Modelling Demand-Based Runway Reconfigurations

A Symbolic AI Approach

by

J.J.P. van den Berg

To obtain the degree of Master of Science
at the Delft University of Technology

Image source[adapted]: Artificial Intelligence - Resembling Human Brain, by www.iqlect.com, licensed under CC BY-SA 2.0

Modelling Demand-Based Runway Reconfigurations

A Symbolic AI Approach

by

J.J.P. van den Berg

To obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday March 6, 2020 at 10:00 AM.

Student number:	4094166	
Project duration:	October 20, 2018 – March 6, 2020	
Thesis committee:	Dr. ir. B.F. Lopes dos Santos,	TU Delft, ATO, chair
	Dr. O.A. Sharpanskykh,	TU Delft, ATO, supervisor
	Ir. M. Tielrooij,	To70, supervisor
	Ir. O. Stroosma,	TU Delft, C&S, external examiner

Contents

Nomenclature vi

List of Figures ii

List of Tables iv

1 Introduction 1

2 Research Context 3
- 2.1 Operational Context 3
  - 2.1.1 Runway and Airspace Structure 3
  - 2.1.2 Operational Constraints 4
  - 2.1.3 Inbound Planning 6
  - 2.1.4 Outbound Planning 8
  - 2.1.5 Runway Reconfiguration Process 9
- 2.2 High-Level Research Requirements 11

3 Literature Overview 13
- 3.1 State of the Art 13
  - 3.1.1 Existing Runway Configuration Management Models 13
  - 3.1.2 Research Gap 14
- 3.2 Modelling Technique Alternatives 15
  - 3.2.1 Modelling and Simulation Techniques 15
  - 3.2.2 Cognitive Concepts 16
  - 3.2.3 Negotiation Technique 18
  - 3.2.4 Reasoning Technique 20

4 Research Objective and Methodology 23
- 4.1 Research Objective 23
- 4.2 Scope 24
- 4.3 Methodological Steps 25

5 Conceptual Model 27
- 5.1 Overview 27
- 5.2 Environment 28
  - 5.2.1 Clock Variable and Global State 28
  - 5.2.2 Airspace 28
  - 5.2.3 Aerodrome 28
  - 5.2.4 Meteorological Elements 29
- 5.3 Domain Model 29
  - 5.3.1 Agent Types 29
  - 5.3.2 Inbound Domain Model 31
  - 5.3.3 Outbound Model 34
- 5.4 Control Model 36
  - 5.4.1 Overview 36
  - 5.4.2 External Supervisor Agent Elements 38
  - 5.4.3 Internal Supervisor Agent Elements 39
- 5.5 Interactions 42
  - 5.5.1 Interaction Among Agents 42
  - 5.5.2 Interaction Between Agents and Environment 42

6 Model Description 43
6.1 Formal Languages . . . 43
6.1.1 LEADSTO Language. . . . 43
6.1.2 Algorithmic Description . . . 43
6.1.3 Defeasible Logic Programming Language . . . 44
6.1.4 Dialogue Game Protocol . . . 44
6.2 Global Clock . . . 44
6.3 Flight Agent . . . 45
6.3.1 Characteristics . . . 45
6.3.2 Perception Capabilities . . . 45
6.3.3 Actions and Cognitive Processes . . . 45
6.4 Planner Agent . . . 46
6.4.1 Characteristics . . . 46
6.4.2 Perception Capabilities . . . 46
6.4.3 Actions and Cognitive Processes . . . 46
6.5 Air Traffic Control Agent . . . 47
6.5.1 Characteristics . . . 47
6.5.2 Perception Capabilities . . . 47
6.5.3 Actions and Cognitive Processes . . . 47
6.6 Supervisor Agent . . . 48
6.6.1 Characteristics . . . 49
6.6.2 Perception Capabilities . . . 49
6.6.3 Actions and Cognitive Processes . . . 49

7 Simulation Scenario 57
7.1 Selection of Simulation Scenario . . . 57
7.2 Scenario Parameters . . . 58
7.2.1 Scenario Schedule . . . 58
7.2.2 Weather Scenario. . . . 59
7.2.3 Trajectory Distances and Taxi Times . . . 59
7.2.4 Aircraft Speeds . . . 60
7.2.5 Start and Landing Intervals . . . 60

8 Validation and Results 61
8.1 Internal Validation . . . 61
8.1.1 Inbound Model . . . 61
8.1.2 Outbound Model . . . 62
8.1.3 Control Model . . . 63
8.2 Simulation Results . . . 65
8.2.1 Validity of the Domain Model . . . 65
8.2.2 Reasoning Procedures . . . 69

9 Conclusions and Recommendations 77
9.1 Main Findings . . . 77
9.2 Contribution . . . 79
9.3 Recommendations for Further Research. . . . 80

A Schematic Rules 81

B Protocol 85

C Trajectory Prediction Model 93
C.1 FIR boundary- IAF segment . . . 93
C.2 IAF - Runway segment . . . 94

