ENHANCING A DOMAIN-SPECIFIC MODELING LANGUAGE FOR RUNWAY ASSIGNMENT SYSTEMS WITH EXPERIENTIAL KNOWLEDGE BASED MODEL GUIDANCE METHODS

Dagi Troegner
German Aerospace Center e.V., Institute of Flight Guidance, Lilienthalplatz 7,
38108 Braunschweig
Leibniz University Hannover, Department of Simulation,
Institute of Systems Engineering, Welfenplatz 1, 30167 Hannover

Keywords: *Airport Modeling, Experiential knowledge, Domain-Specific Modeling*

Abstract

The efficient use of runway capacities continues to gain significance in modern Air Traffic Management (ATM). Over the past years an increase in air traffic has been observed, which leads in conjunction with limited airport capacities to cost-intensive waiting times and an increased workload for air traffic controllers. Decision support systems should enable a strategic decision-making process even in periods of high traffic and therefore an efficient degree of utilization of the runway capacities. The adaptation of a support system to new airports is associated with a considerable expenditure of work and the precondition of an adequate knowledge of the existing domain constraints. The development of a domain-specific modeling language (DSL) for the purpose of arrival management enables a development at domain level and in combination with code transformation procedures a generic creation of airport-specific systems. Especially the generation of a valid and optimized runway assignment system is subject to a variety of constraints that have to be taken into account. To cope with constraint restrictivity and large solution spaces in a model, in this approach airport-specific data, legal requirements and the experiential knowledge of domain experts (air traffic controllers and developers of arrival management systems) is gathered and captured in a knowledge base. Validation of a model is carried out making an alignment between an actual model state and the referring constraints in the knowledge base. Further constraint reasoning methods can be triggered to support the modeling process by deriving conceivable modeling steps to guide the modeler towards a valid model state. The developed modeling language AMAN-ML (Arrival Management Modeling Language) for the runway assignment component of an arrival management system will be exemplified in this paper including its composition and functionality. Furthermore, the integration of factual and experiential knowledge into model guidance methods and the resulting modeling process will be introduced.

1 Introduction

In consequence of the fact that airports increasingly turn into bottlenecks of the entire air transportation system, the runway assignment as a component of the arrival management becomes an increasingly decisive factor regarding capacity optimization at airports. The task of a runway assignment is to assure an efficient allocation of arriving aircrafts to the available runways with the ambition of reducing delays and using runways up to their capacities, while airport-specific conditions and legal requirements have to be considered.

The domain-specific modeling language AMAN-ML provides a solution for an intuitive development of models for airport systems. This enables the generic creation of executable code for runway assignment systems with respect to the individually applicable constraints.

Arrival management systems evolve from longtime research and gradually developments based on experiential knowledge of the domain experts. Capturing this knowledge together with the airports' basic conditions makes relevant information accessible for model checking and guidance methods, which taps the full potential of designing optimized systems for new airports.

The paper is organized as follows: Section 2 describes the Modeling Language AMAN-ML in its composition and functionality. Furthermore, an overview about the concept of Domain-Specific Modeling and the integration of AMAN-ML into an Arrival Management System is given; Section 3 presents the approach of including experiential knowledge in the process of model creation; and Section 4 presents concluding remarks and outlines future work.

2 The Modeling Language AMAN-ML

2.1 Domain-Specific Modeling

The concept of Domain-Specific Modeling (DSM) raises the level of abstraction beyond programming by specifying the solution in a design language that directly uses concepts and rules from a specific problem domain [1]. The modeling language hereby represents besides the model's elements the semantics of the given domain and makes use of the already established domain's notation. A high problem specificity and level of abstraction can be obtained, which enables domain experts to create models intuitively. The models created underlie the specified rules of the domain and therefore ensure the semantic validity of resulting code generations already in early stages of development. Encapsulating domain-specific data into a modeling language comprises the visualization of domain-specific correlations and constraints, which facilitates the analysis of a domain and the evaluation of parameter modifications.

2.2 Runway Assignment in the Arrival Management System 4D- CARMA

The approaching air traffic in the vicinity of an airport is monitored by air traffic controllers. Their responsibility comprises establishing an arrival sequence, assigning arrival times and runways, projecting horizontal and vertical approach ways as well as transferring these parameters into corresponding advisories, while the aircraft's safety has to be assured at any time.

