Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 64 (2022) 328–336

Transportation Research Procedia

www.elsevier.com/locate/procedia

International Scientific Conference "The Science and Development of Transport - Znanost i razvitak prometa"

Creating an ATC knowledge graph in support of the artificial situational awareness system

Michael Schrefl *[a], Bernd Neumayr[a], Sebastian Gruber[a], Marlene Hartmann[a], Ivan Tukarić *[b], Tomislav Radišić[b]

[a] *Institute of Business Informatics - Data and Knowledge Engineering, Johannes Kepler University of Linz, Linz, Austria*
[b] *Department of Aeronautics, Faculty of Transport and Traffic Sciences, University of Zagreb, Zagreb, Croatia*

Abstract

Automation has been recognized as a possible solution for increasing air traffic controller workload trends. This paper presents a methodology for creating an air traffic control knowledge graph, which is used as part of a hybrid artificial intelligence system for air traffic control operations. The system combines machine learning and symbolic reasoning with the purpose of achieving artificial situational awareness in a narrow domain of en-route air traffic control operations. This approach allows the use of user-defined knowledge alongside existing knowledge repositories. The novel knowledge graph development methodology is universal for any area of air traffic management which relies on the aeronautical information exchange models. In this paper we also present the open-source tools which were developed to make this approach possible and system performance evaluations. Future work should address achieving real-time operation and additional task automation, accompanied by appropriate ontology and graph expansion.

Peer-review under responsibility of the scientific committee of the International Scientific Conference „The Science and Development of Transport - Znanost i razvitak prometa –ZIRP2022

Keywords: Knowledge graph; air traffic control; AIXM; FIXM; artificial situational awareness; hybrid system

1. Introduction

Automation is often touted as a solution for the capacity problem in air traffic management (ATM) SESAR Joint Undertaking (2019). Up to 2020, European air traffic was steadily increasing with many of the air navigation service providers (ANSP) creating high levels of air traffic flow management (ATFM) delay EUROCONTROL Performance Review Commission (2020). Although some pandemic-induced changes in travel behaviors will undoubtedly remain

* Corresponding authors:
E-mail address: schrefl@dke.uni-linz.ac.at, itukaric@fpz.unizg.hr

Peer-review under responsibility of the scientific committee of the International Scientific Conference „The Science and Development of Transport - Znanost i razvitak prometa –ZIRP2022
10.1016/j.trpro.2022.09.037

permanent, traffic is expected to recover in medium-term EUROCONTROL (2021). It is certain therefore that the capacity issues will re-appear sooner or later which will push all the efforts to automate ATM services into focus again.

The issue with machine learning (ML) approach to automation is that some very often used methods, such as neural networks, lack transparency and explainability. On the other hand, older methods such as predicate systems or rule-based systems are often unable to learn. Hybrid systems combining neural architectures with symbolic reasoning might be able to overcome or at least mitigate these deficiencies.

SESAR Exploratory Research project AISA - AI Situational Awareness Foundation for Advancing Automation aims to explore such hybrid systems to create a situationally aware air traffic control (ATC) system which could be a basis for introducing more automation to ATM AISA Project Consortium (2021a).

This approach combines a reasoning engine employing first-order logic and rule-based reasoning with a machine learning approach for prediction and estimation. ML modules are used at a lower level to predict small-scale probabilistic events (e.g., aircraft conflict detection), while the inference mechanism is used at a higher level to draw conclusions from the state of the system. By combining the reasoning mechanism with ML, we believe it will be possible for AI to assess complex interactions between objects, draw conclusions, explain reasoning, and predict future system states AISA Project Consortium (2021a).

To enable research on such hybrid systems, it is first necessary to develop a framework to represent knowledge specific to ATC (i.e., a domain-specific knowledge graph). The knowledge graph (KG), in this context, should not be seen as just another form of data encoding- it is also a representation of all attributes, rules, relationships, axioms, etc. in the ATC field. Its purpose is to be used as a basis for drawing conclusions about the state of the system and gaining new knowledge, both at the level of individual components and as a whole AISA Project Consortium (2021a).

