

Chapter 20

A Rule-Based Approach for Air Traffic Control in the Vicinity of the Airport

Theodoros Mitsikas, Petros Stefaneas and Iakovos Ouranos

Abstract The constantly augmenting loads of the aviation industry inevitably define the evolutions in the field of Air Traffic Control. Relevant regulations are changing, in order to accommodate the increase in passenger, flight, and cargo numbers. This paper presents the design of an innovative rule base for the Air Traffic Control regulations during the take-off and landing phases, covering both current and future separation standards of ICAO and FAA. The rule base consists of the rules implementing the Air Traffic Control regulations, and a database containing characteristics of airports and aircraft. The proposed rule base constitutes a flexible tool for the computation of the aircraft separation according to current and future regulations, useful in the fields of conflict detection, conflict avoidance, and scheduling aircraft landings. A further application will be as a decision support tool in real-time environments, guaranteeing the enforcement of all the separation standards.

Keywords RuleML · Rule base · Aircraft separation · Air traffic control

AMS Subject Classifications 68Q60 · 68N30 · 68T30 · 03B70

20.1 Introduction

Air Traffic Control (ATC) is an important factor for the airliner operations, guiding the aircraft on the air and on the ground. ATC is responsible for organizing the air traffic flow in an efficient way, while ensuring the safety.

T. Mitsikas (✉) · P. Stefaneas
Department of Applied Mathematics, National Technical University of Athens,
Zografou Campus, 15780 Athens, Greece
e-mail: mitsikas@central.ntua.gr

P. Stefaneas
e-mail: petros@math.ntua.gr

I. Ouranos
HCAA, Heraklion International Airport “N.Kazantzakis”, 71601 Heraklion, Greece
e-mail: iouranos@central.ntua.gr

© Springer International Publishing AG 2017
S. Lambropoulou et al. (eds.), *Algebraic Modeling of Topological and Computational Structures and Applications*, Springer Proceedings in Mathematics & Statistics 219, https://doi.org/10.1007/978-3-319-68103-0_20

The *separation* of aircraft is the concept of keeping a minimum distance between aircraft to avoid collisions and possible accidents caused by *wake turbulence*. The lift that the aircraft's wing is designed to produce directly affects the intensity and lifespan of the generated vortex. Therefore, the *separation minima* is based upon wake vortex categories of the preceding and the following aircraft which, in turn, are derived from the *maximum takeoff weight* (MTOW) or the *maximum takeoff mass* (MTOM) [8, 10, 13, 14].

Due to the increasing traffic and congested airports, regulation changes are currently being planned, with the aim of increasing airport capacity [6, 12, 14, 17]. The first step is the wider adaptation of RECAT, which recategorizes aircraft and sets new standards for wake turbulence separation minima. Therefore, the wingspan is used as an additional to MTOW/MTOM parameter. As a result, aircraft are placed into six wake vortex categories, common for departure and arrival separation, which enhance both safety and efficiency [8, 11, 14]. The second step is a static separation matrix of distance and time for both arrivals and departures for the common commercial aircraft, called RECAT-2 [7]. The third step, RECAT-3, will provide dynamic pair-wise spacing that will vary with atmospheric conditions and aircraft performance [7]. However, if in the future the concepts of *Free Flight* and *Self-Separation* are employed, the management of air traffic will be revolutionized. They consist of decentralized methods of ATC, using computer communication to reserve parts of the airspace dynamically and automatically in a distributed way [20].

The above mentioned changes mandate the modernization of the current infrastructure, as their application necessitates the operation of relevant computer systems. For example, a specialized tool will be required to apply RECAT-2, because of the size of the separation matrix. Besides, after the realization of Free Flight and Self-Separation concepts, conflict avoidance will be totally automated, relying only on computer systems and computer communication between aircraft.

Our research project aspires to support the evolution of ATC. We propose a rule-based approach to model ATC regulations in the terminal airspace. Our approach allows the validation and verification of its formal properties [15], without compromising the compliance with the regulations. We have used the reference Naf Hornlog implementation of RuleML rule language, OO jDREW, taking advantage of its suitable built-in predicates and functions, and its vast multi-agent and distributed systems compatible API. Furthermore, our approach can serve as a tool for the application of both current and future regulations on the field.

Our main contributions are as follows: (i) we developed rules, derived from the regulations, for the definition of wake vortex categories; (ii) we developed rules concerning the separation minima during the landing phases; (iii) a large database of more than 200 types of aircraft was developed, containing the characteristics required to compute the above. Additionally, we developed a reference airport database, containing characteristics of airports in Greece; (iv) we developed the mathematical background within the rule base, capable of handling the heading of aircraft as well as angles and angle comparison, as a base for future work; (v) we developed rules and a methodology of handling not fully established future regulations, covering different orientations that may follow.

