 1-dimensional primitive. Curves are continuous, connected, and have a measurable length in terms of the coordinate system" [15].

- gml:Curve -> gml:segments – "A curve is composed of one or more curve segments. Each curve segment within a curve may be defined using a different interpolation method. The curve segments are connected, with the endpoint of each segment except the last being the start point of the next segment in the segment list. The orientation of the curve is positive. The element segments encapsulate the segments of the curve" [15].

- gml:GeodeticString -> gml:posList – "A Geography Markup Language (GML) element that specifies coordinates" [13].

5.3 MILITARY_AIRSPACE.ttl

There are two RDF instances that describe two relevant military airspaces in the LSAZM567 sector. The input data include name and type, activation time and status, as well as the description of the military sectors' geometry. The sources for populating the RDF instances were Aeronautical Information Publication (AIP) chapters received from SkyGuide and NEST software.

```
<LST_TRA_east>
       a aixm:Airspace;
       aixm:timeSlice file:TS1 .
file:TS1
       a aixm:AirspaceTimeSlice;
       gml:validTime file:AVT1 ;
       aixm:interpretation [rdf:value "BASELINE"] ;
       aixm:sequenceNumber [ rdf:value "2"^^xsd:unsignedInt ] ;
       aixm:correctionNumber [ rdf:value "0"^^xsd:unsignedInt ] ;
       aixm:featureLifetime file:AFLT1 ;
       aixm:type [rdf:value "TRA"];
       aixm:designator [rdf:value "TRA HIGH"];
       aixm:localType [rdf:value "MILITARY OPERATING AREA"];
       aixm:name [rdf:value "LS-T/TRA"];
       aixm:controlType [rdf:value "JOINT"];
       aixm:activation file:ACTV ;
       aixm:geometryComponent file:GEO_COMP .
file:AVT1
       a gml:TimePeriod ;
       gml:beginPosition [rdf:value "2019-07-04T00:00:00Z"^^xsd:dateTime] ;
       gml:endPosition [gml:indeterminatePosition "unknown"] .
file:AFLT1
       a gml:TimePeriod ;
       gml:beginPosition [rdf:value "2019-07-04T00:00:00Z"^^xsd:dateTime] ;
       gml:endPosition [gml:indeterminatePosition "unknown"] .
file:ACTV
       a aixm:AirspaceActivation;
       aixm:timeInterval file:t_interval;
       aixm:activity [rdf:value "MILOPS"];
       aixm:status [rdf:value "ACTIVE" ].
file:t_interval
       a aixm:Timesheet;
       aixm:timeReference [rdf:value "UTC"];
       aixm:startDate [rdf:value "01-05"];
       aixm:endDate [rdf:value "01-10"];
       aixm:day [rdf:value "ANY"]; +    aixm:startTime [rdf:value "00:00"];
       aixm:endTime [rdf:value "00:00"];
       aixm:daylightSavingAdjust [rdf:value "NO"];
       aixm:excluded [rdf:value "NO"].
file:GEO_COMP
       a aixm:AirspaceGeometryComponent;
       aixm:theAirspaceVolume file:vol .
file:vol
       a aixm:AirspaceVolume;
       aixm:lowerLimit [aixm:uom "FL"; rdf:value "295"];
       aixm:lowerLimitReference [rdf:value "MSL"];
       aixm:upperLimit [aixm:uom "FL"; rdf:value "660"];
       aixm:upperLimitReference [rdf:value "MSL"];
       aixm:horizontalProjection file:hor_projection.
file:hor_projection
       a aixm:Surface;
       gml:patches file:ID1.
file:ID1
       a gml:PolygonPatch;
       gml:exterior file:ID2.
```

```
file:ID2
       a gml:Ring;
       gml:curveMember file:ID3.
file:ID3
       a gml:Curve;
       gml:segments file:ID4.
file:ID4
       a gml:GeodeticString;
       gml:posList [rdf:value "47.310833 9.583056",
              "47.266667 9.533333",
              "47.186944 9.486389",
              "47.096667 9.519444",
```

Used attributes:

- aixm:AirspaceTimeSlice -> gml:validTime - “A convenience property element” [16].
- aixm:AirspaceTimeSlice -> aixm:interpretation – “Property indicating how the timeslice is to be interpreted” [3].
- aixm: AirspaceTimeSlice -> aixm:sequenceNumber – “Used as unique key for the identification of the Time Slice concerned” [3].
- aixm: AirspaceTimeSlice -> aixm:correctionNumber – “A (positive) number of similar items” [3].
- aixm: AirspaceTimeSlice -> aixm:featureLifetime – “The start and end of life of the feature” [14].
- aixm: AirspaceTimeSlice -> aixm:type – “An indicator of the general structure or characteristics of a particular airspace”[3].
- aixm: AirspaceTimeSlice -> aixm:designator – “A published sequence of characters allowing the identification of the airspace”[3].
- aixm: AirspaceTimeSlice -> aixm:localType – “A type designator used locally (inside a State or a Region) for a particular airspace sub-category”[3].
- aixm: AirspaceTimeSlice -> aixm:name – “The name given to an airspace by a responsible authority. It should be written as published, with no significance to upper or lower case letters” [3].
- aixm: AirspaceTimeSlice -> aixm:controlType – “The primary organization type in terms of civil or military, providing air traffic services within a designated airspace” [3].
- aixm: AirspaceTimeSlice -> aixm:AirspaceActivation – “Information about the operational status of the airspace” [3].
- aixm: AirspaceTimeSlice -> aixm:activation – “The role of the component in the airspace geometry. If the geometry of an airspace is composed of single volume, then the attributes of this association class may be left empty”.
- aixm: AirspaceTimeSlice -> aixm:geometryComponent – “The role of the component in the airspace geometry. If the geometry of an airspace is composed of single volume, then the attributes of this association class may be left empty” [3].

- aixm:AirspaceActivation -> aixm:timeInterval – “A component of the schedule” [3].
- aixm:AirspaceActivation -> aixm:activity - “The primary situation or reason on the ground or in the air, which may have an impact on airspace usage”[3].
- aixm:AirspaceActivation -> aixm:status – “The activation status of the airspace”[3].
- aixm:Timesheet -> aixm:timeReference – “An indicator of the time reference system or reference system offset (for example, Coordinated Universal Time (UTC), or UTC minus 7 hours), for time values specified in a timesheet”[3].
- aixm:Timesheet -> aixm:startDate – “The start date of the validity of a timesheet. Used together with a timesheet's 'EndDate' to express repetitive time periods like 15 May to 15 November”[3].
- aixm:Timesheet -> aixm:endDate – “The end date of the validity of a timesheet. Used together with a timesheet's 'StartDate' to express repetitive time periods like 15 May to 15 November” [3].
- aixm:Timesheet -> aixm:day – “An indicator of the day the timesheet is referring to” [3].
- aixm:Timesheet -> aixm:startTime – “The time of the day (specified in the timesheet's 'day' attribute) when the period described in the timesheet starts” [3].
- aixm:Timesheet -> aixm:endTime – “The time of the day when the period described in the timesheet ends” [3].
- aixm:Timesheet -> aixm:daylightSavingAdjust - “Indicates that the 'StartTime' and 'EndTime' values of a timesheet have to be decreased by one hour when Daylight Saving is in force (summer time)” [3].
- aixm:Timesheet -> aixm:excluded – “An indicator that the time block indicated by a timesheet is excluded (subtracted) from the total time interval(s) defined by one or more additional timesheet(s)” [3].
- aixm:GeometryComponent ->aixm:AirspaceVolume – “A defined volume in the air, described as horizontal projection with vertical limits” [3].
- aixm:AirspaceVolume -> aixm:lowerLimit – “The vertical distance of the lower bound (floor) of an airspace volume. The vertical distance is measured relative to a separately specified lower limit reference” [3].
- aixm:AirspaceVolume -> aixm:loverLimitReference – “The reference surface (Mean Sea Level, Ground, standard pressure, etc.) used for determining the value of the lower limit of an airspace volume” [3].
- aixm:AirspaceVolume -> aixm:upperLimit – “The vertical distance of the upper bound (ceiling) of an airspace volume. The vertical distance is measured relative to a separately specified upper limit reference” [3].

- aixm:AirspaceVolume -> aixm:upperLimitReference – “The reference surface (Mean Sea Level, Ground, standard pressure, etc.) used for determining the value of the lower limit of an airspace volume” [3].

- aixm:AirspaceVolume -> aixm:horizontalProjection – “A 2-dimensional primitive and is composed of one or more surface patches as specified in ISO 19107:2003, 6.3.17.1, which is extended with properties that represent the vertical position (elevation, datum, accuracy)” [3].

- aixm:Surface -> gml:patches - “The patches property element contains the sequence of surface patches. The order of the elements is significant and shall be preserved when processing the array” [15].

- gml:PolygonPatch -> gml:exterior - “Exterior boundary of a surface that consists of a number of rings” [15].

- gml:Ring -> gml:curveMember - “A curve is a 1-dimensional primitive. Curves are continuous, connected, and have a measurable length in terms of the coordinate system” [15].

- gml:Curve -> gml:segments – “A curve is composed of one or more curve segments. Each curve segment within a curve may be defined using a different interpolation method. The curve segments are connected to one another, with the end point of each segment except the last being the start point of the next segment in the segment list. The orientation of the curve is positive. The element segments encapsulates the segments of the curve” [15].

- gml:GeodeticString -> gml:posList – “A GML element that specifies coordinates” [13].