The Institute of Flight Guidance at the German Aerospace Center in Braunschweig (DLR e.V.) has developed an arrival management system, 4D-CARMA (4-Dimensional Cooperative Arrival Manager), that supports air traffic controllers in accomplishing these tasks. The runway assignment of 4D-CARMA is part of a sequencing component that plans and optimizes aircraft sequences. Given an arrival sequence, the runway assignment determines a runway and a precise arrival time for each aircraft. These assignments are highly dependent on several factors including the configuration of available runways, operation modes, traffic density and separation requirements to preceding aircraft. When using runways for mixed-mode operations (simultaneous usage for arrival and departure operations), runway blockings for departures in terms of arrival-free intervals (AFI) have to be included in the calculations. An efficient allocation of approaching aircraft to available capacities can be obtained by assigning runways according to various predefined strategies.

2.3 DSL for a Runway Assignment System

AMAN-ML is a domain-specific modeling language with the purpose of designing and implementing a generic arrival management system. The modeling language consists of multiple hierarchically structured components that can be individually adapted to airport specific requirements [2]. Each of the components applies its own syntax and semantics and can be modeled graphically as *AirportFamily*, *RunwayConfiguration*, *AssignmentStrategy* and *AirspaceStructure*.

At the uppermost layer a general model of an airport is created (*AirportGroup*). This model includes all necessary basic data of an airport (e.g. IATA-Code, geographic reference coordinates) and particularly the references to a selected runway configuration, assignment strategy, airspace structure and optionally heuristic procedures that can be applied. The last-mentioned references relate to individual components that are hierarchically linked with the airport model per decomposition. With these decompositions a more detailed description of the airport system is enabled while maintaining the clarity in the airport model.

The component *AirspaceStructure* contains a model for creating airspace structures for an airport. The vicinity of an airport can be designed in this model comprising waypoints and flight paths from TMA (Terminal Maneuvering Area) entry points to the runways. All elements may be associated with corresponding constraints regarding aircraft speed and altitude.

Another component of AMAN-ML, the *AssignmentStrategy*, provides modeling elements to create strategies for the runway assignment. Existing heuristic strategies like ‘Staggered’, ‘Displaced’ or ‘TypeSplitting’ as well as variants of the mathematic optimization strategy ‘JoinTheLeastLoad’ can be selected for an airport model. The strategy ‘Staggered’ corresponds to a strictly alternating assignment of aircraft to the set of available runways. With ‘Displaced’ assignments can be made to runways that include displaced thresholds, whereas assignment decisions depend on resulting wake vortex separations and the runways assigned to preceding aircraft. Furthermore, ‘TypeSplitting’ describes a strategy where assignments are solely dependent to the wake vortex category of an aircraft. This enables a modeling of runway preferences for different weight classes. Another option is ‘JoinTheLeastLoad’, an optimization procedure from queuing theory [3]. In this procedure the actual state of each available runway is taken into account and used for calculating correspondent workloads. The state of a runway is derived from the remaining waiting time, calculated as the sum of separations of the aircraft planned on this runway, and the weight class of the aircraft at the end of the runway queue. The runway with the lowest workload will then be assigned to the considered aircraft. The described strategies can be individually adapted to the airport-specific requirements and conditions and therefore contribute to the calculation of optimized arrival sequences [2].

A fundamental component of AMAN-ML is the *RunwayConfiguration*. It contains a model that represents the layout of an airport’s runway system. Figure 1 illustrates an exemplary model of the component *RunwayConfiguration* for the airport Hamburg-Fuhlsbüttel (IATA-Code: HAM) in Germany. The basic elements of a *RunwayConfiguration* are the runway thresholds containing runway designator, bearing, geographic coordinates and operation mode (arrival, departure or multi mode) as attributes. These are essential data for subsequent calculations. The created runway thresholds can be related to each other using different relationship types to enhance the level of information for the modeler and consequently for the subsequent generation of components for the arrival management system (code generation). Two opposing thresholds of one runway can be related by a relationship of type ‘OnePhysicalRunway’ in the category ‘OppositeOperationalDirections’. Using this relationship with the category ‘DualThreshold’ labels a displaced threshold for a runway. Moreover, degrees of dependencies can be visualized corresponding to different runway layouts. The layouts can be differentiated as V-layout, near-parallel, parallel or crossing according to ICAO Doc 9643 [4]. A V-layout names diverging or converging runways with more than 15° bearing difference, whereas bearing differences between 3° and 15° determine near-parallel layouts. When the bearing differs less than 3° two runways can be referred to as parallel. Crossings are specified by two intersecting centerlines of runways. In Figure 1 the intersecting runways in Hamburg-Fuhlsbüttel are displayed.