Creating a KG, even for a very narrow domain (e.g., in AISA for en-route ATC), is a very time-consuming effort. It is sometimes casually stated that as a rule of thumb '10,000 pairs of hands' are needed to create a viable KG system Sheth (2019). Even if such estimates are an order of magnitude too pessimistic, the effort needed is still prohibitively high. The issue is furthermore compounded by the dynamic nature of ATC operations, where the traffic situation changes at every moment creating a traffic situation unique to that specific time and airspace. Both of these issues are highlighted in the article describing the development of a NASA/FAA KG for ATM Keller (2019).

Fortunately, in the aviation domain there is a comprehensive effort underway to enable modern standardized exchange of information. For example, Aeronautical Information Exchange Model (AIXM) is designed to provide, in digital format, the aeronautical information that is in the scope of Aeronautical Information Services (AIS) EUROCONTROL and FAA (2022a). Similarly, Flight Information Exchange Model (FIXM) is a global exchange standard capturing Flight and Flow information EUROCONTROL and FAA (2022b). These standards are made of logical models, defined in the Unified Modelling Language (UML), which capture all the constructs needed to exchange aeronautical data. An example of an AIXM UML model is shown in Fig. 1 by Porosnicu (2021).

Such constructs are also programmatically converted to a physical representation in Extensible Markup Language (XML) – the Schema Definition (XSD). Most importantly, with appropriate treatment, schemas defined in AIXM or FIXM can be repurposed to build the ATC knowledge graph, thus significantly cutting down on the effort needed to create the basis of the hybrid system.

The possibility of building ontologies based on AIXM, FIXM, and other exchange models has been demonstrated by Egami et al. (2020). The authors describe the creation of ontologies as well as simple tests and query performance assessments. While the test results show excellent query run-times, their limited nature needs to be taken into account. Moreover, domain and "common" ontologies described in their work do not include knowledge defined outside of the exchange models.

In this paper, an overview of the methodology used to create the ATC knowledge graph will be presented. Unlike existing research, the approach expands AIXM and FIXM knowledge with knowledge identified by subject matter experts to be necessary for ATCOs to achieve situational awareness. The resulting knowledge graph is flexible in regards to accepting knowledge related to additional system functions. Along with the broader function of studying KGs in ATM, the KG plays an important role in the proof-of-concept system as explained in the following chapters. Open-source tools which were developed to enable the system will be shown as well (with links to github repositories). Since AISA is an ongoing project, final results are not yet available, however, initial system tests were performed and presented herein.

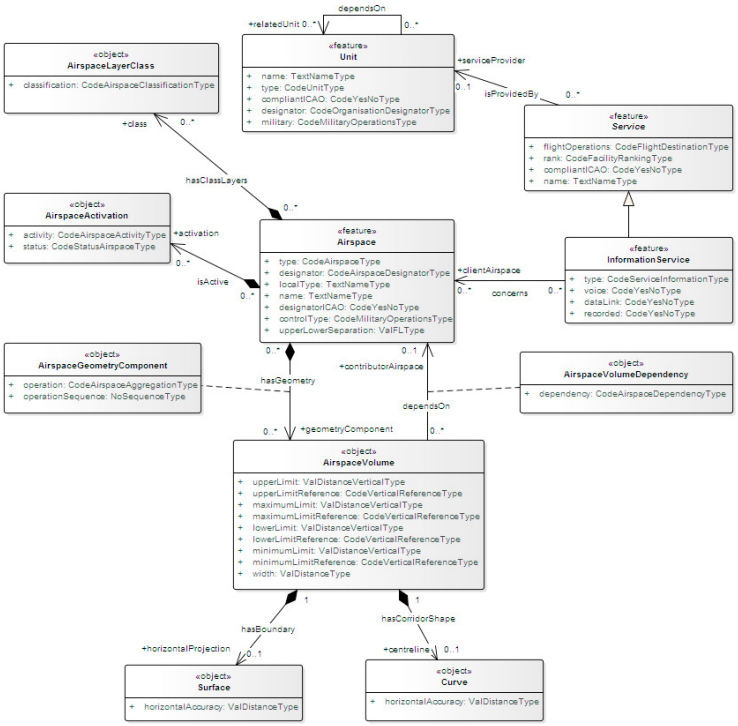

Fig. 1. AIXM UML model for the ”Airspace” concept Porosnicu (2021)

2. ATC Knowledge Graph System

In this section, an overview of the ATC knowledge graph system as defined in the AISA project will be given. The purpose of the KG system, within the scope of the project, is to create a system with artificial situational awareness to be used as a foundation for implementing further automation projects. As defined by Endsley, *‘Situational awareness or situation awareness (SA) is the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their future status’* Endsley (1995).