The rest of the paper is organized as follows. Section 20.2 presents related work, while Sect. 20.3 introduces RuleML. Section 20.4, the main part of our paper, presents the design of the rule-base. Finally, Sect. 20.5 concludes the paper and proposes future work.

20.2 Related Work

To the best of our knowledge, no strict rule-based approach for ATC in the airport area exists. However, a convergence point can be found in other lines of research concerning decision support tools, landing scheduling, and Free Flight. Below, the most related previous work is discussed.

Prototype decision support tools for terminal area and controllers, such as FAST [5], have been developed and evaluated in operation with live air traffic. More complete systems have been developed and tested, such as [17]. An overview of a method for formal requirements capture and validation, in the domain of oceanic ATC is presented in [16]. The obtained model focuses on conflict prediction, while being compliant to the regulations governing aircraft separation in oceanic airspace. The design approach, the specification structure, and some examples of the rules and axioms of the formal specification are provided. Those examples, expressed in Many-Sorted First Order Logic or in the Prolog notation, include rules about conflict prediction and aircraft separation. Supplementary, the model was validated by automated processes, formal reasoning and domain experts.

The objective of landing scheduling is to position all arriving aircraft to a runway and to a specific time window, while respecting separation constraints and minimizing delays [2, 5, 6, 18, 19, 21]. Different algorithms and heuristics are used, the simplest one being first-come first-served (FCFS). Aircraft separation is used as a scheduling constraint. However, it has to be underlined that the regulations requirements are not strictly followed, but constant values or simplified forms are used.

Research in the field of Free Flight concept focuses on model principles and different algorithms for conflict detection and avoidance. In [20], a general platform (NAMA), which is oriented towards agent-based Free Flight implementations, was presented. In the same work, a conflict detection and resolution, based on utility functions without any negotiations between agents, was proposed. However, the separation constraints were not explained. In [12], the authors focused on airport airspace. They developed a multi-agent architecture, and a software prototype. The latter implements an ATC system with distributed policy of conflict resolution, predictive analysis and P2P interaction-based autonomous coordination of aircraft's motions. Nevertheless, aircraft classes serve only for the estimation of airspeed range and not for the differences in separation between classes.

20.3 RuleML Basics

RuleML is a markup language, designed to express both top-down and bottom-up rules in XML schema; a shorthand is POSL, that follows a Prolog-like syntax. More specifically, we used OO jDREW, a RuleML implementation that follows Horn Clauses in implication form, supports negation as failure, and is written in the Java programming language.

A Horn Clause in implication form is written as $h \leftarrow p \wedge q \wedge \dots \wedge r$, where h , p , q , r are atoms. An atom is of the form $r(t_1, \dots, t_n)$, where r is a predicate of arity n , and t_i are terms.

In POSL, a clause has the general form:

```
h :- p, q, ..., r.
```

The head of the clause is h and the body is p, q, \dots, r . A clause is called a fact if the body is empty. Neither disjunction nor negation is supported on the body [3]. In this paper, we take advantage of OO jDREW's built-in equality and inequality predicates, as well as of math functions. The English sentence: "A customer is premium if their spending has been min 5000 euro in the previous year", part of the classic RuleML example which can be found in [4], can be written in POSL form, using inequality predicates, as follows:

```
premium(?customer) :-
  spending(?customer, ?amount, "previous year"),
  greaterThanOrEqual(?amount, 5000^^Real).
```

The relation atoms are `premium` and `spending`, `?customer` and `?amount` are variables representing customer's name and amount spent, respectively, while "previous year" and 5000 are constants and `^^Real` is the type. The built-in predicate `greaterThanOrEqual` is satisfied iff the first argument is equal to or greater than the second argument. By asserting e.g., the following fact in the Knowledge Base:

```
spending(Peter, 6000^^Real, "previous year").
```

the query `premium(?x)` binds `Peter` to the variable `?customer`, as shown in Fig. 20.1.

The notation used in this paper is: (i) variables are denoted by x_k , (ii) constants and bindings are lower or upper case alphanumericals, (iii) atoms, predicates and functions can be either in the form $R_k^n(t_1, \dots, t_n)$, or in the simplified form —e.g. $(t_1 < t_2)$ — for equality and inequality predicates.