D Scheduling Algorithms 95
D.1 Arrival Scheduling . . . 95
D.1.1 Entering a Flight in the Pre-Planning Phase . . . 95
D.1.2 Runway Assignment . . . 95
D.1.3 Update the Planning Status . . . 96
D.1.4 Sort Flights in the Planning . . . 96
D.1.5 Assign Landing Intervals . . . 96

D.1.6 FCFS Scheduling 96
D.2 Departure Scheduling 97
D.2.1 Entering a Flight in the Planning 97
D.2.2 Runway Assignment 97
D.2.3 Sorting the Flights Before Planning 97
D.2.4 Assigning a Start Interval 98
D.2.5 Scheduling of the Flights 98

E Data Analysis 99
E.1 Input Data 99
E.2 Trajectory and Distance Determination 99
E.3 Speed Parameters 101
E.4 Taxi Times 101
E.5 Weather Parameters 101
E.6 Data Tables 102

F Example of Simulation Trace 109

G Graphical User Interface 111

Bibliography 113

Nomenclature

List of Abbreviations

AAA	Amsterdam Advanced Air Traffic Management
AAS	Amsterdam Airport Schiphol
ABN	Argumentation-Based Negotiation
ACC	Area Control Center
AMAN	Arrival Manager
APP	Approach
ASAP	Advanced Schiphol Arrival Planner
ATC	Air Traffic Control
ATM	Air Traffic Management
ATS	Air Traffic Service
CDA	Continuous Descent Approach
CPDSP	Collaborative Pre-Departure Sequence Planning
CTA	Control Area
CTOT	Calculated Take-Off Time
CTR	Control Zone
DeLP	Defeasible Logic Programming
EAT	Expected Approach Time
EOBT	Estimated Off-Block Time
ETA	Estimated Time of Arrival
FAF	Final Approach Fix
FAFS	First Arrive First Serve
FIR	Flight Information Region
FL	Flight Level
IAF	Initial Approach Fix
ICAO	International Civil Aviation Organization
LAS	Latest Assigned Slot
LIV	Landing Interval
LPwNF	Logic Programming without Negation as Failure
LVNL	Air Traffic Control the Netherlands
MUAC	Maastricht Upper Airspace Control
NM	Nautical Mile
OVV	Onderzoeksraad Voor Veiligheid (Dutch Council For Safety)
RETD	Revised Estimated Time of Departure
RNAV	Area Navigation
SID	Standard Instrument Departure
STAR	Standard Arrival Route
SUP	Supervisor
SVM	Support Vector Machine
TMA	Terminal Control Area
TOBT	Target Off-Block Time
TSAT	Target Start-Up Approval Time
TTOT	Target Take-Off Time
TWR	Tower
UDP	Uniform Daylight Period
UTA	Upper Control Area

List of Symbols

$\leftarrow$	Strict relation
$\twoheadrightarrow$	LEADSTO dynamic property operator
$-\!\!\prec$	Defeasible relation
$\wedge$	Logical conjunction
$\vee$	Logical disjunction
$\in$	Set membership
$\forall$	Universal quantification
$\exists$	Existential quantification
$\neg$	Logical complement
$\cup$	Set union
$\cap$	Set intersection
$\sim$	Strong negation
$\succ$	Complete and transitive preference
$\vdash$	Syntactical consequence
$\mid\!\sim$	Defeasible consequence
$\subset$	Proper subset
$\subseteq$	Subset

List of Figures

2.1 Layout of the runway system of AAS 4
2.2 Two-dimensional top view of airspace areas 4
2.3 Vertical structure of airspace layers 5
2.4 Illustration of the route structure in and above the Dutch FIR 5
2.5 Overview of the possible temporal-spatial displacement options [22] 6
2.6 Graphical representation of the phases in the ASAP planning system[69] 7
2.7 Schematic overview of slot assignment 7
2.8 Schematic overview of the outbound planning process 8
2.9 Example traffic demand overview [70] 10
2.10 Southerly and northerly runway use 10
2.11 Inbound trajectory interference caused by runway reconfiguration 11
2.12 Example of departure flight trajectory interaction 11

3.1 Cognitive architecture for anticipation[12] 17
3.2 Cognitive architecture for ABN agent [58] 18

4.1 Schematic overview of the methodological steps 25

5.1 High-level architecture of the conceptual model 27
5.2 Schematic overview of the airspace elements 28
5.3 Schematic overview of Aerodrome elements 28
5.4 Action selection for the flight agent 29
5.5 Action selection for the ATC agent 30
5.6 Action selection for the planning agent 31
5.7 Conceptual inbound model overview 32
5.8 Overview of the trajectory legs for an arrival flight agent 33
5.9 Overview of the nominal paths used for trajectory prediction 33
5.10 The FAFS queue with one active inbound runway 34
5.11 The FAFS queue with two active inbound runways 34
5.12 Overview of the conceptual outbound model 35
5.13 The trajectory of a departure flight agent 35
5.14 Departure planning queues 36
5.15 Overview of the relation of departure runways and sectors 36
5.16 Overview of the conceptual model elements of the agent-based model 37
5.17 Sense of urgency for the initiator 39
5.18 Option generation example for activating a runway 40
5.19 Alternative selection 41
5.20 Agent interaction 42
5.21 Supervisor agent protocol 42

