Final version available at IEEEXplore : http://dx.doi.org/10.1109/DASC.2017.8102002

# Semantic Data Containers for Realizing the Full Potential of System Wide Information Management

Bernd Neumayr
Johannes Kepler University Linz
Linz, Austria
neumayr@dke.uni-linz.ac.at

Eduard Gringinger
Frequentis AG
Vienna, Austria
eduard.gringinger@frequentis.com

Christoph G. Schuetz
Johannes Kepler University Linz
Linz, Austria
schuetz@dke.uni-linz.ac.at

Michael Schrefl
Johannes Kepler University Linz
Linz, Austria
schrefl@dke.uni-linz.ac.at

Scott Wilson
EUROCONTROL
Brussels, Belgium
scott.wilson@eurocontrol.int

Audun Vennesland
SINTEF
Trondheim, Norway
audun.vennesland@sintef.no

*Abstract*— **In order to unleash the full potential of System Wide Information Management (SWIM), the BEST project (Achieving the Benefits of SWIM by Making Smart Use of Semantic Technologies) proposes the semantic container approach which shields service and application developers from the complexities of data provisioning in Air Traffic Management (ATM). In combination with SWIM, semantic containers facilitate the emergence of a marketplace of value-added information services, and allow for complex derivation chains of data sets. Along these derivation chains, existing data are intelligently filtered and prioritized as well as combined and annotated with additional information.**

***Keywords— Aeronautical Information Management; Air Traffic Management; Semantic Web Technologies; OWL***

## I. Introduction

In order to foster common situational awareness among ATM stakeholders, the upcoming SWIM concept promotes a service-oriented architecture for information sharing in the aeronautical domain. To this end, SWIM relies on standardized exchange models such as the Aeronautical Information Exchange Model (AIXM) [1], the Flight Information Exchange Model (FIXM) [2], and the Weather Information Exchange Model (WXXM) [3], as well as semantic data models such as the ATM Information Reference Model (AIRM) [4], which facilitate information exchange among stakeholders. The SWIM registry of information services becomes the central hub for information exchange. In this regard, SWIM can be thought of as a gigantic whiteboard which different authorities fill with messages, the standardized exchange models being the language of the messages.

While the SWIM concept in its current form already positively affects software architecture and software development in the aeronautical domain, there are certain limitations that deter SWIM from realizing its full potential. For example, by itself, SWIM lacks explicit facilities for filtering and aggregation of data sets. SWIM also lacks administrative metadata for capturing quality, provenance, and semantics as well as temporal and spatial facets of the data. Developing value-added data services and applications in SWIM will encompass finding, selecting, filtering and composition of data from different sources (the 'data logic'). Without dedicated support for these tasks, the data logic will likely be hard-coded in applications and service implementations, intertwined with business and presentation logic, which hinders reuse and negatively affects scalability. The complexities of the data logic will likely absorb most of the developer's attention, restraining her from developing novel applications and value-added services. Returning to the gigantic whiteboard metaphor, the gigantic whiteboard which authorities fill with messages becomes difficult for stakeholders to overlook, who then lose focus on the messages that are relevant and required for a specific task. Picking the appropriate messages from the whiteboard becomes a tedious and error-prone activity.

The BEST project's semantic container approach as presented in this paper aims to provide a data-centric perspective on information services in SWIM. A semantic container encapsulates the data logic, clearly separated from business and presentation logic. Each semantic container provides an application or service with all the relevant and required data, and hides the complexity of data provisioning. Semantic containers come with metadata that allow users, services, and applications to judge the freshness and quality of the data. Based on a formal ontology-based specification of an information need for a specific operational scenario, the semantic container system discovers semantic containers as well as the missing processing steps necessary to generate a semantic container that fulfills the specified information need. In this regard, each semantic container serves as a pair of magic goggles for looking at the gigantic whiteboard that contains the authorities' messages, displaying only the relevant and required messages for a specific task.

The remainder of this paper is organized as follows. Section II describes an operational scenario in ATM that serves to motivate and illustrate the semantic container approach. Section III presents the main concepts of the semantic container approach. Section IV presents experimental evaluation of the scalability of using standard ontology language OWL for semantic container discovery.

## II. RELATED WORK

Different takes on applying semantic web technologies and ontologies for aeronautical information management exist; we refer to Keller [12]for a survey of previous research efforts in this direction. In particular, Keller et al. [13] investigate the use of RDF triple stores for integrating various types of ATM data from multiple sources. This ATM data integration architecture allows for the translation of various data sources into RDF format using an "RDF-based ontology" before loading these data into an RDF triple store. The integrated aeronautical data can then be queried using the SPARQL query language. Scalability of the presented approach, however, is unclear. In contrast, the semantic container approach does not propagate conversion of individual data items into an ontology, but describes characteristics of sets of data items, the expected amount of data to be handled being thus considerably lower.

Ongoing research [14] aims at extending the WSDOM ontology, which allows for the semantic description of web service interfaces in the aeronautical domain, with support for geospatial concepts. To this end, GeoSPARQL serves as representation and query language for web service discovery. The WSDOM ontology as well as the proposed framework for handling geospatial information are orthogonal to the semantic container approach as presented in this paper. While semantic containers are frequently the result of web services, we do not focus on the web services as such but on the management and discovery of data sets. To this end, we introduce the notion of semantic containers and employ ontologies for the semantic description of container contents.

In previous work [15] we present an abstract data model for the semantic container approach, identify types of administrative metadata, and motivate the approach using as demo case the handling of derivation chains of Digital NOTAM (Notices to Airmen) containers. We disregard composite data containers combining different data item types, e.g., NOTAMs and METARs (meteorological data), and do not focus on implementation aspects. With respect to previous work, in this paper, we now consider a more advanced operational scenario which requires composite semantic containers of different data item types – data provisioning for flight rerouting – and investigate scalability of the semantic container approach when using the Web Ontology Language (OWL) and the corresponding automatic reasoners.