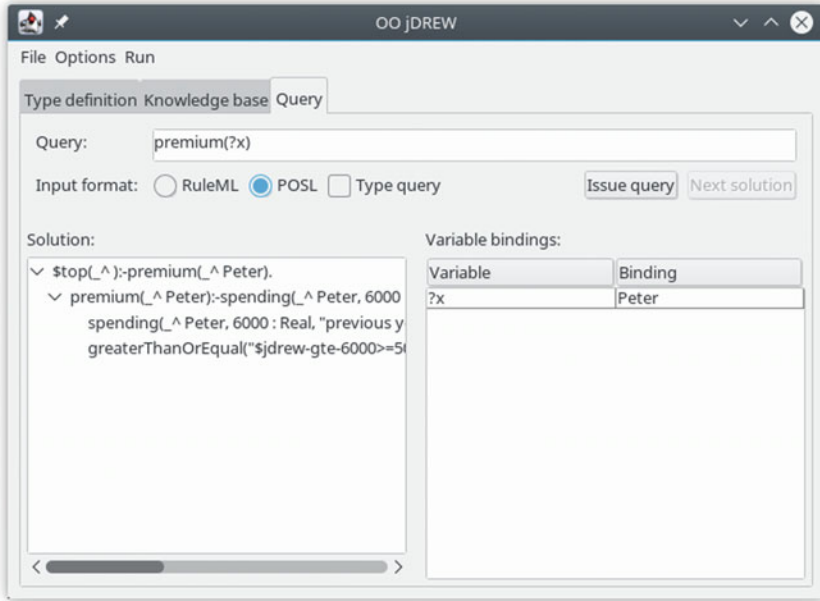


Fig. 20.1 The query execution tab of OO JDREW

20.4 The Rule Base

In this section we present the design of our approach. We express in terms of rules, ATC regulations concerning aircraft separation standards in the airport vicinity under instrument flight rules (IFR). We follow weight classes and the separation of aircraft of different classes according to International Civil Aviation Organization (ICAO) and Federal Aviation Administration (FAA) regulations. Additionally, we describe rules regarding airport layout, heading and turns, and weather conditions in order to cover future regulations or complex cases.

The design of the rule base aims to (i) implement ATC regulations adhering to ICAO and FAA standards. (ii) study complex cases (iii) be easily integrated with any existing or future framework (iv) be flexibly adapted to each framework’s requirements.

Handling intersection departures, intersecting runways, intersecting flight path operations and parallel approaches is presented as example of how our proposed rule base efficiently solves complex cases.

20.4.1 Database

The database, written in RuleML/POSL, contains more than 200 types of aircraft defined by ICAO aircraft designation (a three- or four-character alphanumeric code designating aircraft), with the characteristics needed to compute the separation minima. The data were obtained from [9]. Each entry has a general form of an atom of arity 4 as follows:

$$\mathit{aircraftChar}(x_{type}, x_{kg}, x_{ft}, x_{kt}) \quad (20.1)$$

where x_{type} denotes the type of the aircraft as defined by ICAO aircraft designation, x_{kg} denotes the MTOM in kilograms, x_{ft} the wingspan of the aircraft measured in feet, and x_{kt} is the approach speed, measured in knots. The corresponding code, for the Airbus A321 (ICAO aircraft designation code: A321), in RuleML/POSL, is as follows:

`aircraftChar(A321, 93500^^Real, 111.9^^Real, 138^^Real) .`

For each characteristic an individual atom of arity 1 is obtained by the following Horn Clauses, written in an implication form:

$$\mathit{mtom}(x_{kg}) \leftarrow \mathit{aircraftChar}(x_{type}, x_{kg}, x_{ft}, x_{kt}) \wedge \mathit{aircraft}(x_{type}) \quad (20.2)$$

$$\mathit{wingspan}(x_{ft}) \leftarrow \mathit{aircraftChar}(x_{type}, x_{kg}, x_{ft}, x_{kt}) \wedge \mathit{aircraft}(x_{type}) \quad (20.3)$$

$$\mathit{appSpeed}(x_{kt}) \leftarrow \mathit{aircraftChar}(x_{type}, x_{kg}, x_{ft}, x_{kt}) \wedge \mathit{aircraft}(x_{type}) \quad (20.4)$$

where $\mathit{aircraft}(x_{type})$ must be defined —e.g. `aircraft(A321)` for the Airbus A321— in the knowledge base as a fact.

Similar atoms are defined for the preceding aircraft:

$$\mathit{mtomPreceding}(x_{kg}) \leftarrow \mathit{aircraftChar}(x_{type}, x_{kg}, x_{ft}, x_{kt}) \wedge \mathit{precedingAircraft}(x_{type}) \quad (20.5)$$

$$\mathit{wingspanPreceding}(x_{ft}) \leftarrow \mathit{aircraftChar}(x_{type}, x_{kg}, x_{ft}, x_{kt}) \wedge \mathit{precedingAircraft}(x_{type}) \quad (20.6)$$

where $\mathit{precedingAircraft}(x_{type})$ must be defined as a fact.

