SEMANTICS-BASED SUMMARIZATION OF ATM DATA TO MANAGE INFORMATION OVERLOAD IN PILOT BRIEFINGS

Christoph G. Schuetz* , Bernd Neumayr* , Michael Schrefl* ,
Eduard Gringinger** , Scott Wilson***
*Johannes Kepler University Linz , **Frequentis AG , *** EUROCONTROL

Keywords: *Air Traffic Management, System Wide Information Management, Semantic Containers, Semantic Web Technologies, Business Intelligence, Online Analytical Processing*

Abstract

Pilot briefings, in their traditional form, drown pilots in a sea of information. Rather than unfocused swathes of air traffic management (ATM) information, pilots require only the information for their specific flight. In this paper, we introduce the notion of ATM information cubes. We propose a conceptual framework with merge and abstraction operations for the combination and summarization of the information that is organized in ATM information cubes. A merge operation combines ATM information from individual cells of an ATM information cube. An abstraction operation summarizes the data items within a cell, replacing individual data items by more abstract data items with summary information. The result is a management summary of relevant information.

1 Introduction

A Pre-flight Information Bulletin (PIB) provides pilots with current Notices to Airmen [7] but may also include other types of messages relevant for a flight [1], e.g., meteorological information (METARs). A Notice to Airmen (NOTAM) notifies aviation personnel about temporary changes regarding flight conditions [8], e.g., temporary closure of runways. PIBs traditionally have been delivered on paper in textual form, with limited possibilities for structuring the data. In order to alleviate this problem, electronic or (digitally) enhanced PIBs have been introduced.

An electronic or (digitally) enhanced PIB (ePIB) contains messages in digital form, which allows for improved representation and packaging of relevant information [13, 12]. Digital NOTAMs (DNOTAMs) allow for automated filtering as well as classification of messages along different dimensions (or facets), e.g., importance, geographic area, flight phase, and event scenario, that can be employed to flexibly structure the ePIB [15]. For example, using the classification rules developed in the *Semantic NOTAM* (SemNOTAM) project [15], DNOTAMs can be packaged into *semantic data containers* [10], each container comprising, e.g., the DNOTAMs relevant for a certain flight on a particular date. Consider, for example, the semantic containers on the left-hand side of Figure 1. These containers hold the relevant DNOTAMs for different segments in a flight information region (FIR), importance levels, and flight phases. The first container holds the relevant DNOTAMs for the *EDDU-01* segment of the *EDDU* FIR classified as reports of an *operational restriction* for the *cruise* flight phase. The second container holds the relevant DNOTAMs for the *EDDU-02* segment classified as *flight critical* for the *descent* flight phase. Similar rule-based approaches could also be devised for messages other than DNOTAMs. Indeed, an electronic flight bag (EFB) platform may display various kinds of relevant information for a flight [6].

In this paper, we propose a conceptual framework for combination and summarization of information packaged into semantic containers, which

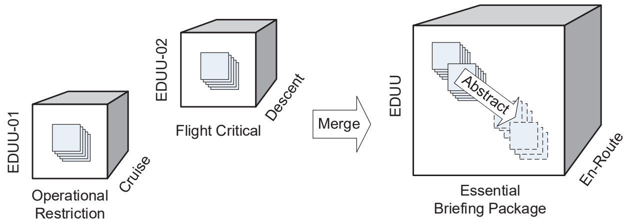

Fig. 1 Illustration of the proposed theoretical framework for semantic container operations

employs *merge* and *abstraction* operators in order to provide *management summaries* of relevant information. To that end, we adapt the well-established concept of data cubes from data warehousing and online analytical processing (OLAP) [17]. We hence propose the notion of *ATM information cube*, which hierarchically organizes semantic containers along different dimensions relating to the container content, e.g., the geographic and temporal applicability, flight criticality, and flight phase that the container content is relevant for. We assume the existence of appropriate rule-based filtering mechanisms to collect ATM information into containers. The individual containers can be merged in order to obtain more comprehensive containers of ATM information. For example, individual containers with flight critical DNOTAMs for the EDDU-02 segment when the flight is in descent phase and DNOTAMs about operational restrictions for the EDDU-01 segment when the flight is in cruise phase, respectively, are merged into a container with DNOTAMs that comprise the essential briefing package for the EDDU FIR when the flight is in an en-route phase (Figure 1). The messages themselves can also be further abstracted.

The remainder of this paper is organized as follows. In Section 2, we present background information. In Section 3, we define the notions of ATM information cube and cube of ATM information cubes. In Section 4, we define operators for flexibly combining semantic containers. In Section 5, we define operators for abstracting data items (messages) within semantic containers. We conclude with a summary and an outlook on future work.

