A Comparative Study of Two Complex Ontologies in Air Traffic Management

Eduard Gringinger*, Richard M. Keller†, Audun Vennesland‡, Christoph G. Schuetz§ and Bernd Neumayr§
*Corporate Research, Frequentis AG, Vienna, Austria
Email: eduard.gringinger@frequentis.com
†Intelligent Systems Division, NASA Ames Research Center, Moffet Field, CA, USA
Email: rich.keller@nasa.gov
‡Software Engineering, Safety and Security, SINTEF, Trondheim, Norway
Email: audun.vennesland@sintef.no
§Institute of Business Informatics – Data & Knowledge Engineering, Johannes Kepler University Linz, Linz, Austria
Email: {schuetz,neumayr}@dke.uni-linz.ac.at

***Abstract*—Over the past 25 years, multiple different data models have been introduced to standardize information management and facilitate data exchange and integration in the aviation domain. As a next step in the evolution of aviation data management, ontologies capturing the semantics (concepts, properties, and relationships) have been produced based on those models. In this paper, we describe a study comparing two recently released and independently-developed complex ontologies focused on Air Traffic Management (ATM) – the NASA ATM Ontology and an ontology derived from the ATM Information Reference Model. We develop a methodology for manually comparing two ontologies and identifying what we describe as exact, light, and mismatches between concepts in the two ontologies. We also describe a classification scheme that characterizes mismatches in terms of the general reason for the mismatch. This approach can be applied to improve existing ATM ontologies and foster interoperability, which will benefit aviation stakeholders.**

***Index Terms*—Information Exchange Models, ATM Information Reference Model, NASA ATM Ontology, Ontology Matching**

I. Introduction

In knowledge representation, ontologies are used to formally specify knowledge about a specific domain, which allows to build "intelligent" applications. Recently, two independently-developed ontologies for the Air Traffic Management (ATM) domain were released: the NASA ATM Ontology (ATMONTO) and AIRM-O, an ontology derived from the ATM Information Reference Model (AIRM). In an effort to capture the semantic relations between concepts in these two ontologies, we set out to establish a mapping between the two ontologies.

Despite their separate development and the thus resulting differences, there is considerable conceptual overlap between the two ontologies, and aviation stakeholders can benefit from the comparison and harmonization of these models. For example, a comparison can surface errors or important omissions in one ontology or the other. A detailed comparison and analysis can suggest possibilities for harmonization of partitions of these ontologies. Furthermore, such a comparison may even provide a roadmap towards developing a single consolidated ontology, resulting in stronger joint product that enables industry stakeholders to better integrate their systems and solutions between the different aviation architectures in Europe and the United States.

In our comparative study, we evaluate the similarities and differences between ATMONTO and AIRM-O by performing a manual ontology alignment using six human experts. The experts mapped concept terms from ATMONTO to concept terms in AIRM-O while indicating the degree of match using a simple categorical scale. After evaluating the degree of match among experts, we produced an alignment that contains the term mappings with highest concurrence. We then compared the ontologies once again, this time using automated, general-purpose ontology matching tools produced by ontology alignment researchers in the semantic web community. In general, the automated methods did not produce alignments that were in close agreement with our manual alignment.

The remainder of this paper is organized as follows. In Section II, we give background information. In Section III, we present the two ATM ontologies. In Section IV, we describe the alignment of the two ATM ontologies. In Section V, we present examples for matches and mismatches between classes of the two ATM ontologies. In Section VI, we discuss the results. Section VII concludes the paper.

II. Background

In this section, we present background information on ATM information (exchange) models, knowledge representation and ontologies, and ontology matching; we also review related work on semantic technologies in ATM.

A. ATM Information (Exchange) Models

The first ATM ontology in our mapping exercise is derived from multiple information exchange models that have been established as standards for the global aviation community. The Aeronautical Information Exchange Model (AIXM) [1], one of the first exchange models, standardizes exchange of data pertaining to relatively static aeronautical infrastructure resources, including air routes, air spaces, aerodromes, etc. Similarly, the Flight Information Exchange Model (FIXM) [2] and the ICAO Meteorological Information Exchange Model (IWXXM) [3] have been defined to trade flight and weather information between ATM systems. The AIRM, on the other hand, incorporates concepts from each of these existing information exchange models in a harmonized way, acting as common reference model [4], [5].

B. Knowledge Representation and Ontologies

In knowledge representation, the notion of ontology refers to a "formal, explicit specification of a shared conceptualization" [6]. An ontology is typically a type of graph-structured data model that captures a set of entities, properties, and relationships in a given domain. The properties either store data associated with the entities or link to other entities via named relationships. Entities can be typed and organized into hierarchical structures that enable property inheritance to entities lower down in the hierarchy. Ontologies were developed within the artificial intelligence and semantic web communities to represent the formal semantics of data and thereby enable intelligent agents to interpret data and draw appropriate logical inferences from the data. Due to this formal underpinning, ontologies can be used to support different types of use cases and applications beyond simple data storage, including "intelligent" applications that reason about the data and infer new data from existing data.

C. Ontology Matching

This paper describes two separate ontologies developed independently to support different ATM use cases. Although the differing use cases influence the content of these ontologies, their overall conceptual coverage of entities, properties, and relations is quite similar. Performing a detailed comparison of these two ontologies brings numerous benefits to both. First, it serves a means of cross-validating and debugging independent work. By matching one ontology to the other, we are able to discover gaps in coverage by identifying non-matching entities, properties, or relations. In this way, ontology comparison forces a detailed reconciliation between the two complementary conceptualizations. In addition, by examining a differently structured ATM domain model, we are able to view alternative choices and consider adopting improvements. In areas where one ontology lacks detail or omits concepts, the other may provide guidance for extending coverage. Finally, comparison may encourage generalization of concepts within either ontology, and even motivate the creation of a merged ontology covering both use cases. As a concrete example of this last point, our ontologies include concepts specific to the air navigation regions for which they were developed (ATMONTO for the US and AIRM-O for Europe) and they reflect regulatory and operational differences between those regions. For instance, consider how an aircraft model is classified into one of several operational wake turbulence categories based on its aircraft characteristics. The classification criteria are different between Europe and the US, and so those details differ between the ontologies. But the fundamental notion of a wake turbulence category is in common across the ontologies and could be generalized in a merged ontology to cover the US, European, and other definitions.

