Building a Knowledge Graph for the Air Traffic Management Community

Richard M. Keller
Intelligent Systems Division, NASA Ames Research Center, Moffett Field, CA, USA rich.keller@nasa.gov

ABSTRACT

Historically, most of the focus in the knowledge graph community has been on the support for web, social network, or product search applications. This paper describes some of our experience in developing a large-scale applied knowledge graph for a more technical audience with more specialized information access and analysis needs – the air traffic management community. We describe ATMGRAPH, a knowledge graph created by integrating various sources of structured aviation data, provided in large part by US federal agencies. We review some of the practical challenges we faced in creating this knowledge graph.

CCS CONCEPTS

• Computing methodologies~Semantic networks • Information systems~Information integration

KEYWORDS

knowledge graph, ontology, semantic web, air traffic information management

1 Introduction and Motivation

Every day, global aviation industry data providers generate a vast array of aviation information. When taken together, these data characterize the functioning of the global aviation system. The availability of this data provides the tantalizing possibility that we might analyze these data and discover new ways to improve air transportation economics, efficiency, and safety, for the benefit of all. However, aviation data are highly heterogeneous and are produced by a multitude of different providers in different formats and encodings. Improvements in the performance of the overall aviation system therefore depend on our ability to integrate, query, and analyze this information.

At the US National Aeronautics and Space Administration (NASA), we have been examining the use of knowledge graph technologies to create an integrated dataset for query and analysis of aviation data. Such a resource would be of potential interest to many aviation stakeholders, including public policy makers, airspace operators, flight controllers, airline carriers, aerospace researchers, and aviation industry service providers. These stakeholders require multiple sources of information to assist them in making decisions that impact current or future aviation systems operation.

We have focused our initial work on providing a resource for aerospace researchers who study air traffic management (ATM) procedures and systems. These researchers access a core set of ATM data sources generated by a handful of different providers, extract the data they require, and then write code to integrate the data – all before beginning their specific analyses. For researchers whose expertise lies in aeronautics – not data management – the overhead required to achieve data integration can be considerable. The data sources lack standardization and incorporate varying data formats, nomenclature, and organizational structure. As a result, data integration is a significant bottleneck to research productivity, anecdotally consuming up to 50% or more of the total data processing effort – effort that could be better spent analyzing data.

To demonstrate the utility of knowledge graphs, we first set about designing an ontology to model the contents of the core ATM data sources. Then we constructed a corresponding knowledge graph populated with instances that were derived from actual ATM data generated by these core sources during one month of air traffic operations in the New York metropolitan area. Finally, we worked with researchers to demonstrate how the knowledge graph could be queried (using SPARQL) to help answer active research questions. Some of the representative research-related queries we generated include the following:

- Find Newark Airport flight arrivals that passed through the PENNS airspace fix[1] and landed during rainy and windy conditions in July 2014;
- Find which airspace sector controlled the most flights in the US during the 9am Eastern hour on 7/15/14;
- List all aircraft types that were used in commercial airline flights departing or arriving JFK Airport on 7/15/14;
- Find flights subject to ground delay advisories on 7/15/2014

This paper describes some of our experiences building an ATM knowledge graph to help answer these types of questions, and presents some of the challenges and pain points we faced. The next section begins by introducing the ontology and knowledge graph in more detail.

2 NASA's ATM Knowledge Graph

The NASA ATM Ontology (ATMONTO) defines key classes of entities pertaining to the US National Airspace System and the management of air traffic through that system. The primarily RDF-based[2] ontology describes a wide variety of aviation entities, and features more than 150 classes, 150 datatype properties, and 100 object properties. The ontology is fully documented in [1] and available for download with accompanying sample data [2].
Briefly summarizing, the ATMONTO classes represent:

- **airspace infrastructure entities**: airports, runways, terminals, airways, waypoints, air sectors, air traffic control facilities, and air control regions;
- **flight-related entities**: points of origin and destination, airline operators, flight plans, flight trajectories, aircraft, aircraft subsystems, and airframe manufacturers; and
- **flight operating conditions**: current and forecast airport weather conditions, systemwide ATM advisories, and routing constraints due to weather, facility, or other disruptions.