5.4 AIRPORT.ttl

AIRPORT.ttl is an RDF instance containing information about airport availability and its features. This particular example describes the LSZH (Zurich) airport. It includes meteorology data that is purely runway related (Runway Visual Range (RVR), visibility and flight conditions). The most important parts of the instance are about airport operability and possible issued warnings by the airport (bird hazard, surface painting, rubber removal etc.).

```
s1:TS_LSZH
       a aixm:AirportHeliportTimeSlice;
       gml:validTime s1:vtnull0;
       aixm:interpretation [rdf:value "BASELINE"];
       aixm:sequenceNumber [rdf:value "1"^^xsd:unsignedInt];
       aixm:correctionNumber [rdf:value "0"^^xsd:unsignedInt];
       aixm:featureLifetime s1:ltnull0;
       aixm:designator [rdf:value "LSZH"];
       aixm:name [rdf:value "Zuerich Airport" ];
       aixm:locationIndicatorICAO [rdf:value "LSZH"];
       aixm:designatorIATA [ rdf:value "ZRH"];
       aixm:fieldElevation [aixm:uom "M"; rdf:value "432"];
       aixm:magneticVariation [rdf:value -1.0];
       aixm:servedCity s1:servedCity;
       aixm:ARP s1:elpointLSZH;
       aixm:transitionAltitude [ aixm:uom "FT" ; rdf:value "5000" ] ;
       aixm:transitionLevel s1:AHTL1 ;
       aixm:availability s1:AHA1 .
s1:vtnull0
```

```
        a gml:TimePeriod;
        gml:beginPosition [rdf:value "2020-01-01T00:00:00Z"^^xsd:dateTime];
        gml:endPosition [gml:indeterminatePosition "unknown"].
s1:ltnull0
        a gml:TimePeriod;
        gml:beginPosition [rdf:value "2009-01-01T00:00:00Z"^^xsd:dateTime];
        gml:endPosition [gml:indeterminatePosition "unknown"].
s1:servedCity
        a aixm:City;
        aixm:name [rdf:value "Zurich"].
s1:elpointLSZH
        a aixm:ElevatedPoint;
        gml:pos [rdf:value "47.458056 8.548056"].
s1:AHTL1
        a aixm:ValFLType ;
        aixm:uom "FL" ;
        rdf:value 70 .
s1:AHA1
        a aixm:AirportHeliportAvailability ;
        aixm:operationalStatus [ rdf:value "NORMAL" ] ;
        aixm:usage s1:AHU1 .
s1:AHU1
        a aixm:AirportHeliportUsage ;
        aixm:operation [ rdf:value "ALL" ] ;
        aixm:limitation s1:AHL1 .
s1:AHL1
        a aixm:UsageCondition ;
        aixm:contact s1:AHC1 ;
        aixm:selection s1:AHS1 .
s1:AHC1
        a aixm:ContactInformation ;
        aixm:name [ rdf:value "SKYGUIDE"];
        aixm:title [ rdf:value "AIR_NAVIGATION_SERVICE_PROVIDER" ] .
s1:AHS1
        a aixm:ConditionCombination ;
        aixm:weather s1:AHW1 .
s1:AHW1
        a aixm:Meteorology ;
        aixm:flightConditions [ rdf:value "VMC" ] ;
        aixm:runwayVisualRange [ aixm:uom "M" ; rdf:value "1500"^^xsd:decimal ] ;
        aixm:runwayVisualRangeInterpretation [ rdf:value "AT_OR_BELOW" ] ;
        aixm:visibility [ aixm:uom "KM" ; rdf:value "10"^^xsd:decimal ] ;
        aixm:visibilityInterpretation [ rdf:value "AT_OR_ABOVE" ] .
```

Used attributes:

- aixm:AirportHeliportTimeSlice -> gml:validTime - “A convenience property element” [16].
- aixm:AirportHeliportTimeSlice -> aixm:interpretation – “Property indicating how the timeslice is to be interpreted” [3].
- aixm:AirportHeliportTimeSlice -> aixm:sequenceNumber - “Used as unique key for the identification of the Time Slice concerned” [3].
- aixm:AirportHeliportTimeSlice -> aixm:correctionNumber “A (positive) number of similar items” [3].
- aixm:AirportHeliportTimeSlice -> aixm:featureLifetime - “The start and end of life of the feature” [14].
- aixm:AirportHeliportTimeSlice -> aixm:designator – “A designator for an airport/heliport” [3].

- aixm:AirportHeliportTimeSlice -> aixm:name – "The primary official name of an airport/heliport as designated by an appropriate authority" [3].
- aixm:AirportHeliportTimeSlice -> aixm:locationIndicatorICAO – "The International Civil Aviation Organization (ICAO) location indicator of the airport/heliport, as listed in ICAO DOC 7910" [3].
- aixm:AirportHeliportTimeSlice -> aixm:designatorIATA – "The identifier that is assigned to a location in accordance with rules (resolution 767) governed by the International Air Transport Association (IATA)" [3].
- aixm:AirportHeliportTimeSlice -> aixm:magneticVariation – "The angular difference between True North and Magnetic North measured at a given position and date" [3].
- aixm:AirportHeliportTimeSlice -> aixm:servedCity – "The location that is served by the airport" [3].
- aixm:AirportHeliportTimeSlice -> aixm:transitionAltitude – "The altitude at or below which the vertical position of an aircraft is controlled by reference to altitudes at the airport/heliport" [3].
- aixm:AirportHeliportTimeSlice -> aixm:transitionLevel – "The lowest flight level available for use above the transition altitude at the airport/heliport" [3].
- aixm:AirportHeliportTimeSlice -> aixm:availability – "Information about the operational status of the airport/heliport" [3].
- aixm:City -> aixm:name – "The primary official name of an airport/heliport as designated by an appropriate authority" [3].
- aixm:AirportHeliportTimeSlice -> aixm:fieldElevation – "The vertical distance above MSL of the highest point of the landing area" [3].
- aixm:ARP -> aixm:ElevatedPoint – "Airport reference point" [3].
- aixm:elevatedPoint -> gml:pos - "Coordinates in GML the coordinates of geometry objects" [13].
- aixm:AirportHeliportAvailability -> aixm:operationalStatus – "Indicates the availability of the airport/heliport for specific flight operations" [3].
- aixm:AirportHeliportAvailability -> aixm:warning – "A reason for caution when operating at the airport/heliport" [3].
- aixm:AirportHeliportAvailability -> aixm:usage – "Rule which regulates the usage of the AirportHeliport" [3].
- aixm:AirportHeliportUsage -> aixm:operation – "A type of activity for which an Airport/Heliport usage rule is specified" [3].
- aixm:AirportHeliportUsage -> aixm:limitation – "The nature of the limitation (such as a permission or an interdiction) which determine rules for usage of an airport/heliport or one of its surfaces" [3].

- aixm:UsageCondition -> aixm:contact – "The contact information for a person or group" [3].
- aixm:usageCondition -> aixm:selection – "The selection of flights, environmental conditions and times for which the usage is specified" [3].
- aixm:ContactInformation -> aixm:name – "The official name of the contact" [3].
- aixm:ContactInformation -> aixm:title – "The official title of the contact" [3].
- aixm:ConditionCombination -> aixm:weather – "Weather conditions for which the usage is specified" [3].
- aixm:Meteorology -> aixm:flightConditions – "An overall indication whether the meteorological conditions allow visual or instrumental operations" [3].
- aixm:Meteorology -> aixm: runwayVisualRange – "The distance over which a pilot of an aircraft on the centreline of the runway can see the runway surface markings or the lights delineating the runway or identifying its centre line"[3].
- aixm:Meteorology -> aixm: runwayVisualRangeInterpretation – "Indicates whether the Meteorology Runway Visual Range is a minimum or maximum value" [3].
- aixm:Meteorology -> aixm:visibility – "The greatest distance at which lights of 1,000 candelas can be seen and identified against an unlit background (as reported by the airport/heliport)" [3].
- aixm:Meteorology -> aixm: visibilityInterpretation – "Indicates whether the meteorology visibility is a minimum or a maximum value" [3].

5.5 AIRCRAFT_CHARACTERISTIC.ttl

Aircraft characteristics instance with actual data for describing different aircraft types and their capabilities. Input regarding aircraft are mostly trusted sites such as SKYbrary [17]. The example below describes an Airbus A-330-300.

```
<aero:airm:1.0.0:ContextualModel:ATMBusinessTerms:AircraftCharacteristic>
       a aixm:AircraftCharacteristic ;
       aixm:aircraftLandingCategory [ rdf:value "C" ] ;
       aixm:speed [ aixm:uom "KT" ; rdf:value "475"^^xsd:decimal ] ;
       aixm:typeAircraftICAO [ rdf:value "A333" ] ;
       aixm:verticalSeparationCapability [ rdf:value "RVSM" ] ;
       aixm:wakeTurbulence [ rdf:value "HIGH" ] ;
       aixm:weight [ aixm:uom "KG" ; rdf:value "242000"^^xsd:decimal ] .
```

Used attributes:

- aixm:AircraftCharacteristic -> aixm:speed – "An Indicated Air Speed (IAS) that can be maintained by the aircraft" [3].
- aixm:AircraftCharacteristic -> aixm:aircraftLandingCategory – "Classification of aircraft based on 1.3 times stall speed in landing configuration at maximum certified landing mass" [3].

- aixm:AircraftCharacteristic -> aixm:typeAircraftICAO – "A coded aircraft type designator assigned by ICAO to all aircraft heavier than micro-/ultra-light" [3].
- aixm:AircraftCharacteristic -> aixm:verticalSeparationCapability – "An indication of the aircraft capability to maintain a specified vertical separation" [3].
- aixm:AircraftCharacteristic -> aixm:wakeTurbulence – "The wake turbulence category of the aircraft" [3].
- aixm:AircraftCharacteristic -> aixm:weight – "The maximum aircraft take-off weight (MTOW)" [3].

5.6 ENTRY_EXIT_POINTS.ttl

This RDF instance is a description of a point in the airspace. Points described can either be an entry point, an exit point or both. Such information is relevant considering ATCOs (Air Traffic Control Officers) know by heart which point is which but for AISA this needs to be written down. The example shown below defines the entry point PUNSA.

```
<entry_PUNSA>
       a aixm:SignificantPointInAirspace;
       gml:validTime s1:vt_punsa_entry;
       aixm:interpretation [rdf:value "BASELINE"];
       aixm:sequenceNumber [rdf:value 2];
       aixm:featureLifetime s1:lt_punsa_entry;
       aixm:type [rdf:value "ENTRY"];
       aixm:relativeLocation [rdf:value "BORDER"];
       aixm:containingAirspace
<aero:airm:1.0.0:ConceptualModel:Subjects:AirspaceInfrastructure:Airspace:LSAZM567>
       aixm:location s1:loc_punsa.
<aero:airm:1.0.0:ConceptualModel:Subjects:AirspaceInfrastructure:Airspace:LSAZM567>
a aixm:Airspace;
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2019-07-04T00:00:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:lt_punsa_entry
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2019-07-04T00:00:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:loc_punsa
       a aixm:Point;
       gml:pos [rdf:value "46.078611 8.025833"];
       aixm:horizontalAccuracy [aixm:uom "M"; rdf:value "100"].
```

Used attributes:

- aixm:SignificantPointInAirspace -> gml:validTime - "A convenience property element" [16].
- aixm:SignificantPointInAirspace -> aixm:interpretation – "Property indicating how the timeslice is to be interpreted" [3].
- aixm:SignificantPointInAirspace -> aixm:sequenceNumber – "Used as unique key for the identification of the Time Slice concerned" [3].
- aixm:SignificantPointInAirspace -> aixm:featureLifetime – "The start and end of life of the feature" [14].