Ontology Matching Systems aim to provide computational support for aligning ontologies, using a wide range of techniques to do so. The techniques used by these systems are commonly categorized into *terminological*, *structural* and *lexical* techniques [7]; a single system often includes multiple techniques from all three categories. Terminological techniques are typically based on some variant of string matching [8], such as edit distance or n-grams. Structural techniques exploit the graph representation of ontologies, and derive semantic relations between concepts on the basis of their taxonomic position relative to neighboring concepts (ancestors and children). Lexical techniques, sometimes called context-based techniques, use external lexical sources such as WordNet [9], Wikipedia or other ontologies in order to infer semantic relations.

D. Related Work

Using ontologies as the basis for data management within an aviation data system enables future extensibility to support these intelligent applications. Some specific examples of intelligent aviation applications that recently have been explored include [10], [11][1]:

- Integration/aggregation of heterogeneous ATM and air transportation industry data to enable system-wide query and analysis
- Terminology and definition standardization to support system interoperability, including:
  - Unification of vocabulary used in different aviation industry standard data exchange models (e.g.: AIXM, FIXM, IWXXM)
  - Unification of aircraft make model series taxonomies maintained by different parties: US Federal Aviation Administration (FAA), International Civil Aviation Organization (ICAO), International Air Transport Association (IATA), and air frame manufacturers
  - Inter-operation of data entry systems and user interfaces for FAA Traffic Management Initiative information
- Markup and information enhancement of unstructured and semi-structured US FAA internal documents and Federal Aviation Regulations (FARs) to facilitate indexing, retrieval, search, and analysis
- Decision support for regulatory analysis and compliance monitoring, including:
  - Text content analysis and information extraction applied to FAA documents (e.g., Advisory Circulars, FARs, Aviation Safety Reports)
  - Aviation safety case validation
  - Post-incident analysis support for safety, maintenance, ATM events

[1] Many of these examples come from presentations made at the 2015 SWAT (Semantic Web for Air Transportation) workshop and are only available from the SWAT website: https://www.faa.gov/air_traffic/technology/swim/governance/service_semantics/#SWAT-Special-Interest-Group

- Filtering and prioritization of operational information, including customization of pre-flight and in-flight pilot briefings
- Publication of interlinked aviation data, including
  - Interchange of System Wide Information Management (SWIM) service description data
  - Publication of integrated archival ATM data

III. ATM Ontologies

In this section, we briefly present the NASA Air Traffic Management Ontology as well as AIRM-O, an OWL ontology derived from AIRM.

A. NASA Air Traffic Management Ontology

The first ATM ontology involved in the mapping exercise is ATMONTO [12]–[14], which was developed independently from AIRM-O. ATMONTO supports semantic integration of ATM data being collected and analysed at NASA for research and development purposes. The ontology functions as an integrative superstructure upon which to overlay data from multiple stove-piped aviation data sources, thus enabling cross-source queries and analyses that would be otherwise time-consuming and cost prohibitive to accomplish. ATMONTO includes a wide range of classes, properties, and relationships covering aspects of flight and navigation, aircraft equipment and systems, airspace infrastructure, meteorology, air traffic management initiatives, and other areas. Data from the US Federal Aviation Administration (FAA), the US Department of Transportation, the National Weather Service, and other sources is loaded into ATMONTO by transforming native data sources into RDF via a series of source-specific scripts.

In contrast to AIRM-O, ATMONTO was developed manually, following a classic knowledge modeling approach. First, domain experts identified a core set of three aviation data resources to be integrated. After an analysis of these data resources, a proposed set of ATM objects, properties, and relations was developed and presented to the experts for critique. Revisions were incorporated and the result became the initial version of ATMONTO. Because this version was built in a bottom-up fashion and was driven by a need to accommodate all of the data stored in the core data resources, this initial ontology was quite narrow and did not represent the full complexity of the ATM domain. Gradually, additional data sources were incorporated and the ontology was extended and revised as necessary to cover missing objects, properties, and relations required for each additional data source. By the end of the development process, more than ten different data sources were covered by the ontology, and the the structure of the ontology had been generalized well beyond those sources to incorporate many additional ATM objects, properties, and relations not strictly required for data coverage. As a result, the present version of ATMONTO is the product of both bottom-up and top-down design forces. Although ATMONTO is a general model of the ATM domain, it was heavily driven by application requirements and limited by the resources available to a small applied research project. Since AIRM-O is derived from a more comprehensive set of inputs and was based on a more systematic methodology, AIRM-O's scope is overall broader than ATMONTO's.

B. AIRM Ontology

As part of the exploratory research project BEST[2] an AIRM ontology (AIRM-O) [15], [16] was created, extracted from the AIRM UML diagrams [17], along with ontologies for AIXM and IWXXM. EUROCONTROL, who leads the AIRM development, was a member of the BEST project consortium. The BEST ontologies have been used in combination with semantic reasoning for supporting retrieval and filtering of ATM information. In this work, we focus on AIRM-O.