Three key concepts are mentioned in this definition, namely, perception, comprehension of meaning, and projection of future status. Out of the three, in principle, KG-based systems are only useful in terms of providing the necessary semantics (meaning). Perception is accomplished by sensors which, in ATM, are filtered and checked for integrity before the data is fed into any of the ATM systems. Projection of future status is not something that KG-based systems excel at, so the system needs to be expanded with this capability via other means, such as ML-based sub-systems.

In Fig. 2, the overall architecture of the AISA project-level system is presented (numbers in the parentheses correspond to the numbers in Fig. 2). The traffic and airspace data (1) obtained from the OpenSky Network Schäfer et al. (2014) and EUROCONTROL’s DDR2 is converted (2) to Resource Description Framework (RDF) models. On the other hand, AIXM, FIXM, and other (3) vocabulary sources are used to create an RDF schema and Shape Constraint Language (SHACL) constraints (4). Afterwards, the traffic and airspace RDF data is checked for conformance against the RDF schema (RDFS) and SHACL constraints (5). The data is then fed into the knowledge graph (6) by creating a single RDF model for each timestamp of a traffic scenario.

The same sources of aeronautical data as those used to populate the knowledge graph are also used to train the ML modules (7). In all, there are three ML modules in the proof-of-concept system, one each for trajectory prediction, conflict detection, and complexity. They are used to make predictions or estimate probabilistic events/values. Their use fulfills the ‘projection of future status’ prerequisite for situational awareness, as mentioned previously. Both results of the ML modules and the modules’ meta-data (describing the training set and model performance) are fed into the knowledge graph, alongside the traffic data.

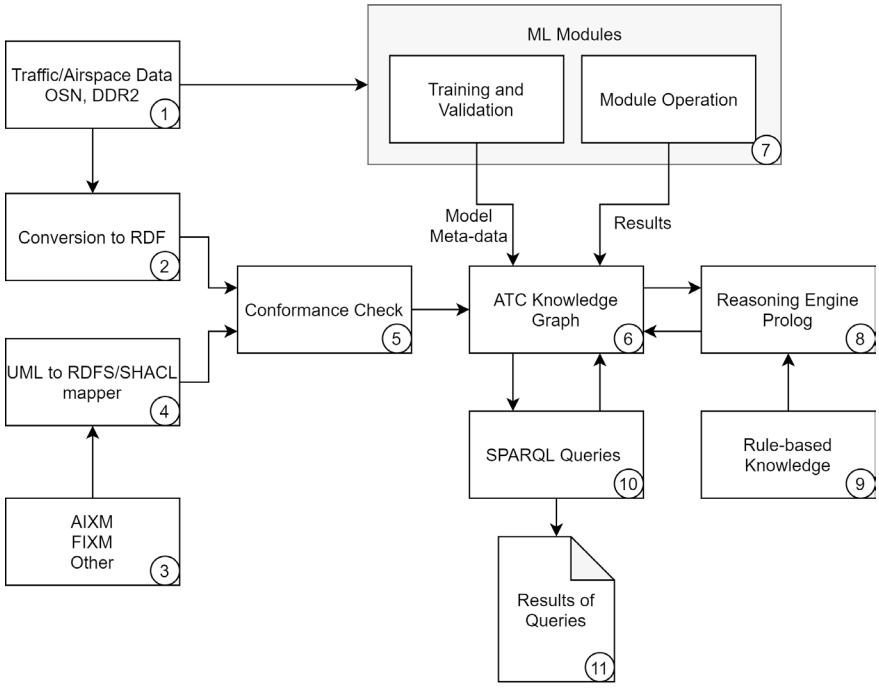

Fig. 2. The Architecture of the AISA System

For advanced reasoning over the knowledge graph, either Prolog (8) or Java programs can be used. They use rules (9) which encode air traffic controller's (ATCO) knowledge about the procedures in a specific airspace. Since the graphs are in RDF format, RDF query language SPARQL is used to query (10) the KG in order to access data, which is then used to provide conclusions about the traffic situation. KG can be queried about any possible aspect of a traffic situation if it was properly encoded in the graph. For the purpose of this project queries are mostly used to automate the ATCO's monitoring tasks, with several tasks aimed at monitoring the status of ML modules. Results of the queries can be displayed to the operator for use or analysis (11).