2 Background

The semantic container approach as developed in the course of the BEST project[1] is a flexible way of compartmentalizing ATM information which complements the service-oriented architecture of SWIM (System Wide Information Management) with techniques for ontology-based data description and discovery [10, 11]. A *semantic container* consists of content and description. The content is a set of data items of a specific type, e.g., DNOTAMs, METARs. The container description defines a membership condition [10]: The data items that fulfill a container's membership condition constitute that container's content.

The membership condition describes multiple facets of the container's content. In this regard, the membership condition may refer to geographic and temporal facets of container content, but also various other semantic facets. For example, a DNOTAM container may contain all the DNOTAMs with a specific spatial and temporal scope, e.g., Vienna airport on 14 May 2018, and which refer to a specific scenario, e.g., taxiway closure. For each facet, a semantic container's membership condition hence associates a concept from an ontology. An ontology is a "formal, explicit specification of a shared conceptualisation" [16] of a real-world domain of interest and consists of multiple concepts which are hierarchically organized. For example, the *LOWW* concept represents Vienna airport and is under the *LOVV* concept which represents the Austrian FIR. Various knowledge representation languages may serve to define these ontologies. From the hierarchy of ontology concepts that make up the faceted membership descriptions derives a hierarchy of semantic containers. For example, a semantic container with DNOTAMs for Vienna airport on 14 May 2018 is more specific than a container with DNOTAMs for the Austrian FIR in May 2018. The hierarchy of semantic containers then serves to discover existing semantic containers that most closely satisfy a certain information need by a specific end user or application.

[1] `http://project-best.eu/`

Information processing and reasoning techniques at the instance level complement the metadata-centric semantic container approach – in order to fill the semantic containers with actual content. The SemNOTAM approach [15, 2], for example, employs a formal rule system to filter and annotate DNOTAMs with importance levels according to a user's interest specification. The SemNOTAM engine receives a set of DNOTAMs as input and the user's interest specification as argument. The SemNOTAM engine further translates the input into a representation that suits knowledge-based reasoning, and selects from the SemNOTAM knowledge base the relevant set of rules which the knowledge-base reasoner executes against the input ATM information. The term "filtering", in this context, refers to the disregarding of DNOTAMs from the input in the result set whereas "annotation" refers to the assignment of importance levels, e.g., flight critical, to DNOTAMs. The result of the reasoning process – a filtered and enriched set of DNOTAMs – is provided to the pilot or air traffic controller. The filtered and enriched set of DNOTAMs could also become the content of a semantic container, with the argument interest specification constituting the membership condition.

Traditional OLAP works on multidimensional models with numeric measures [17]. Going beyond numeric measures, InfoNetOLAP [3] associates weighted graphs with dimension attributes. Topological and informational roll-up are the basic kinds of operations, which are akin to merge and abstraction operations presented in this paper. The focus of InfoNetOLAP are weighted directed graphs which are unsuitable for schema-rich ATM information.

The concept of ATM information cubes builds on the ideas developed in our previous work [14] on the use of business model ontologies for the management and summarization of complex information in OLAP cubes. The cells of such an OLAP cube are associated with business knowledge that is valid in a particular context, as defined by the dimensions of the cube. The Resource Description Framework (RDF) serves as the representation language.

3 ATM Information Cubes

A semantic data container is a flexible data structure for storing data items of various different kinds (see Section 2); the concept is central to the notion of ATM information cubes. Note that we employ the terms "semantic data container" and "semantic container" synonymously.

We arrange semantic containers in ATM information cubes along multiple dimensions (or facets) of content description. For that arrangement of semantic containers, we borrow the data cube metaphor from data warehousing and OLAP: The dimensions of the ATM information cube span a multidimensional space where each point associates a set of ATM data items, e.g., DNOTAMs or METARs, rather than numeric values as in traditional data cubes. Each semantic container hence becomes associated with a point in a multidimensional space according to the container's membership condition. Consider, for example, the three-dimensional ATM information cube in Figure 2. Individual DNOTAMs are collected into semantic containers along geography, importance, and scenario dimensions. Each semantic container in that cube hence contains a set of DNOTAMs describing a specific scenario [4] for a specific geographic segment within a FIR with some importance, e.g., operational restriction or flight critical, for the flight and date which the cube has been defined for. Note that the flight and date are fixed for that cube, which constitutes additional context information necessary to correctly interpret that cube.

