Formalizing Air Traffic Control Regulations in PSOA RuleML

Theodoros Mitsikas[1], Sofia Almpani[1], Petros Stefaneas[2], Panayiotis Frangos[1], and Iakovos Ouranos[3]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, Heroon Polytechniou 9, 15780 Zografou, Greece,
mitsikas[AT]central[DOT]ntua[DOT]gr, salmpani[AT]mail[DOT]ntua[DOT]gr, pfrangos[AT]central[DOT]ntua[DOT]gr ,
[2] School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Heroon Polytechniou 9, 15780 Zografou, Greece,
petros[AT]math[DOT]ntua[DOT]gr
[3] Hellenic Civil Aviation Authority, Heraklion International Airport "N.Kazantzakis", 71601 Heraklion, Greece,
iouranos[AT]central[DOT]ntua[DOT]gr

Abstract. The formalization of Air Traffic Control Regulations for the separation of aircraft during approach and departing phases is discussed. Our aim is to introduce rules in Positional-Slotted, Object-Applicative (PSOA) RuleML syntax that capture those regulations. This rulebase is combined with aircraft facts, resulting in a complete Knowledge Base for the computation of the required separation of aircraft. We provide examples of queries posed in the open-source PSOATransRun system and we show the capabilities and limitations of the Knowledge Base and PSOATransRun.

Keywords: Air Traffic Control, Regulation Formalization, Knowledge Base, RuleML, PSOA

1 Introduction

The primary purpose of Air Traffic Control (ATC) is to prevent collisions between aircraft, organize and expedite the flow of air traffic, and provide information and other support for pilots [8].

Collision prevention is realized by ensuring a minimum distance between aircraft, a concept also called *separation minimum*. Separation of aircraft serves an additional important role, which is the avoidance of *wake turbulence*. As aircraft pass through the air, the pressure difference between the lower and upper surface of the wing creates powerful counter-rotating vortices. They can cause a wing of a following aircraft to lose lift and potentially cause an accident. As the strength and lifespan of the vortices depends on the pressure difference (directly related to the lift that the wings produce), the current regulations assume that the maximum weight/mass that these wings can lift represents the intensity and lifespan of the generated vortices. Similarly, this maximum weight/mass is considered representative of how much an aircraft will be affected by the turbulent air of a leading aircraft. Therefore, for the purpose of categorizing the aircraft into classes according to the required separation minima, the characteristic that is taken into account is the *Maximum Take-Off Weight* (MTOW, for the regulations of FAA[4]), or the *Maximum Take-Off Mass* (MTOM, for the countries that follow the regulations of ICAO[5]) [7,8,11,12].

Despite the success (in terms of safety) of the regulations used for many years, due to the increasing traffic and congested airports, regulation changes are currently being planned, aiming at increasing the airport capacity [4,10,12,15]. The first step is the wider adaptation of *RECAT*, which re-categorizes aircraft and sets new standards for wake turbulence separation minima. RECAT uses the wingspan 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 [7,9,12]. 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 [5]. The third step, RECAT-3, will provide dynamic pair-wise spacing that will vary with atmospheric conditions and aircraft performance [5].

The purpose of our work is to capture the above regulations[6] and formalize them in PSOA RuleML. Examples of formalizing ATC regulations are [13] and [14]. The former presented an overview of a method for formal requirements capture and validation, in the domain of oceanic ATC. It was developed within the context of the FAROAS project, a research project funded by the U.K. Civil Aviation Authority. The obtained model focused 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 were provided. Those examples, expressed in Many-Sorted First Order Logic or in the Prolog notation, included rules about conflict prediction and aircraft separation. Supplementary, the model was validated by automated processes, formal reasoning and domain experts. [14] focuses on capturing ATC regulations valid in the airport area. The authors formalized the separation minima mandated by ICAO, FAA, and RECAT regulations in POSL RuleML form. A background for further expansion was implemented, aiming at cases of reduced separation minima that is allowed on certain conditions. However, the authors did not utilize all RuleML features (e.g. slots).