While AIRM-O and ATMONTO both cover the ATM domain, there are some important differences between the ontologies. AIRM was developed as a strategic institutional resource and was designed top-down to comprehensively cover the broad set of ATM terms and concepts employed in documents, procedures, and software requirement specifications developed by EUROCONTROL, the coordinating air traffic management authority for Europe. AIRM underwent an extensive internal review process, which resulted in a set of revisions over the course of several years. In contrast, ATMONTO was the product of a small research project and was designed bottom-up to cover a much narrower set of concepts necessary to represent the various sources of data chosen for integration. Still, there is considerable conceptual overlap between the two ontologies.

The transformation of the AIRM UML diagrams into an OWL ontology followed the Object Management Group's guidelines in the Ontology Definition Metamodel [18], using the XML Metadata Interchange (XMI) representation of the UML diagrams. The UML diagrams in the AIRM Logical Data Model [17] served as the fundamental for the construction of AIRM-O. The reason for choosing the Logical Data Model as the fundamental for AIRM-O construction lies in the required level of detail in order to be useful for practical applications. This level of detail is achieved by transforming the properties and associations from the Logical Data Model in addition to the entities, i.e., the UML classes. The XSLT scripts were made available online along with the ontology.

One of the objectives of the BEST project was to develop strategies applying semantic technology for supporting data distribution, supporting the SWIM target of making information exchange in aviation more efficient and precise. For performance reasons these strategies used smaller modules of the AIRM-O ontology to match data demand with available data in order to offer more accurate information services in ATM. The ontology modules were automatically extracted from the AIRM-O ontology using principles from syntactic locality module extraction [19].

A critical – and labor intensive – task of governance in ATM is ensuring compliance between AIRM and information exchange models. This is a task that is performed manually

[2]http://www.project-best.eu/

following a compliance framework that in detail specifies how to interpret compliance at different levels (e.g. semantic equality, generalisation, etc.). To reduce the manual burden, the BEST project developed a *AIRM Compliance Validator* application that used ontology matching techniques in combination with AIRM-O and ontological representations of the exchange models to verify that elements from information exchange models were in line with the semantic constructs specified in the AIRM [20].

IV. Matching the ATM Ontologies

Matching AIRM-O and ATMONTO was very challenging; six judges spent significant time and effort over several weeks to arrive at a suitable reference alignment. In general, these ontologies describe a complex, technical domain with a large number of highly specialized terms and concepts, e.g., *AirspaceFlowProgramTMI*, *StandardInstrumentDeparture*. Typical knowledge resources used in automated ontology alignment, e.g., WordNet or DBpedia, do not cover the ATM topic and terminology in any depth. Although the size of the two ontologies in numbers of classes and properties is moderate, the complexity of the ATM ontologies is more similar to large biomedical terminologies than to moderately-sized ontologies. Furthermore, these ontologies include many abstract concepts, such as *AircraftFlowCapacity* and *CloudLayerProfile*, which are difficult to match based solely on class names since naming is more arbitrary and less standardized for complex, abstract concepts.

The judges – each of whom had some degree of familiarity with both aviation and ontologies – were asked to match each of the 154 entity classes in ATMONTO to corresponding classes in the larger AIRM-O. As part of this process, the judges could make use of their domain knowledge as well as all available input, including descriptive class and property annotations in the ontologies plus any other informative web resources, e.g., Skybrary[3]. After the initial matches were compiled, two of the six judges who were specifically aviation subject matter experts then reviewed the matches for each ATMONTO class and produced a consensus result. Each result was assigned to one of three match categories describing the relationship between the ATMONTO and AIRM-O classes:

1) Exact Match: Since a 100% match of all properties between two independently-defined entities is quite unlikely, an *Exact Match* was defined as one in which the modeling depth, scope and definition is highly similar between the classes.

2) Light Match: A *Light Match* means that both ontology classes capture the same concept but in a different way. For example, the concepts could differ in depth, focus on different standards, vary in the way the class was modelled, etc.

[3]https://www.skybrary.aero

3) Mismatch: A *Mismatch* is the case where a majority of the six judges misidentified two classes as matching, but the aviation expert judges found that there was no match.

For the light match category, we had planned to rate each result on a numeric degree-of-match scale from 0-1, but in practice, a simple numeric metric did not seem very illuminating. Instead, we began to focus on the mismatches and characterizing the underlying reasons why the judges incorrectly paired a class from the source ontology with one in the target ontology. As expected, judges – especially those with less aviation background – tended to match those classes with similar class names. However, similar class names were no guarantee of a correct match. In fact, in approximately 25% of the exact match pairs, the class names did not contain words in common, while in approximately 40% of the light match pairs, the class names did contain words in common. This partly explains why automated alignment techniques that focus on class name similarity do not perform particularly well on this data set. Based on our analysis of the misidentified pairs of classes, we developed a generic classification for mismatches.

A. Generic Mismatch Classifications

During our analyses we discovered, among others, the following generic classification of mismatches, which was helpful for further manual matching processes.

1) Different Standards: Different standards that were used as baseline for the ontologies are one reason for differences. In our case, ATMONTO classes cover the needs of the FAA and AIRM classes cover European needs. This can lead to differences according to the underlying standards.

2) Different Level of Abstraction: Classes can have different levels of abstraction, wherein at each level, a different degree of information content is captured in the ontologies.

3) Same Level of Abstraction, Different Level of Detail: Another generic group is defined by the same level of abstraction but with a different level of detail.

4) Classes vs. Properties: Due to different design decisions that were made for the two ontologies, often classes from ATMONTO can be mapped to AIRM properties.

A more comprehensive classification of mismatches relating to existing literature on ontology mismatches will be the subject of a separate publication.

V. Matching Results

This section discusses the manual matching results performed by ATM experts. In particular, we present examples of identified exact and light matches as well as mismatches. We refer to the entity descriptions in the respective ontologies [13], [15] and the AIRM UML diagrams [17].