3. UML to RDFS/SHACL Mapper

UML to RDFS/SHACL Mapper (available at: https://github.com/jku-win-dke/AISA-XMI-Mapper/) is used to convert the selected classes between UML Class Diagrams and RDF Schema (RDFS)/Shape Constraint Language (SHACL) documents. RDFS defines the required vocabulary of the domain described by UML class diagrams, that is, classes and class hierarchies. SHACL, on the other hand, defines the structural constraints of the domain, such as types and value ranges of certain values. This means that RDFS defines concepts such as 'Flight' and its attributes (e.g., 'callsign', 'flight level'), while SHACL checks if 'flight level' value is defined as a number (and not as text).

The mapper was specifically created to map AIXM and FIXM models that fit a particular modeling style; however, this does not mean that other models cannot be mapped. It just means that the models provided to the mapper must meet certain semantic and syntactic requirements. By default, UML models (such as those describing the classes missing from the aeronautical models) must comply with these semantic requirements AISA Project Consortium (2021b):

- Names of classes must be uniquely present in a model. Each model can have a class of the same name as another model (e.g., both AIXM and FIXM having a class called Route) but two different UML classes called the same in one model (e.g., two Route classes within FIXM) are not allowed.
- Models should only contain directed associations.

- Similarly to class names, role names (at the target) of associations with the same source class must be unique within the source class.
- Role names must be present in cases where there is more than one association between classes (source and target classes). In cases with only one association and no role name provided, the role name will be defined using the name of the target class.

Only the first requirement is validated by the UML to RDFS/SHACL Mapper and throws an exception if violated. Other semantic requirements are assumed to be inherited from the UML model requirements and are thus implied [13]. Only one syntactic requirement is made, although quite a broad one. Models that are mapped must be exported to a single XML Metadata Interchange (XMI) file by the Enterprise Architect (version 14.1) AISA Project Consortium (2021b). XMI files are the most common way of exchanging UML diagrams.

The mapper architecture is presented in Fig. 3. The configuration file (top left) refers to XMI files and maintains lists of selected UML classes. A single configuration file serves as input to the mapper. Based on the configuration file, selected subsets of the models (AIXM, FIXM or others) are extracted by the extraction module. Extracted subsets are then mapped by model-specific plug-ins to RDFS/SHACL documents as RDF/XML files AISA Project Consortium (2021b).

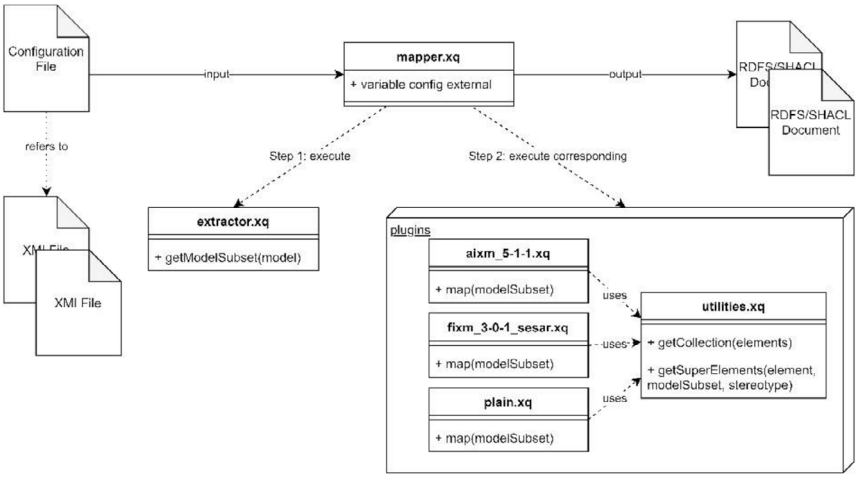

Fig. 3. The Architecture of the UML to RDFS/SHACL Mapper AISA Project Consortium (2021b)

From AIXM, classes such as *'AirspaceVolume'* and *'SignificantPoint'* were chosen. FIXM supplied important concept classes such as *'FlightType'* or *'FlightRouteInformation'*. Only subsets of classes were chosen because some concepts were not useful for en-route operation, e.g. *'DangerousGoods'* or *'AirsportSuppliesService'*.