Our work aims to modernize the latter, by exploring the capabilities of Positional-Slotted, Object–Applicative (PSOA) RuleML [2,3,16] in capturing all the above regulations that are given in controlled natural language. The resulting

[4] FAA: Federal Aviation Administration. The United States of America national authority that regulates all aspects of civil aviation.

[5] ICAO: International Civil Aviation Organization. A UN specialized agency, for civil aviation.

[6] RECAT-2 and RECAT-3 exist only as a concept (or are under development).

Knowledge Base (KB) consists of rules capturing the above regulations, combined with aircraft facts (database) also in PSOA RuleML form. We explored the resulting KB with systematic queries in PSOATransRun. The KB served also as a case study for the evaluation of the newly developed SWI Prolog back-end of PSOATransRun.

The rest of the paper is organized as follows: Section 2 presents the regulations we modelled, along with their formalization in PSOA RuleML presentation syntax. Section 3 describes the form of the self-contained database entries (aircraft facts), while Section 4 demonstrates queries and evaluates the results. Finally, Section 5 concludes the paper.

2 Capturing the Regulations

PSOA RuleML presentation syntax generalizes RIF-BLD and POSL RuleML by a homogeneous integration of relationships and frames into ***p**ositional-**s**lotted **o**bject-**a**pplicative* (***psoa***) *terms*, for the often used single-tuple case having these forms ($n \geq 0$ and $k \geq 0$):

$$\textbf{Oidless}: \quad \mathtt{f(t_1 \ldots t_n\ p_1\text{->}v_1 \ldots p_k\text{->}v_k)} \tag{1}$$

$$\textbf{Oidful}: \mathtt{o\#f(t_1 \ldots t_n\ p_1\text{->}v_1 \ldots p_k\text{->}v_k)} \tag{2}$$

Both (1) and (2) apply a function or predicate `f` (acting as a relator) – in (2) identified by an OID `o` via a membership, `o # f`, of `o` in `f` (acting as a class) – to a tuple of arguments $\mathtt{t_1} \ldots \mathtt{t_n}$ and to a bag of slots $\mathtt{p_j\text{->}v_j}$, $\mathtt{j} = 1, \ldots, \mathtt{k}$, each pairing a slot name (attribute) $\mathtt{p_j}$ with a slot filler (value) $\mathtt{v_j}$.

Variables in PSOA are '`?`'-prefixed names, e.g., `?x`. The most common atomic formulas are psoa atoms in the form of (1) or (2). Compound formulas can be constructed using the Horn-like subset of First-Order Logic.

A PSOA KB consists of clauses, mostly as ground facts and non-ground rules: While facts are psoa atoms, rules are defined – within `Forall` wrappers – using a Prolog-like *conclusion* `:-` *condition* syntax, where *conclusion* can be a psoa atom and *condition* can be a psoa atom or a prefixed conjunction of psoa atoms [17].

The implementation of the rulebase in PSOA RuleML capturing the ICAO, FAA, RECAT regulations for separation minima during approaches and departures is shown below[7].

2.1 ICAO Regulations

Current regulations of ICAO categorize aircraft as follows[11,12]:

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.

[7] The complete KB coupled with the database source and the Python script for converting the database in to a PSOA RuleML code can be found at http://users.ntua.gr/mitsikas/ATC_KB/.

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

In PSOA RuleML, e.g. the **Medium** category is formalized as follows:

```
Forall ?a (
  :AircraftIcaoCategory(?a :Medium) :-
    And(?a#:Aircraft(:mtom->?w)
        math:lessThan(?w 136000)
        math:greaterThan(?w 7000))
)
```

The conditions predicate `:Aircraft` is a frame atom, where the hash infix `#` denotes *class membership* by typing an OID with its predicate, while the arrow infix "`->`", pairs each predicate-independent slot name with its filler. The predicate `:AircraftIcaoCategory` is a relationship that links the aircraft with the corresponding ICAO category it pertains. The OID `a` represents the ICAO aircraft type designator (in lowercase). The predicates having the `math:` prefix are defined in the imported mathematics library `http://psoa.ruleml.org/lib/math.psoa`. They are shortcuts for external built-in calls in PSOA.