.

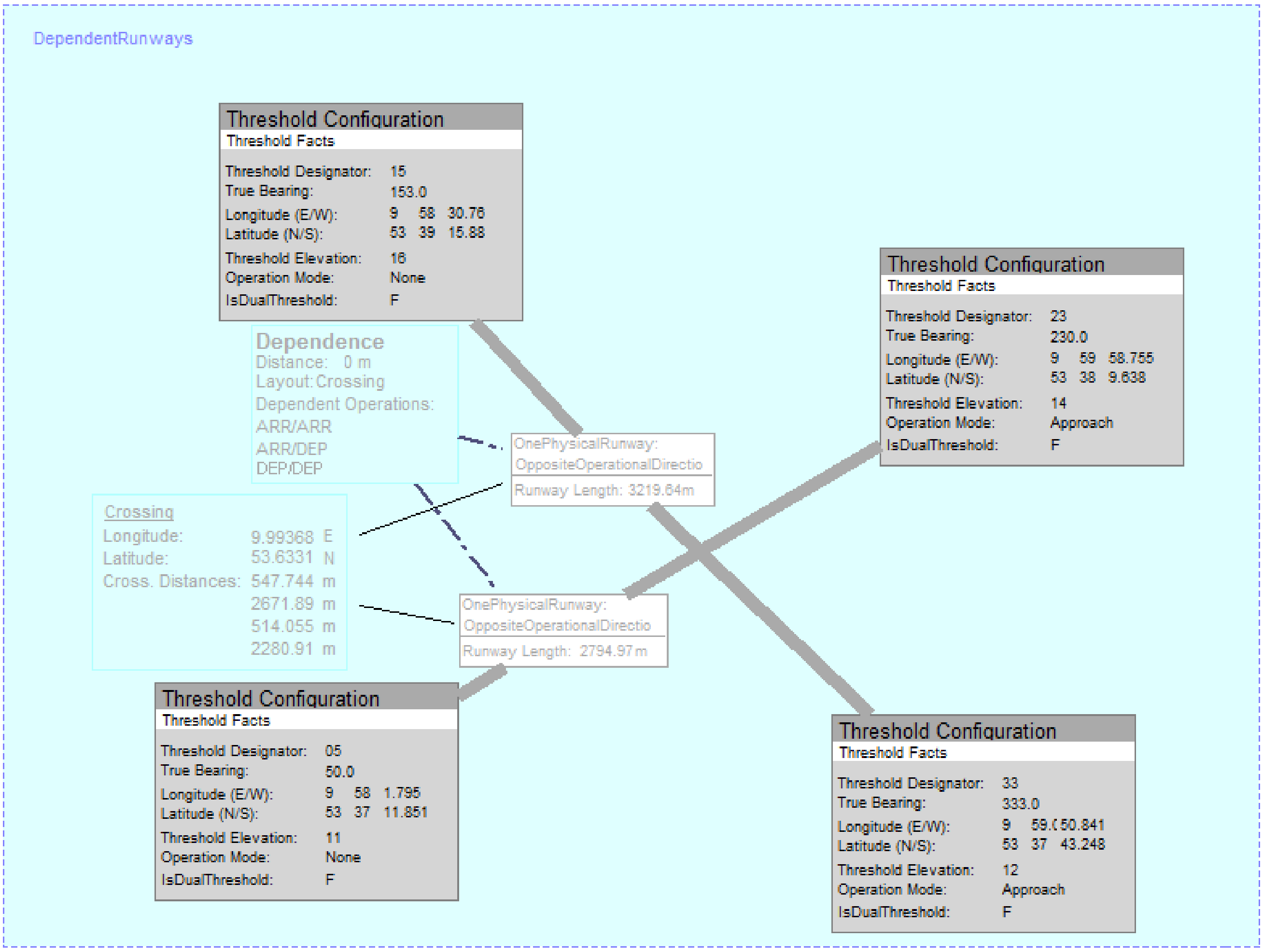

Fig. 1. RunwayConfiguration for Hamburg-Fuhlsbüttel, Germany

The 'Crossing'-relationship provides further information regarding the crossing coordinates and the distances from the thresholds to the intersection point. A crossing implies dependencies for all operation modes for the runways involved.