The dimensions characterize the ATM information cube: Their members identify points (or cells) in the cube. In order to allow for roll-up operations, i.e., viewing the content at different granularity levels (see Section 4), a cube employs hierarchically organized dimensions. Consider, for example, the dimension hierarchies in Figure 3, which illustrates the dimension hierarchies for the cube from Figure 2 with importance, geography, and scenario dimensions. The importance dimension hierarchy follows the importance classification system for DNOTAMs from SemNOTAM [15]. The scenario dimension hier-

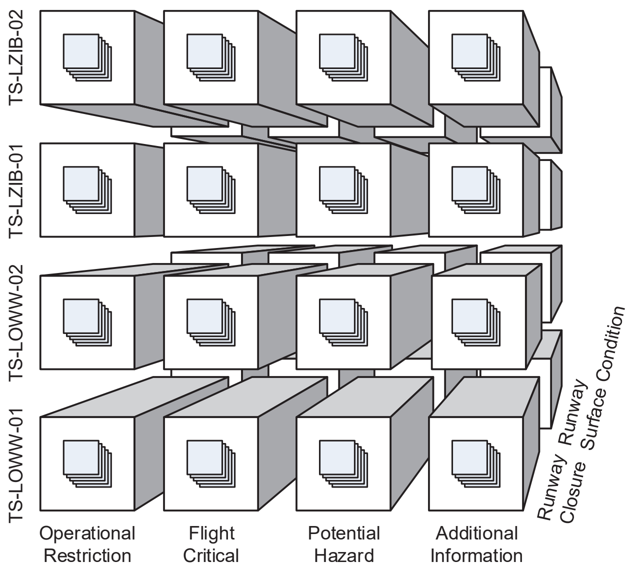

Fig. 2 An example ATM information cube with geographic, importance, and scenario dimensions

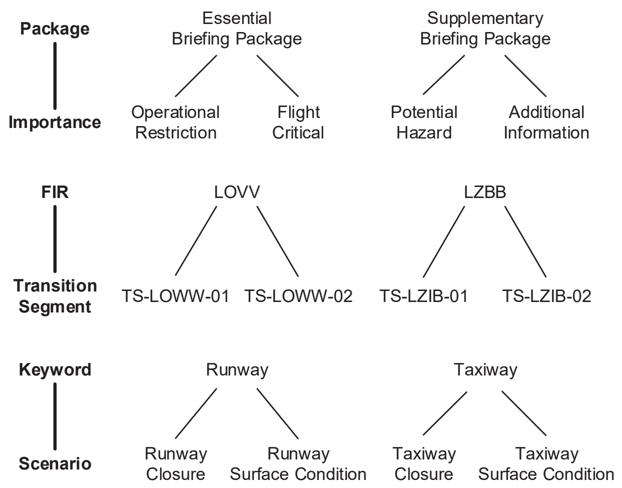

Fig. 3 Example dimension hierarchies of an ATM data cube: levels (in boldface) and level members

archy follows the organization of the FAA's specification of airport operation scenarios [4]. The geography dimension hierarchy consists only of transition segments to airports, which are assigned to a FIR. Using the roll-up relationships of the dimension hierarchies, an analyst may view, e.g., DNOTAMs per FIR rather than individual transition segments. We note that alternative roll-up relationships could be defined, e.g., to support alternative geographic classifications.

The coordinates of a container correspond to a semantic description of the data items inside the container – the container's membership condition. For example, the point identified by *TS-LOWW-01*, *Flight Critical*, and *Runway Closure* indicates that the associated semantic container comprises the DNOTAMs about runway closures that are flight critical for the *TS-LOWW-01* transition segment. Now, the attentive reader will notice two things. First, nowhere in the model has the data item type been fixed to "DNOTAM". Second, the importance of a DNOTAM depends on many things – first and foremost on the particular flight and date. Yet, the cube has no dimension for indicating flight and date. In the example, the data item type, flight, and date are implicit constants that set the context for the cube. For each flight and date, a separate cube of DNOTAMs would exist. The pilot could dynamically select containers of DNOTAMs along the dimensions within that context only. Furthermore, a cube of ATM information cubes may organize multiple individual cubes and explicitly represent the otherwise tacit context information (see Section 3.2).

The ATM information cube is potentially sparse, i.e., not every cell at the base granularity has a semantic container attached. In the following, we introduce the notion of multigranular ATM information cubes.

3.1 Multigranular ATM Information Cubes

While the example cube in Figure 2 shows an ATM information cube that associates semantic containers only with a single, base granularity, we may well imagine the existence of a multigranular ATM information cube that also associates semantic containers with coarser levels of granularity. For example, in some cases, individual DNOTAMs may not fall unambiguously into a single importance category such as flight critical or operational restriction. Consider then the ATM information cube in Figure 4: a cube that associates data items explicitly with the coarser *supplementary briefing package* and the *essential briefing package* granularity levels.