## III. OPERATIONAL SCENARIO - REROUTING OF FLIGHTS

The Operational Scenario: Rerouting of Flights is a very good example where the full potential of BEST comes to light. This is especially the case when various data streams are aggregated together. Rerouting of flights can have various reasons. One situation that can lead to a rerouting is due to drastic fluctuations in the available capacity of airspaces. In this particular case it may be necessary to use a different flight path, which is called rerouting. Airspace can be closed because of poor weather conditions. Such extreme weather conditions can reduce the capacity of a specific airspace or airport, in worst cases the capacity can drop to zero. As a result, flight controllers have to reroute the aircraft via alternate routes, in order to accommodate the changes in capacity.

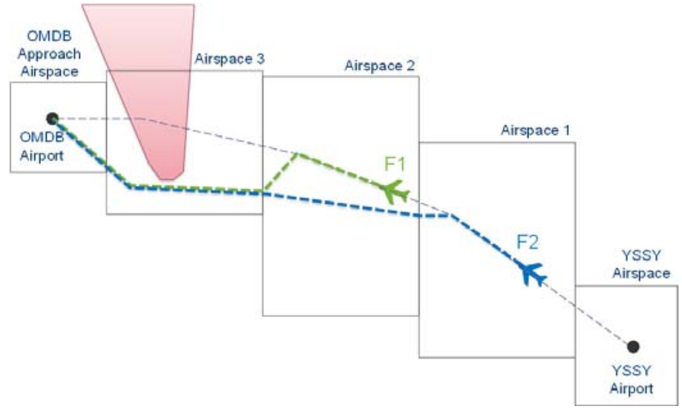


Fig. 1. Rerouting Scenario as today.

In this operational scenario due to extreme weather conditions, a closed airspace is created, and two flights need to be rerouted. As soon as an aircraft is informed about the imminent rerouting, the new flight-path can be selected without adding more delay than necessary. In Fig. 1, Flight F1 is already very near to the newly closed airspace and therefore has to take a sharper detour that adds more to its remaining trajectory as flight F2, which being further away, can select a trajectory that extends the original remaining one by a smaller percentage and doesn't require sharp maneuvering. This scenario of simulating an airspace closure was selected in order to compare today's procedures and information exchanges with the improved ones expected to be put in place when ATM will use the SWIM enabled infrastructure, and subsequently perform the comparison with the BEST concept added in place.

### A. *Current Operational Method*

Currently, the rerouting decision making relies on the experience of air traffic controllers and is not done using a formal optimization model. Flight Operations creates flight plans, prepares weather information for the flights, and considers the Notices To AirMen (NOTAM) for the flight path. For long-haul flights, the "pilot briefing" (flight-crew preparation) takes place there. The flight documents are transported by a document driver to the aircraft parking position and stored there in a document compartment. The Ramp Agent is then responsible for handing over the documents to the flight crew. For the operational process the whole ATC network available is used.

A reroute is an alternative offer to an airspace user in case of substantial delay, unavailability of a filed route or flight efficiency purposes. "What-if" reroute and group rerouting are functions within the Eurocontrol Network Manager (NM) run by Eurocontrol that are designed to assist the Network Manager Operations Centre staff to find viable alternative routes. The Enhanced Tactical Flow Management System (ETFMS) considers the routes as well as the possible flight level limitations and gives the consequent result in terms of delay, miles to fly, fuel, and route charge information. In case of significant disruption to the Network, and in order to reduce delays in a particular area or assist a flight suffering a disproportionate delay, according to the NM [5].

Fig. 1 shows the reroute scenario as handled today. The scenario implies two flights starting in Sydney (YSSY) heading to Dubai (OMDB). While the first flight enters the second airspace, the second flight is still in the first en-route airspace. Due to an extreme weather condition the parts of the Airspace 3 are closed. This influences the planed flight plan as both flights have to get around the closed airspace. During the tactical phase, the NM monitors the delay situation and where possible, identifies flights subject to delays that would benefit from a reroute. When the Network situation permits, re-routing proposals can be sent to propose more efficient routes to airspace users. This is achieved by selecting a flight and then either choose an alternative route or process all possible options by the ETFMS.

In both cases the ETFMS considers the routes as well as the possible flight level limitations and give the consequent result in terms of delay, miles to fly and Central Route Charges Office route charge information. The NM may, depending on the circumstances, consult the Aircraft Operator (AO) concerned about their final selection. Once the final decision is taken, the NM will then propose the selected route which will result in the booking of a slot for that flight and at the same time trigger the sending of a Rerouting Proposal (RRP) message to the originator, associated with the appropriate comment. AOs who wish to benefit from the offer shall consequently modify their flight plan (either with a Change (modification) message (CHG) or a flight plan cancellation message and refile using the replacement flight plan procedure. To secure the new CTOT, the CHG / new filed flight plan should be received before the respond by time in the RRP. Upon the reception of the new route in the flight plan, the ETFMS shall merge the new route with the proposal. Then messages like slot revision message or slot requirement cancellation shall be transmitted by the NM as appropriate.

Most of today's communication is based on voice. The operational process takes some time for the information exchange between all the different stakeholders involved. The Aeronautical Information Management (AIM) server publishes the NOTAM regarding the closed airspace. The involved Area Control Centres have to proceed with the rerouting of the two flights. During this time flight 1 and flight 2 are still en-route travelling according to their schedules flight plan. This will lead to a rerouting that is not as optimal as possible. In this example it leads to a larger rerouting as actually needed. In the worst case it can lead to the situation that the flight cannot be rerouted but has to travel through the bad weather condition. The following subsections show the difference with SWIM and BEST in place.

### B. *SWIM enabeld Operational Method*

SWIM was and is advertised as new paradigm for sharing ATM information. Nevertheless the chosen technology stack has proven its advantages in other domains. Instead of focusing on real benefits (e.g. better or new operational processes) SWIM is a technical enabler. Besides commonly agreed and understood data standards and information models, the new technologies used within the SWIM concept are also the enabler to do more advanced things than the operational processes can cover today.

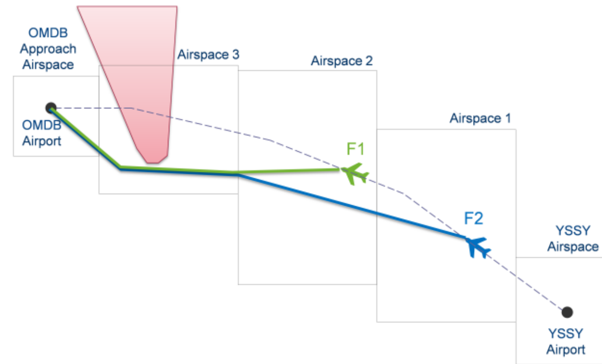


Fig. 2. SWIM enabled rerouting scenario

From an operational point there will be a lot of changes with the envisioned SWIM. Most of the changes are on the technical side. With the SWIM information exchanges in place new knowledge can be gained by combining, aggregating and semantically enhancing it. Most of the SESAR 1 projects addressed the tip of the iceberg that the SWIM concept is capable of.