Since the schemas are mapped only once, performance of the mapper is not crucial in AISA. Therefore, the XQuery code was not optimized for performance. Nevertheless, preliminary studies to get a feeling for the mapper's performance were made. The performance studies have been conducted with a Lenovo Thinkpad T470p using the provided configuration files and running mapper.xq with the BaseX GUI. In Table 1 there are three columns, one for each of the three test files used, each showing the duration of one mapper execution in milliseconds AISA Project Consortium (2021b).

4. AIXM, FIXM, and the Missing Links

As previously mentioned, using the existing aeronautical information exchange models (AIXM, FIXM) enables fast creation of the KG schema. Also, this process makes sure that the created schema is of high integrity which is particularly important for the safety-oriented tasks such as those of the ATCO. Using existing standards enables effortless merging of information from different originators, too. This approach, however, leaves in the KG *gaps* related to those areas of ATC operations which are not covered by any of the existing standards.

Table 1. Results of Preliminary Performance Studies for the AISA XMI Mapper

Execution	AIXM_DONLON.xml	AIXM_COCESNA.xml	FIXM_EDDF-VHHH.xml
1	32 821 ms	102 323 ms	12 555 ms
2	34 069 ms	102 758 ms	12 336 ms
3	33 137 ms	103 382 ms	12 524 ms
4	33 875 ms	103 906 ms	12 333 ms
5	34 443 ms	104 194 ms	12 335 ms
Average	33 669 ms	103 313 ms	12 417 ms

The knowledge graph could theoretically be extended indefinitely, never achieving the level of complexity or comprehensiveness similar to the level of an ATCO's mental model. However, some of the tasks which the KG-based system needs to perform might be directly or indirectly dependent on exactly the knowledge which is missing. It is therefore necessary to fill in those gaps while keeping in mind to add only the necessary data.

The process of adding necessary data to the KG consists of three consecutive steps:

1. Identifying missing knowledge
2. Establishing connections between existing and missing knowledge
3. Defining vocabulary for new knowledge

4.1. Identifying missing knowledge

This step can be achieved by: i) interviewing subject matter experts (in this case, ATCOs), ii) observing actual operations and noting the process of knowledge sourcing and utilization, iii) defining the tasks that the system needs to perform and creating well-defined requirements.

In AISA, all three methods were used iteratively. Multiple rounds of interviews and observations of ATC operations were followed by drafting the requirements. This resulted in the identification of missing knowledge in the following areas: ATC procedures (on a tactical level), letters of agreement, aircraft performance (general performance for each aircraft type), and air traffic flow and capacity metrics (sector capacity, traffic complexity etc.). Additionally, as the project goal is to make a hybrid system combining ML and symbolic reasoning, vocabulary to describe the ML modules, their meta-data and results, is also needed.

To observe actual ATC operations, specific scenarios based on real traffic have been designed. The scenarios were developed with the help of subject matter experts and enable the system to collect a variety of data. The ATCOs performed exercises lasting between 5 and 23 minutes in an airspace that was well-known to them. The same scenarios were later used to compare the SA of ATCOs and the KG-based system.

4.2. Establishing connections with existing knowledge

The task of connecting the identified gaps with the rest of the KG, especially in very narrow domains such as en-route ATC operations, is a straightforward task because there is a lot of overlap between areas. The descriptions in this section, appropriately, have a lot of overlap with the process of defining and adding new vocabulary, which is described in more detail in the following section.

An important part of the system are outputs from the conflict detection ML module, which is used to detect pairs of flights whose trajectories might bring them closer than the prescribed separation minima (horizontally 5 NM, vertically 1000 ft). A class *'AircraftPairs'* could be created and connected to existing *'Flight'* classes, inherited from FIXM.

Name of each *'AircraftPairs'* class node was the concatenation of flight callsigns, divided by an underscore (e.g., "MAERO_BAW881V"). These nodes did not need to be connected to flight data by predicates because string parsing allowed the pair name to be divided back into separate callsigns, which were then used to find specific flight data.