The containers associated with coarser granularities are *composite containers*. For example,

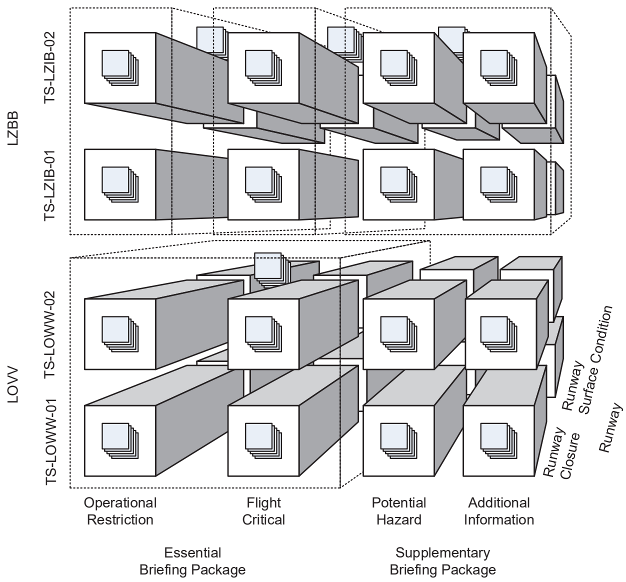

Fig. 4 A multigranular ATM information cube

in Figure 4, the cell identified by the point *LOVV*, *Essential Briefing Package*, and *Runway* (denoted by dotted lines) associates a composite container that consists of the eight component containers at the finer *segment-importance-scenario* granularity along with data items associated specifically with the coarser *FIR*, *package*, and *keyword* granularity. Therefore, on the one hand, a semantic container at a coarser granularity also (transitively) comprises the data items packaged at finer granularities. For example, if some message is flight critical for a segment of the LOVV region then, all other things remaining unchanged, that message is also in the essential briefing package for the entire LOVV region. On the other hand, the component containers "inherit" the data items that the composite container explicitly associates with the coarser granularity level: The data items propagate from the composite container to the component containers. For example, the data items generally classified as part of the essential briefing package should likewise be included in packages for operational restriction and flight critical, respectively. Similarly, the data items relevant for an entire FIR should also be included in the packages for individual transition segments within that FIR, necessitating a top-down data sharing mechanism along the level hierarchies.

Concerning the materialization of data sharing and container composition within ATM information cubes, we note the following. In theory, each possible granularity level in a cube could have a composite container associated, having component containers from the more finely grained points underneath. A composite container, when selected, should return the data items proper of the composite container as well as the component containers' data items. Materialization of these composite sets of data items would speed up performance. In practice, however, the materialization of composite containers at every possible granularity level may be infeasible due to combinatorial explosion. A common solution in data warehousing is the selection of beneficial aggregate views for materialization [5]. Materialization of downward propagation, however, is unproblematic performance-wise when the ATM information is predominantly available at the base granularity.

3.2 Cubes of ATM Information Cubes

We propose ATM information cubes being built for a certain operational context, e.g., a specific flight on a particular date. In the previous sections, a cube's operational context was not explicitly defined in the model but assumed to be implicit constants outside the model. Thus, in order to externalize that context, we propose to arrange the ATM information cubes themselves into multidimensional structures (Figure 5).

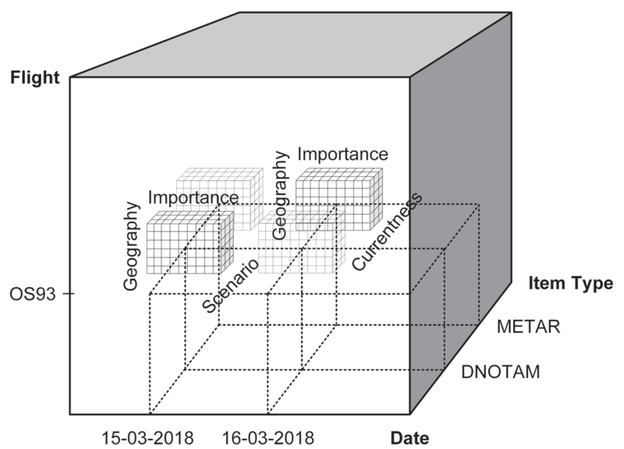

Fig. 5 A cube of ATM information cubes (metacube)

A cube of ATM information cubes – a *metacube* – hence consists of several cubes, the sets of dimensions of which will typically overlap but not necessarily be equal. For example, the metacube in Figure 5 has ATM information cubes with different dimensionality depending on the data item type. Cubes of DNOTAMs have geography, importance, and scenario dimensions whereas cubes of METARs have geography, importance, and currentness dimensions. The currentness, in this respect, refers to the precise time of the METAR's underlying observation. A drill-accross operator allows to combine the different cubes, joining via the common dimensions, with all non-common dimensions considered rolled up at the implicit *all* level (see Section 4.2). While the dimensions can be manifold, we assume data item type, flight, and date/time as the typical candidates for dimensions. A point in such a metacube may contain, e.g., a cube of DNOTAMs relevant for flight OS93 on 15 March 2018.

4 Operations on ATM Information Cubes

In this section, we present operations that allow for combining individual semantic containers that are organized in ATM information cubes.