Furthermore, the database contains a reference implementation of airports. Each entry in the database is of the form:

$$\mathit{airportChar}(x_{code}, x_{or_1}, x_{or_2}, x_{or_3}, x_{or_4}, x_{name_1}, x_{name_2}, x_{name_3}, x_{name_4}, x_d, x_{rules}) \quad (20.7)$$

where x_{code} is representing the ICAO airport code (a four-character alphanumeric code designating airports), x_{or_1} , x_{or_2} , x_{or_3} and x_{or_4} are representing the the exact heading of each runway in degrees in ascending order respectively, x_{name_1} , x_{name_2} , x_{name_3} , and x_{name_4} are the variables representing the runway designation, x_d is the distance between the two runways and, the regulations applicable to each airport are denoted by x_{rules} .

Those characteristics are needed to compute the airport layout, the heading during landing, and the initial heading after take-off. Currently, the rule base supports up to two runways in both directions.

As above, atoms of arity 1 are obtained for each characteristic, by the following Horn Clauses, written in an implication form:

$$\mathit{runwayOneOr}(x_{or_1}) \leftarrow \mathit{airportChar}(x_{code}, x_{or_1}, \dots) \wedge \mathit{airportName}(x_{code}) \quad (20.8)$$

$$\vdots$$

$$\mathit{runwayOneName}(x_{name_1}) \leftarrow \mathit{airportChar}(x_{code}, \dots, x_{name_1}, \dots) \wedge \mathit{airportName}(x_{code}) \quad (20.9)$$

$$\vdots$$

$$\mathit{distanceBtwnRunw}(x_d) \leftarrow \mathit{airportChar}(x_{code}, \dots, x_d, \dots) \wedge \mathit{airportName}(x_{code}) \quad (20.10)$$

where $\mathit{airportName}(x_{code})$ must be defined as a fact in the knowledge base, e.g. $\mathit{airportName}(\text{LGTS})$ for the Thessaloniki Airport ‘‘Macedonia’’ (ICAO airport code: LGTS), an airport with two intersecting runways (10/28 and 16/34). The corresponding entry in the database, in RuleML/POSL, is

```
airportChar(LGTS, 103.9^^Real, 166.3^^Real,
  283.9^^Real, 346.3^^Real, 10^^String, 16^^String,
  28^^String, 34^^String, 0^^Real, icao).
```

Weather information can be useful for simultaneous operations on different runways, or for future expansion to support RECAT-3, Weather Dependent Separations (WDS), and Time-Based Separation (TBS) in strong headwind conditions [7]. The form of the atoms in the database concerning weather information is as follows:

$$weather(x_{code}, x_{kt}, x_{dir}) \quad (20.11)$$

where x_{code} is representing the ICAO airport code, x_{kt} denotes the wind speed, measured in knots, and x_{dir} denotes the wind direction.

The following is an example according to METAR information of March, 21st, 18:20UTC, for LGTS:

$$weather(LGTS, 3, variable). \quad (20.12)$$

As above, atoms of arity 1 can be obtained by the following Horn Clauses, written in an implication form:

$$windSpeed(x_{kt}) \leftarrow weather(x_{code}, x_{kt}, x_{dir}) \wedge airportName(x_{code}) \quad (20.13)$$

$$windDirection(x_{dir}) \leftarrow weather(x_{code}, x_{kt}, x_{dir}) \wedge airportName(x_{code}) \quad (20.14)$$

20.4.2 ICAO Regulations

Current regulations of ICAO categorize aircraft as follows [13, 14]:

Light MTOM of 7000 kg or less.

Medium MTOM of greater than 7000 kg, but less than 136000 kg.

Heavy MTOM of 136000 kg or greater.

Super - A separate designation that currently only refers to the Airbus A380 (MTOM 575000 kg, ICAO designation A388).

The categorization can be specified using the following Horn Clauses, (for **Light** and **Medium** categories):

$$icaoCategory(light) \leftarrow mtom(x_{kg}) \wedge (x_{kg} \leq 7000) \quad (20.15)$$

$$icaoCategory(medium) \leftarrow mtom(x_{kg}) \wedge (x_{kg} > 7000) \wedge (x_{kg} < 136000) \quad (20.16)$$

Airbus A380 would normally belong to **Heavy** category. Consequently, the definition of **Heavy** and **Super** classes is additionally using the aircraft type.

$$icaoCategory(heavy) \leftarrow mtom(x_{kg}) \wedge (x_{kg} \geq 136000) \wedge aircraft(x_{type}) \wedge (x_{type} \neq A380) \quad (20.17)$$

$$icaoCategory(super) \leftarrow aircraft(A380) \quad (20.18)$$

Table 20.1 Current ICAO weight categories and associated separation minima [14]

ICAO separation standards (nautical miles)					
		Follower			
		Super	Heavy	Medium	Light
Leader	Super	MRS	6	7	8
	Heavy	MRS	4	5	6
	Medium	MRS	MRS	MRS	5
	Light	MRS	MRS	MRS	MRS

MRS: Minimum Radar Separation

Inequality predicates are built-ins of OO jDREW. The computation of aircraft's category can be made by defining the type of an aircraft, which must be present at the database as a fact —e.g. `aircraft(A321)`— and executing the query `icaoCategory(?x)`. Similar rules exist for the preceding aircraft, defining the atom `icaoCategoryPreceding(xclass)`.