ATMONTO serves as the representational foundation for the NASA ATM Knowledge Graph (ATMGRAPH). ATMGRAPH is populated with over 38M instances and 260M triples derived from infrastructure, flight, and weather data collected for the three largest New York area airports (JFK, Newark, and LaGuardia) during July 2014. Included are data from approximately 100K flights arriving or departing from these airports during this month. The data are stored in an instance of OntoText's GraphDB triple store running at NASA.[3]
To give a flavor for how the ATM Knowledge Graph is structured, consider Figure 1. At the center of this graph fragment is a node representing a specific instance of a flight: UAL535 on 2014-07-15 departing at 00:19:00. The flight is linked to a variety of associated nodes via object properties: the specific aircraft flown (the aircraft with registration number N589UA, a Boeing 757 model 222), the carrier (United Airlines), the flight's departure and arrival airports (JFK and LAX), the planned and actual flight route between these airports (both represented as linked lists of either airways or actual trajectory points en route). The flight and the other instances in the graph fragment contain values for various datatype properties defined in ATMONTO. (Note: the datatype properties and values are not shown in the Figure.) For example, the flight has properties corresponding to the flight identifier ('UAL535'), the flight category ('commercial'), and the departure and arrival date/time. Although the knowledge graph depicted in Figure 1 focuses on the representation of a single flight, the actual graph contains data for 100K flights and is extremely densely

[1] An airspace navigation fix, or waypoint, is a named geospatial location defined at an intersection or point along a designated airway or above a surface landmark. Fixes are used in aircraft guidance and navigation.

[2] ATMONTO uses a limited set of OWL constructs, including property restrictions.

[3] A subset of the data covering 100 actual flights has been released for documentary purposes [2]. The triple store is not currently accessible to the public.

connected. For example, over 38K flights are linked to JFK as either their departure or arrival airport during July 2015; and 59 flights flown during that period used the aircraft registered as N589UA. Furthermore, the fragment in Figure 1 illustrates only a small fraction of the total number of classes and link types found in the overall graph.

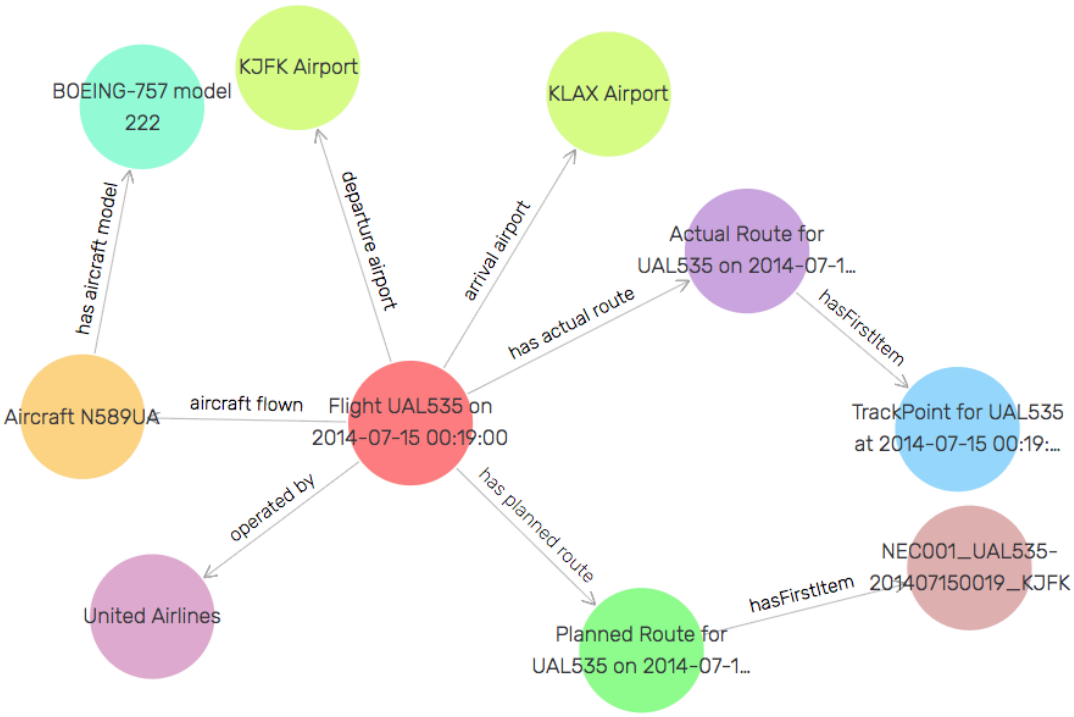

Figure 1: Fragment taken from ATMGRAPH illustrating some of the linkages centered around a flight. All nodes reside within one of ATMONTO's defined namespaces. Datatype properties -- such as flight start and end time -- are not shown.