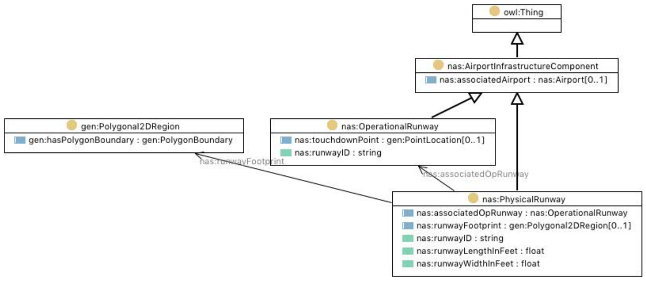

Fig. 1. ATMONTO: PhysicalRunway

A. Exact Match Example: PhysicalRunway and Runway

The class *PhysicalRunway*[4] in ATMONTO is defined as a "delimited rectangular surface region of the airport", each "associated with two operational runways, 180 degrees apart", representing the use of the runway for taking off or landing in either direction. Figure 1 illustrates the *PhysicalRunway* class and its relationships.

The AIRM-O class *Runway*[5] matches *PhysicalRunway* in ATMONTO and denotes a "rectangular area on a land aerodrome prepared for the landing and take-off of aircraft" as defined by ICAO Annexes 1 and 14. Figure 2 illustrates the *Runway* class and its relationships in AIRM while Figure 3 illustrates the corresponding representation in AIRM-O using VOWL notation (see [21]). A *Runway* has *RunwayDirection*, is part of *ManoeuvringArea* and situated at an *Aerodrome*. The *Runway* class is also connected with *Taxiway*.

Both the *PhysicalRunway* in ATMONTO and the *Runway* class in AIRM-O have a similar perspective and have been classified as an exact match. AIRM-O, however, models the notion of runway in greater depth.

B. Light Match Example #1: GroundDelayProgramTMI and ATFMMeasure/GroundDelayProgramme

The class *GroundDelayProgramTMI*[6] in ATMONTO represents a Ground Delay Program (GDP) traffic management initiative (TMI). A TMI "is an orchestrated air traffic management procedure implemented as needed to control the flow of air traffic in the NAS based on capacity and demand"[7]. A GDP delays an aircraft's departure airport for the purpose of managing demand and capacity at the aircraft's destination airport.

In ATMONTO, all TMIs have the same basic properties in common and properties specific to a particular type of TMI are defined in the various subclasses of *TrafficManagementInitiative*. All TMIs express a set of conditions under which the TMI is valid, and a set of constraints on aircraft, airports, and/or airspace facilities to which the TMI applies. In order to describe these constraints, ATMONTO employs the same underpinning set of abstract, reusable classes across all TMIs where possible. For example, both *GroundDelayTMI* and *GroundStopTMI* are associated with a class called *DelayModel*. The class *DelayModel* comprises parameters associated with the computational delay model used in determining and assigning delay times to the aircraft involved in the ground delay or ground stop. The classes *AirportSpec* and *FlightSpec* are used to constrain the set of airports or flights to which a given TMI pertains. The class *ReRouteSegment* links a reroute TMI with the set of reroute flight paths that are authorized for this reroute.

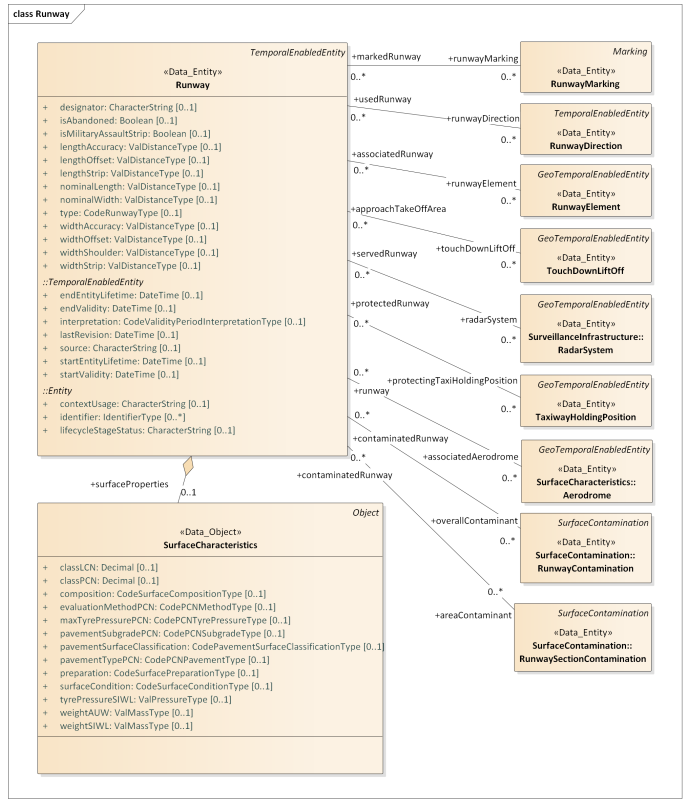

Fig. 2. *Runway* class in AIRM and its relationships, excerpt from the AIRM UML model [17]

When matching ATMONTO and AIRM-O / AIRM, two classes in AIRM-O / AIRM are candidates for a light match with *GroundDelayProgramTMI*. First, the class *ATFMMeasure*[8] (where type is Ground_Delay) matches *GroundDelayProgramTMI*, but the depth of representation is different. In ATMONTO, the details of the TMI are made explicit using various properties to specify its conditions and constraints. In

[4]https://data.nasa.gov/ontologies/atmonto/NAS#PhysicalRunway, see https://data.nasa.gov/ontologies/atmonto/doc/nas_PhysicalRunway.html