ICAO separation standards for flights on instrument flight rules (IFR) are presented at Table 20.1.

Those separations standards can be represented by Horn Clauses. For instance, the derived Horn Clauses for the categories **heavy**, **medium** are as follows:

$$icaoSeparation(mrs) \leftarrow icaoCategory(heavy) \wedge icaoCategoryPreceding(heavy) \quad (20.19)$$

$$icaoSeparation(5) \leftarrow icaoCategory(heavy) \wedge icaoCategoryPreceding(heavy) \quad (20.20)$$

$$icaoSeparation(mrs) \leftarrow icaoCategory(heavy) \wedge icaoCategoryPreceding(heavy) \quad (20.21)$$

$$icaoSeparation(4) \leftarrow icaoCategory(heavy) \wedge icaoCategoryPreceding(heavy) \quad (20.22)$$

By defining the type of the preceding and following aircraft, the needed separation can be obtained by executing the query `icaoSeparation(?x)`.

20.4.3 FAA Regulations

The methodology for constructing the rules concerning aircraft classes and separation according to FAA regulations is similar. The FAA is using the following classes [10]:

Small - Aircraft of 41000 pounds (19000 kg) or less MTOW.

Large - Aircraft of more than 41000 pounds (19000 kg) MTOW, up to, but not including, 300000 pounds (140000 kg).

Heavy - Aircraft capable of takeoff weights of 300000 pounds (140000 kg) or more.

Super - A separate designation that currently only refers to the Airbus A380 and the Antonov An-225

B757 - Different separation standards are applied for the Boeing 757.

The Horn Clauses defining the above regulations, after conversion of pounds to kilograms has been applied, are:

$$faaCategory(small) \leftarrow mtom(x_{kg}) \wedge (x_{kg} \leq 19000) \quad (20.23)$$

Boeing 757 would normally belong to **Large** class, while Airbus A380 and Antonov A225 would belong to **Heavy** class. Therefore, similar to ICAO categorization, the definition of those classes additionally needs the aircraft type:

$$faaCategory(large) \leftarrow aircraft(x_{type}) \wedge (x_{type} \neq B757) \wedge mtom(x_{kg}) \wedge (x_{kg} > 7000) \wedge (x_{kg} < 136000) \quad (20.24)$$

$$faaCategory(heavy) \leftarrow mtom(x_{kg}) \wedge (x_{kg} \geq 136000) \wedge aircraft(x_{type}) \wedge (x_{type} \neq A380) \wedge (x_{type} \neq A225) \quad (20.25)$$

$$faaCategory(super) \leftarrow aircraft(A380) \vee aircraft(A225) \quad (20.26)$$

$$faaCategory(B757) \leftarrow aircraft(B757) \quad (20.27)$$

The lack of disjunction in the head of Horn Clauses and in RuleML/POSL, mandates two rules for **B757** and **Super** classes, one for each ICAO aircraft code designation (B752, B753 for Boeing 757, A225 for Antonov An-225). Executing the query $faaCategory(?x)$, answers the class of the aircraft. Similar rules exist for the preceding aircraft, defining the atom $faaCategoryPreceding(x_{class})$.

The separation standards at the runway threshold for flights under IFR are defined by the Table 20.2.

The derived Horn Clauses for e.g. the categories **heavy**, **B757**, are as follows:

$$faaSeparation(5) \leftarrow faaCategory(B757) \wedge faaCategoryPreceding(heavy) \quad (20.28)$$

$$faaSeparation(4) \leftarrow faaCategory(heavy) \wedge faaCategoryPreceding(B757) \quad (20.29)$$

$$faaSeparation(4) \leftarrow faaCategory(heavy) \wedge faaCategoryPreceding(heavy) \quad (20.30)$$

Table 20.2 FAA wake separation standards (nautical miles, at the threshold) [8]

Leader/Follower	Super	Heavy	B757	Large	Small
Super	MRS	6	7	7	8
Heavy	MRS	4	5	5	6
B757	MRS	4	4	4	5
Large	MRS	MRS	MRS	MRS	4
Small	MRS	MRS	MRS	MRS	MRS

MRS: minimum radar separation

$$faaSeparation(4) \leftarrow faaCategory(B757) \wedge faaCategoryPreceding(B757) \quad (20.31)$$

20.4.4 RECAT Regulations

For the purposes of wake turbulence separation minima, aircraft are categorized as Category A through Category F. Each aircraft is assigned a category based on wingspan, and maximum takeoff weight (MTOW) [8, 11]:

Category A. Aircraft capable of MTOW of 300,000 pounds or more and wingspan greater than 245 feet.