6.1 Warranting procedure 53
6.2 Structure of a locution 54

7.1 Runway combinations for northerly runway use at AAS 57
7.2 Schematic overview of the intersection points of an arrival trajectory 59
7.3 Schematic overview of Aerodrome elements and an example departure trajectory 59
7.4 Distribution of the Mode-s meteorological data 59

8.1 Interface used for verifying the ATC instructions 62
8.2 Visualisation used for verifying the inbound planning 62
8.3 Visualisation used for verifying the outbound planning 63
8.4 Visualisation of the upcoming demand peak 63

8.5 Visualisation of the delay values in the planning 63
8.6 Visualisation used for verifying the argumentation module 64
8.7 Deviation between actual arrival times and simulation arrival times for runway 06 65
8.8 Deviation between actual arrival times and simulation arrival times for runway 36R 65
8.9 Deviation of departure times with realistation for runway 36L 66
8.10 Deviation of departure times with realistation for runway 36C 66
8.11 Comparison of the actual time and simulation time for runway reconfigurations 67
8.12 Planning information for the first decision scenario 70
8.13 Prediction of the state evolution of the outbound system 70
8.14 Planning information for negotiation interaction 2 72
8.15 Prediction of the state evolution of the inbound and outbound system for negotiation interaction 2 72
8.16 Results of mental simulation for deactivating a departure runway before flight TFL191 74

C.1 Trajectory prediction overview 93

D.1 Process flow for the scheduling update 95
D.2 The FAFS queue in a one-runway active scenario 96
D.3 The FAFS queue in a two-runway active scenario 96
D.4 Process flow for the departure scheduling update 97
D.5 Distribution of flights for two active departure runways 98

E.1 Heatmap of the ADS-B track data 99
E.2 Density plot with grid for the mode-s data 99
E.3 Visualisation of splitting ADS-B trajectories for analysis 100
E.4 Examples of ACC flight paths 100
E.5 Examples of TMA flight paths 100
E.6 Speed profile for Boeing 737-800 101

F.1 Visualisation of simulation trace for the APP supervisor agent 109
F.2 Visualisation of simulation trace for the TWR supervisor agent 110
F.3 Visualisation of state of observation values for the supervisor agents 110

G.1 Graphical user interface 111

List of Tables

2.1 Preferential runway table for daytime operations [2] 5
2.2 Preferential runway table for nighttime operations [2] 5
2.3 Demand peak schedule for AAS 9

6.1 Reification pattern 51

7.1 Values used for landing interval 60
7.2 Values used for start interval 60

8.1 Sample mean and standard deviation for deviation of arrival times 65
8.2 Sample mean and standard deviation for deviation of departure times 66
8.3 Comparison of the actual and simulation times for inbound runway reconfiguration 67
8.4 Comparison of the actual and simulation times for outbound runway reconfiguration 67

E.1 ACC trajectory data for flights to ARTIP 102
E.2 ACC trajectory data for flights to RIVER 103
E.3 ACC trajectory data for flights to SUGOL 104
E.4 TMA trajectory data 104
E.5 Aircraft speeds per type part 1 105
E.6 Aircraft speeds per type part 2 106
E.7 Inbound taxi times for runway 06 107
E.8 Inbound taxi times for runway 36R 107
E.9 Outbound taxi times for runway 36L 107
E.10 Outbound taxi times for runway 36C 107
E.11 Weather data sample 108

1

Introduction

Amsterdam Airport Schiphol (AAS) saw a record number of passengers during the summer of 2018, with an average of 220,000 passengers per day[61]. Over recent years, the airport has recorded an increase in passenger numbers every year. Projections of the International Air Transportation Association show a significant increase in demand for air travel worldwide in every of the projection scenarios[38]. It is therefore highly likely that the demand for flying from AAS will continue to increase. This poses a problem because the airport is surrounded by densely populated municipalities. Apart from passenger numbers, the airport also broke records in terms of complaints from the surrounding community[53].

To mitigate nuisance, the airport is assigned a fixed quota in terms of environmental pollution and external safety. This leads to the airport being restricted to a certain amount of flights per year and the operations being highly constrained by noise abatement procedures. A major factor in these noise abatement procedures is the "Fourth runway rule". This rule is aimed at minimising operations with four runways in use simultaneously, by mandating that under normal circumstances no more than 40 flights per day may be handled using the "fourth runway"[3]. As a result, it is required to operate predominantly using configurations with three active runways throughout the day. Facilitating the alternating pattern of inbound and outbound peaks at AAS, using only three active runways, requires switching between using a second runway for inbound traffic and using a second runway for outbound traffic. These type of reconfigurations are referred to as demand-based runway reconfigurations.