- aixm:SignificantPointInAirspace -> aixm:type – “A code indicating the type of association between a significant point and an airspace. For example, entry point, exit point, etc.” [3].
- aixm:SignificantPointInAirspace -> aixm:relativeLocation – “A code indicating the location of a significant point in relation to airspace. For example, In, Out, On Border” [3].
- aixm:SignificantPointInAirspace -> aixm:containingAirspace – “The Airspace containing the SignificantPointInAirspace” [3].
- aixm:SignificantPointInAirspace -> aixm:location – “The location of the SignificantPointInAirspace” [3].
- aixm:Point -> gml:pos - “Coordinates in GML represent the coordinates of geometry objects” [13].
- aixm:Point -> aixm:horizontalAccuracy – “The difference between the recorded horizontal coordinates of a point and its true position referenced to the same geodetic datum expressed as a circular error at 95 percent probability” [3]

5.7 NAVAIDS.ttl

A list of all navaids in the chosen sector (LSAZM567) and slightly outside of it for a greater scope and better precision around borders. The listed navaids have been found via Swiss airspace charts and mapped accordingly. This example contains data on a VOR/DME and a Non-Directional Beacon (NDB).

```
s1:NAV_GVA
       gml:validTime s1:nav1901_gva;
       aixm:interpretation [rdf:value "BASELINE"];
       aixm:sequenceNumber [rdf:value 1];
       aixm:featureLifetime s1:ltnav1901_gva;
       aixm:designator [rdf:value "GVA"];
       aixm:flightChecked [rdf:value "YES"];
       aixm:name [rdf:value "Geneva Cointrin VOR-DME"];
       aixm:purpose [rdf:value "ENROUTE"];
       aixm:type [rdf:value "VOR_DME"];
       aixm:servedAirport s1:airp_gva;
       aixm:location s1:locGVA;
       aixm:navaidEquipment s1:GVA_VOR , s1:GVA_DME .
s1:nav1901_gva
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2020-01-19T08:38:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:ltnav1901_gva
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2010-01-01T00:00:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:airp_gva
       a aixm:AirportHeliport;
       aixm:name [rdf:value "Geneva Cointrin International Airport"];
       aixm:locationIndicatorICAO [rdf:value "LSGG"];
       aixm:designatorIATA [rdf:value "GVA"].
s1:locGVA
       a aixm:ElevatedPoint;
       aixm:elevation [aixm:uom "M"; rdf:value "420"];
       gml:pos [rdf:value "46.253899 6.132220"].
s1:GVA_VOR
       a aixm:NavaidComponent ;
```

```
      aixm:collocationGroup [ rdf:value "1"^^xsd:unsignedInt ] ;
      aixm:theNavaidEquipment s1:vor_gva .
s1:vor_gva
      a aixm:VOR ;
      aixm:timeSlice s1:vor1TimeSlice1 .
s1:vor_gvaTimeSlice1
      a aixm:VORTimeSlice ;
      gml:validTime s1:vtnull0_gva_vor;
      aixm:interpretation [rdf:value "BASELINE"];
      aixm:sequenceNumber [rdf:value "1"^^xsd:unsignedInt];
      aixm:correctionNumber [rdf:value "0"^^xsd:unsignedInt];
      aixm:frequency [aixm:uom "MHZ"; rdf:value "115.75"];
      aixm:type [rdf:value "VOR"] .
s1:vtnull0_gva_vor
      a gml:TimePeriod;
      gml:beginPosition [rdf:value "2009-01-01T00:00:00Z"^^xsd:dateTime];
      gml:endPosition [gml:indeterminatePosition "unknown"].
s1:GVA_DME
      a aixm:NavaidComponent ;
      aixm:collocationGroup [ rdf:value "1"^^xsd:unsignedInt ] ;
      aixm:theNavaidEquipment s1:dme_gva .
s1:dme_gva
      a aixm:DME ;
      aixm:timeSlice s1:dme1TimeSlice1 .
s1:dme:gvaTimeSlice1
      a aixm:DMETimeSlice ;
      gml:validTime s1:ltnull0_gva_dme;
      aixm:interpretation [rdf:value "BASELINE"];
      aixm:sequenceNumber [rdf:value "1"^^xsd:unsignedInt];
      aixm:correctionNumber [rdf:value "0"^^xsd:unsignedInt];
      aixm:frequency [aixm:uom "MHZ"; rdf:value "115.75"];
      aixm:type [rdf:value "VOR_DME"].
s1:ltnull0_gva_dme
       a gml:TimePeriod;
       gml:beginPosition [rdf:value "2009-01-01T00:00:00Z"^^xsd:dateTime];
       gml:endPosition [gml:indeterminatePosition "unknown"].
s1:NDB_GLA
      gml:validTime s1:gla2701;
      aixm:interpretation [rdf:value "BASELINE"];
      aixm:sequenceNumber [rdf:value 2];
      aixm:featureLifetime s1:ltgla2701;
      aixm:designator [rdf:value "GLA"];
      aixm:name [rdf:value "Gland NDB"];
      aixm:frequency [aixm:uom "KHZ"; rdf:value "375"];
      aixm:purpose [rdf:value "ENROUTE"];
      aixm:type [rdf:value "NDB"];
      aixm:servedAirport s1:airp_gla;
      aixm:navaidEquipment s1:naveq_gla;
      aixm:location s1:locGLA.
s1:gla2701
      a gml:TimePeriod;
      gml:beginPosition [rdf:value "2020-01-22T08:38:00Z"^^xsd:dateTime];
      gml:endPosition [gml:indeterminatePosition "unknown"].
s1:tgla2701
      a gml:TimePeriod;
      gml:beginPosition [rdf:value "2010-01-01T00:00:00Z"^^xsd:dateTime];
      gml:endPosition [gml:indeterminatePosition "unknown"].
s1:airp_gla
      a aixm:AirportHeliport;
      aixm:name [rdf:value "Geneva Cointrin International Airport"];
      aixm:locationIndicatorICAO [rdf:value "LSGG"];
      aixm:designatorIATA [rdf:value "GVA"].
s1:naveq_gla
      a aixm:NavaidEquipment;
      aixm:designator [rdf:value "GLA"];
```

```
        aixm:flightChecked [rdf:value "YES"];
        aixm:mobile [rdf:value "NO"].
s1:locGLA
        a aixm:ElevatedPoint;
        aixm:elevation [aixm:uom "M"; rdf:value "411"];
        gml:pos [rdf:value "46.408699 6.244250"].
```

Used attributes:

- aixm:Navaid -> aixm:timeSlice – “An abstract class representing a feature Time Slice” [3].
- aixm:NavaidTimeSlice -> gml:validTime – “A convenience property element” [16].
- aixm:NavaidTimeSlice -> aixm:interpretation – “Property indicating how the timeslice is to be interpreted” [3].
- aixm:NavaidTimeSlice -> aixm:sequenceNumber - “Used as unique key for the identification of the Time Slice concerned” [3].
- aixm:NavaidTimeSlice -> aixm:featureLifetime – “The start and end of life of the feature” [14].
- aixm:NavaidTimeSlice -> aixm:designator – “The coded identifier given to the navaid system” [3].
- aixm:NavaidTimeSlice -> aixm:name – “The long name given to the composite navaid” [3].
- aixm:NavaidTimeSlice -> aixm:frequency – “The frequency value of a navigation system” [3].
- aixm:NavaidTimeSlice -> aixm:purpose – “Indication of the intended usage of the navaid (enroute or terminal)” [3].
- aixm:NavaidTimeSlice -> aixm:type – “Type of the navaid service such as Instrument Landing System (ILS), Microwave Landing System (MLS), Very High Frequency Omni-Directional Range/ Distance Measuring Equipment (VOR/DME), etc.” [3].
- aixm:NavaidTimeSlice -> aixm:servedAirport – “The AirportHeliport served by the Navaid” [3].
- aixm:NavaidTimeSlice -> aixm:navaidEquipment – “The component of the navaid system” [3].
- aixm:AirportHeliport -> aixm:name – “The primary official name of an airport/heliport as designated by an appropriate authority” [3].
- aixm:AirportHeliport -> aixm:locationIndicatorICAO – “The International Civil Aviation Organization (ICAO) location indicator of the airport/heliport, as listed in ICAO DOC 7910” [3].
- aixm:AirportHeliport -> aixm:designatorIATA – “The identifier that is assigned to a location in accordance with rules (resolution 767) governed by the International Air Transport Association (IATA)” [3].
- aixm:NavaidEquipment -> aixm:designator – “The identifying code given to the navaid” [3].
- aixm:NavaidEquipment -> aixm:flightChecked – “Indicates if the navaid equipment has been flight checked” [3].
- aixm:NavaidEquipment -> aixm:mobile – “Indicates if the navaid equipment is mobile” [3].

- aixm:ElevatedPoint -> aixm:elevation – “The vertical distance of the point measured from MSL” [3].
- aixm:ElevatedPoint -> gml:pos – “Coordinates in GML represent the coordinates of geometry objects” [13].
- aixm:NavaidComponent -> aixm:collocationGroup – “Set of navaid equipments that are collocated. All Navaid equipment that has the same value for this property are collocated together. For example an ILS might have a Localizer with collocation Group number = 1 and collocated DME and Glidepath sharing collocationGroup number = 2” [3].
- aixm:NavaidComponent -> aixm:theNavaidEquipment – “A physical navaid equipment like VOR, DME, localizer, Tactical Air Navigation System (TACAN), etc.” [3].
- aixm:VOR -> aixm:VORTimeSlice - “A short-range, very-high-frequency omnidirectional beacon which provides an indication in the aircraft of the bearing of the beacon, or left-right track indication” [3].
- aixm:VORTimeSlice -> gml:validTime - “A convenience property element” [16].
- aixm:VORTimeSlice -> aixm:interpretation – “Property indicating how the timeslice is to be interpreted” [3].
- aixm:VORTimeSlice -> aixm:sequenceNumber – “Used as unique key for the identification of the Time Slice concerned” [3].
- aixm:VORTimeSlice -> aixm:correctionNumber – “A (positive) number of similar items” [3].
- aixm:VORTimeSlice -> aixm:frequency – “The value of the frequency” [3].
- aixm:VORTimeSlice -> aixm:type – “A code indicating a type of VOR, e.g. conventional VOR or Doppler VOR” [3].

5.8 DESIGNATED_POINTS.ttl

An RDF instance that records all the designated points within the LSAZM567 sector and around it. NEST values have been used to properly determine their positions.

```
s1:DPN_SOSAL
   a aixm:DesignatedPointTimeSlice;
   gml:validTime s1:vtSOSAL;
   aixm:interpretation [rdf:value "BASELINE"];
   aixm:sequenceNumber [rdf:value 2];
   aixm:featureLifetime s1:ltSOSAL;
   aixm:designator [rdf:value "SOSAL"];
   aixm:type [rdf:value "ICAO"];
   aixm:name [rdf:value "SOSAL"];
   aixm:location s1:pSOSAL.
s1:vtSOSAL
   a gml:TimePeriod;
   gml:beginPosition [rdf:value "2017-07-01T00:00:00Z"^^xsd:dateTime];
   gml:endPosition [gml:indeterminatePosition "unknown"].
s1:ltSOSAL
   a gml:TimePeriod;
```

```
    gml:beginPosition [rdf:value "2009-01-01T00:00:00Z"^^xsd:dateTime];
    gml:endPosition [gml:indeterminatePosition "unknown"].
```

Used attributes:

- aixm:DesignatedPoint -> aixm:DesignatedPointTimeSlice – "A geographical location not marked by the site of a radio navigation aid, used in defining an ATS route, the flight path of an aircraft or for other navigation or ATS purposes" [3].
- aixm:DesignatedPointTimeSlice -> gml:validTime "A convenience property element" [16].
- aixm:DesignatedPointTimeSlice -> aixm:interpretation – "Property indicating how the timeslice is to be interpreted" [3].
- aixm:DesignatedPointTimeSlice -> aixm:sequenceNumber – "Used as unique key for the identification of the Time Slice concerned" [3].
- aixm:DesignatedPointTimeSlice -> aixm:featureLifetime – "The start and end of life of the feature" [14].
- aixm:DesignatedPointTimeSlice -> aixm:designator – "The coded identifier of a designated point" [3].
- aixm:DesignatedPointTimeSlice -> aixm:type - The kind of point designator, indicating the rules by which the designator has been created.
- aixm:DesignatedPointTimeSlice -> aixm:name – "The full textual name of a designated point, if any. For example, 'GOTAN Intersection' for GOTAN. Name may also be used to identify an unnamed point (See ARINC 424 for unnamed points)" [3].
- aixm:DesignatedPointTimeSlice -> aixm:location – "The geographical location of the designated point" [3].
- aixm:Point -> gml:pos – "The location of the designated point" [3].
- aixm:Point -> aixm:horizontalAccuracy – "The difference between the recorded horizontal coordinates of a point and its true position referenced to the same geodetic datum expressed as a circular error at 95 percent probability" [3].