Category B. Aircraft capable of MTOW of 300,000 pounds or more and wingspan greater than 175 feet and less than or equal to 245 feet.

Category C. Aircraft capable of MTOW of 300,000 pounds or more and wingspan greater than 125 feet and less than or equal to 175 feet.

Category D. Aircraft capable of MTOW of less than 300,000 pounds and wingspan greater than 125 feet and less than or equal to 175 feet; or aircraft with wingspan greater than 90 feet and less than or equal to 125 feet.

Category E. Aircraft capable of MTOW greater than 41,000 pounds with wingspan greater than 65 feet and less than or equal to 90 feet.

Category F. Aircraft capable of MTOW of less than 41,000 pounds and wingspan less than or equal to 125 feet, or aircraft capable of MTOW less than 15,500 pounds regardless of wingspan, or a powered sailplane.

The derived definite clauses in implication form, after conversion of pounds to kilograms are as follows:

$$recat(A) \leftarrow mtom(x_{kg}) \wedge (x_{kg} \geq 136000) \wedge wingspan(x_{ft}) \wedge (x_{ft} > 245) \quad (20.32)$$

$$recat(B) \leftarrow mtom(x_{kg}) \wedge (x_{kg} \geq 136000) \wedge wingspan(x_{ft}) \wedge (x_{ft} \leq 245) \wedge (x_{ft} > 175) \quad (20.33)$$

$$\begin{aligned} \text{recat}(C) \leftarrow & \text{mtom}(x_{kg}) \wedge (x_{kg} \geq 136000) \wedge & (20.34) \\ & \text{wingspan}(x_{ft}) \wedge (x_{ft} \leq 175) \wedge (x_{ft} > 125) \end{aligned}$$

$$\begin{aligned} \text{recat}(D) \leftarrow & (\text{wingspan}(x_{ft}) \wedge (x_{ft} \leq 125) \wedge (x_{ft} > 90)) \vee & (20.35) \\ & (\text{mtom}(x_{kg}) \wedge (x_{kg} < 136000) \wedge \\ & \text{wingspan}(x_{ft}) \wedge (x_{ft} \leq 175) \wedge (x_{ft} > 125)) \end{aligned}$$

$$\begin{aligned} \text{recat}(E) \leftarrow & \text{mtom}(x_{kg}) \wedge (x_{kg} > 18000) \wedge & (20.36) \\ & \text{wingspan}(x_{ft}) \wedge (x_{ft} \leq 90) \wedge (x_{ft} > 65) \end{aligned}$$

$$\begin{aligned} \text{recat}(F) \leftarrow & (\text{mtom}(x_{kg}) \wedge (x_{kg} < 7000)) \vee & (20.37) \\ & (\text{mtom}(x_{kg}) \wedge (x_{kg} \leq 18000) \wedge \text{wingspan}(x_{ft}) \wedge (x_{ft} \leq 125)) \end{aligned}$$

The lack of disjunction in the body of Horn Clauses and in RuleML/POSL, mandates two rules for categories **D** and **F**. Executing the query `recatCategory(?x)` answers about aircraft's category. Similar rules exist for the preceding aircraft, defining the atom `recatPreceding(xclass)`.

RECAT separation standards for IFR flights are presented in Table 20.3. For instance, the separation values for the pair **B**, **C** are captured from the following Horn Clauses:

$$\text{recatSeparation}(mrs) \leftarrow \text{recat}(B) \wedge \text{recatPreceding}(C) \quad (20.38)$$

$$\text{recatSeparation}(4) \leftarrow \text{recat}(C) \wedge \text{recatPreceding}(B) \quad (20.39)$$

Table 20.3 RECAT wake separation standards (nautical miles) [8, 11]

		Follower					
		A	B	C	D	E	F
Leader	A	MRS	5	6	7	7	8
	B	MRS	3	4	5	5	7
	C	MRS	MRS	MRS	3.5	3.5	6
	D	MRS	MRS	MRS	MRS	MRS	4
	E	MRS	MRS	MRS	MRS	MRS	MRS
	F	MRS	MRS	MRS	MRS	MRS	MRS

MRS: Minimum Radar Separation

20.4.5 Airport Layout

ICAO and FAA rules are covering cases such as intersecting runway/intersecting flight path separation, intersection departures, parallel approaches, etc. Furthermore, different separation standards may exist when aircraft operate in different runways separated by at least 2500 ft or 760 m [10, 13]. Finally, the rule base must cover situations where a runway is closed for maintenance or for emergency reasons. In order to implement the above, it is necessary to define rules concerning the airport layout.