ATMGRAPH is constructed principally from eight different structured data sources consisting of approximately 50 different low-level data products (e.g., database tables, custom data files, html files, spreadsheets). These data were produced by various governmental agencies (including the US Federal Aviation Administration (FAA), the US Department of Transportation, and the US National Oceanographic and Atmospheric Administration), non-governmental organizations (such as the International Civil Aviation Organization (ICAO)), and other web-based content providers (e.g. openflights.org). These structured sources are highly heterogeneous in format and structure. Python scripts and Java code were developed to transform these data from their original format into RDF triples conforming to the ATMONTO ontology, and the triples were loaded into the GraphDB triple store. As with all real-world applications, the data are noisy, contain errors, and must be extensively preprocessed to ensure the quality of the information in the knowledge graph.

In the next section we review some of the many practical challenges we faced in constructing ATMGRAPH.

3 Challenges

3.1 Entity Naming, Resolution, and Linking

Key components of the overall process necessary to transform the source data into linked data involve entity naming, resolution, and linking. Although these aspects are generally thought to be relatively straightforward when dealing with structured data sources versus unstructured text, they were non-trivial with the structured sources produced in the ATM domain. The difficulties relate back to the heterogeneity of data producers,

data formats, and data encodings – and to the lack of standardization in data products overall. In the next subsections we give examples of these issues.

3.1.1 Assembling a Flight Instance

Although a flight would seem to be a key entity in ATM domain, there is no single authoritative source of flight information to be found within any of the core data sources used by the researchers or published by the FAA. Instead, information about flights must be pieced together by accumulating data from multiple sources in order to synthesize the properties of a flight instance and link it appropriately. A principal source of information we used to bootstrap the flight data amalgamation process was the ASDI (Aircraft Situation Display to Industry) flight track data, a source generated by FAA and used widely by commercial flight tracking web sites, such as flightaware.com and flightview.com. An ASDI file contains one line for each 60-second reporting period for every flight aloft during the timeframe covered by the file. The tracking data provides proof positive that a flight was flown, and provides some essential data such as the flight identifier, the departure/arrival airports, and tracking information (altitude, latitude, longitude, airspeed, etc.). But key additional information is missing from ASDI files, including the aircraft registration number, the aircraft manufacturer, and the airline carrier, among other items. Those data must be inferred from the encoded information and combined using auxiliary data dictionaries. For example, the airline carrier must be inferred from the flight identifier (e.g., in 'UAL535', 'UAL' is the encoding for United Airlines).

3.1.2 Naming the Flight Instance

The creation of a unique flight identifier is something that the aviation industry has struggled with for some time [3]. FAA and EUROCONTROL (FAA's counterpart in Europe) have come to an understanding over the past decade that integration of flight data from multiple sources requires the use of a Globally Unique Flight Identifier (GUFI), and are developing registry services for providing a GUFI. However, GUFIs are not yet in widespread usage within the aviation industry, creating challenges for entity resolution and making it difficult to determine which data from multiple data products pertains to the same flight.

3.1.3 Multiple Standards in Use

Even though many of the core ATM data products are generated by the same government agency (FAA), they do not consistently encode references to a given entity, and in some cases, there are multiple standards in use for the encoding [4]. As a simple example, the airport code for John F. Kennedy International Airport can be expressed using the ICAO standard (yielding 'KJFK'), the IATA (International Air Transport Association) standard ('JFK'), or the FAA standard, which in this case – but not always – is the same as the IATA code.

3.2 Spatial and Temporal Representation

Aviation information involves spatial and temporal aspects that must be adequately addressed by any representation [5]. Some examples of specific temporal requirements on entities in the ATM domain include: periodic updates to aircraft routes; time-limited air traffic initiatives; temporary airport obstacles; scheduled runway closures; and forecast weather phenomena. In addition to temporal aspects, many of the problems addressed by ATM researchers involve spatial reasoning. As a simple example, it is often necessary to determine whether an aircraft has passed through a defined region of the airspace, such as a flight control sector or a restricted airspace. Or it may be necessary to calculate the closest distance between an aircraft and a given navigation airway. Our representation of a flight trajectory (i.e., a flight path) illustrates some of the issues we faced, and the tradeoffs that must be made between expressivity and efficiency.

3.2.1 Representing a Flight Trajectory

A trajectory in ATMONTO is represented as a sequence of explicit track point instances. Each track point corresponds to a specific reporting time[4] when an aircraft's speed and navigation fix (its latitude, longitude, and altitude) are captured and relayed to ground systems. Unfortunately, this representation, while adequate

[4] ATMONTO supports temporal aspects using either time points (modeled as datetime properties) or intervals (modeled as classes with start/end time points).

for many needs, is verbose and leads to a proliferation of track points that undermines the efficiency of SPARQL query responses. Fully 70% of the 38M nodes in ATMGRAPH are either track points or navigation fixes, as described in [6]. Furthermore, performing the geometric calculations necessary to compute line crossings or geometric distance calculations in SPARQL is awkward and often impossible without escaping to custom-coded programming language functions.