The "Fourth runway rule" combined with varying wind conditions at AAS, results in the airport seeing 18 runway reconfigurations per day on average[54]. According to investigations by the Dutch council for safety(OVV), the high amount of runway reconfigurations poses elevated safety risks[55]. Apart from these elevated safety risks, runway reconfigurations are considered disruptive events that can lead to substantial capacity reductions for the airport system [4]. Considering these facts, it is critically important to establish an adequate understanding of the phenomenon thereby enabling the estimation of the effects and generating the potential for mitigating these effects. The research described in this report is aimed at developing this understanding by exploring the dynamics, decision-making processes, quantitative aspects, and qualitative aspects related to the runway reconfiguration phenomenon.

For an airport like AAS, with a complex runway structure, performing a safe and efficient runway reconfiguration is no trivial task. This task is in reality performed through an interaction between two highly skilled and experienced air traffic control operators, that bear responsibility for mutually exclusive control areas of the inbound-outbound system. Investigating this complex socio-technical phenomenon is highly challenging due to the many interactions, uncertainties, and qualitative aspects involved. Especially the aspect of human intelligence involved in this phenomenon is hard to capture and fully understand.

Cognitive modelling is a research field that is concerned with exploring the processes and structures of the human mind. Cognitive modellers are capable of explaining an aspect of human intelligence by creating a model of it[17]. In light of this observation, it is hypothesized that developing a better understanding of the runway reconfiguration phenomenon can be best achieved by modelling it using concepts of cognitive modelling. This leads to the central research question of how this should be done or more specifically:

> *How can a model be developed that accurately reflects the demand-based runway reconfiguration phenomenon and allows for gaining a detailed understanding of the processes involved by building the model?*

To answer this question, a closer look is taken at the context of the research and what high-level research requirements follow from this context in chapter 2. Translating these requirements into a capable model requires establishing a set of suitable modelling techniques. This is done by investigating the capabilities of modelling techniques used in existing models, as well as investigating alternative modelling techniques that could potentially resolve discrepancies between capabilities of existing models and the posed high-level modelling requirements, i.e. the research gap. This is described in chapter 3 and leads to following an agent-based approach with a focus on cognitive concepts for anticipation, negotiation, and argumentation. After selecting the modelling techniques, a structure for carrying out the required research steps using these techniques is defined in chapter 4.

To provide a high-level overview of the model and elaborate upon the design decisions, a conceptual model is developed in chapter 5. This is followed by a more detailed definition of the model in chapter 6. Using the developed model, it is aimed to investigate how well the model reflects the decisions taken in reality. To this end, a simulation scenario is defined in chapter 7. An analysis of the operationalisation of the model for the defined simulation scenario is provided in chapter 8. In this analysis, it is found that the model constructed using agent-based techniques and cognitive architectures for anticipation and argumentation is capable of emulating the timing of runway configuration decisions taken in reality well. A maximum deviation of 14 minutes and an average deviation of only four minutes is observed when comparing the runway configuration timing of the model with the actual operations for a selected scenario day. To conclude the report, conclusions and recommendations for further research are discussed in chapter 9.

2

Research Context

As a first step towards establishing a better understanding of runway reconfigurations at AAS, a closer look is taken at the operational aspects of the phenomenon. This operational context is comprised of the involved infrastructural elements, actors, and procedures. After gaining a clearer view of these aspects, requirements can be posed on which of these aspects must be included in the research. Exploration of the operational context combined with a definition of the requirements, together serve as a basis for this research.

2.1 Operational Context

AAS is selected as the airport of interest for studying the phenomenon of a runway reconfiguration. To this effect, it is desired to obtain insights into relevant aspects of the operations at AAS. These insights are presented in this section, starting with the runway and airspace structure designated for AAS. This is followed by a description of the operational constraints, the inbound/outbound planning process, and the procedure for runway reconfigurations.

2.1.1 Runway and Airspace Structure

Varying wind conditions at AAS, give rise to having runways in multiple compass directions to facilitate headwind landings and take-offs. This results in a runway structure that is relatively complex compared to other airports with similar capacities. The infrastructure layout of AAS is structured around a centrally located traffic area that consists of 90 gates, 73 passenger aprons, and 25 cargo aprons. This area is surrounded by a derived version of a tangential runway system, with six runways, aimed at avoiding runway intersections.

The runway system is depicted in figure 2.1, where designations are based on the compass directions of the runway ends. If a runway is used towards the south and the runway end in that direction is pointing towards 180 degrees, it is designated as 18. For parallel runways this designation can be supplemented by a letter L, R or C indicating the left, right or centre runway. Five out of the six runways at AAS have lengths varying between 3300 and 3800 meters and are capable of accommodating all types of aircraft[26]. Runway 04-22, is shorter and is mostly used for general aviation traffic. The runways are most often used in segregated mode and can be used in both directions, with two exceptions. For safety reasons, runway 36L and 18L are only used for starts and 36R and 18R are only used for landings.