In addition to the runway layout the model enables a visualization of *Dependence* (and *Independence)* relationships between thresholds (parallelism, crossing or V-layout). These relationships can be created manually or completed automatically by running a model analysis feature providing additional information on distance, dependent operation modes, resulting separation requirements and potential crossing coordinates of runways.

A valid airport model with assigned runway configuration and selected strategy can be translated into executable code via a code generator adapted to the modeling language.

3 Integration of experiential knowledge into model guidance mechanisms

The components specified in AMAN-ML are airport-specific, underlie legal requirements or can be described using heuristics. This variety of domain knowledge originates from different resources and experts and can be enlarged by the results of the model analysis. Gathering this information in a knowledge base makes them accessible for model checking and guidance methods. Standard model checking techniques based on meta-model constraints are enhanced in this approach by the data and rules provided with the knowledge base. Large solution spaces in completing a model and restrictive constraints in complex application domains can be addressed using this approach. The modeling process is simplified and comprehensible.

AMAN-ML is enhanced with model guidance methods leading the modeler through the modeling process towards a valid model state. Reasoning techniques are triggered when the creation of model elements is initiated or modifications on the model are made. The model guidance application developed for AMAN-ML is based on rules formulated in natural language.

3.1 Experiential knowledge

The domain knowledge in arrival management is composed by air traffic controllers, airport operators and developers of arrival management systems and is fundamental for creating reasonable and reliable support systems. Heuristics (domain-specific procedures and facts formulated by domain experts) arise due to continual engagement in a specific area of application. Integrating this knowledge into the process of model creation increases the level of information for the modeler and adds expressiveness to the software components to be generated. The model and the generated software become more intuitive for the users as their knowledge is represented within the adapted model notation and element specification.

Air traffic controllers are the major users of arrival management systems. Their acceptance is therefore the crucial factor in accomplishing a sustained utilization. An intuitive operability and the possibility to adapt a system to individual preferences add significantly to a high acceptance. When an air traffic controller is enabled to integrate his everyday knowledge into his operating system, the system becomes more reflective of an air traffic controller's behavior. Possible heuristics of an air traffic controller's knowledge base could be:

- preferred runways of certain airlines,
- sequencing of different aircraft types or
- direct routings for specific aircraft.

Another group of domain experts for arrival management systems are the airport operators. Their knowledge about airport design and operations planning enhances the correctness of the requirements, implicated by an airport layout, and acts as a basis for arrival planning calculations. Heuristics of an airport operator's knowledge base could therefore be:

- operation mode implications of a given runway layout,
- preferable taxiways for a runway or
- favored approach procedures at the airport.

Developers of arrival management systems may as well increase the fundamental knowledge that is integrated into a modeling language and the corresponding operating system. Arrival management systems evolve from extensive research over long periods of time and modular developments based on experiential knowledge of several domain experts. Especially aircraft sequencing optimizations, display adaptations or runtime issues are part of a domain's knowledge provided by software developers in this area of research. The following examples can be given as heuristics from the knowledge base of developers in arrival management:

- efficiency of different optimization algorithms for arrival sequencing,
- implications of different runway assignment strategies or
- ergonomic layout design of air traffic controller displays.

The domain knowledge in AMAN-ML is gathered in a combined knowledge base, which can be used to support a modeler in creating new models or to ensure a model's correctness. The domain knowledge is extended by an airport's specification data. These specifications originate from the Aeronautical Information Publication (AIP) [5], where the reference coordinates, runway coordinates, allowed operation modes and approach procedures are defined. Further legal requirements that have to be applied in arrival management like wake vortex separations [7], dependencies of runways according to their distances and operation modes [6] or constraints regarding speed and altitude for TMA waypoints [5] are saved in the knowledge base to verify the compliance of an airport model with the prevailing legislation.