An interesting case is the exception in the **Heavy** category, in which Airbus A380 would normally belong. In PSOA RuleML negation is restricted to numeric disequality (`math:notEq`). Therefore, in order to implement exceptions for all special cases, we used an additional slot indicating whether an aircraft is subject of an exception[8]. Special cases are the Airbus A380 for the ICAO and FAA regulations, while FAA regulations have additionally the special cases for the Antonov An-225 (ICAO designation A225) and the Boeing 757 (see section 2.2). This mandates the extra slot `:specialCase` in all aircraft facts in the database, having the value `:No` in general, and a self-referential value e.g. `:A380` for the special cases. The formalization in PSOA RuleML syntax is as follows:

```
Forall ?a (
  :AircraftIcaoCategory(?a :Heavy) :-
    Or(And(?a#:Aircraft(:mtom->?w :specialCase->:No)
           math:greaterEq(?w 136000))
        ?a#:Aircraft(:mtom->?w :specialCase->:A225)
        )
    )
```

And, for the **Super** category[9]:

[8] While potential issues may arise with the addition of more exceptions, the regulations are generally stable and new types of aircraft e.g. in the **Super** category are not currently in active development

[9] In order to avoid the self-referential values in rules where disequality or negation is not used, this rule can also be formalized as

```
Forall ?a (
  :AircraftIcaoCategory(?a :Super) :-
    ?a#:Aircraft = :a388#:Aircraft)
```

```
Forall ?a (
   :AircraftIcaoCategory(?a :Super) :-
       ?a#:Aircraft(:specialCase->:A380))
```

Boeing 757 is categorized as **Medium**, therefore is not necessary to include it in the above rules.

ICAO separation minima for flights on instrument flight rules (IFR)[10] during arrivals and departures[11], are presented at Table 1. Due to the fact that in aviation industry it is very common to use nautical miles (1 NM = 1852 m) and feet as units of length, no conversion to SI is applied.

Table 1. Current ICAO weight categories and associated separation minima [12]

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.

MRS is the Minimum Radar Separation, which is 3 NM or 2.5 NM, depending on operational conditions unrelated to wake turbulence (good visibility, clean surfaces, and high speed turnoffs) [12].

As aircraft categorization according to ICAO regulations can be computed, a chaining derivation can be used for the formalization of Table 1. Disjunction support of PSOA RuleML allows for compact rules covering many pairs of aircraft, resulting in a total of six rules. For example all combinations of leaders and followers that must be separated by MRS are formalized as follows:

```
Forall ?x ?y (
  :icaoSeparation(:leader->?x :follower->?y :miles->:Mrs):-
```

[10] Separation minima for arrivals and departures for flights on visual flight rules (VFR) are time-based. Additionally, this time-based separation can be applied between arriving IFR flights executing visual approach when the aircraft has reported the preceding aircraft in sight and has been instructed to follow and maintain own separation from that aircraft [8,11]. For their formalization, similar rules can be used.

[11] The minima set out at Table 1 shall be applied when: a) an aircraft is operating directly behind another aircraft at the same altitude or less than 300 m (1 000 ft) below; or b) both aircraft are using the same runway, or parallel runways separated by less than 760 m (2 500 ft); or c) an aircraft is crossing behind another aircraft, at the same altitude or less than 300 m (1 000 ft) below [11].

```
 Or(
    And(:AircraftIcaoCategory(?x :Medium)
        :AircraftIcaoCategory(?y :Medium))

    And(:AircraftIcaoCategory(?x :Medium)
        :AircraftIcaoCategory(?y :Heavy))

    :AircraftIcaoCategory(?x :Light)

    :AircraftIcaoCategory(?y :Super)
    )
)
```

2.2 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 [8]:

Small - Aircraft of 41000 pounds (≈19000 kg) or less MTOW.
Large - Aircraft of more than 41000 pounds MTOW, up to, but not including, 300000 pounds (≈140000 kg).
Heavy - Aircraft capable of takeoff weights of 300000 pounds 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.