An example of direct linking (by predicates) was the creation of a *'Performance'* class within the *'FlightIdentification'* class. As shown in Fig. 4 (top), it consists of *'performanceAltitude'*, *'machNumber'*, *'rateOfClimb'*, *'rateOfDescent'*, and *'trueAirSpeed'* attributes. Their values can all be pulled from the Base of Aircraft Data (BADA).

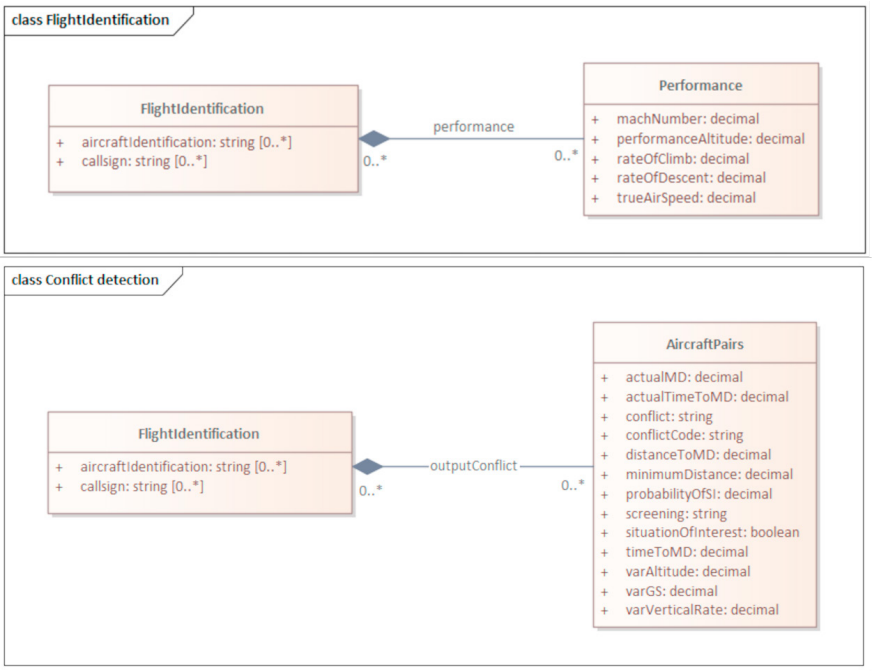

Fig. 4. The Performance class, created to represent aircraft performance data (top) and the Conflict detection class, created to represent results of the CD ML module (bottom)

4.3. Defining new vocabulary

With the gaps in the knowledge graph identified and the links between the existing and missing knowledge specified, finally new vocabulary can be defined and added. Our approach was to start with the UML diagram modelled in accordance with the rules described in the UML to RDFS/SHACL Mapper chapter. An example of the new vocabulary is shown in Fig. 4 (bottom) which presents an UML diagram describing the input-output information of the conflict detection ML module. The UML diagram was then used to create an RDF Schema and SHACL constraints.

By conforming to both the schema and constraints, the data instances (i.e. the RDF graphs representing a traffic scenario) can be created either manually (for static data) or programmatically (for dynamic data). Static data, which either changes rarely or does not change through the scenario, includes airspace point coordinates, ML module metadata and others. Dynamic data, on the other hand, represents rapidly changing data such as flight position or conflict detection module outputs.

5. Populating the Knowledge Graph

Observations of ATC operations were not used exclusively for UML vocabulary modification. EUROCONTROL's real-time ATC simulator ESCAPE Light, was used to perform human-in-the-loop (HITL) simulations. ESCAPE Light allows for data export, which can then be converted used for RDF graph creation. Simulation data was the principal source of KG data, complemented by EUROCONTROL's network strategic tool (NEST) and other sources.

Simulations were in fact scenarios based on real traffic data and situations. Since participating ATCOs were employed at Switzerland's SKYGUIDE, Swiss airspace (or more specifically, sector LSAZM567) was used. Four different scenarios had been designed, each of which had one specific situation of interest (e.g., pilots' non-compliance

with ATCO instructions, potential conflict outside sector border etc.). An additional training scenario, which each participating ATCO completed before working on actual scenarios, was designed for familiarization with the ESCAPE Light simulator.