4.1 Merge of Semantic Containers

Individual semantic containers may be aggregated along the hierarchically ordered dimensions of an ATM information cube. In this regard, the essential operation is merge-union. The *merge-union* operation takes an input cube and returns a cube with a specified coarser base granularity where the lower-level containers from the input cube are merged. The *merge-union* operation produces flat containers comprising the data items from multiple semantic containers but unlike composite containers do not preserve component containers.

Figure 6 illustrates the result of applying the merge-union operator on a three-dimensional input cube with *segment-importance-scenario* base granularity (Figure 3). The result cube has a coarser base granularity than the input cube, namely *FIR-package-keyword* granularity. The

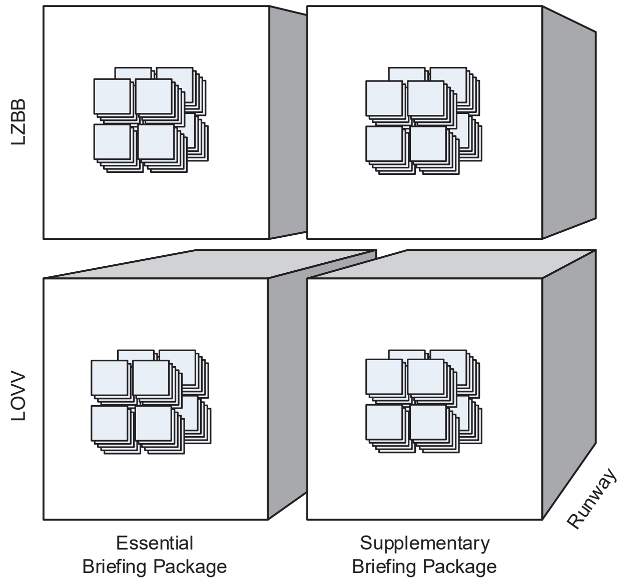

Fig. 6 Merge of the semantic data containers from Figure 2 using the hierarchies from Figure 3

containers in the output cube contain the same data items as in the input cube's containers. For example, the semantic container for *LZBB*, *Essential Briefing Package*, and *Runway* in the output cube comprises the data items from eight base containers of the input cube, which roll up to the point identified by *LZBB*, *Essential Briefing Package*, and *Runway*.

The *merge-intersect* operation aims at analyzing the information contained in multiple semantic containers by creating the intersection of the involved containers' data items, leaving the base granularity of the cube unchanged. The merge-intersect operation serves to identify the common data items of a set of multiple semantic containers. For example, a merge-intersect operation on the cube from Figure 2 with a *FIR-package-keyword* merge granularity, in order to obtain the data items for the container at the point identified by *LZBB*, *Essential Briefing Package*, and *Runway*, selects the intersection of data item sets from all semantic containers at points that roll up to that specific point. The containers at the coarser merge granularity thus receive additional data items from the containers underneath at finer granularities whereas the base containers remain unchanged by that operation.

4.2 Drill Across the Metacube

The drill-across operation combines different cubes within a metacube along the dimension hierarchies of the metacube, using the cubes' common dimensions to join the cubes. The drill-across operation takes an input metacube and returns an output metacube with a specified coarser base granularity where the lower-level cubes from the input cube are joined over their common dimensions. Consider, for example, the metacube in Figure 7, which shows the result of a drill-across operation on the three-dimensional metacube from Figure 5. The drill-across operation, in this example, changes the metacube granularity such that the data item type dimension is rolled up to the dimension's top (or *all*) level, and hence effectively reduces the dimensionality of the metacube (although formally the dimension is still there). The DNOTAM cubes in the input metacube have *Geography*, *Importance*, and *Scenario* dimensions whereas the METAR cubes have *Geography*, *Importance*, and *Currentness* dimensions. The cubes from the input metacube about DNOTAMs and METARs are joined over their common *Geography* and *Importance* dimensions. The drill-across operation first applies the merge-union operation on the DNOTAM and METAR cubes from the input metacube in order to obtain a common granularity by rolling up the *Scenario* and *Currentness* dimensions to the *all* level before obtaining cubes with both DNOTAMs and METARs.

In a way, the drill-across operation is the metacube counterpart of the merge-union operation. The drill-across operation changes a metacube's base granularity and combines the contents associated with the merged points. In case of the drill-across operation that contents are ATM information cubes. In order to sensibly combine cubes from the different points, the cubes must be joined over common dimensions, with non-common dimensions rolled up to the implicit *all* level. The drill-across operation, just like the merge-union operation, preserves all data items from the cubes in the input metacube – only their organization into ATM information cubes is different in the output cube.