The SWIM-enabled view of the operational scenario is covering Information Service Reference Model (ISRM) service usage, weather information made available on-board the aircraft and airspace closure. The operational scenario is handled via the distribution of an airspace closure notification. This is a major step to change the operational process from voice centric to service centric solution. This results in rerouting of flights involving five different systems, an AIM server, various Approach and En-Route Air traffic control (ATC) centres, a SWIM user interface displaying the bad weather condition and the trajectories received. The assumed cause for the airspace closure was an extreme weather condition, which led to a closed air space. An airspace closure triggered sending a Digital NOTAM via the SWIM infrastructure that was consequently received by all ATC systems that subscribed for receiving such information. The ATC system in charge then recalculated the trajectory and updated the shared Flight Information. In a further step, the destination airport calculated an updated the Estimated Time of Arrival (ETA), which was then sent via the SWIM infrastructure. A messaging gateway received these updates and sent text messages with the revised ETA for the affected flights to subscribers in the audience.

### C. *BEST enabeld Operational Method*

The BEST prototype will be used to demonstrate that the exchange and management of ATM information, as foreseen by the SWIM concept, can be enhanced by adding support for the filtering and aggregation of the data for specific purposes, and providing information on the data quality. BEST's semantic container concept enriches data and integrates it into a set of data items labelled with semantic meta-data adding information about, for example, freshness, quality aspects, localization, time. SWIM applications can use these enriched data sets without processing over and over again the same things that the pre-selected containers offer in a generic way.

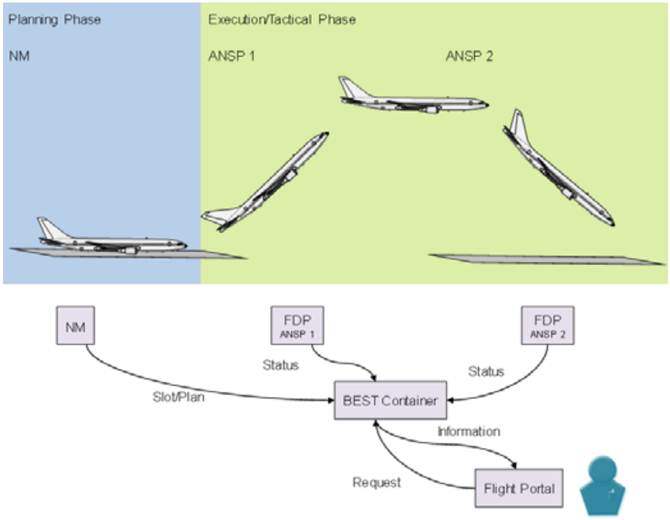


Fig. 3. BEST-enabled Flight Service

Today's problem of application and service developer is that most of the time filtering and composition is hard coded in the applications and therefore are not reusable. The BEST semantic container technique envisioned in BEST forms information needed for an operational scenario (e.g.: filtering, composition, quality attributes). It automatically identifies the missing processing steps to generate a semantic container that fulfils the specification needs. With the BEST concept in place for each of the main routes a container will exist collecting the weather, aeronautical and other relevant data. The BEST container will collect the needed information for all main inner routes and provide all SWIM applications, which will process flight plans with the pre-calculated data sets. This will save all SWIM-enabled applications this prior filtering step. Since the collection of data in a container is operational independent the BEST container can be used in a generic way. The container for a specific flight will also contain information about standard approach routes.

A BEST-enabled Flight Plan Service (see Fig. 3) intends to provide information about flights to prospecting partners via pre-calculated containers. To be able to do so, information from two other partners NM and Flight Data Processing (FDP) is required. The fact, that one partner (NM) is an organization and the other (FDP) is a system within another organization, an Air Navigation Service Provider (ANSP), is irrelevant for the example. The Flight Plan Service collects all information and refines it. The flight-plan containers will collect all available information for the main ATS routes. Upon request, the service provides the information in restructured form to a portal where users are able to query information about flights of their interest. To allow for the highest degree of transparency, all communication is established by using SWIM. This leads to improved data quality, improved information selection and prioritization, improved information presentation and reduced effort and time for ATM Users.

## IV. The Semantic Container Approach

This section discusses the key concepts of the semantic container approach along the operational scenario. First, the notion of semantic container is defined and an overview of the different kinds of metadata used to describe a semantic container is given. Next, the key reasoning task of the semantic container approach, namely matching information needs with available semantic containers is explained. The feasibility and scalability of this data discovery approach will be evaluated experimentally in Section V. In addition to this core of the semantic container approach, the remainder of this section also outlines the basic ideas of combining different semantic containers into a composite semantic container, of semantic-container-based management of value-added data, and of chains of semantic container derivation activities.

### A. *What is a Semantic Container*

A semantic container has content and description. The content is a set of data items (such as NOTAM or METAR). The description includes a membership condition and administrative metadata, such as provenance, quality, and technical metadata. A semantic container should contain all and only data items that fulfil the membership condition, quality metadata (such as a last-update timestamp) gives indications of possible deviations of actual content from membership condition.

Constituents of a semantic container:

- *Set of data items.* The *content* of the semantic container.
- *Membership condition*. Every data item that fulfils the condition is member of the set of data items.
- *Administrative metadata*. Quality, freshness, provenance, and technical metadata.

An *elementary semantic container* is a semantic container with all contained data items being instances of the same *data item type* and having the same *origin* (which are part of the membership condition). Composite semantic containers (see subsection IV.D) overcome this limitation. Note, the content and the administrative metadata of a semantic container may change over time, but the membership condition remains stable.