5.9 PLAIN_DATA.ttl

An RDF instance containing all the data and connections that do not show up in AIXM nor FIXM, at least not in a relationship favourable for this project and its KG. It contains known attributes such as datalink (initialCall, frequencyChange, equipmentDegradation etc.), meteorology data (precipitation, cloudCoverage, windspeed etc.), FLAS (Flight Level Allocation Scheme) regarding routes and points on those routes, as well as other miscellaneous data that cannot be congregated by other means.

Figure 2 shows a basic UML class diagram that represents the concept for plain data input. This one and the following UML diagrams are used to generate an RDF schema with the mapper (the validation program is further explained in the following chapter).

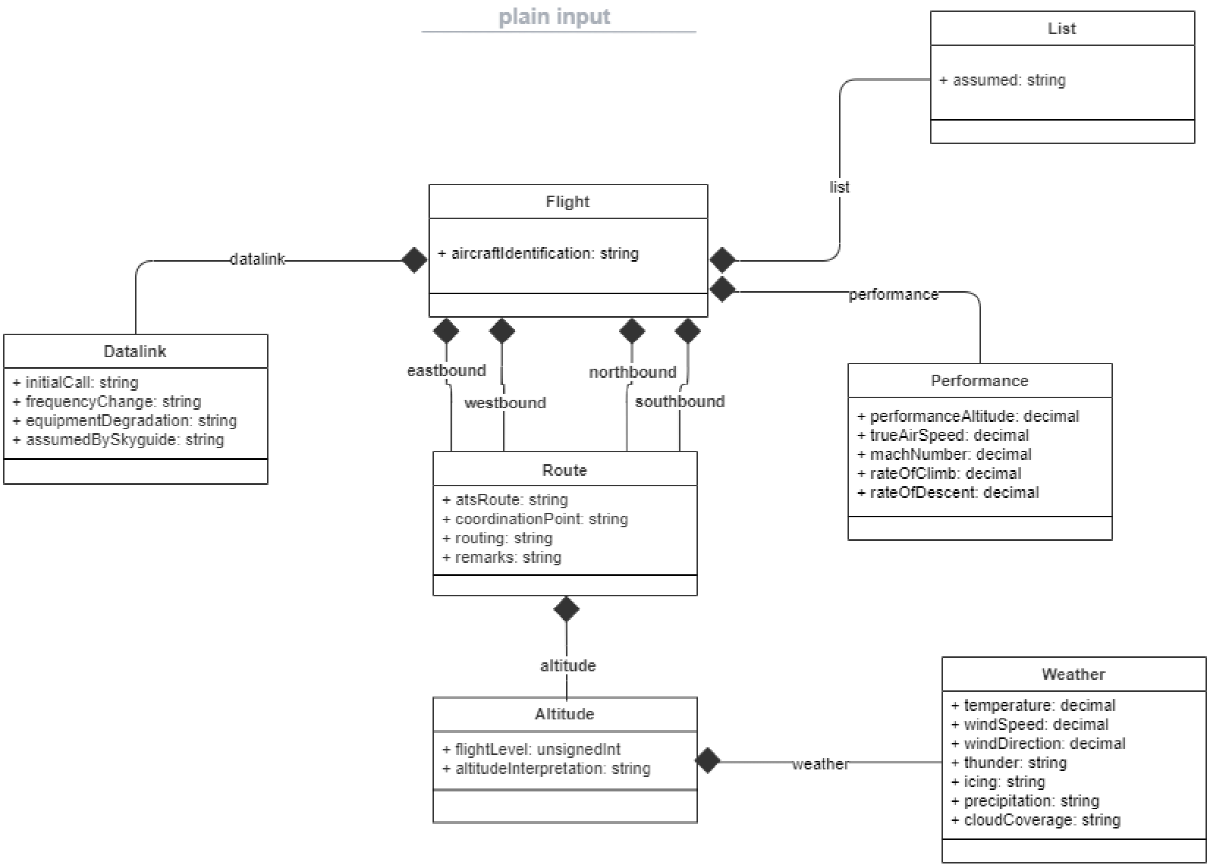


Figure 2. Plain input concept UML diagram

With a flight at the centre of it, the diagram branches out to describe other relevant data. This is still a concept and is susceptible to changes, such as delegation of a list of assumed aircraft to the AI itself, without the need to code it in. The example below is the plain data instance.

```
file:flight
        a plain:Flight ;
        plain:aircraftIdentification [ rdf:value "AA15263566" ] ;
        plain:eastbound file:eastbound ;
        plain:westbound file:westbound ;
        plain:northbound file:northbound ;
        plain:southbound file:southbound ;
        plain:datalink file:datalink ;
        plain:performance file:perf370 ;
        plain:list file:list .
file:westbound
        a plain:Route ;
        plain:atsRoute [ rdf:value "Z57" ] ;
        plain:coordinationPoint [ rdf:value "GUDAX" ] ;
        plain:routing [ rdf:value "GUDAX DOPIL ERMUS" ] ;
        plain:remarks [ rdf:value "ADES LSZH during Mil OFF, In case of subsequent
arrivals non-standard FL may be used, Not available for flights ADEP LSGG to ADES
LSZH, GENEVA ACC may clear flights DCT DOPIL provided it remains within the release
box limits" ] ;
        plain:altitude file:alt370 .
file:alt370
        a plain:Altitude ;
        plain:flightLevel [ plain:uom "FL" ; rdf:value "370" ] ;
        plain:altitudeInterpretation [ rdf:value "AT" ] ;
        plain:weather file:weather .
file:weather
        a plain:Weather ;
        plain:temperature [ plain:uom "CELSIUS" ; rdf:value "-57"^^xsd:decimal ] ;
        plain:windSpeed [ plain:uom "MILES_PER_HOUR" ; rdf:value "55"^^xsd:decimal ];
        plain:windDirection [ plain:uom "DEGREES" ; rdf:value "000" ] ;
        plain:thunder [ rdf:value "NO" ] ;
        plain:icing [ rdf:value "NO" ] ;
        plain:precipitation [ rdf:value "NO" ] ;
        plain:cloudCoverage [ rdf:value "SKC" ] .
file:perf370
        a plain:Performance ;
        plain:performanceAltitude [uom:FL ; rdf:value "370" ];
        plain:trueAirSpeed [plain:uom "KNOTS" ; rdf:value "447"] ;
        plain:machNumber [plain:uom "MACH" ; rdf:value "0.78"] ;
        plain:rateOfClimb [plain:uom "FEET PER MINUTE" ; rdf:value: "405" ] ;
        plain:rateOfDescent [plain:uom "FEET PER MINUTE" ; rdf:value "2922"] .
file:datalink
        a plain:Datalink ;
        plain:initialCall [ rdf:value "YES" ] ;
        plain:frequencyChange [ rdf:value "NO" ] ;
        plain:equipmentDegradation [ rdf:value "NO" ] ;
        plain:assumedBySykguide [ rdf:value "YES" ] .
file:list
        a plain:List ;
        plain:assumed [ rdf:value "YES" ] .
```

Used attributes:

- plain:Flight -> plain:aircraftIdentification – “Name used by ATS units to identify and communicate with an aircraft” [4].
- plain:Route -> plain:atsRoute – “The coded designator for a published ATS route or route segment” [4].
- plain:Route -> plain:coordinationPoint – Point on a route at which coordination occurs.
- plain:Route -> plain:routing – A sequence of a flight containing the coordination point.
- plain:Route -> plain:remarks – “Free text remarks” [3].
- plain:Altitude -> plain:flightLevel - “Current aircraft's altitude at standard air pressure, expressed in hundreds of feet” [12].
- plain:Altitude -> plain:altitudeInterpretation - Indicates whether the altitude (FL) is interpreted as AT, AT OR ABOVE or AT OR BELOW altitude (FL).
- plain:Weather -> plain:temperature – “The value of a temperature” [3].
- plain:Weather -> plain:windspeed – “The speed of the wind” [18].
- plain:Weather -> plain:windDirection – “Direction from which wind is blowing” [18].
- plain:Weather -> plain:thunder – Indicates whether thunder is present.
- plain:Weather -> plain:icing – Indicates if icing conditions are met.
- plain:Weather -> plain:precipitation – Indicates whether there is precipitation present.

- plain:Weather -> plain:cloudCoverage – Indicates the type of cloud cover conditions, such as SKC (sky clear) and OVC (overcast).
- plain:Datalink -> plain:initialCall – An indication of whether an initial call has been made via datalink.
- plain:Datalink -> plain:frequencyChange – An indication of whether a frequency change has been made via datalink.
- plain:Datalink -> plain:equipmentDegradation - An indication of whether equipment degradation has been mentioned via datalink.
- plain:Datalink -> plain:assumedBySykguide - An indication of whether the aircraft has been assumed by Skyguide.
- plain:List -> plain:assumed – Indicates whether the aircraft has been assumed and adds it to the list of assumed aircraft.
- plain:Performance -> plain:performanceAltitude – Altitude on which aircraft performances are based on.
- plain:Performance -> plain:trueAirSpeed – "The Airspeed type represents any airspeed measurement, in metric, imperial, or Mach units, as specified by the *uom* attribute" [4].
- plain:Performance -> plain:machNumber – "A dimensionless quantity in fluid dynamics representing the ratio of flow velocity past a boundary to the local speed of sound" [19].
- plain:Performance -> plain:rateOfClimb – "An aircraft's vertical speed – the positive rate of altitude change with respect to time" [20].
- plain:Performance -> plain:rateOfDescent – "An aircraft's vertical speed – the negative rate of altitude change with respect to time" [20].

5.10 CONFLICT_DETECTION_MODULE.ttl

This RDF instance contains inputs and outputs of the Conflict detection module developed for the AISA project, which serve as one of the inputs for the knowledge graph. Figure 3 is a diagram used to connect the various pieces of information in the conflict detection module and store the triples in the knowledge graph.