An alternative and more efficient approach involves the use of a geospatial representation to capture the trajectory as a segmented line, employing something similar to the standards-based WKT (Well-Known Text) or GML (Geographic Markup Language) geometry representations used in geospatial databases, such as PostGIS and Oracle Spatial. These databases are optimized for performing fast spatial queries. This type of geometry representation is much less verbose than the representation used in ATMONTO; rather than being represented by separate instances in the knowledge graph, all of the track points in a trajectory would be specified using a single text string (e.g., '[$point1_{lat}$, $point1_{lon}$, $point1_{alt}$] [$point2_{lat}$, $point2_{lon}$, $point2_{alt}$] …') stored in a datatype property of a trajectory instance. We experimented with this type of representation as part of the GeoSPARQL facility supported by our triple store[5]. GeoSPARQL is the Open Geospatial Consortium's standard for representing and querying geospatial linked data [7]. But GeoSPARQL only represents 2-D geometries, and 3-D was necessary to represent the altitude dimension of the flight trajectories and other three-dimensional regions of the airspace.

Even if GeoSPARQL supported 3-D geometries, there would still be an issue of representational granularity to consider. Because the compact geometry representation in GeoSPARQL buries the individual track point information within a text string stored in a datatype property, it is not possible to make statements about individual track points. For example, an important concept in ATM is the 'top of descent' – the spot at which an aircraft begins its descent to the airport. It would be preferable to mark a specific track point as the top of descent using a property on a track point instance. But with the GeoSPARQL representation, this is not possible. Similarly, if we want to record multiple aircraft passing through the same track point at different times or to record successive weather conditions at the track point, this would not be possible using GeoSPARQL.

3.3 Scaling

We have reported previously [6] about our efforts to benchmark performance of two different triple stores as we scaled up the amount of data in ATMGRAPH from one day to one month of New York flight operations. The results indicate that for 60% of our 17 benchmark queries, the execution time increases roughly linearly in the number of triples. (For one benchmark query, however, execution time increased exponentially.) In only 30% of the queries was execution time was not impacted by the increase in triples. Linear and exponential increases in execution time signal problems ahead as we attempt to scale up ATMGRAPH to include ever larger amounts of flight data. NASA aeronautics researchers sometimes perform multi-year ATM analyses across the entire US, so one month of data for New York is quite limiting. Hybrid approaches in which a portion of the flight data is stored in either a high-performance big data system or a geospatial database may help alleviate performance issues. Nevertheless, the performance of state-of-the-art triple stores is not sufficient to support real-time querying of the knowledge graph – at least for 70% of our benchmark queries. Furthermore, query optimization tools that might allow us to hand-tune performance of a SPARQL query engine are relatively primitive and difficult to use, at least in our experience.

3.4 Visualization

Visualization plays an important exploratory role for those knowledge graph users who wish to closely examine query results and navigate through specific chains of instances in the graph. Our ATM researchers often wish to examine anomalous flights that are being flown outside of normal operating parameters, or flights flown under specific operating conditions, such as severe convective weather. In these circumstances, it would be useful for users to examine specific flight instances in ATMGRAPH as an adjunct to other

[5] GraphDB includes native GeoSPARQL support via a plugin.

analyses they are performing. Our experience with current tools is that visualization remains challenging in large, densely-connected graphs, a fact that has been well-documented [8-10]. Navigating through a graph in which hundreds or thousands of links emanate from a single node requires specific filtering techniques that allow users to selectively choose the link or links they wish to traverse and the nodes they wish to view. While the GraphDB environment we are using implements some filtering capabilities (and even allows users to write their own filters using SPARQL) these are not sufficient to support our users' graph exploration needs.

4 Related Work

There is a body of published work on the use of ontologies, and more recently knowledge graphs, in aviation-related applications, including ATM [11-14], aviation data management [15], aviation safety [16, 17], and avionics [18]. The Graph of Things [19] incorporates flight information and track points similar to the data incorporated in ATMGRAPH. Despite this related work, reuse of other ontologies was not generally practical in our real-world application and setting. In some cases, ontology details were not published; in other cases, ontologies were too simplistic and/or mismatched to the needs of our ATM application in terms of scope or level of detail. And finally, often the effort involved in locating, augmenting, and reusing ontologies is significant and outweighs their overall utility.

5 Summary

This paper has described our efforts toward developing a knowledge graph resource for the air traffic management community, including some of the challenges we faced in producing a prototype system for use by aerospace researchers. Of the challenges described, scaling is the most serious barrier to deployment, as a narrowly-scoped knowledge graph will be of limited use to NASA researchers. Further, without more intuitive knowledge graph query languages and visualization tools, non-experts will struggle to use the technology to its full potential.