Interesting cases due to absence of negation or negation-as-failure in PSOA RuleML are the **Large**, **Heavy**, **Super** and **B757**, formalized as follows:

```
Forall ?a (
  :AircraftFAACategory(?a :Large) :-
  And(?a#:Aircraft(:mtow->?w :specialCase->:No)
       math:greaterThan(?w 41000)
       math:lessThan(?w 300000)
       )
   )

Forall ?a (
  :AircraftFAACategory(?a :Heavy) :-
  And(?a#:Aircraft(:mtow->?w :specialCase->:No)
       math:greaterEq(?w 300000)
       )
   )

Forall ?a (
  :AircraftFAACategory(?a :Super) :-
```

```
  Or(?a#:Aircraft(:specialCase->:A380)
     ?a#:Aircraft(:specialCase->:A225))
    )

Forall ?a (
  :AircraftFAACategory(?a :B757) :-
    ?a#:Aircraft(:specialCase->:B757)
    )
```

The Boeing 757 would normally belong to **Large** category, while the Airbus A380 and the Antonov An-225 would belong to **Heavy** category. Therefore, special cases of A380 and A225 are not required in the rule that captures the **Large** category.

The separation standards at the runway threshold for flights under IFR are defined by the Table 2.

Table 2. FAA wake separation standards (nautical miles, at the threshold) [7]

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					

The formalization of the above separation minima regulations consists of six rules. For example, all pairs that require a minimum separation of MRS are formalized in PSOA RuleML as follows:

```
Forall ?x ?y (
:faaSeparation(:leader->?x :follower->?y :miles->:Mrs):-
    Or(
        And(:AircraftFAACategory(?x :Large)
            :AircraftFAACategory(?y :Large))

        And(:AircraftFAACategory(?x :Large)
            :AircraftFAACategory(?y :B757))

        And(:AircraftFAACategory(?x :Large)
            :AircraftFAACategory(?y :Heavy))
```

```
    :AircraftFAACategory(?x :Small)
    :AircraftFAACategory(?y :Super)
    )
)
```

2.3 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) [7,9]:

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

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

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

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

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

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

RECAT separation standards for IFR flights are presented in Table 3.

Table 3. RECAT wake separation standards (nautical miles) [7,9]

		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

As the above regulations do not have exceptions, the previously used slot `:specialCase` is no longer needed. Consequently, the categorization is formalized as in the following example of PSOA RuleML code for the classification of aircraft of RECAT **A** category:

```
Forall ?a (
  :AircraftRecatCategory(?a :A) :-
  And(?a#:Aircraft(:mtow->?w :wingspan->?s)
       math:greaterEq(?w 300000)
       math:greaterThan(?s 245))
)
```

The formalization in PSOA RuleML e.g for the pairs that have a separation minimum of 6 NM is:

```
Forall ?x ?y (
:recatSeparation(:leader->?x :follower->?y :miles->6):-
    Or(
        And(:AircraftRecatCategory(?x :A)
            :AircraftRecatCategory(?y :C))

        And(:AircraftRecatCategory(?x :C)
            :AircraftRecatCategory(?y :F))
        )
)
```

3 Aircraft Facts

Each aircraft entry in the database (which is also implemented in PSOA RuleML), is identified by its ICAO aircraft type designator, which is a two-, three- or four-character alphanumeric code. The aircraft characteristics that are required for the separation minima computation were obtained from [6]. Although this data has not been fully verified, it can serve as a reliable —for the purpose of this paper— source. It contains many "duplicate" entries, as different models with the same ICAO designation exist, having also differences in their characteristics. Whether those differences are important for the obtained results is evaluated in the next Section.