[5]AIRM-O: https://w3id.org/airm-o/ontology#Runway, AIRM UML: urn:aero: airm:1.0.0:ConceptualModel:Subjects:BaseInfrastructure:Aerodrome Infrastructure:Runway

[6]https://data.nasa.gov/ontologies/atmonto/ATM#GroundDelayProgramTMI, see https://data.nasa.gov/ontologies/atmonto/doc/atm_GroundDelayProgram TMI.html

[7]See https://data.nasa.gov/ontologies/atmonto/doc/atm_TrafficManagement Initiative.html

[8]AIRM-O: https://w3id.org/airm-o/ontology#ATFMMeasure / AIRM UML: urn:aero:ses:eurocontrol:airm:1.0.0:ConceptualModel:Subjects:AirTrafficOp erations:DemandAndCapacityBalancing:ATFMMeasure

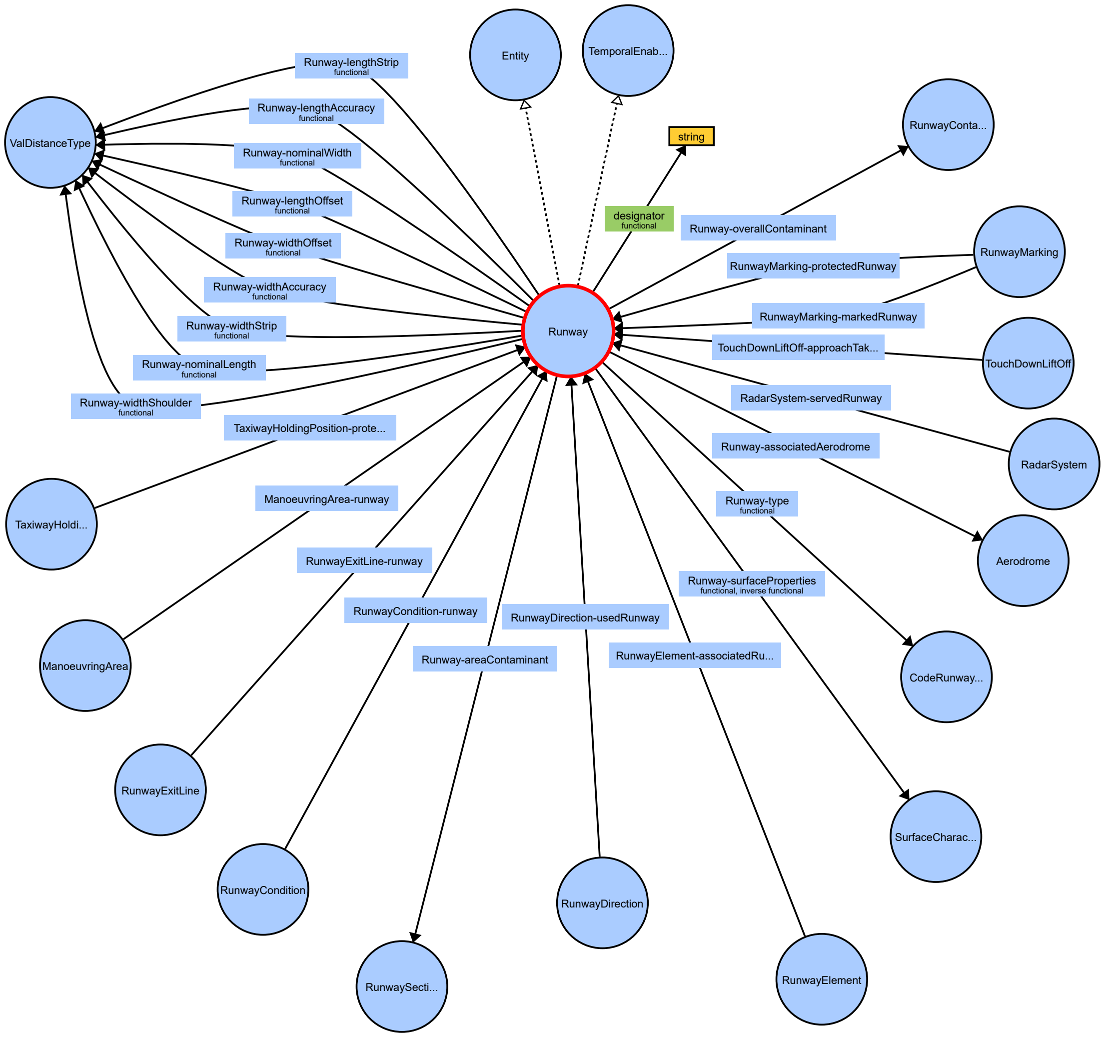

Fig. 3. AIRM-O: Runway

AIRM-O, there is much less explicit information about the program. Second, the AIRM class *GroundDelayProgramme*[9], which is not part of AIRM-O, represents a strategic, pre-tactical, or tactical Air Traffic Flow Management (ATFM) measure in the course of which an aircraft is ordered to stay on the ground for the purposes of managing capacity and demand in a specific volume of airspace or at a specific airport. This class might be the better match but still does not fulfill the criteria of an exact match.

C. Light Match Example #2: Sector and Airspace/NavigationAreaSector

The class *Sector*[10] represents a "defined volume in the airspace of an [Air Route Traffic Control Centers (ARTCC)] or [Terminal Radar Approach Control (TRACON)] whose flight traffic is typically controlled by a single controller". A sector contains multiple layers, each of which is a polygonal volume defined by an identical surface boundary polygon on the top and bottom, and vertical sides (called a *ShearSidedPolygonalVolume*). The sector is modeled as a vertical stack of these volumes forming a type of jagged layer cake structure. Each layer is linked to the sector via the property *hasSectorLayer*. Any sector immediately adjacent to (i.e., touching) the sector is linked via the property *adjacentSector*, and the ARTCC in which the sector is located is linked via the property *locatedInCenter*.