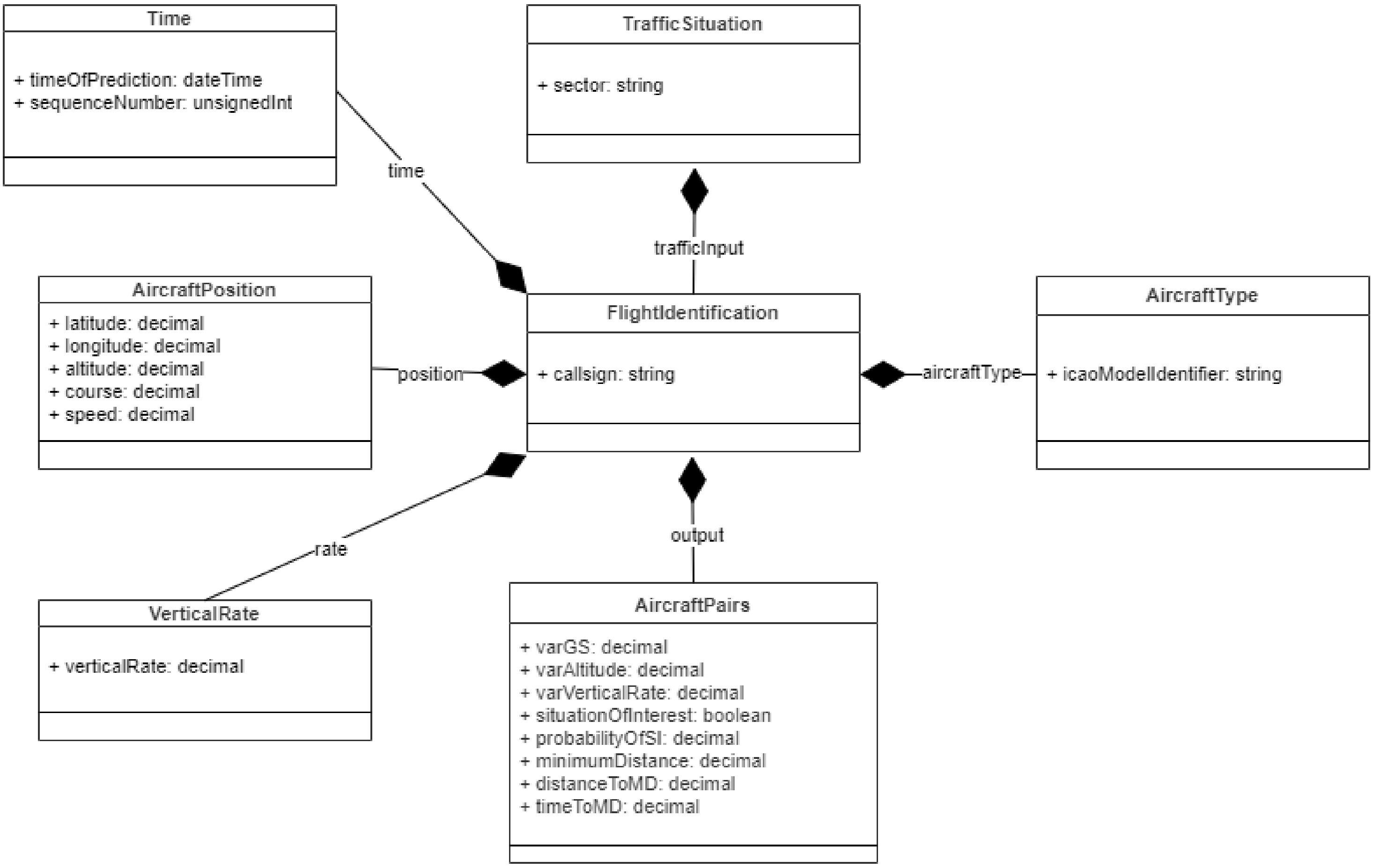


Figure 3. Conflict detection module UML diagram

The RDF instance below is an example of the conflict detection module input and output for two flights in LSAZM567 sector.

```
file:trafficSituation
       a plain:TrafficSituation ;
       plain:sector [rdf:value "LSAZM567" ] ;
       plain:trafficInput file:acft1, file:acft2 .
file:acft1
       a plain:FlightIdentification ;
       plain:callsign [ rdf:value "A6CAS" ] ;
       plain:time file:time1 ;
       plain:aircraftType file:type1 ;
       plain:position file:pos1 ;
       plain:rate file:rate1 ;
       plain:output file:output12 .
file:time1
       a plain:Time ;
       plain:timeOfPrediction [rdf:value "2019-07-04T12:18:00.000Z"^^xsd:dateTime] ;
       plain:sequenceNumber [rdf:value 1 ] .
file:type1
       a plain:AicraftType ;
       plain:icaoModelIdentifier [rdf:value "A320" ] .
file:pos1
       a plain:AircraftPosition ;
       plain:latitude [ rdf:value "47,134746"^^xsd:decimal ] ;
       plain:longitude [ rdf:value "9,146963"^^xsd:decimal ] ;
       plain:altitude [ plain:uom "FL" ; rdf:value "390"^^xsd:decimal ] ;
       plain:speed [ plain:uom "KNOTS" ; rdf:value "470"^^xsd:decimal ] ;
       plain:track [plain:uom "DEGREES" ; rdf:value "121,591516"^^wsd:decimal].
file:rate1
       a plain:VerticalRate ;
       plain:verticalRate [ plain:uom "FEET_PER_MINUTE ; rddf:value "0" ] .
file:acft2
       a plain:Aircraft ;
       plain:callsign [ rdf:value "TOM4BA" ] ;
       plain:aircraftType file:type2 ;
       plain:position file:pos2 ;
       plain:rate file:rate2 ;
       plain:output file:output12 .
file:type2
       a plain:AircraftType ;
       plain:icaoModelIdentifier [rdf:value "B734" ] .
file:pos2
       a plain:AircraftPosition ;
       plain:latitude [ rdf:value "46,656049"^^xsd:decimal ] ;
       plain:longitude [ rdf:value "10,291649"^^xsd:decimal ] ;
       plain:altitude [ plain:uom "FL" ; rdf:value "390"^^xsd:decimal ] ;
       plain:track [plain:uom ; rdf:value "296,689472"] ;
       plain:speed [ plain:uom "KNOTS" ; rdf:value "412"^^xsd:decimal ] .
file:rate2
       a plain:VerticalRate ;
       plain:verticalRate [ plain:uom "FEET_PER_MINUTE" ; rdf:value "0" ] .
file:output12
       a plain:AircraftPairs ;
       plain:varGS [ plain:uom "METERS_PER_SECOND" ; rdf:value "0"^^xsd:decimal ] ;
       plain:varAltitude [ plain:uom "FL" ; rdf:value "0"^^xsd:decimal] ;
       plain:varVerticalRate [ plain:uom "FEET_PER_MINUTE" ; rdf:value
"0"^^xsd:decimal ] ;
       plain:situationOfInterest [rdf:value "1"^^xsd:boolean];
       plain:probabilityOfSI [rdf:value "1"^^xsd:decimal ] ;
       plain:minimumDistance [plain:uom "NM" ; rdf:value "1,21"^^xsd:decimal ] ;
       plain:distanceToMD [plain:uom "NM" ; rdf:value "28,8"^^xsd:decimal ] ;
       plain:timeToMD [plain:uom "SECONDS" ; rdf:value "220"^^xsd:decimal ] .
```

Used attributes:

- plain:TrafficSituation -> plain:sector – “A subdivision of the airspace” [4].
- plain:FlightIdentification -> plain:callsign – “A group of alphanumeric characters used to identify an aircraft in air-ground communication” [18].
- plain:Time -> plain:timeOfPrediction – The time stamp of the prediction.
- plain:Time -> plain:sequenceNumber - “Used as unique key for the identification of the Time Slice concerned” [3].
- plain:AircraftPosition -> plain:latitude – “The angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth” [21].
- plain:AircraftPosition -> plain:longitude – “The angle east or west of a reference meridian to another meridian that passes through that point” [21].
- plain:AircraftPosition -> plain:altitude – “A mean of static pressure readings” [22].
- plain:AircraftPosition -> plain:track – “The direction the aircraft is flying, over the ground, relative to true north. It is the heading of the aircraft as impacted by the wind” [4].

- plain:AircraftPosition -> plain:speed – “An Indicated Air Speed (IAS) that can be maintained by the aircraft” [3].
- plain:AircraftType -> plain:icaoModelIdentifier – Alphanumerical code describing the model of the aircraft.
- plain:VerticalRate -> plain:verticalRate – “An expression of an aircraft's vertical rate of change (climb if positive, descent if negative)” [4].
- plain:AircraftPairs -> plain:varGS – “Ground speed variation between an aircraft pair” [7].
- plain:aircraftPairs -> plain:varAltitude – “The altitude difference between an aircraft pair” [7].
- plain:aircraftPairs -> plain:varVerticalRate – “The vertical-rate variation between an aircraft pair” [7].
- plain:aircraftPairs -> plain:situationOfInterest – “When an aircraft pair is expected to intersect with a horizontal separation lower than a pre-defined separation and infringe the vertical separation minima” [7]. The determined separation is chosen to be 10 NM horizontally and 1000 ft vertically.
- plain:aircraftPairs -> plain:probabilityOfSI – “A numerical variable that provides information about the prediction probability. This variable is calculated based on ML algorithms and represents a confidence value of the prediction” [7].
- plain:aircraftPairs -> plain:minimumDistance – “A numerical variable that provides information about the minimum separation calculated for each aircraft pair” [7].
- plain:aircraftPairs -> plain:distanceToMD – “A numerical variable that provides information about the distance to reach the minimum separation calculated for each aircraft pair” [7].
- plain:aircraftPairs -> plain:timeToMD – “A numerical variable that provides information about the time to reach the minimum separation calculated for each aircraft pair” [7].

5.11 COMPLEXITY_ESTIMATION_MODULE.ttl

A complexity estimation module has also been developed under Work Package 3 (WP 3) and is yet another input for the KG. Aircraft data is used as an input while an output gives an idea of a potential conflict that affects the complexity. Figure 4 shows a diagram depicting the module.