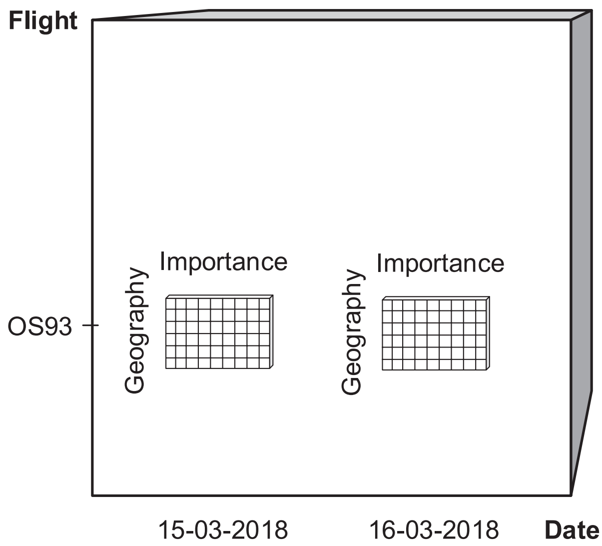

Fig. 7 A drill across the metacube from Figure 5 over the item type dimension

5 Operations on Semantic Containers

The notion of *abstraction* serves as an umbrella term for a wide variety of different operations. Abstraction, as opposed to merge and drill-across, denotes operations that produce new data items and links between data items. Originally proposed for working with RDF data [14], the principle of abstraction is independent from any concrete data or information model.

We employ UML object diagrams to illustrate the principle of the abstraction operation. Consider an object diagram (Figure 8) that shows DNOTAMs according to the AIXM information model. In particular, the object diagram shows two DNOTAMs about the surface conditions of runways as well as two DNOTAMs about the closure of runway directions. The *LOWW-16/34* runway has two layers of contaminants with an overall extent of 0.31 m: dry snow (0.29 m) and ice (0.02 m). The *LOWW-11/29* runway has a layer of ice with an extent of 0.01 m. Both runways, however, have a specified length and width already cleared of contaminants while the remainder of the runway has winter services going on, thus leading to the closure of one runway direction from each runway. The closures are due to snow and ice removal, respectively, and for each runway the