The class *Airspace*[11] almost matches *Sector*, even though the notion of airspace seems more general than sector, intuitively. An airspace in AIRM is a "defined three dimensional region of air space relevant to air traffic". Yet, the *controlSector* property of *SectorConfiguration*, which links to an *Airspace*, indicates that it is "a subdivision of a designated control area within which responsibility is assigned to one

[9]urn:aero:airm:1.0.0:ConceptualModel:Subjects:AirTrafficOperations:DemandAndCapacityBalancing:GroundDelayProgramme

[10]https://data.nasa.gov/ontologies/atmonto/NAS#Sector, see https://data.nasa.gov/ontologies/atmonto/doc/nas_Sector.html

[11]AIRM-O: https://w3id.org/airm-o/ontology#Airspace, AIRM UML: urn:aero: airm:1.0.0:ConceptualModel:Subjects:AirspaceInfrastructure:Airspace:Airspace

controller or to a small group of controllers". This matches the *Sector* definition in ATMONTO.

The class *NavigationAreaSector*[12] represents a subdivision of a *NavigationArea* and is also a close fit to ATMONTO's *Sector*. The relationship between *NavigationArea* and *Airspace* is different; as a matter of fact it is a light match.

In addition three other classes have also been identified as possible candidates but in the end have not been selected. AIRM-O's *SectorConfiguration* is a specification of how the airspace is configured and defines the sectors via the *controlSector* property. *Sector* in ATMONTO matches *Airspace* most closely. In AIRM, *Sector* is represented as the *controlSector* property of *SectorConfiguration*, which links to an *Airspace*.

The notion of a *Sector* in ATMONTO seems less general than the representation in AIRM-O because AIRM-O has a more general notion of airspace, and sectors are just one flavor of airspace; consequently, there is a representation mismatch here. Overall, the approach taken in AIRM-O seems to be to avoid making lots of subclasses and to instead keep the concepts general. The *Sector* class from ATMONTO should probably be considered a subclass of *Airspace* in AIRM-O. In AIRM-O, such subclass relationships are defined implicitly using specific subclassing properties that point to the general class *Airspace*. This approach is also used, for example, with the *SignificantPoint* concept. ATMONTO has many subclasses of AIRM-Os *SignificantPoint*, including the different types of fixes: *MeterFix*, *VORfix*, *IntersectionFix*, etc.), but the closest matching class is probably *PointLocation*.

D. Mismatch Example #1: AirportRoute

An *AirportRoute*[13] in ATMONTO is "a route within a SID or STAR that connects the common route to one of multiple airports that use the SID/STAR". SIDs and STARs are defined as an approach or departure pattern that includes three components: a set of entry routes into the SID/STAR, a common route flown by all aircraft in the SID/STAR, and a set of exit routes. An *AirportRoute* corresponds to an entry route (for a SID, on departure) or an exit route (for a STAR on arrival). Its superclass is *AirspaceRoute*, defined as a sequence of navigational elements (e.g., named fixes, routes) defining a path through the airspace. Note that this notion of a route was based on a representation employed in one specific FAA data source, and therefore may not adequately reflect an expert conceptual view of SIDs, STARs, and routes.

During the manual matching process, several classes had been identified as match candidates but none of them actually matches the ATMONTO *AirportRoute* class. One of the AIRM-O classes identified was *ProcedureTransition* which seemed to be the most closely analogous class because it involves a "group of consecutive segments that are part of a branch on an approach procedure, SID or STAR". This is what an *AirportRoute* actually represents. But the way these consecutive segments are represented differs between AIRM-O and ATMONTO. Another class taken into account was *Procedure* but this class is too general compared to *AirportRoute*, which is more likely a portion of a procedure (as in *ProcedureTransition*). *Route* was another class looked into, which describes named routes through the airspace and serves as a container for *RouteSegments* and *DirectRoutes*. This makes *Route* in AIRM-O somewhat similar to *NavigationPath* in ATMONTO (*AirportRoute* is subsumed by *NavigationPath*). Route is in any case too general to be a match for *AirportRoute*. The AIRM-O class *RouteSegment* describes a route between two specified points plus some constraints on flying between the two points. Also this class is not a match for *AirportRoute* because *AirportRoute* defines multiple points and a path among them. *FlightRestrictionRoute* is a class which has the same idea of a route as given by *AirspaceRoute*. A sequence of routing elements, where the routing elements are points, route portions, or airspace(s). The elements in ATMONTO, however, do not include airspace components, so the sequence of elements here is more abstract than permitted by *AirspaceRoute*. *FlightRestrictionRoute* seems to implement a similar type of sequence, but it is done for a different purpose – namely to specify a constraint/restriction on a route, rather than to specify the specific route. *FlightRoutingElement* are the elements associated with a *FlightRestrictionRoute*; both are in some way related to the ATMONTO class, but neither is a match. *Trajectory* was the last class examined. It is certainly an over-generalization of *AirportRoute* and does not specify a sequence of *SignificantPoints* through the airspace, as does *AirportRoute*.

During discussions conducted after the manual matching exercise, different ATM experts had different viewpoints on which was the best match for *AirportRoute*: *RouteSegment*[14] or *ProcedureTransition*[15]. This diversity of opinion reflects something deep about the ambiguity of the term 'route', and its different meaning in ATMONTO versus AIRM. In ATMONTO, a route is a specified path through the airspace specified by a sequence of points or (recursively) sub-routes. There is no information on altitude, speed, or heading associated with an ATMONTO route; the route is simply a geometric representation of the path. Whether there is a corresponding notion in AIRM is unclear. It is also important to mention that the ATMONTO notion of route is a non-expert notion and may be misnamed. This is important to note if automated matches will be relying on lexical matches.