Each of the 20 ATCOs completed the rest of the scenarios in random order. The ESCAPE simulator system records exercise logs in XML format. Logs contain data on navigation points, sector boundaries, military sector boundaries and aircraft flight plans - this data doesn't change during the exercise. Alongside that data, aircraft position is recorded together with its heading, Mach number, True Airspeed, ground speed, actual flight level, requested flight level, cleared flight level, etc. The simulator also records changes that had occurred due to ATCO actions on a flight, ensuring an automatically generated revised flight trajectory. ATCO actions are also important because, even though the initial traffic situations were identical for all ATCOs, they applied different strategies which influenced the traffic. The result of the experiment was a set of 76 different exercises – 4 per participating ATCO, with several removed for technical reasons.

The task of converting the data from exercise XML logs to RDF files allowed for a certain degree of automation, but initially the conversion was performed manually. To encode relevant parts of the exercises, between 15 and 58 RDF graphs were used, one for each of the relevant timestamps. Each timestamp represents a snapshot of traffic situations at a different moment. The steps between timestamps range from 5 to 60 seconds – graphs are denser during high interest situations (e.g., during changes to aircraft trajectories, during non-standard situations) and sparser when there were no changes to the traffic situations.

For example, exercise CH4-ATCO18 contains 54 RDF graphs, covering a 23 min interval. Sector demand increases as the exercise progresses, and each additional aircraft correlates to additional triples in the corresponding graph. This increases both the time needed to upload the triples to the KG server and the processing time of each task (because they are generally applied to all aircraft in the graph).

Table 2. Results of Preliminary Performance Studies for the AISA XMI Mapper

Graph	Timestamp	Graph processing time	Graph	Timestamp	Graph processing time	Graph	Timestamp	Graph processing time
g0	11:54:00	2924 ms	g18	11:58:40	8184 ms	g36	12:08:20	15542 ms
g1	11:54:15	2663 ms	g19	11:58:55	8175 ms	g37	12:08:40	16441 ms
g2	11:54:30	2579 ms	g20	11:59:40	8329 ms	g38	12:09:00	19044 ms
g3	11:54:50	2558 ms	g21	12:00:10	8780 ms	g39	12:09:20	17924 ms
g4	11:55:00	2761 ms	g22	12:00:25	9254 ms	g40	12:09:50	17980 ms
g5	11:55:20	3057 ms	g23	12:01:25	10095 ms	g41	12:10:20	16305 ms
g6	11:55:40	3154 ms	g24	12:02:00	10365 ms	g42	12:10:50	17762 ms
g7	11:56:20	3532 ms	g25	12:02:25	10686 ms	g43	12:11:25	17787 ms
g8	11:56:30	3727 ms	g26	12:03:00	11845 ms	g44	12:12:00	19357 ms
g9	11:56:40	3974 ms	g27	12:03:20	11958 ms	g45	12:12:40	18238 ms
g10	11:56:50	4605 ms	g28	12:04:20	12521 ms	g46	12:13:10	19159 ms
g11	11:57:00	4836 ms	g29	12:04:45	14007 ms	g47	12:14:00	19319 ms
g12	11:57:10	5256 ms	g30	12:05:00	14191 ms	g48	12:14:25	19812 ms
g13	11:57:20	5529 ms	g31	12:05:40	14006 ms	g49	12:15:00	20921 ms
g14	11:57:45	5681 ms	g32	12:06:05	14105 ms	g50	12:15:30	24347 ms
g15	11:57:55	5601 ms	g33	12:06:25	15563 ms	g51	12:15:55	19539 ms
g16	11:58:10	6948 ms	g34	12:07:00	16085 ms	g52	12:16:30	19611 ms
g17	11:58:20	7473 ms	g35	12:07:45	14951 ms	g53	12:17:00	18484 ms
Median runtime per graph		11732 ms						

The KG system processing times also depend on the number of tasks applied to each graph. The times shown in Table 2 include 46 of the 58 tasks defined in earlier stages of the project. The system is run from the Eclipse IDE for Java Developers (version 2021-06, 4.20.0), on a Dell Vostro 15 3000 laptop. Further testing on more powerful configurations might yield improved runtimes, but current performance already hints that real-time use is possible.

6. Conclusions and Future Work

In this paper we have presented our approach to generating a domain-specific knowledge graph based on the existing exchange models and the use of that knowledge graph as part of a larger hybrid system combining ML and symbolic reasoning aimed at providing situational awareness in en-route ATC operations.