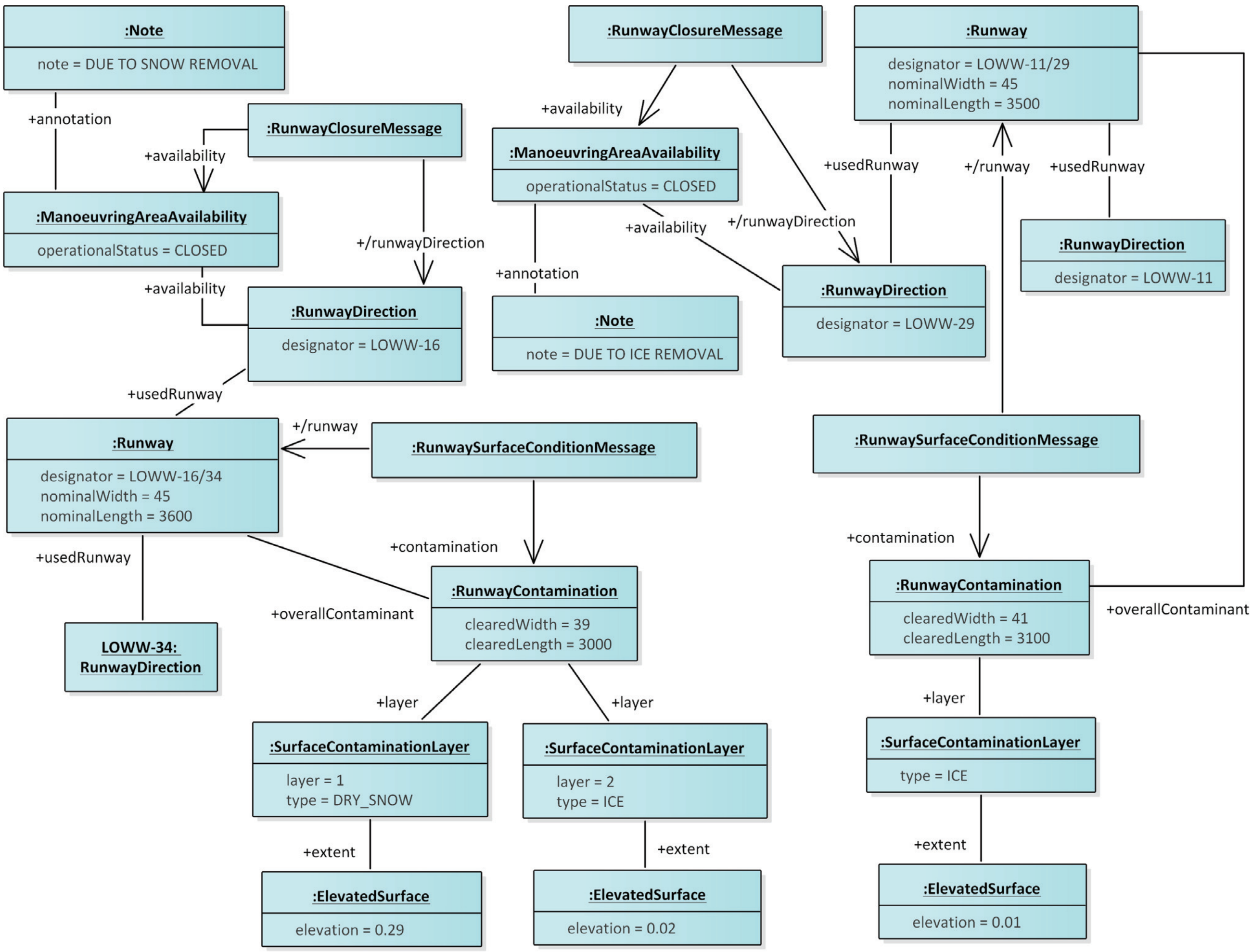

Fig. 8 An object diagram illustratings DNOTAMs according to the (simplified) AIXM metamodel

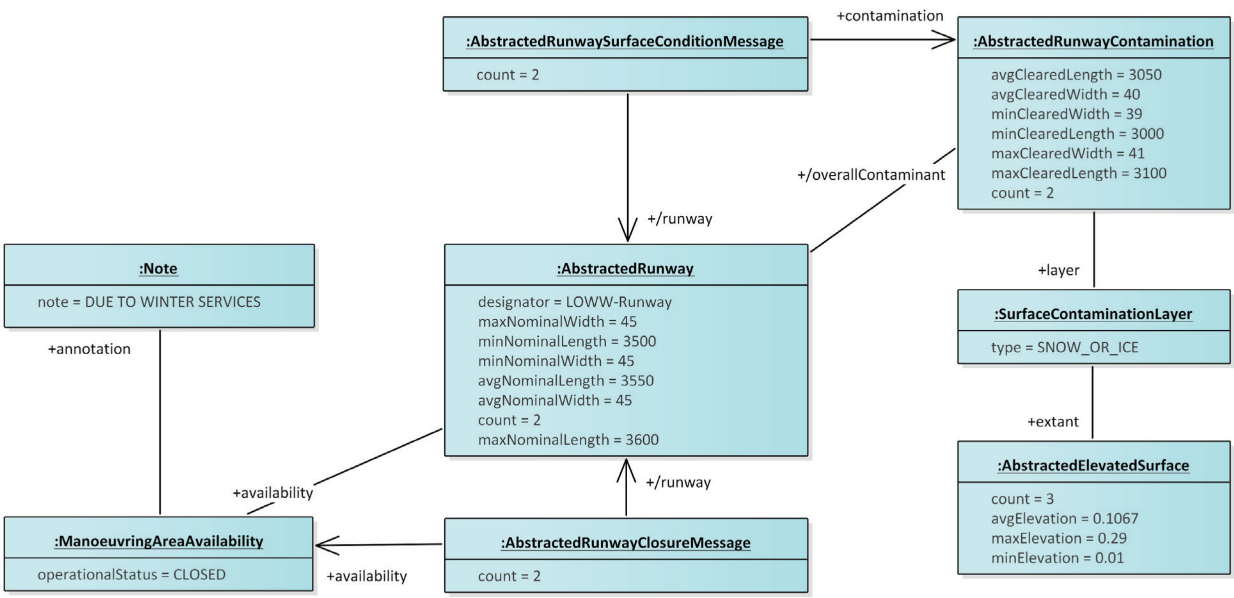

Fig. 9 An example of abstracted DNOTAM information obtained from the DNOTAM information in Fig. 8

closure affects only one runway direction whereas the other remains open.

Assume the DNOTAMs in Figure 8 concern the destination airport of a particular flight, e.g., OS93 from Washington-Dulles to Vienna. At the beginning of a flight, detailed information about the destination are not interesting for the pilot preparing a flight. Rather, the pilot may prefer a *management summary* suitable for display during the preparation and early phases of that flight which might only show abstracted DNOTAMs that alert the pilot that wintry conditions await at the destination airport with runway closures in place. The summary may also include average, minimum, and maximum of the contaminant's extent in order to allow the pilot to get a grasp of the severity of the situation at a single glance.

Consider then the abstracted DNOTAM information in Figure 9, which is the result of applying the abstraction operation with some abstraction function that combines DNOTAMs about surface contamination and runway closures, respectively. The application of such function to the DNOTAMs from Figure 8 produces an *AbstractedRunwayClosureMessage* with abstracted information about the runway closures as well as an *AbstractedRunwaySurfaceConditionMessage* with abstracted information about runway contamination. The abstracted DNOTAMs reference a generic *LOWW-Runway* rather than specific runway directions. The abstracted DNOTAMs summarize the attributes such as the cleared length and width of the contaminated runways, providing average, minimum, and maximum values for these numeric attributes. The objects' *count* attributes preserve information about the size of the input model. For example, while the abstracted DNOTAM about surface condition has only one generic runway contamination, the *count* attribute documents the number of runway contaminations in the input model. The type of contamination is "SNOW_OR_ICE", which subsumes the "SNOW" and "ICE" layers from the input DNOTAMs. Information about such subsumption relationships could be derived from ontologies, e.g., the NASA ATM Ontology [9]; attribute values would then be concepts from ontologies.