An example semantic container is depicted in Fig. 4. It contains all data items of type METAR originating from Bureau of Meteorology's Aviation Weather Service (bom.gov.au) and relevant for flight route YSSY-OMDB (Sydney to Dubai) on Juni 1, 2017. The content is serialized in XML format and was last changed at 11am. Since then, new METARs may have been published by the Bureau of Meteorology and have not been included in the content of the semantic container.

How a semantic container is populated (filled with data items) is not prescribed by the semantic container approach. Its membership condition is to be understood as a contract, that the container contains all data items that fulfil the membership condition, how this is realized is in the responsibility of the provider of the semantic container (MET office, information service providers that aggregate and refine source data, etc.) and indicated as part of the administrative metadata of the semantic container. It is possible that a container is filled by a human expert who does for example the filtering. In simple settings, a membership condition or parts of the membership condition may be translated to an SQL query which can be directly executed on a database. In many cases the membership condition will be too complex to be directly transformed to a database query, instead a rule-based system that encodes expert knowledge (with SemNOTAM [7] as a notable example) can be used for the population of semantic containers.

### *B. Using semantic containers for data description*

The semantic description of an elementary semantic container consists of its *membership condition* (or content definition) and *administrative metadata*.

We distinguish administrative metadata into technical metadata, quality metadata, and provenance metadata. Technical metadata include the description of a semantic container's data format. The data format itself consists of syntax and data model. Syntax may be JSON, XML, some RDF notation, etc. The data model may be AIXM, WXXM, FIXM, some RDFS/OWL ontology. Quality metadata includes timestamps describing the freshness of the data, such as last update, last check. Provenance metadata may include the service that was used to populate the container with data items.

Independent of a semantic container's quality, provenance, and technical metadata, a semantic container's content is defined by a membership condition. This content definition can be understood as a contract of what data items the semantic container contains, in the sense that every data item that fulfils the content definition is part of the semantic container. The membership condition consist of the *data item type* and *semantic*, *temporal* and *spatial facets.*

There are many different data item types in SWIM, such as METAR, TAF, and NOTAM. A primary data item type typically corresponds to a class in AIRM of stereotype IMMessage. The semantic container approach further supports secondary data item types, such as annotations of type 'NOTAM Importance'.

The membership condition defines which data items go into a semantic container and, therefore, characterizes the content of a semantic container. Membership conditions may be used to populate a semantic container or reason about subsumption of semantic containers, i.e., if one semantic container is more specific than the other, with the more general container having all the content that is in the more specific container, and possibly more.

The membership condition (e.g. 'METAR originating from Bureau of Meteorology's Aviation Weather Service and relevant for flight route YSSY-OMDB on Juni 1st, 2017') is to be understood as a defined concept which can be split up into orthogonal facets (e.g., 'data item relevant for route YSSY-OMDB'). Splitting up membership conditions into orthogonal facets supports ontology modularization. This kind of modularization allows for splitting up the reasoning tasks into smaller independent subtasks which is a key to improved scalability. Membership reasoning (for populating containers) can be done separately for each facet and then combined. Similarly, subsumption reasoning (for deriving a hierarchy of semantic containers) can be done separately for each facet and then combined.

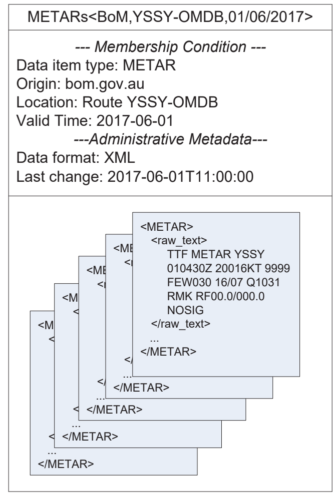


Fig. 4. A semantic container

The facet values (such as 'location:Route YSSY-OMDB' or 'validtime:2017-06-01') are to be understood as concepts (e.g., 'data item relevant for route YSSY-OMDB' or 'data item valid on 2017-02-23') that can be interpreted as sets of data items. These facet values/concepts are organized in subsumption hierarchies (e.g., 'Valid time:2017-06-01' is subsumed by 'Valid time:2017' meaning that every 'data item valid on June 1st' is also a 'data item valid in 2017').

The facets of a membership condition are grouped into temporal facets, such as valid time, spatial facets, such as location, and semantic facets, an umbrella term for other facets such as the aircraft type that the data is relevant for. The facets characterize the contents of semantic containers, and are predefined in order to allow for a decentralized network of semantic containers that can be queried. Every facet refers to an ontology. For some ontologies, the subsumption hierarchy of concepts can be generated automatically. For other ontologies, an external reasoner must compute subsumption hierarchies. For example, external reasoners are used to derive subsumption hierarchies of GML shapes or of complex temporal concepts. The choice of reasoner depends on the facet, each facet may come with a specific reasoner, which may be an automatic reasoning engine or a human domain expert.

Based on facet-specific subsumption hierarchies, membership conditions of semantic containers may be organized into subsumption hierarchies by automated reasoners. These subsumption hierarchies of membership conditions yield a hierarchization of the corresponding semantic containers from more general to more specific. Consider, for example, the membership conditions of semantic containers in Fig. 5. The membership condition of the container with label 'NOTAMs<FAA,YSSY-OMDB,2017>' subsumes the membership condition of the container with label 'NOTAMs<FAA,YSSY-OMDB,01/06/2017>'. The former semantic container contains all NOTAMs relevant for flight route YSSY-OMDB valid in year 2017. The latter is more specific a date description and contains NOTAMs relevant for the same route but only those with a valid time intersecting with day 01/06/2017.