The Dutch flight information region (FIR), which has the ICAO designation EHAA, is divided into three vertically separated layers [26]. The layer above flight level (FL) 245 is called upper airspace and is controlled by Maastricht Upper Airspace Control (MUAC). The lower airspace layer is located below the upper airspace layer between FL 95 and FL 245. This layer is controlled by Air Traffic Control the Netherlands (LVNL) and is constituted by a part of the upper control area (UTA) below the division flight level, FL 245, and a part of the control area (CTA) between FL 195 and FL 95. The airspace around the airport is the lowest airspace layer, which ranges from the ground to FL 95. Both the terminal control area (TMA) between FL 95 and 1500 feet and the Control zone (CTR) between 3000 feet and the ground are located in the airspace around the airport and are controlled by LVNL. An overview of these control areas is provided in figure 2.2 and figure 2.3, depicting their vertical as well as their geographical boundaries.

Apart from dedicated airspace areas, dedicated routes through EHAA are available for aircraft arriving at AAS or departing from AAS. An overview of the different route structures in or above the FIR is presented in figure 2.4. The highest elevated routes are called air traffic service(ATS) routes. These can be regarded as highways in the upper airspace and are connecting a global network of airports [26]. If a flight arrives from one of the ATS routes and requests its top of descent to enter EHAA, the flight will normally be directed to one of the standard arrival

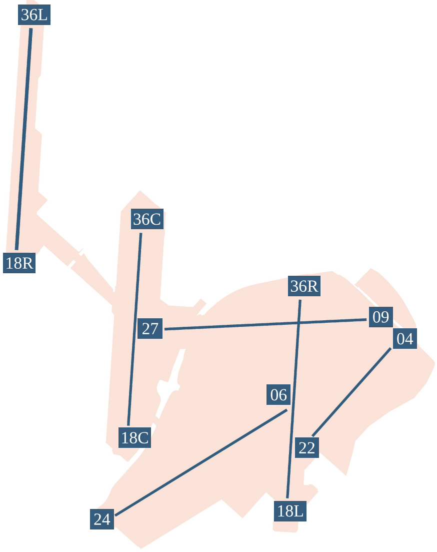


Figure 2.1: Layout of the runway system of AAS

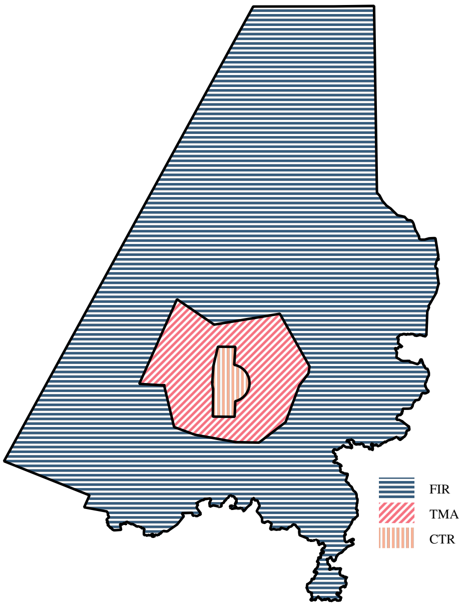


Figure 2.2: Two-dimensional top view of airspace areas

routes (STAR). A STAR is used to guide flights towards the initial approach fix (IAF). This IAF is a geographical point at a specified altitude, where the STARs for inbound traffic from different directions merge. For EHAA three IAFs are defined called ARTIP, RIVER and SUGOL. The geographical location of the IAFs is presented in figure 2.4. These points can be used to generate desired delays for inbound flights by instructing holding patterns.

When an aircraft leaves the IAF, it is guided towards a final approach fix (FAF) where it will line up for the final approach segment. During the daytime, flights at AAS are generally guided from the IAF to the FAF by instructing radar vectors. The only runway for which an area navigation (RNAV) procedure is available during the daytime, that could be used instead of radar vectors, is runway 36R. During nighttime (20:30-04:30 UTC), all inbound traffic is guided from the IAF towards the FAF using RNAV procedures in combination with a continuous descent approach (CDA) to aid in noise abatement.

For outbound traffic, standard instrument departure (SID) routes are available. These SID routes are specified for each runway and guide aircraft towards one of the ATS routes. Depending on the traffic situation, different SIDs can be used between a departure runway and a ATS route. The selection of such a SID is determined by the active runway combination and the departure direction of a flight. A visualisation for some of these SIDs specified for runway 24 is presented in figure 2.4.

2.1.2 Operational Constraints

In controlled airspace, minimum separation distances between aircraft are required to ensure safe operation. Aircraft can either be separated vertically or laterally. The minimum vertical separation distance in the Dutch FIR is 1000 feet. The lateral separation is in general 3 NM (Nautical Mile) for distances less than 30 NM from the radar source and 5 NM for distances greater than 30 NM from the radar source. There are however circumstances in which an increased lateral separation distance is required. An increased lateral radar separation minimum applies when wake turbulence is an issue.

Wake turbulence becomes an issue when two aircraft are behind each other or crossing and are less than 1000 feet vertically separated. Separation minima for wake turbulence mostly depend on the weight and wingspan of the aircraft. ICAO has defined four categories for aircraft defining their wake turbulence category. These categories are SUPER, HEAVY, MEDIUM and LIGHT. Depending on the wake turbulence categories of aircraft pairs, different separation minima are defined ranging between 4 NM and 8 NM.