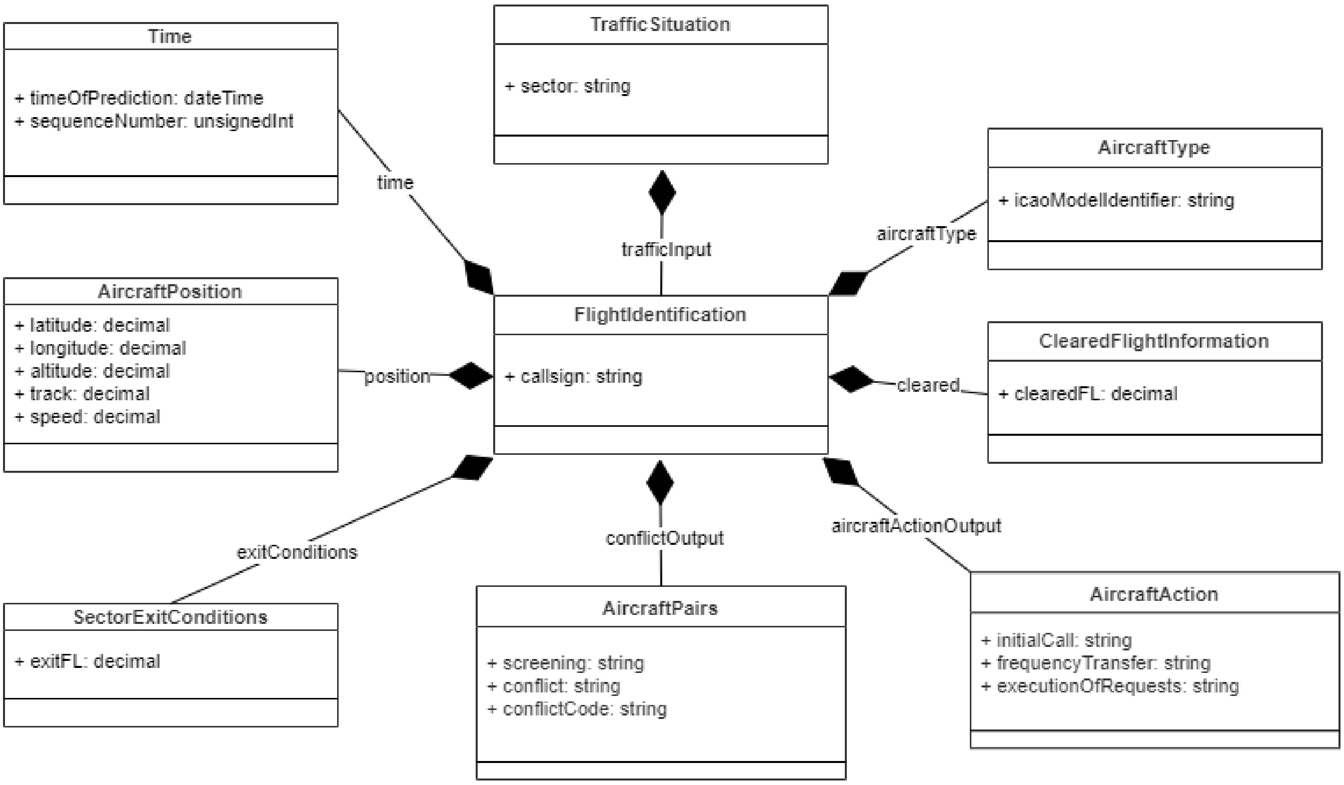


Figure 4. Complexity estimation module UML diagram

Written below is an example of complexity estimation using an aircraft pair CTN4847 and DLH643.

```
file:trafficSituation
       a plain:TrafficSituation ;
       plain:sector [rdf:value "LSAZM567" ] ;
       plain:trafficInput plain:aircraft1, plain:aircraft2 .
file:aircraft1
       a plain:FlightIdentification ;
       plain:callsign [ rdf:value "CTN4847" ] ;
       plain:time file:time1 ;
       plain:aircraftType file:Type1 ;
       plain:position file:Pos1 ;
       plain:exitConditions file:ext1 ;
       plain:cleared file:clr1 ;
       plain:aircraftActionOutput file:acout1 ;
       plain:conflictOutput file:conf12 .
file:time1
       a plain:Time ;
       plain:timeOfPrediction [rdf:value "2019-07-04T12:25:00.000Z"^^xsd:dateTime] ;
plain:sequenceNumber [rdf:value 1 ] .
file:Type1
       a plain:AircraftType ;
       plain:icaoModelIdentifier [rdf:value "A320" ] .
file:pos1
       a plain:AircraftPosition ;
       plain:latitude [ rdf:value "44,9089"^^xsd:decimal ] ;
       plain:longitude [ rdf:value "14,2482"^^xsd:decimal ] ;
       plain:altitude [ plain:uom "FL";rdf:value "355"^^xsd:decimal ] ;
       plain:speed [ plain:uom "KNOTS" ; rdf:value "450"^^xsd:decimal ] ;
       plain:track [ plain:uom "DEGREES" ; rdf:value "111,5"^^xsd:decimal ] .
file:ext1
       a plain:SectorExitConditions ;
       plain:exitFL [ plain:uom "FL" ; rdf:value "355" ] .
```

```
file:clr1
       a plain:ClearedFlightInformation ;
       plain:clearedFL [ plain:uom "FL" ; rdf:value "355" ] .
file:acout1
       a plain:AircraftAction ;
       plain:initialCall [ rdf:value "NO" ] ;
       plain:frequencyTransfer [ rdf:value "NO" ] ;
       plain:executionOfRequests [ rdf:value "NO" ] .
file:aircraft2
       a plain:FlightIdentification ;
       plain:callsign [ rdf:value "DLH643" ] ;
       plain:aircraftType file:Type1 ;
       plain:position file:Pos2;
       plain:exitConditions file:ext2;
       plain:cleared file:clr2;
       plain:aircraftActionOutput file:acout2 ;
       plain:conflictOutput file:confl2 .
file:Type2
       a plain:AircraftType ;
       plain:icaoModelIdentifier [rdf:value "A320" ] .
file:pos2
       a plain:AircraftPosition ;
       plain:latitude [ rdf:value "45,1332"^^xsd:decimal ] ;
       plain:longitude [ rdf:value "14,0915"^^xsd:decimal ] ;
       plain:altitude [ plain:uom "FL"; rdf:value "355"^^xsd:decimal ] ;
       plain:speed [ plain:uom "KNOTS" ; rdf:value "475"^^xsd:decimal ] ;
       plain:track [ plain:uom "DEGREES" ; rdf:value "123,1"^^xsd:decimal ] .
file:ext2
       a plain:SectorExitConditions ;
       plain:exitFL [ plain:uom "FL" ; rdf:value "355" ] .
file:clr2
       a plain:ClearedFlightInformation ;
       plain:clearedFL [ plain:uom "FL" ; rdf:value "355" ] .
file:acout2
       a plain:AircraftAction ;
       plain:initialCall [ rdf:value "NO" ] ;
       plain:frequencyTransfer [ rdf:value "NO" ] ;
       plain:executionOfRequests [ rdf:value "NO" ] .
file:output12                     #CTN4847 with DLH643
       a plain:AircraftPairs ;
       plain:screening [ rdf:value "SI" ] ;
       plain:conflict [ rdf:value "YES" ] ;
       plain:conflictCode [ rdf:value "CS231442222211144" ] .
```

Used attributes:

- plain:TrafficSituation -> plain:sector – “A subdivision of the airspace” [4].
- plain:FlightIdentification -> plain:callsign – “A group of alphanumeric characters used to identify an aircraft in air-ground communication” [18].
- plain:Time -> plain:timeOfPrediction – The time stamp of the prediction.
- plain:Time -> plain:sequenceNumber - “Used as unique key for the identification of the Time Slice concerned” [3].
- plain:AircraftType -> plain:icaoModelIdentifier – Alphanumerical code describing the model of the aircraft.

- plain:AircraftPosition -> plain:latitude – “The angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth” [21].

- plain: AircraftPosition -> plain:longitude – “The angle east or west of a reference meridian to another meridian that passes through that point” [21].

- plain: AircraftPosition -> plain:altitude – “A mean of static pressure readings” [22].

- plain: AircraftPosition -> plain:speed – “An Indicated Air Speed (IAS) that can be maintained by the aircraft” [3].

- plain: AircraftPosition -> plain:track – “The direction the aircraft is flying, over the ground, relative to true north. It is the heading of the aircraft as impacted by the wind” [4].

- plain:SectorExitConditions -> plain:exitFL – “The flight level at which an aircraft will leave the Flight Information Region (FIR) or the control area” [24].

- plain:ClearedFlightInformation -> plain:clearedFL – “The flight level to or at which a flight has currently been cleared by ATC” [25].

- plain:AircraftAction -> plain:initialCall – “An aircraft is about to enter the controlled airspace and is 20 NM or less from the controlled airspace boundary” [26].

- plain:AircraftAction -> plain:frequencyTransfer – “The aircraft that is in the controlled airspace is 15 NM or less until exiting that airspace” [26].

- plain:AircraftAction -> plain:executionOfRequests – Indicates whether there is a difference between current FL and exit FL.

- plain:AircraftPairs -> plain:screening – Indicates whether there is interactive conflict screening (SI), non-interactive conflict screening (SN) or potential interactive conflict screening (SP).

- plain:AircraftPairs -> plain:conflict – Indication of whether there is a conflict between the aircraft pair.

- plain:AircraftPairs -> plain:conflictCode – A code specifying the type of conflict, clarified in deliverable D3.1 Air traffic complexity module [9].

5.12 4DT_PREDICTION_MODULE.ttl

A 4D trajectory prediction module was also developed under WP 3 and is also used as an input for the KG. Flight and trajectory data is used as an input while an output gives a predicted future position along with its probability and predicted altitude. Figure 5 below shows a diagram depicting the module. The RDF instance example below is a part of the trajectory prediction for one flight.