The five slots are `:mtom`, `:mtow`, `:wingspan`, `:appSpeed`, as well as the additional slot `:specialCase` which serves as a "not equal" or a negation-as-failure substitution. Slots `:mtom`, `:mtow` define the Maximum Take-Off Mass (in kilograms), and the Maximum Take-Off Weight (in pounds), respectively. The choice of having both present stems from the desire for more accurate regulation formalization. Due to the automated generation of the relevant PSOA RuleML code (through a script) this was not a problem for the database construction. The slot `:wingspan` defines the wingspan of each aircraft, given in feet, while `:appSpeed` serves as an extra characteristic for future expansion of the KB towards Time-Based-Separation [5] support. An example of a database entry is the following PSOA RuleML aircraft fact:

```
:a321#:Aircraft(:mtom->93499.9
            :mtow->206132.0
            :wingspan->111.88
            :appSpeed->140.0
            :specialCase->:No)
```

This entry is about Airbus A321, which has the ICAO designation code `A321`. An example of the multiple entries presence in the database is the five different occurrences of `:a321` entries, with differences in `:mtom`, `:mtow`, and `:wingspan`. The database contains more than 440 entries in total, while more than 261 are unique.

4 Queries

In the following, we pose representative queries to the KB and demonstrate the answers obtained by PSOATransRun.

Representative queries about aircraft categorization according to ICAO, FAA, and RECAT regulations, along with the obtained answers, are given below:

```
:AircraftIcaoCategory(:a388 ?x)
Answer(s):
?x=<http://psoa.ruleml.org/usecases/ATC_KB#Super>

:AircraftFAACategory(:b752 ?x)
Answer(s):
?x=<http://psoa.ruleml.org/usecases/ATC_KB#B757>

:AircraftRecatCategory(:a346 ?x)
Answer(s):
?x=<http://psoa.ruleml.org/usecases/ATC_KB#B>
```

These queries ask about the categorization of the aircraft with ICAO code `A388`, `B752`, `A346`, according to ICAO, FAA and RECAT regulations, respectively. All answers are in accordance with the actual aircraft classification of Airbus A380 (ICAO designation code: `A388`), Boeing 757-300 (ICAO code: `B753`), and Airbus A340-600 (ICAO code: `A346`). Notice that the first two queries are about special cases discussed in Section 2 and show the correct classification of the aircraft.

Queries about all aircraft belonging to a specific category can also be answered, e.g. using output variable ?x to deduce aircraft that belong to category **B** according to RECAT regulations (some results are omitted):

```
:AircraftRecatCategory(?x :B)
Answer(s):
?x=<http://psoa.ruleml.org/usecases/ATC_KB#c5>
?x=<http://psoa.ruleml.org/usecases/ATC_KB#a332>
?x=<http://psoa.ruleml.org/usecases/ATC_KB#a333>
?x=<http://psoa.ruleml.org/usecases/ATC_KB#a342>
?x=<http://psoa.ruleml.org/usecases/ATC_KB#a343>
...
```

The next queries ask for the separation minimum of aircraft pairs. Three different cases obeying to ICAO, FAA and RECAT regulations are examined for the pairs (leader - follower respectively) Airbus A320 (`A320`) - ATR 72 (`AT72`), Boeing 757-200 (`B752`) - ATR 72, and Airbus A380 (`A388`) - British Aerospace 146 (`B461`):

```
:icaoSeparation(:leader->:a320 :follower->:at72 :miles->?m)
Answer(s):
?m=<http://psoa.ruleml.org/usecases/ATC_KB#Mrs>

:faaSeparation(:leader->:b752 :follower->:at72 :miles->?m)
Answer(s):
?m=4

:recatSeparation(:leader->:a388 :follower->:b461 :miles->?m)
Answer(s):
?m=7
```