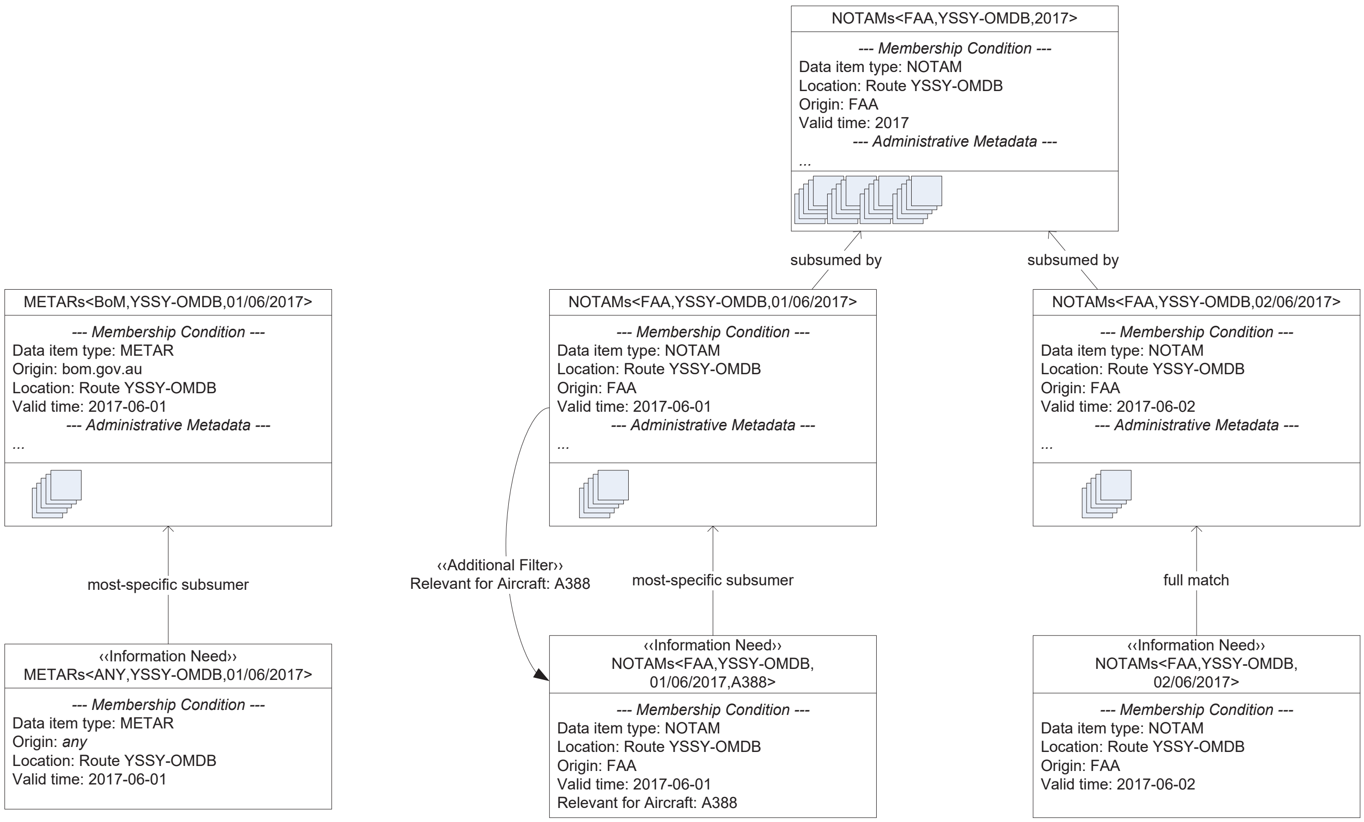


Fig. 5. Data discovery based on semantic reasoning

### C. *Using semantic containers for data discovery*

In order to fulfil a particular task, a service, application or user has a particular information need (i.e., the data/knowledge needed in order to be able to fulfil the task). For example, a pilot preparing for a flight from Sydney to Dubai on the 1st of June needs the METARs and NOTAMs relevant for this route and day.

Key concepts for data discovery in the semantic container approach are:

- *Information need* of a service, application or user; expressed as a membership condition.
- *Full match*. A semantic container that contains exactly the data to fulfil the information need, i.e., the semantic container's membership condition is equivalent to the information need's membership condition.
- *Most-specific subsumer*. A semantic container that contains all the data to fulfil the information need, i.e., the semantic container's membership condition subsumes the information need's membership condition.
- *Additional filters*. A missing processing step necessary to produce from the most-specific subsumer a semantic container that fully matches the information need.

In order to determine the semantic container that best fit the information need required for a particular ATM task, the containers' membership condition must be analysed. To this end, the information need must first be expressed as a membership condition and a subsumption reasoner can determine the semantic container that is the best match of the information need. To provide a wildcard mechanism for information need descriptions, facet-specific ontologies may contain a bottom concept (i.e., a facet-value that is subsumed by all other facet-values). For example, 'Origin:any' is such a bottom concept that is subsumed by all other origins, such as 'Origin:bom.gov.au'.

The best match of an information need may be a full match or a most-specific subsumer. In the latter case, the system should indicate which additional filters are to be applied on the most-specific subsumer to produce a semantic container that fully matches the information need. In case these additional filters cannot be directly expressed in a query or call of an existing service, it is to be understood as an input for the developer of the 'data logic' which will take care of the implementation of the filter.

For example (see Fig. 5), a pilot's information need is expressed by two information needs. Information need 'METARs<ANY,YSSY-OMDB,01/06/2017>' contains wildcard 'any' for facet origin and has the container labelled METARs<BoM,YSSY-OMDB,01/06/2017> as most-specific subsumer, no additional filtering is necessary. For information need 'NOTAMs<FAA,YSSY-OMDB,01/06/2017,A388>' the reasoner discovers container ‘NOTAMs<FAA,YSSY-OMDB,01/06/2017>’ as most-specific subsumer, and identifies ‘aircraft:A388’ as additional filter necessary to produce a full match.

### *D. Composition of semantic containers: key concepts*

Typically, a user’s or service’s information need is not satisfied by a single semantic container with all data items of the same type and from the same origin. Rather a set of semantic containers with data of different types and different provenance is needed. Composite semantic containers allow to express such complex information needs and their realization as semantic containers. Regarding composition, the following types of semantic containers are considered:

- *Elementary Semantic Container*. A set of data items of the same data item type (e.g., METARs) and same data model (e.g., WXXM), data format (e.g., XML), provenance, quality, and freshness. All data items in an elementary semantic container share the same administrative metadata which can in turn be provided ‘in bulk’ with the container.
- *Homogeneous Composite Semantic Container*. A set of data items of the same data item type (e.g., METARs), possibly with differing administrative metadata (provenance, quality, freshness). A homogeneous semantic container is a set of elementary semantic containers (fragments) of the same data item type, data model, and data format.
- *Heterogeneous Composite Semantic Container*. A set of data items of different data item types. A set of homogeneous composite semantic containers.