Given the wind limits and required capacity, the runway combination should be chosen in such a way that the noise pollution for the surrounding municipalities is reduced as much as possible. This is enforced by a preferential runway combination list. The preferential runways during the daytime operations are depicted in table 2.1. It is aimed to use combinations that are highest in this table. Furthermore, L1 should be selected if only one runway is used to handle inbound traffic and S1 should be selected if only one runway is used to handle outbound traffic.

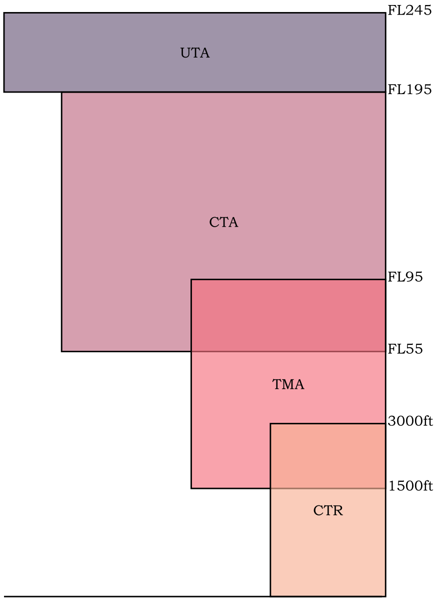


Figure 2.3: Vertical structure of airspace layers

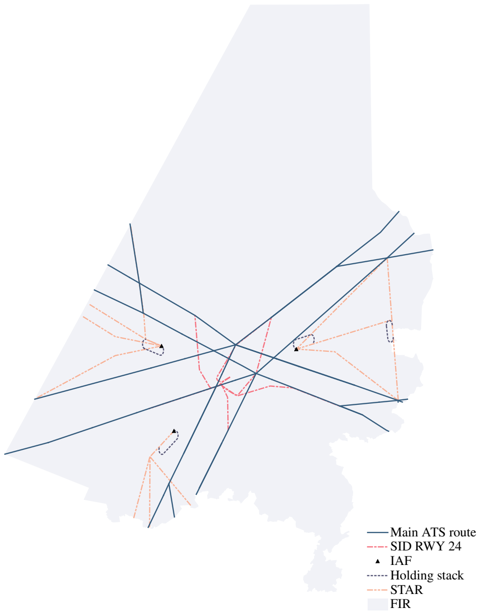


Figure 2.4: Illustration of the route structure in and above the Dutch FIR

The preference order for nighttime operations is presented in table 2.2, where only options with one active runway can be selected for each direction. In general, the runways 18R-36L and 06-24 should be used as much as possible. These runways are therefore called the primary runways and cause the least noise pollution.

The preferential use of runway combinations is determined per scenario. For good visibility conditions in the uniform daylight period (UDP), a feasible runway combination with the highest preference of combination 1-4 in table 2.1 should be selected. For good visibility conditions outside UDP, 5a or 5b should be selected if feasible and for marginal visibility conditions, 6a and 6b may also be selected. For nighttime operations, a feasible runway combination should be selected from table 2.2.

Table 2.1: Preferential runway table for daytime operations [2]

Preference	Land		Start	
	L1	L2	S1	S2
1	06	36R	36L	36C
2	18R	18C	24	18L
3	06	36R	09	36L
4	27	18R	24	18L
5a	36R	36C	36L	36C
5b	18R	18C	18L	18C
6a	36R	36C	36L	09
6b	18R	18C	18L	24

Table 2.2: Preferential runway table for nighttime operations [2]

Preference	Land	Start
1	06	36L
2	18R	24
3	36C	36L
4	18R	18C

Another important factor influencing the selection of runway combinations is the availability of runways. During certain periods of the year, maintenance is carried out. If a runway is under maintenance it cannot be used and the runway combinations including this runway will become unavailable. Other activities like lawn mowing or bird checks can temporarily cause unavailability of runways. The aforementioned events are scheduled events, but runways may also become unavailable due to unexpected events. Bird strikes, technical failures or runway incursions may lead to unexpected temporary unavailability. In these cases, a sudden runway reconfiguration may be required.

2.1.3 Inbound Planning

To ensure efficient use of the runway capacity under the operational constraints, proper inbound planning is required. The inbound planning process is performed iteratively by the approach planner. In the case of AAS, this is the approach supervisor. This iterative process is performed using the following information sources:

- Filed flight plans.
- Trajectory predictions.
- The active runway configuration.
- Separation minima.
- Mix of aircraft types.
- Weather conditions.

For the inbound planning process, the approach planner heavily relies on the Amsterdam Advanced ATM (AAA) system. This system is a central unit that, amongst others, collects and processes the aforementioned information relevant to inbound planning. Part of this system is the Advanced Schiphol Arrival Planner (ASAP), which is used to regulate the inflow of aircraft into the TMA. This is done by planning the arrival times of aircraft at the runway threshold, through the assignment of landing slots. These landing slots can be used to derive the time that an aircraft should leave the IAF and enter the TMA, referred to as the expected approach time (EAT). Area control is responsible for making sure that aircraft achieve this EAT, with some small error margin. To accomplish this, area control can use a variety of temporal displacement options in the flight legs up to the IAF or at the IAF. The possible options for generating these temporal displacements are displayed in figure 2.5.