The subsequent query asks for all aircraft pairs that have a separation minimum of 3 NM under RECAT regulations (many results are omitted):

```
:recatSeparation(:leader->?x :follower->?y :miles->3)
Answer(s):
...
?x=<http://psoa.ruleml.org/usecases/ATC_KB#a333>,
?y=<http://psoa.ruleml.org/usecases/ATC_KB#b743>
...
?x=<http://psoa.ruleml.org/usecases/ATC_KB#b788>,
?y=<http://psoa.ruleml.org/usecases/ATC_KB#a346>
...
```

Unfortunately, similar queries (such as `:recatSeparation(:leader->?x :follower->?y :miles->:Mrs)`) fail[12] due to the possible massive answer size. This also demonstrates that the KB is suitable for "stress testing" of rule engines and their back-ends.

All the above queries also served as tests of the oncoming SWI Prolog back-end of PSOATransRun. Both tested back-ends (SWI Prolog and XSB Prolog) gave identical answers, while they failed at the same queries. SWI Prolog back-end was noticeably slower ($\sim$ 18 sec versus $\sim$ 1 sec) at the computationally demanding queries of the form `:recatSeparation(:leader->?x :follower->?y :miles->3)`, while other queries were answered without any visible difference in the elapsed time.

One of the benefits of using PSOA RuleML for the formalization is the ability to evaluate the database quality. The effect of multiple entries was evaluated with

[12] If PSOATransRun is invoked without the "-a" option (i.e. return all results), the crash does not happen using the XSB back-end, and results in freezes when using the SWI back-end.

queries about a possibility that an ICAO designation code can refer to different aircraft belonging in different categories at the same time, e.g.:

```
And(:AircraftIcaoCategory(?a :Light)
    :AircraftIcaoCategory(?a :Medium))
```

This yields the result that indeed aircraft with the same ICAO code are placed in different categories:

```
Answer(s):
?a=<http://psoa.ruleml.org/usecases/ATC_KB#b350>
?a=<http://psoa.ruleml.org/usecases/ATC_KB#c207>
```

For the first result, the database has two different entries for the similar Beechcraft models King Air 350ER and King Air 350i with different MTOW, despite having the same ICAO designation code. The second result is a case of completely different aircraft, from different manufacturers (Casa C-207A AZOR and the Cessna 207 family (207 Skywagon, T207 Turbo Skywagon, 207A Stationair 8)) that are having the same ICAO designation code, which was confirmed by manual inspection of the source dataset. Whether both cases are causing a problem in the real world is outside of the scope of this paper.

5 Conclusions

In this paper we have demonstrated the formalization of a subset of Air Traffic Control Regulations, given in a semi-controlled natural language form, into a PSOA RuleML KB. The formalized rules are relevant to the airport vicinity, and specifically to the separation minima at arrival and departure of aircraft. This regulation subset includes the categorization of aircraft into classes according to the wake turbulence separation standards of ICAO, FAA, as well as the newest FAA regulations of RECAT. Additionally, it includes the computation of separation minima for all relevant cases.

PSOA RuleML proved to be a suitable environment for KB development, as a large KB consisting of rules —implementing ATC regulations— and aircraft facts —containing the required characteristics— was implemented. The resulting KB is capable of computing the separation minima mandated by ATC regulations, while using the self-contained database of aircraft facts. This use case (coupled with other examples of regulation formalization KB's such as the European Regulation of Medical Devices [1] or the Port Clearance Rules [17]) provides strong evidence that PSOA RuleML is well-suited to capture real-world problems and PSOATransRun is well-suited for KB development. Despite the lack of literal inequalities, negation and/or negation-as-failure (at the time of the KB development), the PSOA RuleML design allowed for implementing exceptions in rules through the usage of slots, with the downside of possibly inducing a computational or memory overhead.