ACKNOWLEDGMENTS

This work was funded by the Airspace Operations and Safety Program from within NASA's Aeronautics Research Mission Directorate. My thanks to the members of the Sherlock Data Warehouse Team at NASA Ames, who supported me in conducting this work, and the aerospace researchers who kindly provided their input. I also want to thank the FAA and the many other aviation data providers who made their data available for this effort.

REFERENCES

[1] Keller, R.M. *The NASA Air Traffic Management Ontology: Technical Documentation*. Technical Memo NASA/TM-2017-219526, National Aeronautics and Space Administration, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/2017 0006095.pdf, 2017.
[2] Keller, R.M. *ATMONTO Ontology*. https://data.nasa.gov/ontologies/atmonto/, release dated March 2018.
[3] EUROCONTROL.*Globally Unique Flight Identifier (GUFI) Requirements*,2014, https://www.fixm.aero/documents/GUFI%20Requirements%20v2%201_Final.pdf.
[4] *Airport Codes*. https://www.aircharteradvisors.com/airport-codes/ Accessed 2/19.
https://fixm.aero/documents/GUFI%20Requirements%20v2%201_Final.pdf, 2014.
[5] EUROCONTROL. *AIXM 5 Temporality Model*. http://aixm.aero/sites/aixm.aero/ files/imce/AIXM51/aixm_temporality_1.0.pdf , 2007.
[6] Keller, R.M., Ranjan, S., Wei, M.Y. and Eshow, M.M. *Semantic Representation and Scale-up of Integrated Air Traffic Management Data*. ACM, in Proceedings of the International Workshop on Semantic Big Data, San Francisco, 2016.
[7] Open Geospatial Consortium. *OGC GeoSPARQL - A Geographic Query Language for RDF Data*. Document # OGC 11-052r4, 2012.
[8] Gómez-Romero, J., Molina-Solana, M., Oehmichen, A. and Guo, Y. Visualizing large knowledge graphs: A performance analysis. *Future Generation Computer Systems*, 89 (2018), 224-238.
[9] Pienta, R., Abello, J., Kahng, M. and Chau, D. H. *Scalable graph exploration and visualization: Sensemaking challenges and opportunities*. IEEE, in Big Data and Smart Computing (BigComp), 2015.
[10] Wills, G. J. NicheWorks—interactive visualization of very large graphs. *Journal of computational and Graphical Statistics*, 8, 2 (1999), 190-212.
[11] Burgstaller, F., Steiner, D., Schrefl, M., Gringinger, E., Wilson, S. and van der Stricht, S. *AIRM-based, fine-grained semantic filtering of notices to airmen*. In IEEE *Integrated Communication, Navigation, and Surveillance Conference (ICNS),* 2015, pp. D3-1-D3-13.
[12] Neumayr, B., Gringinger, E., Schuetz, C. G., Schrefl, M., Wilson, S. and Vennesland, A. *Semantic data containers for realizing the full potential of system wide information management*. In IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), pp. 1-10. IEEE, 2017.
[13] van Putten, B.-J., Wolfe, S. R. and Dignum, V. *An Ontology for Traffic Flow Management*. In *Proc. 8th Aviation Technology, Integration, and Operations Conference*, AIAA, 2008.
[14] Vennesland, A., Neumayr, B., Schuetz, C. G. and Savulov, A. *Experimental ontology modules formalizing concept definition of ATM data*. SESAR Exploratory Research Report, http://www.project-best.eu/downloads/D1.1%20Experimental%20 ontology%20modules%20formalising%20concept%20definition%20of%20ATM%20data.pdf, 2017.
[15] Keller, R. M. *Ontologies for Aviation Data Management*. In Proc. IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 2016.

[16] Ledvinka, M., Lališ, A. and Křemen, P. Toward Data-Driven Safety: An Ontology-Based Information System. *Journal of Aerospace Information Systems*, 16, 1 (2018), 22-36.
[17] Zhao, Q., Li, Q. and Wen, J. Construction and application research of knowledge graph in aviation risk field. In Asia Conference on Mechanical and Aerospace Engineering, EDP Sciences, 2017.
[18] Blasch, E. Ontologies for nextgen avionics systems. In Proc. IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), 2015.
[19] Le-Phuoc, D., Quoc, H. N. M., Quoc, H. N., Nhat, T. T. and Hauswirth, M. The Graph of Things: A step towards the Live Knowledge Graph of connected things. Journal of Web Semantics, 37 (2016), 25-35.