3.2 Integration into AMAN-ML

The heuristics of the domain experts can be formulated by rules and data. A formulation is hereby simplified by enabling natural language for expressing a domain's knowledge. This enables a domain expert to add his knowledge to the design of an arrival management system without the precondition of having to learn a corresponding programming or modeling language with own syntaxes and semantics. The modeling language uses the domain's notation; hence the rules and data for AMAN-ML can be expressed with minor expenditure of work. In this section the formulation, transformation and integration of exemplary rules is described.

A domain expert could define for example boundary values for the length of a runway (either for a specific airport or generally for airports in a certain region). A possible heuristic for this purpose is stated in (1.1) and (1.2).

"The length of a runway in Germany lies between 700 and 4000m"	(1.1)

The rules are first investigated for predefined keywords. The main keywords are reserved for the elements and their corresponding attributes defined in the meta-model of the modeling language. In AMAN-ML, the basic keywords resulting from the *RunwayConfiguration* meta-model are 'Threshold', 'OnePhysicalRunway', 'Dependence', 'Independence' and 'Crossing'. These basic keywords are used to define a context for a rule that specifies an interrelation to the referring model elements. In (1.1) the relevant keywords would be 'runway' and 'length' determining a rule that refers to the relationship 'OnePhysicalRunway' with its attribute 'runwayLength'. Therefore, a constraint will be generated that restricts the length of runways (in this case for all airports in Germany) with the given marginal values. In the example given (1.1), the marginal values are formulated as precise boundaries with a lowest value of 700m and a highest value of 4000m. Heuristics using precise formulations are translated by AMAN-ML into constraints applying the Object Constraint Language (OCL). The Object Constraint Language (OCL) is a declarative language for capturing constraints that can be applied to model specifications conform to Meta-Object Facility (MOF) meta-modeling architectures. As a result the constraints can be correlated directly to the model element specifications.

In principle, natural language and heuristics are not always precise. In (1.2) an example is given that restates the rule given in (1.1) including vague statements. The constraint generation in AMAN-ML thus includes a formalism, which parses for relativizing terms in a rule expression. The set of relativizing terms (modal preposition) includes: "likely", "around", "about", "approximately", "circa" and "nearly". In (1.2) the length of a runway is restricted with imprecise boundary values. A transformation into a corresponding constraint is enabled by an integration of vague concepts into an OCL constraint. Vagueness is expressed by introducing fuzzy variables. Fuzzy logic is a form of multi-valued logic based on the paradigm of inference under vagueness. Accordingly, fuzzy logic variables may have a truth value that ranges between 0 and 1 and are not constrained to the two truth values of a classic propositional logic [8]. Likewise, (2.2) represents the constraint referring to the heuristic using imprecise statements in (1.2).

"The length of a runway in Germany is likely to be around 2900m"	(1.2)

In (1.2) the relativization results from the additional term 'likely'. Therefore, the heuristic rule is translated into a constraint that uses a fuzzy variable for the attribute of the runway length. A constraint that is derived by the constraint generation by AMAN-ML is composed as shown in (2.1) and (2.2).

```
[context
Relationship:OppositeOperationalDirection
inv:
if (OnePhysicalRunway.getType() = =
“OppositeOperationalDirection”){
OppositeOperationalDirection.runwayLength
> = 2000 AND
OppositeOperationalDirection.runwayLength
< 4500}]
```

(2.1)

In (2.1), the corresponding constraint to the heuristic rule formulated in (1.1) is presented.

```
[context
Relationship:OppositeOperationalDirection
if (OnePhysicalRunway.getType() = =
“OppositeOperationalDirection”){
OppositeOperationalDirection.runwayLength=
fuzzy(runwayLength) = 2900}]
```

(2.2)

A mapping of rules formulated in natural language is therefore enabled by generating either OCL or fuzzy constraints in dependence of the existence of relativizing terms in the heuristic expression. All generated constraints are added to the knowledge base and are validated against a model state in case of a model checking initiation or guidance requirement.

3.3 Modeling process adaptation

The rules and data formulated in the knowledge base are translated into corresponding domain constraints as described in section 3.2. Based on the constraints provided, a modeler is supported in the process of model creation by checking a created model’s correctness and guiding the modeler towards valid model states by recommending reasonable modeling steps.