[12] AIRM-O: https://w3id.org/airm-o/ontology#NavigationAreaSector, AIRM UML: urn:aero:airm:1.0.0:ConceptualModel:Subjects:AirspaceInfrastructure:Airspace:NavigationAreaSector

[13] https://data.nasa.gov/ontologies/atmonto/NAS#AirportRoute, see https://data.nasa.gov/ontologies/atmonto/doc/nas_AirportRoute.html

[14] AIRM-O: https://w3id.org/airm-o/ontology#RouteSegment, AIRM UML: urn:aero:airm:1.0.0:ConceptualModel:Subjects:AirspaceInfrastructure:RouteAndProcedure:RouteSegment

[15] AIRM-O: https://w3id.org/airm-o/ontology#ProcedureTransition, AIRM UML: urn:aero:airm:1.0.0:ConceptualModel:Subjects:AirspaceInfrastructure:RouteAndProcedure:ProcedureTransition

E. Mismatch Example #2: MeteorologicalCondition

A *MeteorologicalCondition*[16] is a "representation of the meteorological status for the specified time period, including sky, wind, visibility, and weather subcomponents". The class *MeteorogicalCondition* is the central organizing class for describing weather conditions, including present and projected/forecast conditions. The class *MetCondition* references *MeteorologicalCondition* and its sub-classes, providing details of the sky, surface, weather phenomena, and visibility conditions. This basic class is used to uniformly describe meteorological conditions as reported by Aviation System Performance Metrics (ASPM), METAR (Aviation Routine Weather Report), and Terminal Area Forecasts (TAF). In TAF reports (*TAFreport*), forecasts are represented as sequences (*Sequence*) of meteorological conditions, each with its own validity timeframe.

Only two AIRM-O classes seem to focus on the meteorological condition. First, the class *WeatherCondition*[17] is focused on en-route conditions in an airspace volume or point. *WeatherCondition* describes weather conditions and phenomena primarily. Subclasses of *WeatherCondition*, however, provide details on temperature, pressure, visibility (*AviationCondition*), and the sea state (*SeaCondition*). *WeatherCondition* has the following properties: *analysisTime* defining the start time of the process of the observation or forecast, *changeIndicator* describing the type of change of meteorological conditions, *confidence* defining the "quality of trusting for a forecast expressed as a percentage", *probability* expressing the "relative likelihood of a forecast expressed as a percentage", *phenomenon* indicating the specific weather condition observed or forecast, and *contour* outlining a particular weather situation within the forecast or observation.

The second candidate match for *MeteorologicalCondition* was *AerodromeCondition*[18], which focuses on weather at the airport. *AerodromeCondition* describes sea and cloud conditions. This class is defined in conjunction with *FlightRestriction*, and is not very detailed. The property *qfe* is the atmospheric pressure at the elevation of the aerodrome. Another property is *qnh*, which is the "Q Code corresponding to the derived atmospheric pressure at Mean Sea Level, based on the atmospheric pressure at the reference point converted using the characteristics of the ICAO Standard Atmosphere". The *aerodromeSeaState* conditions are "typically reported together at an aerodrome" whereas the *cloudCondition* are at the aerodrome and the *aerodrome* property refers to the aerodrome that the observation or forecast is about. *MeteorologicalCondition* is more focused on airport weather. Both of the match candidates describe limited aspects of the weather in comparison with ATMONTO *MeteorologicalCondition*, which covers sky, wind, visibility, and weather, and is modelled after METAR, primarily. The ATMONTO representation of these aspects is very granular and explicit in contrast with AIRM-O. *AerodromeCondition* seems to include sky conditions, but not other components found in *MeteorologicalCondition. METARreport* is similar to *AerodromeCondition*. But *MeteorologicalCondition* is the better match candidate for AerodromeCondition. Nevertheless both candidates are mismatches due to the different meteorological standards used by the two ontologies.

[16]https://data.nasa.gov/ontologies/atmonto/data#MeteorologicalCondition, see https://data.nasa.gov/ontologies/atmonto/doc/data_MeteorologicalCondition.html

[17]AIRM-O: https://w3id.org/airm-o/ontology#WeatherCondition, AIRM UML: urn:aero:airm:1.0.0:LogicalModel:Subjects:Meteorology:WeatherCondition

[18]AIRM-O: https://w3id.org/airm-o/ontology#AerodromeCondition, AIRM UML: urn:aero:airm:1.0.0:LogicalModel:Subjects:Meteorology:AerodromeCondition

VI. Discussion and Lessons Learned

Alongside the manual matching and classification we tried to use ontology matching systems for the task of aligning ATMONTO and AIRM-O. A large number of automatic matching systems have been developed over the last two decades and many of them perform very well in the annual benchmarking campaign for such systems – the Ontology Alignment Evaluation Initiative (OAEI) [22]. We experimented with three state-of-the-art automatic matching systems, all of which are top contenders in the OAEI campaign: AgreementMakerLight (AML) [23], LogMap [24], and YAM++ [25]. These systems were evaluated against the manually created mapping described in the previous sections.

When matching ATMONTO and AIRM-O, the employed automated matching systems were generally able to avoid false positives, especially LogMap; the automatice matchers, however, failed to identify the majority of equivalence relations. All three matching systems were able to identify the correct equivalence relations where the source and target class names are exact string matches. Most relations in the manual match, however, could not be detected by the employed automatic matching systems.