The *abstraction function* that conducts the actual abstraction of data items is akin to the *aggregation function* in traditional OLAP. Just like there are different aggregation functions to summarize numeric values, e.g., SUM, MIN, MAX, COUNT, there are different abstraction functions to summarize more complex data items, e.g., DNOTAMs and METARs. There may exist different abstraction functions for different data item types, e.g., DNOTAMs may be summarized differently from METARs. Other abstraction functions may apply to a variety of data item types. The identification and definition of an extensive catalog of abstraction functions merits further investigation and is left for future work.

6 Summary and Future Work

In this paper, we have presented a conceptual framework for the organization of ATM information which allows for the subsequent merging and abstraction of the thus organized ATM information in order to provide a more condensed view – a management summary – of relevant ATM information. The conceptual framework is independent from any particular data format and could be implemented using various technologies, e.g., XML and XQuery, RDF and SPARQL. Future work will identify, define, and implement a set of abstraction functions common for ATM information. Future work will also investigate the applicability of the presented approach in other ATM activities, e.g., Air Traffic Flow and Capacity Management.

References

[1] Pre-flight information bulletin. URL: http://www.ead.eurocontrol.int/ino_resources/webhelp/help/pib_help.html. Accessed: 09-05-2018.

[2] F. Burgstaller, D. Steiner, B. Neumayr, M. Schrefl, and E. Gringinger. Using a model-driven, knowledge-based approach to cope with complexity in filtering of Notices to Airmen. In *Proceedings of the Australasian Computer Science Week Multiconference*, 2016.

[3] C. Chen, F. Zhu, X. Yan, J. Han, P. Yu, and R. Ramakrishnan. InfoNetOLAP: OLAP and mining

of information networks. In *Link Mining: Models, Algorithms, and Applications*, pages 411–438. Springer, New York, 2010.

[4] Federal Aviation Administration. *Federal NOTAM system airport operations scenarios*, 2010. Accessed: 11 January 2018.

[5] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 205–216, 1996.

[6] D. Hiltunen, S. G. Chase, A. Kendra, and Y. J. Jo. Electronic flight bag (EFB) 2015 industry survey. Technical report, John A. Volpe National Transportation Systems Center, 2015.

[7] International Civil Aviation Organization. *Aeronautical Information Services Manual*, 6th edition, 2003.

[8] International Civil Aviation Organization. *Aeronautical Information Services – Annex 15*, 15th edition, 2016.

[9] R. M. Keller. The NASA Air Traffic Management Ontology (atmonto) – release dated March 2018. Technical report, National Aeronautics and Space Administration, 2018. Accessed: 16-05-2018.

[10] I. Kovacic, D. Steiner, C. Schuetz, B. Neumayr, F. Burgstaller, M. Schrefl, and S. Wilson. Ontology-based data description and discovery in a SWIM environment. In *Proceedings of the 17th Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 5A4–1–5A4–13, 2017.

[11] B. Neumayr, E. Gringinger, C. G. Schuetz, M. Schrefl, S. Wilson, and A. Vennesland. Semantic data containers for realizing the full potential of system wide information management. In *Proceedings of the 36th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2017.

[12] R. Pelchen-Medwed and E. Porosnicu. Enhanced pilot situational awareness through the digital/graphical pre-flight briefing concept. *HindSight*, 23:66–69, June 2016.

[13] A. Savulov. D120 - aeronautical information management system description (ts) - digital integrated briefing - final version. Technical report, Frequentis, EUROCONTROL, ENAV, Noracon, Thales, 2016. Accessed: 9 May 2018.

[14] C. Schütz, B. Neumayr, and M. Schrefl. Business model ontologies in OLAP cubes. In C. Salinesi, M. C. Norrie, and O. Pastor, editors, *CAiSE 2013*, volume 7908 of *LNCS*, pages 514–529. Springer, Berlin Heidelberg, 2013.

[15] D. Steiner, I. Kovacic, F. Burgstaller, M. Schrefl, T. Friesacher, and E. Gringinger. Semantic enrichment of DNOTAMs to reduce information overload in pilot briefings. In *Proceedings of the 16th Integrated Communications Navigation and Surveillance (ICNS) Conference*, pages 6B2–1–6B2–13, 2016.

[16] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.

[17] A. Vaisman and E. Zimányi. *Data Warehouse Systems – Design and Implementation*. Springer, Berlin Heidelberg, 2014.

Contact Author Email Address

christoph.schuetz@jku.at

Acknowledgments

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.

Copyright Statement