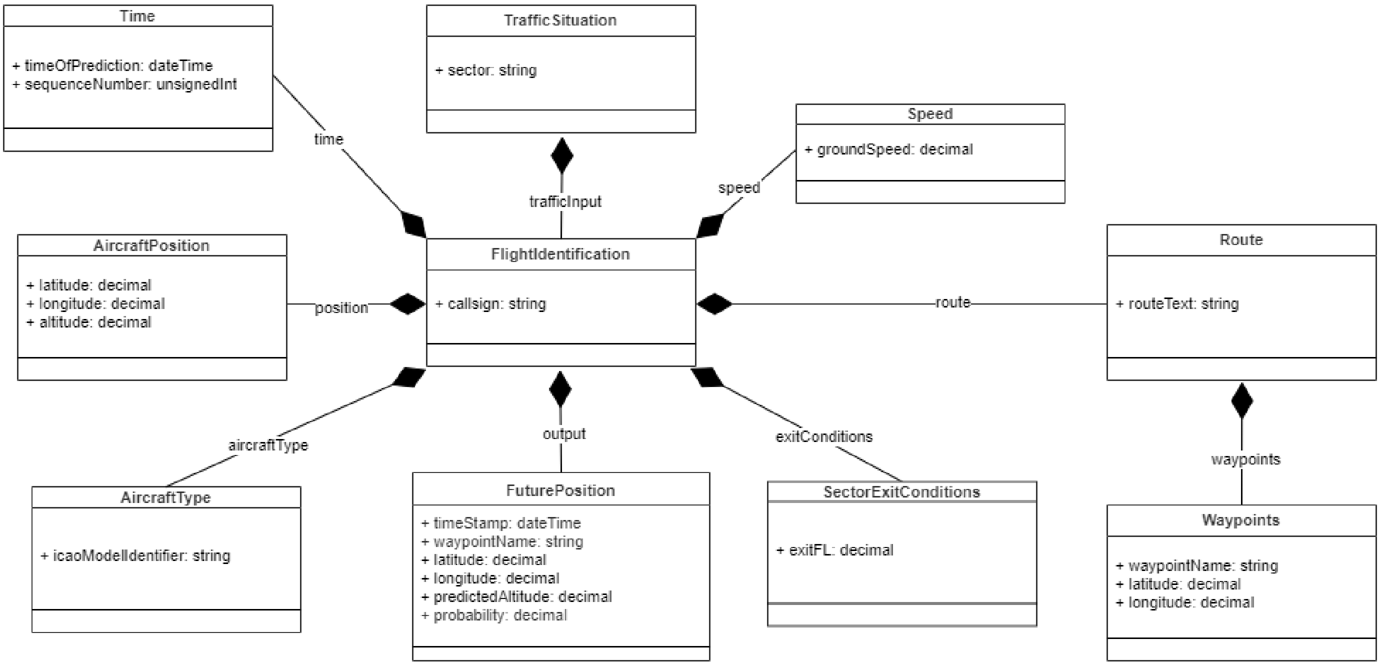


Figure 5 4DT prediction model UML diagram

```
file:trafficSituation
       a plain:TrafficSituation ;
       plain:sector [rdf:value "LSAZM567" ] ;
       plain:trafficInput file:flightIdentification .
file:flightIdentification
       a plain:FlightIdentification ;
       plain:callsign [ rdf:value "TOM51G" ] ;
       plain:time file:time1 ;
       plain:position file:aircraftPosition ;
       plain:aircraftType file:aircraftType ;
       plain:output file:futurePosition1, file:futurePosition2 ;
       plain:exitConditions file:sectorExitConditions ;
       plain:route file:route ;
       plain:speed file:speed .
file:time1
       a plain:Time ;
       plain:timeOfPrediction [rdf:value "2019-07-04T12:38:00.000Z"^^xsd:dateTime] ;
plain:sequenceNumber [rdf:value 1 ] .
file:aircraftPosition
       a plain:AircraftPosition ;
       plain:latitude [ rdf:value "47,05"^^xsd:decimal ] ;
       plain:longitude [ rdf:value "9,298056"^^xsd:decimal ] ;
       plain:altitude [ plain:uom "FL" ; rdf:value "380"^^xsd:decimal ] .
file:aircraftType
       a plain:AircraftType ;
       plain:icaoModelIdentifier [ rdf:value "B738" ] .
file:futurePosition1
       a plain:FuturePosition ;
       plain:timeStamp [rdf:value "2019-07-04T12:58:38.000Z"^^xsd:dateTime];
       plain:waypointName [ rdf:value "11x13" ] ;
       plain:latitude [ rdf:value "47,085696"^^xsd:decimal ] ;
       plain:longitude [ rdf:value "9,011392"^^xsd:decimal ] ;
       plain:predictedAltitude [ plain:uom "FL" ; rdf:value "380"^^xsd:decimal ] ;
       plain:probability [ rdf:value "0,222622"^^xsd:decimal ] .
file:futurePosition2
       a plain:FuturePosition ;
       plain:timeStamp [rdf:value "2019-07-04T12:59:38.000Z"^^xsd:dateTime];
```

```
        plain:waypointName [ rdf:value "12x12" ] ;
        plain:latitude [ rdf:value "47,251675"^^xsd:decimal ] ;
        plain:longitude [ rdf:value "8,76853"^^xsd:decimal ] ;
        plain:predictedAltitude [ plain:uom "FL" ; rdf:value "380"^^xsd:decimal ] ;
        plain:probability [ rdf:value "0,15019"^^xsd:decimal ] .
file:sectorExitConditions
        a plain:SectorExitConditions ;
        plain:exitFL [ plain:uom "FL" ; rdf:value "380"^^xsd:decimal ] .
file:route
        a plain:Route ;
        plain:routeText [ rdf:value "*4KFN LATAN RESIA ARGAX ELMUR MANEG RIPUS DITON
HOC NATLI LUL LUVAL KOTUN PODUK DIDOR NURMO DIPER VESAN RATUK SOVAT SANDY DET LAM
BPK POTON BEDFO EBOTO PIPIN LESTA EMBOR EVSON ADELU MAMUL" ] ;
        plain:waypoints file:waypoints1, file:waypoints2 .
file:waypoints1
        a plain:Waypoints ;
        plain:waypointName [ rdf:value "ARGAX" ] ;
        plain:latitude [ rdf:value "47,05"^^xsd:decimal ] ;
        plain:longitude [ rdf:value "9,298056"^^xsd:decimal ] .
file:waypoints2
        a plain:Waypoints ;
        plain:waypointName [ rdf:value "ELMUR" ] ;
        plain:latitude [ rdf:value "47,156667"^^xsd:decimal ] ;
        plain:longitude [ rdf:value "8,9075"^^xsd:decimal ] .
file:speed
        a plain:Speed ;
        plain:groundSpeed [ rdf:value "523"^^xsd:decimal ] .
```

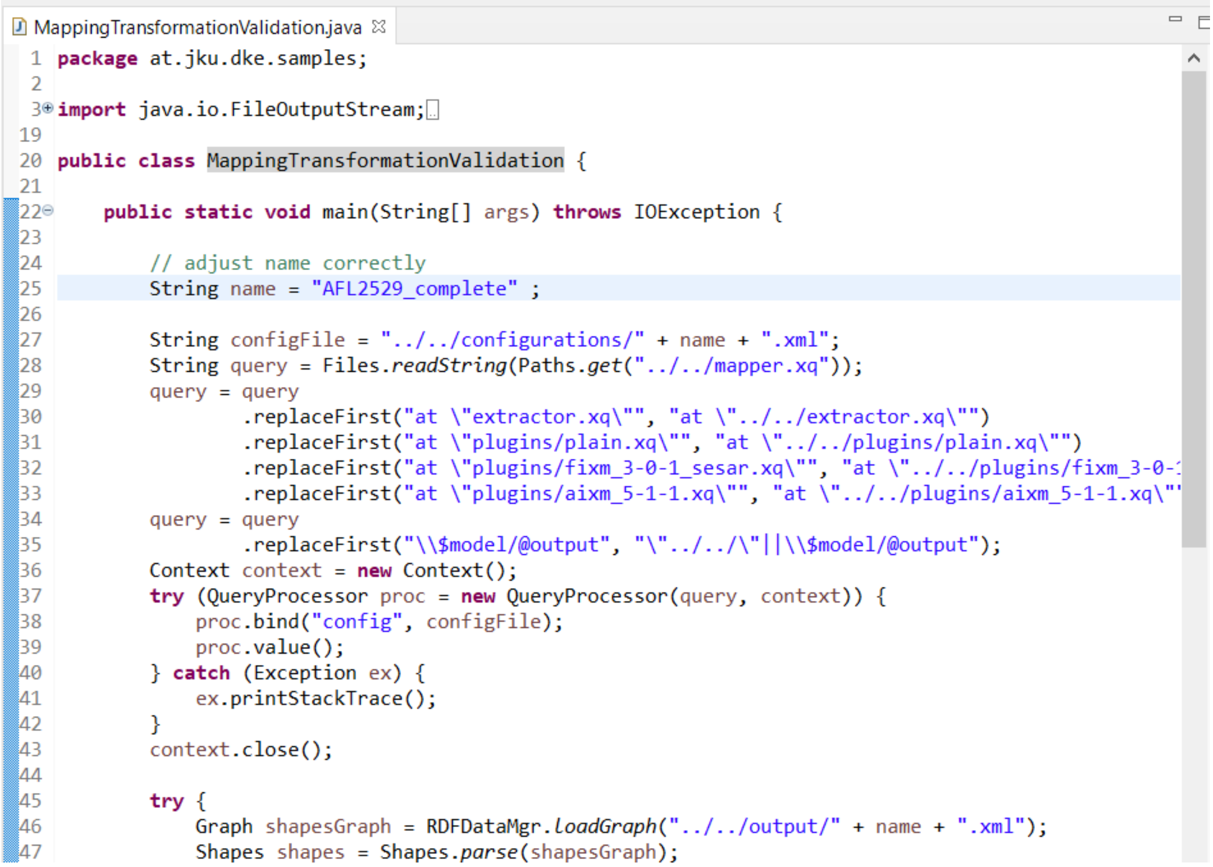
Used attributes:

- plain:TrafficSituation -> plain:sector – “A subdivision of the airspace” [4].
- plain:FlightIdentification -> plain:callsign – “A group of alphanumeric characters used to identify an aircraft in air-ground communication” [18].
- plain:Time -> plain:timeOfPrediction – The time stamp of the prediction.
- plain:Time -> plain:sequenceNumber - “Used as unique key for the identification of the Time Slice concerned” [3].
- plain:AircraftPosition -> plain:latitude – “The angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth” [21].
- plain:AircraftPosition -> plain:longitude – “The angle east or west of a reference meridian to another meridian that passes through that point” [21].
- plain:AircraftPosition -> plain:altitude – “A mean of static pressure readings” [22].
- plain:AircraftType -> plain:icaoModelIdentifier – Alphanumerical code describing the model of the aircraft.
- plain:FuturePosition -> plain:timestamp – A time in the future position that is calculated using aircraft distance and aircraft speed.
- plain:FuturePosition -> plain:waypointName – A name of the waypoint, usually a five-letter word.

- plain:FuturePosition -> plain:latitude – “The angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth” [21].
- plain:FuturePosition -> plain:longitude – “The angle east or west of a reference meridian to another meridian that passes through that point” [21].
- plain:FuturePosition -> plain:predictedAltitude – A prediction of the altitude based on the future position of the aircraft.
- plain:FuturePosition -> plain:probability – A probability that the aircraft will overfly the waypoint in question.
- plain:SectorExitConditions -> plain:exitFL – “The flight level at which an aircraft will leave the Flight Information Region (FIR) or the control area” [24].
- plain:Route -> plain:routeText – “The ICAO route string as depicted from the flight plan” [4].
- plain:Waypoints -> plain:waypointName – A name of the waypoint, usually a five-letter word.
- plain:Waypoints -> plain:latitude – “The angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth” [21].
- plain:Waypoints -> plain:longitude – “The angle east or west of a reference meridian to another meridian that passes through that point” [21].
- plain:Speed -> plain:groundSpeed – “Horizontal speed of an aircraft relative to the Earth's surface” [23].

6 Validation program

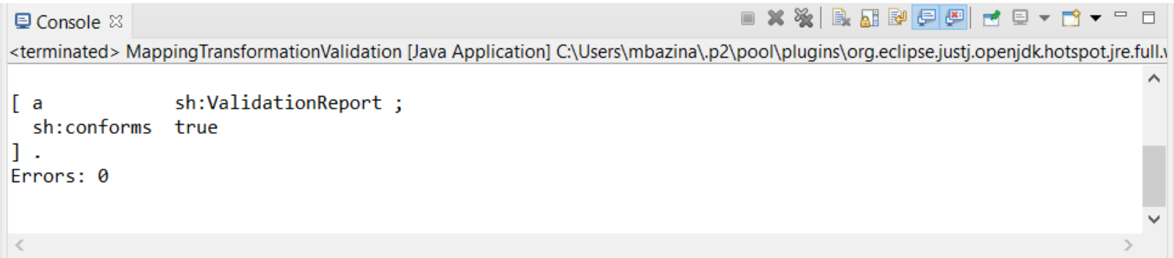
The validation program used was developed by a JKU (Johannes Kepler Universität Linz) team in Java and allows for the validation of the named graph written in turtle (ttl) format. It is used via the Eclipse program and combines the mapper, transformation of XML to ttl, and SHACL (Shapes Constraint Language) validation under a single program named MappingTransformationValidation.java. The version of the program used is Eclipse IDE (Integrated Development Environment) for Java Developers. To run MappingTransformationValidation.java it needs to be imported into Eclipse using the Project Explorer option.



```
 1 package at.jku.dke.samples;
 2
 3 import java.io.FileOutputStream;
19
20 public class MappingTransformationValidation {
21
22     public static void main(String[] args) throws IOException {
23
24         // adjust name correctly
25         String name = "AFL2529_complete" ;
26
27         String configFile = "../../configurations/" + name + ".xml";
28         String query = Files.readString(Paths.get("../../mapper.xq"));
29         query = query
30                 .replaceFirst("at \"extractor.xq\"", "at \"../../extractor.xq\"")
31                 .replaceFirst("at \"plugins/plain.xq\"", "at \"../../plugins/plain.xq\"")
32                 .replaceFirst("at \"plugins/fixm_3-0-1_sesar.xq\"", "at \"../../plugins/fixm_3-0-
33                 .replaceFirst("at \"plugins/aixm_5-1-1.xq\"", "at \"../../plugins/aixm_5-1-1.xq\""
34         query = query
35                 .replaceFirst("\\$model/@output", "\"../../\"||\\$model/@output");
36         Context context = new Context();
37         try (QueryProcessor proc = new QueryProcessor(query, context)) {
38             proc.bind("config", configFile);
39             proc.value();
40         } catch (Exception ex) {
41             ex.printStackTrace();
42         }
43         context.close();
44
45         try {
46             Graph shapesGraph = RDFDataMgr.loadGraph("../../output/" + name + ".xml");
47             Shapes shapes = Shapes.parse(shapesGraph);
```

Figure 6. Part of the MappingTransformationValidation.java program

When the name of the named graph we want to run is written under string name at line 25 (Figure 6), the program can run the validation code and provides us with the output in the true or false form.



```
<terminated> MappingTransformationValidation [Java Application] C:\Users\mbazina\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.
```

```
[ a            sh:ValidationReport ;
  sh:conforms  true
] .
Errors: 0
```

Figure 7. Console displaying that the code is true

References

[1] "What is a Knowledge Graph? | Ontotext Fundamentals," *Ontotext*. https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/ (accessed May 18, 2021).