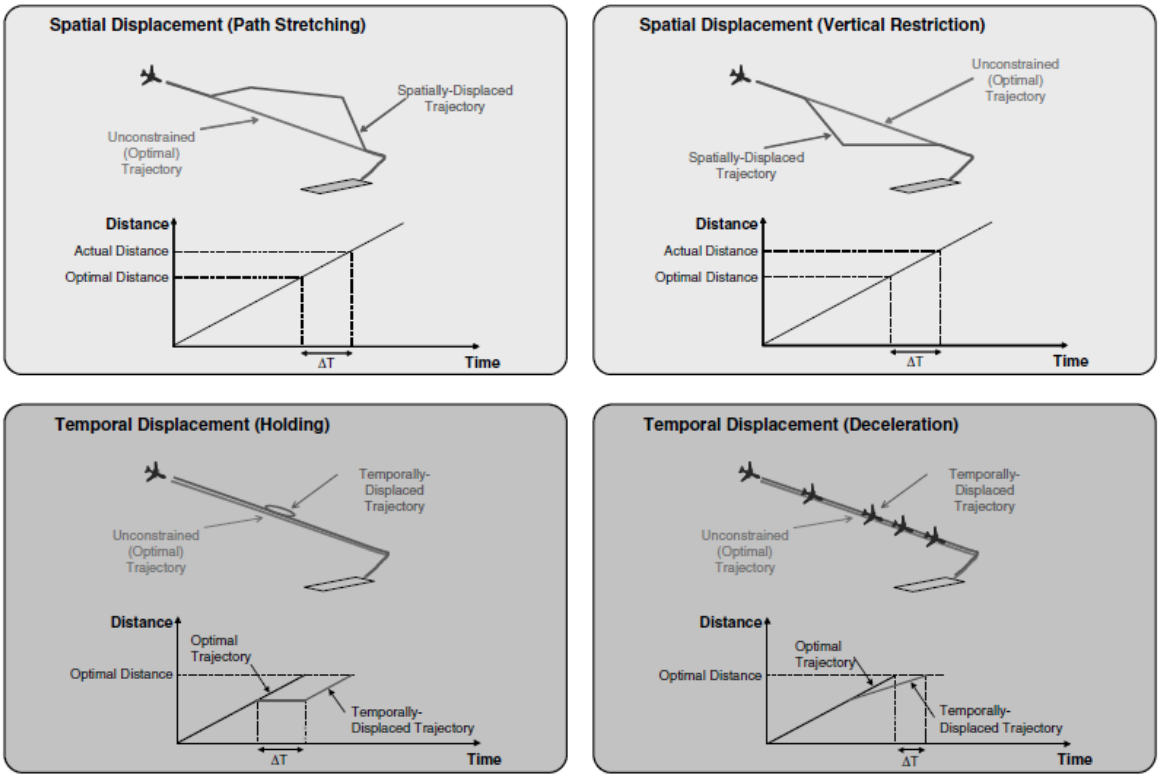


Figure 2.5: Overview of the possible temporal-spatial displacement options [22]

The EAT of a flight is established through a planning process by the ASAP system. This planning process consists of different planning phases depicted in figure 2.6. A flight appears in the planning system three hours before its expected time of arrival, when the flight plan is received from the AAA system. At this point, the flight will not be planned yet. Once the system receives signals that the flight is on its way and is therefore confident that the flight will arrive close to its expected arrival time, the flight enters the pre-planning phase. In the pre-planning phase, flights are assigned a runway and the first available slot that is later than the estimated time of arrival at the runway threshold (ETA) of the flight. This is done by an automated process that computes ETAs using trajectory prediction and schedules on a first come first serve basis. The pre-planning phase ends when the EAT of a flight is within 14 minutes for the first time. At this point, the scheduled arrival time is "frozen". This means that the scheduled arrival sequence can not be changed automatically by the planning process anymore.

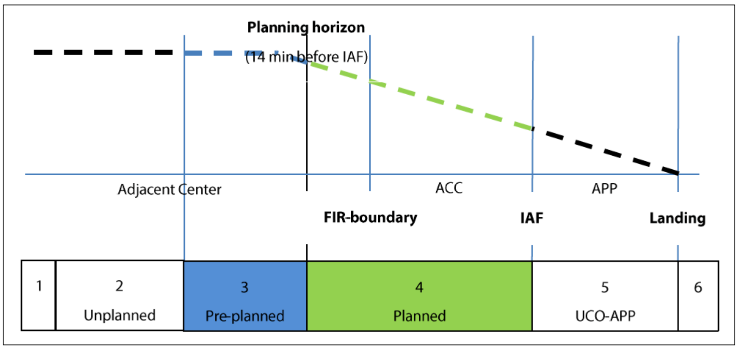


Figure 2.6: Graphical representation of the phases in the ASAP planning system[69]

Once a flight is frozen, it enters the planned phase. In this phase, the sequence can only be changed manually by the approach planner. Revised estimates for arrival times are used in this phase by the ASAP system to adapt the EATs for flights in the frozen sequence. When a flight passes the IAF, the flight is under the responsibility of approach control and the planning system is not leading anymore. From this point onward, approach control uses radar vectors to very precisely provide instructions that ensure aircraft are lined up according to the minimal separation criteria at the FAF, for optimal throughput.