The knowledge graph generation approach can also be used to create knowledge graphs for other uses. The main advantage of this approach is that it reduces the effort needed to create the knowledge graph by repurposing existing aeronautical exchange models. In support of the work done we have developed a tool to automatically map UMLs to RDF schema and SHACL constraints. This tool is shared as open source and can be used under the very permissive MIT license (free for commercial and private use, distribution, and modification).

We also presented briefly the methodology used to identify and patch the gaps in the knowledge graph along with an example of new vocabulary specific to our use case. An important part of identifying knowledge gaps was the analysis of ATC operations and formulation of ATCO tasks. The tasks were then implemented in the KG system using the Java programming language and applied to the same traffic situations which were analyzed as part of knowledge gap identification process.

The addition of task-specific knowledge and its integration with knowledge defined in existing aeronautical exchange models represents a step forward from the ontologies described in articles by Keller (2019) and Egami et al. (2020). Comparison of peformance results shows that inclusion of strictly that knowledge which is relevant for system tasks yields better results than work done by Keller (2019). While the query runtime comparison with Egami et al. (2020) may not be as favorable, it should be noted that the number of queries is vastly different, as is the system architecture. Nevertheless, lessons learned from those articles are informational for future research on this topic.

In our future work more ATCO tasks will be added to the hybrid KG system while expanding the KG with necessary information , such as BADA data, meteorological information, letters of agreement, etc. We will explore data management strategies which could help optimize system performance and counter data accumulation.

Acknowledgment

This paper is part of the AISA project, which has received funding from the SESAR Joint Undertaking under grant agreement No 892618 under European Union's Horizon 2020 research and innovation program.

References

AISA Project Consortium, 2021a. Concept of Operations for AI Situational Awareness. Technical Report. AISA Project Consortium. Available: https://aisa-project.eu/downloads/AISA_D2.1_CONOPS.pdf.

AISA Project Consortium, 2021b. Proof-of-concept KG system. Technical Report. AISA Project Consortium. Available: https://aisa-project.eu/downloads/AISA_4.1.pdf.

Egami, S., Lu, X., Koga, T., Sumiya, Y., 2020. Ontology-based data integration for semantic interoperability in air traffic management, in: 2020 IEEE 14th International Conference on Semantic Computing (ICSC), pp. 295–302. doi:10.1109/ICSC.2020.00059.

Endsley, M.R., 1995. Measurement of situation awareness in dynamic systems. Human factors 37, 65–84.

EUROCONTROL, 2021. Forecast Update 2021-2024 European Flight Movements and Service Units - Three Scenarios for Recovery from COVID-19. Technical Report. EUROCONTROL. Bruxelles, Belgium. Available: https://www.eurocontrol.int/sites/default/files/2021-05/eurocontrol-four-year-forecast-2021-2024-full-report.pdf.

EUROCONTROL, FAA, 2022a. Aeronautical information exchange model (AIXM). Website. Https://www.aixm.aero/.

EUROCONTROL, FAA, 2022b. Flight information exchange model (FIXM). Website. Https://www.fixm.aero/.

EUROCONTROL Performance Review Commission, 2020. Performance Review Report 2019. Technical Report. EUROCONTROL. Bruxelles, Belgium. Https://www.eurocontrol.int/publication/performance-review-report-prr-2019.

Keller, R.M., 2019. Building a knowledge graph for the air traffic management community, in: Companion Proceedings of The 2019 World Wide Web Conference, pp. 700–704.

Porosnicu, E., 2021. Overview - aixm confluence - home - aixm confluence. Website. Https://ext.eurocontrol.int/aixm_confluence/.

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., Wilhelm, M., 2014. Bringing up opensky: A large-scale ads-b sensor network for research, in: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IEEE. pp. 83–94.

SESAR Joint Undertaking, 2019. Single programming document 2019-2021. Technical Report. LU: Publications Office. Available: https://data.europa.eu/doi/10.2829/158329.

Sheth, A., 2019. Why do these knowledge graphs need 10,000 pairs of hands? LinkedIn Article. Https://www.linkedin.com/pulse/why-do-knowledge-graphs-need-10000-pairs-hands-amit-sheth.