[2] L. Ehrlinger and W. Wöß, "Towards a Definition of Knowledge Graphs," 2016.

[3] "AIXM." https://www.aixm.aero/ (accessed May 20, 2021).

[4] "FIXM." https://www.fixm.aero/ (accessed May 20, 2021).

[5] ICAO, *Procedures for air navigation and air traffic management pans-atm doc 4444.* ICAO, 2016.

[6] Lars L. Schmidt, "4D trajectory prediction module," 2021.

[7] Javier A. Pérez Castán, "Conflict detection module." 2021.

[8] "The OpenSky Network." https://opensky-network.org/ (accessed Jun. 15, 2021).

[9] I. Tukarić, "Airspace complexity module." 2021.

[10] "Estimated Off-Block Time - EUROCONTROL ATM Lexicon." https://ext.eurocontrol.int/lexicon/index.php/Estimated_Off-Block_Time (accessed May 28, 2021).

[11] "Flight Information Exchange Model Data Dictionary." Feb. 11, 2015.

[12] "Flight level," *Wikipedia*. May 26, 2021. Accessed: May 28, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Flight_level&oldid=1025213281

[13] "Geography Markup Language," *Wikipedia*. Jan. 07, 2021. Accessed: Jun. 18, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Geography_Markup_Language&oldid=998823234

[14] EUROCONTROL and FAA, "AIXM Temporality Model." 2019. [Online]. Available: https://aixm.aero/sites/aixm.aero/files/imce/AIXM511/aixm_temporality_1.1.pdf

[15] "Datypic." http://www.datypic.com (accessed Jun. 25, 2021).

[16] "Schema Documentation for GML." https://schemas.isotc211.org/19136/-/gml/1.0/gml/ (accessed Jun. 18, 2021).

[17] "SKYbrary Aviation Safety." https://www.skybrary.aero (accessed May 24, 2021).

[18] "The ATM Information Reference Model." https://airm.aero/index.html (accessed Jun. 22, 2021).

[19] "Mach number," *Wikipedia*. Jun. 20, 2021. Accessed: Jun. 24, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Mach_number&oldid=1029512835

[20] "Rate of climb," *Wikipedia*. Feb. 07, 2021. Accessed: Jun. 24, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Rate_of_climb&oldid=1005435163

[21] "Geographic coordinate system," *Wikipedia*. Jun. 23, 2021. Accessed: Jun. 24, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Geographic_coordinate_system&oldid=1030029338

[22] "Barometric Altimeter," *Basic Air Data*. https://www.basicairdata.eu/projects/barometric-altimeter/ (accessed Jun. 24, 2021).

[23] "Ground speed," *Wikipedia*. Oct. 29, 2020. Accessed: Jun. 25, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Ground_speed&oldid=985990818

[24] "Exit flight level - EUROCONTROL ATM Lexicon." https://ext.eurocontrol.int/lexicon/index.php/Exit_flight_level (accessed Jun. 24, 2021).

[25] "Cleared flight level - EUROCONTROL ATM Lexicon." https://ext.eurocontrol.int/lexicon/index.php/Cleared_flight_level (accessed Jun. 24, 2021).

[26] Bruno Antulov-Fantulin, "Air traffic complexity model based on air traffic controller tasks." 2020.

Appendix A Glossary

Abbreviation	Term
ADS-B	Automatic Dependent Surveillance-Broadcast
AIP	Aeronautical Information Publication
AIRM	Air Traffic Management Information Reference Model
AIS	Aeronautical Information Services
AIXM	Aeronautical Information Exchange Model
ATC	Air Traffic Control
ATCO	Air Traffic Control Officer
ATM	Air Traffic Management
ATS	Air Traffic Service
ATSU	Air Traffic Services Unit
DME	Distance Measuring Equipment
EFPL	Extended Flight Plan
FAA	Federal Aviation Administration
FF-ICE	Flight and Flow Information for a Collaborative Environment
FIR	Flight Information Region
FIXM	Flight Information Exchange Model
FL	Flight Level
FLAS	Flight Level Allocation Scheme
FTTS	Sveučilište u Zagrebu Fakultet prometnih znanosti
GML	Geography Markup Language
IAS	Indicated Air Speed
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
ILS	Instrument Landing System
ISO	International Organization for Standardization
JKU	Johannes Kepler Universität Linz
KG	Knowledge Graph
LoA	Letters of Agreement
LOAA	Learn Once Apply Anywhere
ML	Machine Learning
MLS	Microwave Landing System
MSL	Mean Sea Level

NDB	Non-Directional Beacon
NEST	Network strategic tool
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RMSE	Root Mean Squared Error
RVR	Runway Visual Range
SHACL	Shapes Constraint Language
SI	Situation of Interest
TACAN	Tactical Air Navigation System
TTL	Turtle
TUBS	Technische Universität Braunschweig
UML	Unified Modeling Language
UPM	Universidad Politécnica de Madrid
UTC	Coordinated Universal Time
VOR	Very High Frequency Omni-Directional Range
WXXM	Weather Information Exchange Model
XML	Extensible Markup Language

Appendix B ML module as input in KG

This is an example of how the ML module will serve as an input for the KG. The task at hand is 5.3 which was defined under D2.2 Requirements for automation of monitoring tasks via AI SA. Task 5.3 reads as follows: Check which conflicts are to occur within the sector. The requirement for this task is: The system shall infer whether the conflict will occur within the sector. Given the task and requirements, we can safely conclude that the Conflict detection module will be used. A schema given below (Figure 8) shows how it could be implemented.

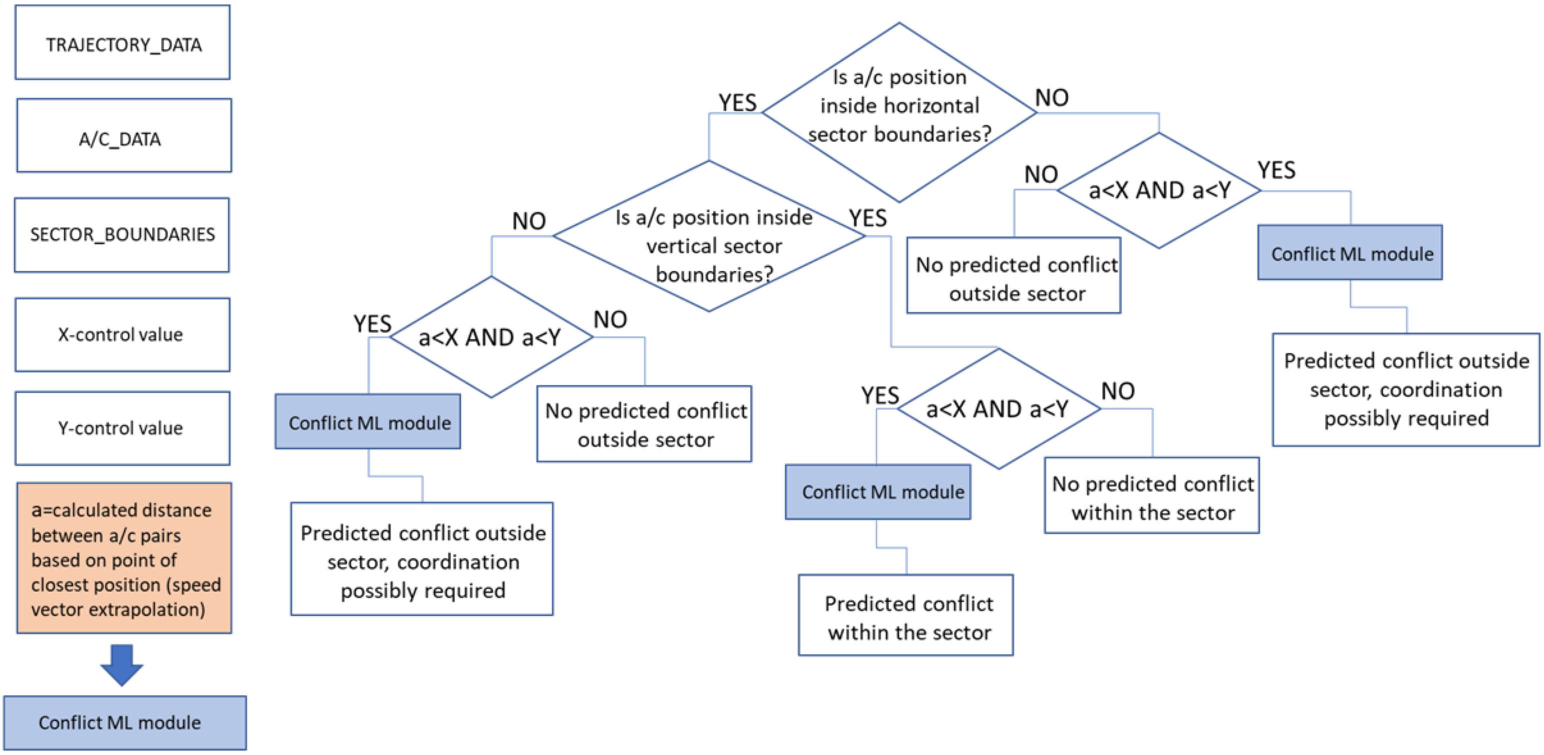

TRAJECTORY_DATA
A/C_DATA
SECTOR_BOUNDARIES
X-control value
Y-control value
a=calculated distance between a/c pairs based on point of closest position (speed vector extrapolation)
Conflict ML module
Is a/c position inside horizontal sector boundaries?
YES
NO
Is a/c position inside vertical sector boundaries?
NO
YES
a<X AND a<Y
YES
NO
Conflict ML module
No predicted conflict outside sector
Predicted conflict outside sector, coordination possibly required
a<X AND a<Y
YES
NO
Conflict ML module
No predicted conflict within the sector
Predicted conflict within the sector
a<X AND a<Y
NO
YES
No predicted conflict outside sector
Conflict ML module
Predicted conflict outside sector, coordination possibly required

Figure 8. Reasoning for task 5.3