A valid model state has to be conform to the rules predetermined by the model’s domain. The actual model state is first compared with the rules given by the meta-model. The meta-model specifies e.g. data types of attributes, cardinalities of relationships or allowed relationships between element types. When the model meets these requirements, additional rules of the knowledge base are considered to verify the model’s correctness. In the model checking process each element of the actual model is verified against the constraints referring to its context. The process can be individually invoked by a modeler and returns a model checking report that contains references to the rules and data in the knowledge base that may not have been met.

In AMAN-ML the modeler needs to specify the basic elements of a new model first before starting the model guidance application. In case of the *RunwayConfiguration* model the thresholds of an airport have to be created and the corresponding attribute values have to be specified to provide the essential data for applying guidance methods. The model guidance is triggered by selecting a relationship type and a first element to be part of that relationship. Reasonable second objects are calculated and highlighted in the model. Due to the context of each constraint, a set of constraints can be derived that can be verified against possible modeling steps. A combined constraint reasoning is invoked based on the information provided by the actual model state. Incorrect relationships are marked giving an error code and an explanation referring to the knowledge base. Incomplete models can be completed gradually following the steps proposed by the guidance system. Incorrect elements or unreasonable element combinations can be located and corrected in early stages of development.

4 Conclusion

The generation of a valid and optimized runway assignment system is subject to a variety of constraints that have to be taken into account. To cope with constraint restrictivity and large solution spaces within a model, in this approach airport-specific data, legal requirements and the experiential knowledge of domain experts (air traffic controllers, airport operators and developers of arrival management systems) is gathered and captured in a knowledge base. The constraints can be formulated in natural language and therefore be included directly into an arrival management system by a domain expert. Possible vagueness in natural language formulations can be recognized and translated into corresponding constraints using fuzzy logic. The validation of a model is carried out making an alignment between an actual model state and the referring constraints in the knowledge base. Further constraint reasoning methods can be triggered to support the modeling process by deriving conceivable modeling steps to guide the modeler towards a valid model state.

Concluding, in this paper, the developed modeling language AMAN-ML for an arrival management system has been exemplified including its composition and functionality. Furthermore, the experiential knowledge in the domain of arrival management is described. The transformation of natural language and the integration of the heuristics into model checking and guidance have been introduced. Finally, an outline of the appliance in the resulting modeling process is given.

Abbreviations

4D-CARMA	4- Dimensional Cooperative Arrival Manager
AMAN-ML	Arrival Management Modeling Language
AFI	Arrival Free Interval
ATM	Air Traffic Management
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DSL	Domain-Specific Modeling Language
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
OCL	Object Constraint Language
MOF	Meta-Object Facility
TMA	Terminal Manoeuvring Area

References

[1] Tolvanen J.-P. Domänenspezifische Modellierung für vollständige Codegenerierung. *JavaSpektrum* 01/2006, SIGS-DATACOM , pp. 9 -12, 2006.

[2] Troegner D. Modellierung und Simulation einer Anflugplanung für die Landebahn Nordwest in Frankfurt. *Deutscher Luft- und Raumfahrtkongress 2010*, Hamburg, Germany, 2010.

[3] Bäuerle N, Engelhardt-Funke O and Kolonko M. On the Waiting Time of Arriving Aircrafts and the Capacity of Airports with One or Two Runways. *European Journal of Operational Research*, Vol. 177, pp. 1180-1196, 2007.

[4] International Civil Aviation Organization (ICAO). Doc9643, Manual on Simultaneous Operations on Parallel or Near-Parallel Instrument Runways (SOIR). 1st Edition, 2004.

[5] DFS Deutsche Flugsicherung. *Aeronautical Information Publication (AIP)* , 2010.

[6] International Civil Aviation Organization (ICAO). Annex 14, Volume I to the Convention on International Civil Aviation: Aerodrome Design and Operations, International Standards and Recommended Practices. Fourth Edition, 2004.

[7] International Civil Aviation Organization (ICAO). *Doc4444, Procedures for Air Navigation Services: Rules of the Air and Air Traffic Services*. Fifteenth Edition, 2007.

[8] Novák V, Perfilieva I and Močkoř J. *Mathematical principles of fuzzy logic*. Dodrecht: Kluwer Academic, 1999.

Contact Author Email Address

Email: dagi.troegner@dlr.de

Copyright Statement