One of the benefits of the semantic container approach is that it allows to keep track of data provenance and quality/freshness metadata to allow judgements about the quality of the data without the need to attach (redundant) metadata to each and every data item. To avoid metadata redundancy, the elementary semantic container is the finest grain where semantic metadata is attached.

Elementary containers with the same data item type but different provenance (and thus maybe different update cycles and data quality) may be combined into a homogeneous composite semantic container. For example (see Fig. 6), five elementary METAR containers with differing values for facets location and/or origin are combined into a homogeneous semantic container labelled “METARs<YSSY-OMDB,01/06/2017>”. All semantic containers in a homogeneous composite container must also have the same data format, possibly achieved through conversion.

The composition of elementary or composite semantic containers of different data item types yields a heterogeneous composite semantic container, such as the electronic flight bag represented in Fig. 6, composed of an elementary semantic container of data item type NOTAM, an elementary semantic container of secondary data item type “NOTAM IMPORTANCE” (explained later), and a homogeneous composite semantic container of data item type “METAR”.

The membership condition of a composite semantic container is simply the union of the membership conditions of

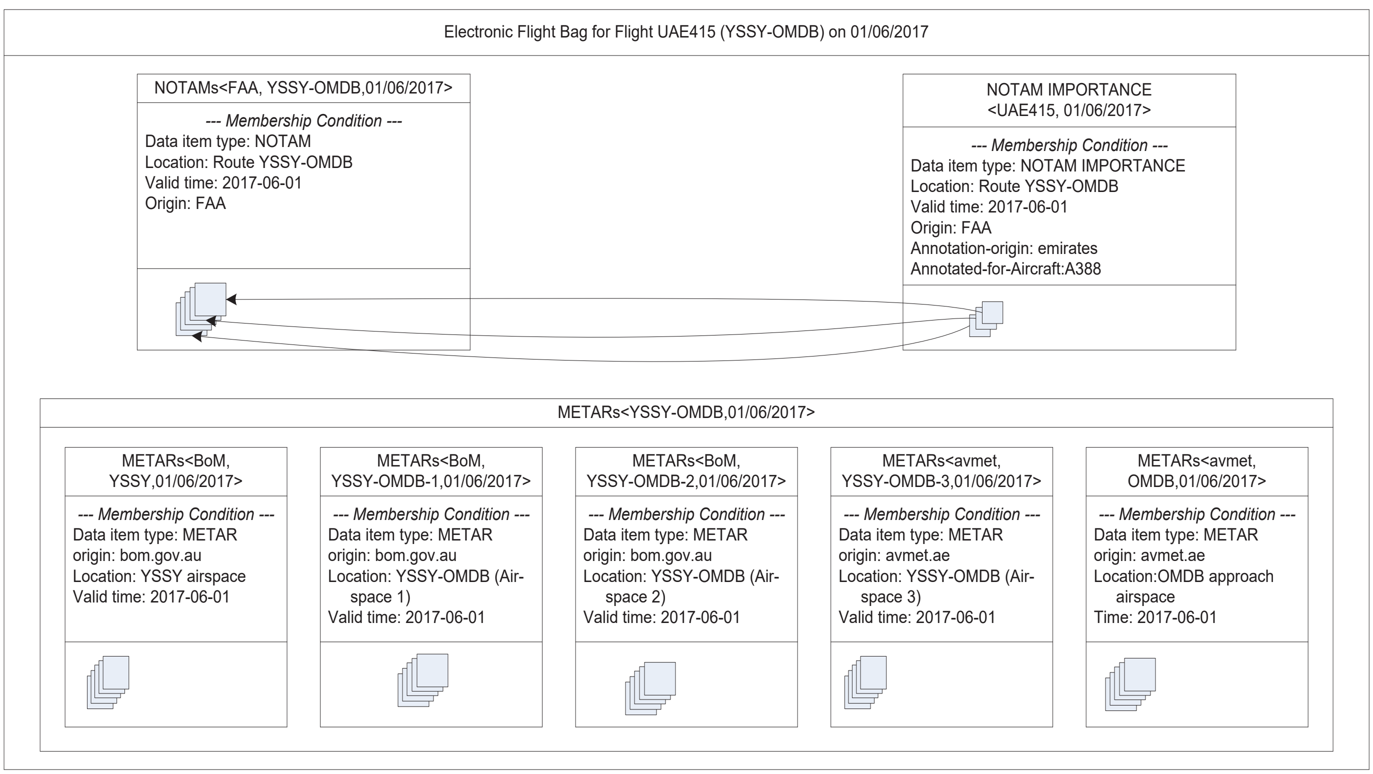


Fig. 6. A composite semantic container

its components.

## E. Value-added data in semantic containers

Primary data items in SWIM are entities and messages standardized by the information exchange models AIXM, FIXM, or WXXM and with AIRM as a common reference model. One of the assumptions of BEST is that information services may also provide secondary data which enrich the primary data. In the semantic container approach, such 'value-added' data are the result of semantic transformation, annotation, or classification. A secondary data item is attached to another data item (which it enriches), typically a primary data item.

Kinds of data items with regard to added value:

- *Primary data items*: A data item (entity, message) from AIXM, FIXM, IWXXM.
- *Secondary data items:* A secondary data item is associated with another data item.

Examples of secondary data items are the event scenario classification of a NOTAM, the transformation of a NOTAM to standard SI units, or an importance annotation of a NOTAM with regard to a specific aircraft. The membership condition of a semantic container of secondary data items is similar to a semantic container of primary data items with the main difference to have a secondary data item type and often additional facets. For example (see Fig. 6), the semantic container labelled "NOTAM IMPORTANCE <UAE415, 01/06/2017>" contains secondary data items of type notam importance. Each of these data items is associated with a NOTAM data item (indicated by the arcs in Fig. 6). The semantic container's membership condition has additional facets, namely "annotation origin" with value "emirates" and "Annotated-for-Aircraft" with "A388" as value.