Due to the database size, some computationally intensive queries invoking combinations of entries that would give thousands of results, caused crashes or freezes. This fact could make the KB a real-world use case benchmark for evaluating reasoning engines performance.

Future work includes the implementation of a more complete set of ATC regulations, such as spatial reasoning applicable to various reduced separation minima cases (e.g. parallel landings), incident management, and aircraft transition zones.

References

1. Almpani, S., Stefaneas, P., Boley, H., Mitsikas, T., Frangos, P.: Computational Regulation of Medical Devices in PSOA RuleML. In: Benzmüller, C., Ricca, F., Parent, X., Roman, D. (eds.) Rules and Reasoning. pp. 203–210. Springer International Publishing, Cham (2018)
2. Boley, H.: A RIF-Style Semantics for RuleML-Integrated Positional-Slotted, Object-Applicative Rules. In: Proc. 5th International Symposium on Rules: Research Based and Industry Focused (RuleML-2011 Europe), Barcelona, Spain. pp. 194–211. Lecture Notes in Computer Science, Springer (Jul 2011)
3. Boley, H.: PSOA RuleML: Integrated Object-Relational Data and Rules. In: Faber, W., Paschke, A. (eds.) Reasoning Web. Web Logic Rules (RuleML 2015) - 11th Int'l Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures. LNCS, vol. 9203. Springer (2015)
4. Erzberger, H.: Design principles and algorithms for automated air traffic management. Knowledge-Based Functions in Aerospace Systems 7, 2 (1995)
5. EUROCONTROL: Future airport operations. `https://www.eurocontrol.int/articles/future-airport-operations`, accessed: 2018-06-16
6. FAA: Aircraft characteristics database. `https://www.faa.gov/airports/engineering/aircraft_char_database/`, accessed: 2018-06-16
7. FAA: Advisory Circular 90-23G - Aircraft Wake Turbulence (2014)
8. FAA: ORDER JO 7110.65V, Air Traffic Control (2014)
9. FAA: Order JO 7110.659C, Wake Turbulence Recategorization (2016)
10. Gorodetsky, V., Karsaev, O., Samoylov, V., Skormin, V.: Multi-Agent Technology for Air Traffic Control and Incident Management in Airport Airspace. In: Proceedings of the International Workshop Agents in Traffic and Transportation, Portugal. pp. 119–125 (2008)
11. ICAO: Doc 4444-RAC/501, Procedures for Air Navigation Services - Rules of the Air and Air Traffic Services
12. 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. AIAA Atmospheric and Space Environments Conference (2010)
13. McCluskey, T., Porteous, J., Naik, Y., Taylor, C., Jones, S.: A requirements capture method and its use in an air traffic control application. Software: Practice and Experience 25(1), 47–71 (1995)
14. Mitsikas, T., Stefaneas, P., Ouranos, I.: A Rule-Based Approach for Air Traffic Control in the Vicinity of the Airport. In: Algebraic Modeling of Topological and Computational Structures and Applications. pp. 423–438. Springer International Publishing, Cham (2017)
15. 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) (jun 2014), `http://dx.doi.org/10.2514/6.2014-2861`

16. Zou, G., Boley, H.: PSOA2Prolog: Object-Relational Rule Interoperation and Implementation by Translation from PSOA RuleML to ISO Prolog. In: Proc. 9th International Web Rule Symposium (RuleML 2015), Berlin, Germany. Lecture Notes in Computer Science, Springer (Aug 2015)
17. Zou, G., Boley, H., Wood, D., Lea, K.: Port Clearance Rules in PSOA RuleML: From Controlled-English Regulation to Object-Relational Logic. In: Proceedings of the RuleML+RR 2017 Challenge. vol. 1875. CEUR (Jul 2017), `http://ceur-ws.org/Vol-1875/paper6.pdf`