Reasoning on airport layout is realized by using information about airport characteristics described in Sect. 20.4.1, combined with additional rules. Currently, the rule base supports up to two runways, used both ways. In accordance with ICAO and FAA regulations, the basic layouts supported are: (i) single runway, (ii) two intersecting runways, (iii) two close parallel runways, less than 2500 feet (760 m for ICAO regulations) apart, (iv) parallel runways more than or equal to 2500 feet, and (v) non parallel and non intersecting runways, denoted as *npni*.

$$\text{layout}(\text{closeparallel}) \leftarrow (R_1(x_1) \wedge R_2(x_2) \wedge (x_1 = x_2)) \wedge R_d(x_3 < 2500) \quad (20.40)$$

$$\text{layout}(\text{farparallel}) \leftarrow (R_1(x_1) \wedge R_2(x_2) \wedge (x_1 = x_2)) \wedge R_d(x_3 \geq 2500) \quad (20.41)$$

$$\text{layout}(\text{npni}) \leftarrow (R_1(x_1) \wedge R_2(x_2) \wedge (x_1 \neq x_2)) \wedge R_d(x_3 \neq 0) \quad (20.42)$$

where R_{layout} is defined by *layout*, R_1 is the heading of the first runway, defined as *runwayOneOr*, R_2 is the heading of the second runway, defined as *runwayTwoOr* and R_d is the distance between runways, defined by the atom *distanceBtwnRunw*. Equality predicate is built-in of OO jDREW. The query *layout(?x)* is shown in Fig. 20.2.

20.4.6 Angles and Heading

Solving complex cases, as mentioned in Sect. 20.4.5, require reasoning over heading and turns, as well as weather information. Representing spatial information for the purpose of an ATC rule base mandates the use of quantitative terms for angle and heading, since qualitative spatial terms cannot be used for precise arithmetic calculations. Previous work (e.g. [1]), can be useful only in cases of fuzzy terms, as wind heading variations as expressed by METAR information.

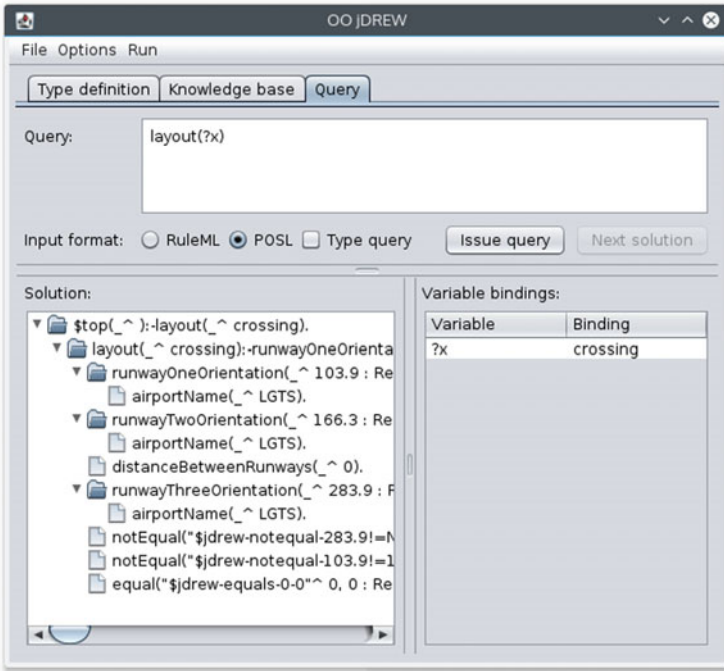


Fig. 20.2 Executing the query about airport layout, for the Thessaloniki Airport “Macedonia”

During landing phase, aircraft’s heading is derived from the orientation of the runway on which the aircraft operates (denoted by the atom $onRunway(x_2)$). Four rules are needed to match runway name with the corresponding heading.

$$\begin{aligned}
 heading(x_{deg}) \leftarrow & runwayOneOr(x_{deg}) \wedge & (20.43) \\
 & runwayOneName(x_1) \wedge onRunway(x_2) \wedge (x_1 = x_2)
 \end{aligned}$$

Heading and angle changes require two Horn Clauses for each computation, due to the possible, greater than 360° result. Angle subtraction also requires additional rules, covering cases from 0° to 180° and from 180° to 360° . Negative values have been taken in to consideration, using the built-in function $abs(x_{abs}, x_1)$, where, given that the first argument is a variable then it will be bound to the absolute value of the second argument.

20.4.7 Future Regulations

The final form of the pair-wise separation matrix of RECAT-2 is not yet known; therefore it was not possible to include those regulations in the rule base. However,

a 120×120 matrix was implemented for testing purposes. The computational time was sufficient for use in real-time environments, or at large simulations (~ 5 ms per query on a laptop with AMD A6-6310 APU, 8GB RAM).

Other planned regulation changes as WDS or TBS can be handled by the rule base and the database, according to known information. One advantage of the current rule base is the inclusion of approach speed for each type of aircraft which makes the precise computation of the time-based separation needed possible, in contrast with [12, 17, 21].