## F. Derivation chains of activities and semantic containers

The main focus of this work is the description and discovery of data products as semantic containers. Additionally, the approach allows to model the whole derivation chain of semantic containers, not only focusing the static aspects of semantic containers but also the activities that derive one container from other containers. We briefly sketch the four kinds of activities for modeling derivation chains:

- *Filter*: reduce the number of data items based on a membership condition
- *Enrich*: an activity that derives secondary data items
- *Combine*: an activity with two or more semantic containers of the same data item type as input and one homogeneous composite container as output
- *Compose*: an activity with two or more semantic containers of possibly different data item types as input and one heterogeneous composite container as output. The composition may involve the filtering of one container with regard to another (including semi-joins)

This modelling approach is agnostic with regard to the implementation of the activities. For example, an activity "Filter NOTAMs relevant for aircraft type A388" could be realized by a knowledge-based system like SemNOTAM or also by a human expert who manually derives importance of NOTAMs with regard to an aircraft type.

# V. Feasibility Study

An approach suitable in a SWIM setting needs to be scalable. The automation of data discovery, matching information needs with available semantic containers, is the core service provided by a semantic container management system. In order to show the feasibility of the semantic container approach we thus need to evaluate the scalability of the associated reasoning task. The experiment focuses on reasoning over elementary semantic containers and information needs. In this section we first explain the OWL-based representation of information needs and of semantic container's membership conditions and the data discovery approach based on subsumption reasoning. We give an overview of the open-source software to realize the experiment and give details about the scalability experiment and its results.

## A. OWL representation of membership conditions and information needs

Semantic containers are represented as OWL classes with their definition (the OWL expression that comes after EquivalentTo) represented as intersection of classes from facet-specific ontologies. For example, the following two class definitions (in OWL manchester syntax) represent two of the semantic containers in Fig. 5.

```
Class: NOTAM_FAA_YSSYOMDB_01062017
EquivalentTo:
   NOTAM and Origin_FAA and
   Location_Route_YSSY_OMDB and
   ValidTime_01062017

Class: NOTAM_FAA_YSSYOMDB_2017
EquivalentTo:
   NOTAM and Origin_FAA and
   Location_Route_YSSY_OMDB and
   ValidTime_2017
```

The subsumption hierarchy of classes from facet-specific ontologies is derived independently and materialized as ontology with subclass axioms, for example:

```
Class: ValidTime_01062017
   SubclassOf: ValidTime_062017

Class: ValidTime_062017
   SubclassOf: ValidTime_2017
```

Based on this knowledge, the reasoner derives that the class NOTAM_FAA_YSSYOMDB_01062017 is subsumed by class NOTAM_FAA_YSSYOMDB_2017, meaning that the actual set of data items in the semantic container represented by the former class is a subset of the set of data items in the semantic container represented by the latter class.

Information needs are regarded as volatile and thus only represented as class expressions and not added as classes to the

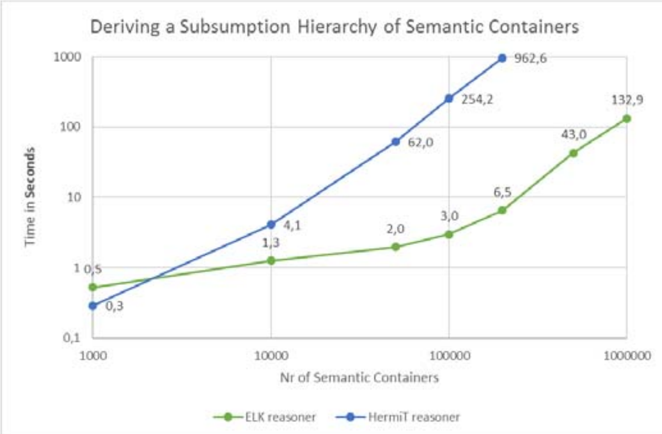


Fig. 7. Experimental results: classifying semantic containers

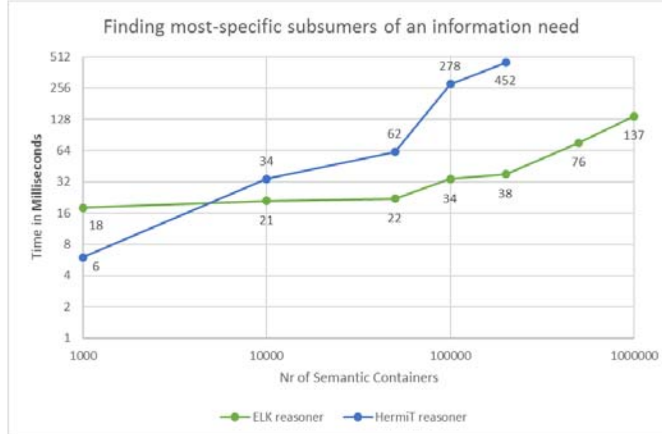


Fig. 8. Experimental results: matching information need with semantic containers

ontology. For example, one of the information needs in Fig. 5 is represented as:

NOTAM and Origin_FAA and
Location_Route_YSSY_OMDB and
ValidTime_01062017 and
RelevantForAircraft_A388

The OWL reasoner retrieves the direct subsumer in the ontology of this expression, namely NOTAM_FAA_YSSYOMDB-_01062017 which represents the semantic container which most closely satisfies the information need. From there it is easy to identify RelevantForAircraft_A388 as additional filter to produce a full match.

This OWL representation of membership conditions and information needs only uses a small subset of the constructs of OWL and is in the OWL EL profile [11]. OWL EL is a profile especially well-suited for very large ontologies and allows for very efficient subsumption reasoning.

### B. *Experimental setting and used open-source software*

The experiments were conducted on a machine with 16 GB of RAM, an Intel® Core™ i7-5600 with 2.6 GHz running Windows 10 Pro, 64 Bit.

The OWL ontologies with defined classes representing semantic containers were generated using Java (JavaSE-1.8) and the OWL API (version 3.4.3). The OWL API allows to create and manipulate OWL ontologies programmatically in Java and to parse and serialize OWL ontologies in the various OWL syntaxes. The OWL API is released as open-source software under the GNU Lesser General Public License (LGPL) or the Apache License.

For subsumption reasoning two off-the-shelf OWL 2 reasoners were evaluated: HermitT and ELK. HermiT [5] (version 1.3.8) is an open-source OWL reasoner, covering all language constructs of OWL 2 [10]. As a reasoner based on Description Logics, HermiT supports OWL 2 direct semantics [9]. The ELK reasoner [7] is an open-source reasoner for OWL 2 EL ontologies and, for this subset of the OWL language, supports very fast subsumption reasoning.