The differences in the two ontologies not only reflect the difference in architectural design between the ATM systems in the US and Europe, but there are also semantic differences, discrepancies in the level of detail, and of course in the design of the ontologies themselves. For automatic matching systems it is not easy to distinguish between concepts that are only related superficially. Even for ATM experts working in the information modeling area for years, it is sometimes hard to tell if there is match or not. Matching complex ontologies requires a mix of automatic matching systems with manual classification and matching support by experts in order to produce truly useful results.

VII. Conclusion

Within this comparative study we evaluated not only the correlations but also the discrepancies between two ATM ontologies – ATMONTO and AIRM-O – in the course of performing a manual matching. To conduct this match, we introduced in Section IV a matching methodology for mapping concept terms from one ontology to the other. As part of this methodology, we defined three categories signifying the degree of match. The results from matching ATMONTO and AIRM-O showed there is considerable conceptual overlap but that there is also diversity and divergence due to differences in the US and European ATM systems. A detailed comparison and analysis has been made to identify possible harmonization actions on both sides. In section V we showed examples of exact matches, light matches and mismatches. This will help aviation stakeholders in the future to build, integrate and adapt their ATM systems more easily.

ACKNOWLEDGMENT

Part of this work was conducted as part of the BEST project. This project received funding from the SESAR Joint Undertaking under grant agreement No 699298 under the European Union's Horizon 2020 research and innovation program. This work was also supported by the NASA Airspace Operations and Safety Program. The views expressed in this paper are those of the authors.

REFERENCES

[1] "The Aeronautical Information Exchange Model (AIXM)," http://aixm.aero/ (Accessed: 12 July 2019).

[2] "The Flight Information Exchange Model (FIXM)," https://fixm.aero/ (Accessed: 12 July 2019).

[3] "Weather Information Exchange Models and Schema (WXCM-WXXM-WXXS)," http://wxxm.aero/ (Accessed: 12 July 2019).

[4] S. Wilson, R. Suzi, and S. Van der Stricht, "The SESAR ATM information reference model within the new ATM system," in *Proceedings of the 2014 Integrated Communications, Navigation and Surveillance Conference (ICNS) Conference*, 2014, pp. L3–1–L3–13.

[5] "The ATM Information Reference Model (AIRM)," http://airm.aero/index.html (Accessed: 12 July 2019).

[6] C. Feilmayr and W. W, "An analysis of ontologies and their success factors for application to business," *Data & Knowledge Engineering*, vol. 101, pp. 1–23, 2016.

[7] P. Arnold and E. Rahm, "Enriching ontology mappings with semantic relations," *Data & Knowledge Engineering*, vol. 93, pp. 1 – 18, 2014, selected Papers from the 17th East-European Conference on Advances in Databases and Information Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169023X14000603

[8] M. Cheatham and P. Hitzler, "String similarity metrics for ontology alignment," in *The Semantic Web – ISWC 2013*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 294–309.

[9] G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[10] R. M. Keller, "Ontologies for aviation data management," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sep. 2016, pp. 1–9.

[11] C. C. Insaurralde and E. Blasch, *Ontologies in Aeronautics*. Cham: Springer International Publishing, 2018, pp. 67–85. [Online]. Available: https://doi.org/10.1007/978-3-319-75058-3_6

[12] R. M. Keller, *The NASA Air Traffic Management Ontology: Technical Documentation, NASA Tech Memo TM-2017-219526, National Aeronautics and Space Administration*, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170006095.pdf, NASA, June 2017.

[13] ——, *The NASA Air Traffic Management Ontology*, https://data.nasa.gov/ontologies/atmonto, March 2018.

[14] ——, "Building a knowledge graph for the air traffic management community," in *Companion of The 2019 World Wide Web Conference*, 2019, pp. 700–704.

[15] A. Vennesland, B. Neumayr, C. G. Schuetz, A. Savulov, S. Wilson, Gringinger, and J. E., Gorman, "AIRM-O – ATM Information Reference Model Ontology," https://w3id.org/airm-o/ontology/, 2017.

[16] A. Vennesland, R. Keller, E. Gringinger, C. G. Schuetz, B. Neumayr, S. Wilson, and J. Gorman, "ATMONTO2AIRM A reference alignment between the NASA ATM Ontolgy and the ATM Information Reference Model Ontology," https://w3id.org/airm-o/atmonto2airm/, 2019.

[17] "AIRM UML Model v1.0.0," http://www.airm.aero/developers/downloads/AIRM_v1.0.0.eap (Accessed: 28 June 2019), 2019.

[18] Object Management Group, *Ontology Definition Metamodel v1.1*, 2014, https://www.omg.org/spec/ODM/1.1/.

[19] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Modular reuse of ontologies: Theory and practice," *Journal of Artificial Intelligence Research*, vol. 31, pp. 273–318, 2008.

[20] A. Vennesland, J. Gorman, S. Wilson, B. Neumayr, and C. G. Schuetz, "Automated compliance verification in ATM using principles from ontology matching," in *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K*, D. Aveiro, J. L. G. Dietz, and J. Filipe, Eds. SciTePress, 2018, pp. 39–50.

[21] S. Lohmann, S. Negru, F. Haag, and T. Ertl, "Visualizing ontologies with VOWL," *Semantic Web*, vol. 7, no. 4, pp. 399–419, 2016.

[22] "Ontology Alignment Evaluation Initiative," http://oaei.ontologymatching.org/ (Accessed: 11 July 2019).

[23] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The agreementmakerlight ontology matching system," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2013, pp. 527–541.

[24] E. Jiménez-Ruiz and B. Cuenca Grau, "LogMap: Logic-based and scalable ontology matching," in *ISWC 2011*, ser. LNCS, vol. 7031. Springer, 2011, pp. 273–288.

[25] D. Ngo and Z. Bellahsene, "Overview of YAM++ – (not) yet another matcher for ontology alignment task," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 41, pp. 30–49, 2016.