20.5 Conclusions and Future Work

A rule base for ATC regulations has been developed for the vicinity of the airport. This rule base, equipped with a large database consisting of characteristics of aircraft and airports, can compute the separation minima during landing phases, as mandated from current and future ATC regulations concerning operations under IFR.

The rule base is derived from interpretation of ATC regulations as Horn Clauses, which allows validation and verification of formal properties. The reasoner used, OO jDREW, provides adequate built-in predicates and functions for the implementation of the subset of ATC regulations studied.

This approach is suitable for building a database of aircraft and airports, containing their characteristics. Furthermore, it is possible to develop rules for categorization of aircraft using only characteristics which are present in the database, according to current ICAO, FAA and RECAT regulations, and subsequently, the implementation of separation tables defined by the above.

Other aspects of regulations and existing models were investigated, leading to the preliminary implementation of a mathematical background capable of handling angles and angle comparison, headings of aircraft, turns, airspeed, airport layout and future regulations.

In the future, we plan to extend the rule base to include a more complete set of regulations, such as cases of separation minima reduction, incident management, and transition zones. Further extensions may include lateral and vertical separation, and 4-D rules, which, given the position, heading, airspeed and time, will generate advisories to avoid possible conflicts. Finally, a collision avoidance method, external or embedded in the rule base, will be necessary to utilize the rule base in Free Flight concepts.

References

1. Batsakis, S.: Reasoning over 2d and 3d directional relations in OWL: a rule-based approach. In: Theory, Practice, and Applications of Rules on the Web, pp. 37–51. Springer Science + Business Media, Berlin (2013)

2. Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Scheduling aircraft landings - the static case. *Transp. sci.* **34**(2), 180–197 (2000)
3. Boley, H.: Integrating positional and slotted knowledge on the semantic web. *J. Emerg. Technol. Web. Intell.* **2**(4), 343–353 (2010)
4. Boley, H., Grosz, B., Tabet, S.: RuleML tutorial. In: *The RuleML Initiative* (2005). <http://www.ruleml.org/papers/tutorial-ruleml.html>. (Cited July 2016)
5. Davis, T., Krzeczowski, K., Bergh, C.: The final approach spacing tool. In: *Automatic Control in Aerospace 1994*, pp. 73–79. Elsevier BV, Amsterdam (1995)
6. Erzberger, H.: Design principles and algorithms for automated air traffic management. *Knowl. Based. Funct. Aerosp. Syst.* **7**, 2 (1995)
7. EUROCONTROL. *Future Airport Operations* (2015). <https://www.eurocontrol.int/articles/future-airport-operations>. (Cited July 2016)
8. FAA. *Advisory Circular 90-23G - Aircraft Wake Turbulence* (2014)
9. FAA. *Aircraft Characteristics Database* (2009). http://www.faa.gov/airports/engineering/aircraft_char_database/. (Cited July 2016)
10. FAA. *Order JO 7110.65V, Air Traffic Control* (2014)
11. FAA. *Order JO 7110.659C, Wake Turbulence Recategorization* (2016)
12. Gorodetsky, V., Karsaev, O., Samoylov, V., Skormin, V.: *Multi-Agent Technology for Air Traffic Control and Incident Management in Airport Airspace* (2008)
13. ICAO. *Doc 4444-RAC/501, Procedures for Air Navigation Services - Rules of the Air and Air Traffic Services* (2007)
14. Lang, S., Tittsworth, J., Bryant, W., Wilson, P., Lepadatu, C., Delisi, D., Lai, D., Greene, G.: Progress on an ICAO wake turbulence re-categorization effort. In: *AIAA Atmospheric and Space Environments Conference* (2010)
15. Ligeza, A.: *Principles of Verification of Rule-Based Systems*. Springer, Berlin (2006)
16. McCluskey, T., Porteous, J., Naik, Y., Taylor, C., Jones, S.: A requirements capture method and its use in an air traffic control application. *Softw. Pract. Exp.* **25**(1), 47–71 (1995)
17. Nikoleris, A., Erzberger, H., Paielli, R.A., Chu, Y.-C.: Autonomous system for air traffic control in terminal airspace. In: *14th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics (AIAA) (2014)
18. Soomer, M., Franx, G.J.: Scheduling aircraft landings using airlines preferences. *Eur. J. Operat. Res.* **190**(1), 277–291 (2008)
19. Tobias, L., Scoggins, J.: *Time-based Air-Traffic Management Using Expert Systems*. Technical report, NASA Ames Research Center (1986)
20. Valkanas, G., Natsiavas, P., Bassiliades, N.: A collision detection and resolution multi agent approach using utility functions. In: *2009 Fourth Balkan Conference in Informatics*. Institute of Electrical and Electronics Engineers (IEEE) (2009)
21. Veidal, E.: *Scheduling Aircraft Landings - The Dynamic Case*. Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark (2007)