### C. *Scalability Experiment*

In the experiment, three facet-specific ontologies, membership conditions of semantic containers, and information needs are artificially generated. Each of the facet-specific ontologies contains 364 classes (possible facet values) arranged in a subclass hierarchy with a depth of five forming a tree. These subclass hierarchies represent the materialization of the result of subsumption reasoning conducted independently for each of the facet-specific ontologies. In the experiment we are not interested in subsumption reasoning over facet-specific ontologies but only in the reasoning over their combinations in membership conditions and information needs. Combining classes from these three facet-specific ontologies allows to generate up to 48228544 different membership conditions and information needs. Membership conditions and information needs are generated randomly, each taking one class as facet-value from each of the three facet-specific ontologies.

To analyze the scalability of semantic container discovery, we ran the experiment with different numbers of semantic containers (1000, 10000, 50000, 100000, 200000, 500000, 1000000 semantic containers) expressed as OWL classes and with two different OWL reasoners (HermiT and ELK).

The first reasoning task was to derive the subsumption hierarchy for the set of semantic containers. As the results of the experiment shows (see Fig. 7), this first task is quite expensive yet scales up to one million semantic containers on a single desktop machine (with ELK taking 133 seconds for the derivation). The more general HermiT reasoner ran out of memory with 500000 semantic containers. Using ELK, it takes only 6.5 seconds to derive the subsumption hierarchy of 200000 containers while HermiT already takes 1000 seconds. For smaller sets of semantic containers (1000-10000 containers) both ELK and HermiT perform well, taking up to 1.3 and 4 seconds, respectively, for deriving the subsumption hierarchy of semantic containers.

The second reasoning task was to find for each of four generated information needs (each expressed as an OWL class expression) the most specific subsumers (i.e., the direct subsumers of the OWL class expressions). Based on the derived subsumption hierarchy, finding the direct subsumers of a class expression is very fast with both reasoners (see Fig. 8):

Given a subsumption hierarchy of 200000 containers and an information need it took HermiT 452 milliseconds and ELK only 38 milliseconds to get the set of most specific subsuming containers. Even with a million of semantic containers, it took ELK only 137 milliseconds to get the set of most-specific subsumers of a given information need

## VI. CONCLUSIONS

In this paper the advantages of SWIM and of BEST's semantic container approach were described along the operational scenario 'rerouting of flights'. The paper introduced the key concepts of the semantic container approach and evaluated the scalability of its realization using standardized ontology language OWL and off-the-shelf OWL reasoners.

The proposed approach facilitates the organization of data of various types and origins relevant for an operational scenario into semantic containers and to keep track of metadata associated with the data. Ontology-based description of information needs and of the contents of semantic containers facilitates automated data discovery, making it possible to reuse already prepared collections of data, avoiding the need to redundantly implement data collection and preparation. Organizing data into semantic containers makes it easier to keep track of changes to the data. This is especially important in operational scenarios like 'rerouting of flights' where a lot of new data items are created and need to be provided in a well-organized manner to affected applications and users.

Employing the semantic container approach in the large-scale setting of SWIM makes scalability a key requirement. The feasibility study showed that based on an adequate ontology-based representation of information needs and of semantic container's membership conditions, automated data discovery is feasible in such a large-scale setting.

## ACKNOWLEDGMENT

This project has received funding from the SESAR Joint Undertaking under grant agreement No 699298 under the European Union's Horizon 2020 research and innovation program. The views expressed in this paper are those of the authors.

## REFERENCES


[1] Eurocontrol, "Aeronautical Information Exchange Model 5.1," 2014. http://www.aixm.aero/. Accessed 28 May 2017.

[2] Eurocontrol, "Flight Information Exchange Model 4.0," 2016. http://www.fixm.aero. Accessed 28 May 2017.

[3] Eurocontrol, "Weather Information Exchange Model 2.0," 2017. http://www.wxxm.aero. Accessed 28 May 2017.

[4] Eurocontrol, "ATM Information Reference Model 4.1.0," 2017. http://www.airm.aero. Accessed 28 May 2017.

[5] Eurocontrol, "ATFCM Operations Manual - Network Operations Handbook," Network Manager, Brussels, 2017.

[6] Glimm, B., Horrocks, I., Motik, B., Stoilos, G., & Wang, Z. (2014). HermiT: an OWL 2 reasoner. *Journal of Automated Reasoning*, *53*(3), 245-269.

[7] Kazakov, Y., Krötzsch, M., & Simančík, F. (2014). The incredible ELK. *Journal of Automated Reasoning*, *53*(1), 1-61.

[8] Steiner, D., Kovacic, I., Burgstaller, F., Schrefl, M., Friesacher, T., and Gringinger, E. 2016. "Semantic enrichment of DNOTAMs to reduce information overload in pilot briefings," in Proceedings of the 16th Integrated Communications Navigation and Surveillance (ICNS) Conference.

[9] W3C 2012a. OWL 2 Web Ontology Language Direct Semantics (Second Edition) - W3C Recommenda-tion 11 December 2012. https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/.

[10] W3C 2012d. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) - W3C Recommendation 11 December 2012. https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

[11] W3C 2012e. OWL 2 Web Ontology Language Profiles (Second Edition) - W3C Recommendation 11 December 2012. https://www.w3.org/TR/owl2-profiles/.

[12] Keller, R. M. 2016. "Ontologies for aviation data management," in Proceedings of the 35th IEEE/AIAA Digital Avionics Systems Conference (DASC).

[13] Keller, R. M., Ranjan, S., Wei, M. Y., and Eshow, M. M. 2016. "Semantic representation and scale-up of integrated air traffic management data," in Proceedings of the International Workshop on Semantic Big Data, pp. 1–6.

[14] Balaban, A. 2016. Testbed-12 Aviation Semantics Engineering Report. http://docs.opengeospatial.org/per/16-039.html. Accessed 28 May 2017.

[15] Kovacic, I., Steiner, D., Schuetz, G., Neumayr, B., Burgstaller, F., Schrefl, M., Wilson, S. 2017. "Ontology-based data description and discovery in a SWIM environment" in Proceedings of the 16th Integrated Communications Navigation and Surveillance (ICNS) Conference.