The EATs are planned by the ASAP system in such a way, that the TMA capacity is never exceeded. This TMA capacity is limited by the available runway capacity for inbound traffic. To ensure that this runway capacity is not exceeded, landing slots are assigned. These slots are assigned based on the ETA of a flight and a required landing interval (LIV), see figure 2.7. A flight that has a ETA later than the latest assigned slot for a runway plus the required LIV, is assigned a landing slot time that is equal to its ETA. However, if the ETA of a flight is earlier than the latest assigned slot plus the LIV, this flight cannot be assigned a slot equal to its EAT without violating the required LIV. This required LIV is based on separation criteria and may not be violated. The slot of the flight is therefore set at a time later than its ETA, such that the required LIV after the latest assigned slot is maintained.

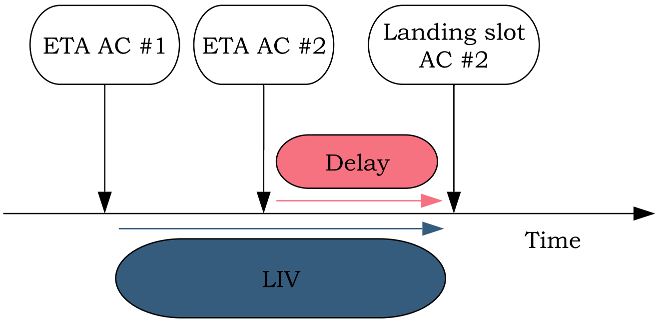


Figure 2.7: Schematic overview of slot assignment

Assigning a slot that is later than the ETA of a flight, means that the flight will have to be delayed by a temporal displacement that is equal to its assigned slot minus its ETA. This delay can already be generated in the flight legs in the CTA by delaying the EAT of a flight. To determine the new EAT, the nominal time for flying from the IAF to the assigned runway is subtracted from the assigned slot. If a delayed EAT implies a temporal displacement that cannot be achieved anymore by a spatial displacement or a deceleration instruction (see figure 2.5), only a holding instruction remains a viable option. As a rule of thumb four minutes can be regarded as the threshold for this.

Instructing a holding pattern is highly undesirable as it introduces a lot of uncertainty, which can cause an increased workload for air traffic controllers. This uncertainty stems from the fact that when a flight is instructed to leave the holding pattern, it becomes unpredictable at what time the flight exactly does this. If this results in an undesirable time for leaving the IAF, it will cause a high workload for air traffic controllers to restore the desired separation between flights.

2.1.4 Outbound Planning

Outbound planning is aimed at optimising the outbound runway capacity. At AAS, the outbound planner is responsible for this task. The outbound planning is performed by assigning a unique planned take-off time to every departure flight, using a process schematically depicted in figure 2.8. In this process, the following time parameters are involved:

- **TOBT**: Time at which the ground handling process is expected to be concluded and the flight is ready to receive its start-up clearance.
- **Taxi-time**: Estimated time for taxiing from the apron to the assigned runway.
- **TTOT**: Time at which the flight is planned for take-off.
- **TTOT'**: The earliest possible time at which a flight can be planned for take-off (TOBT + taxi time).
- **TSAT**: Time at which the flight is planned to receive start-up approval (TTOT - taxi time).

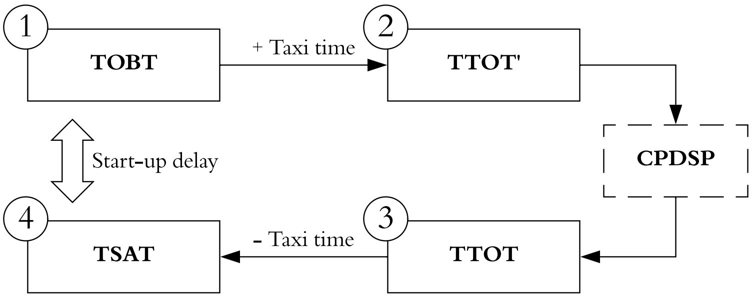


Figure 2.8: Schematic overview of the outbound planning process

An important tool used by the outbound planner in the planning process is the Collaborative Pre-Departure Sequence Planning (CPDSP) module. The CPDSP module takes the TOBT's of departure flights and uses them to establish a sequence of planned take-off times that optimises runway capacity. In the process of establishing this sequence, the following factors are taken into account as constraints:

- **Earliest possible take-off time**: A flight cannot be planned earlier than its earliest possible take-off time.
- **Wake turbulence**: The sequence should respect the separation minima required to avoid wake turbulence issues.
- **Runway capacity**: Throughput of the planned sequence for a runway may not exceed the declared capacity of that runway.
- **Regulated flights**: A flight that is regulated by Eurocontrol should be given priority in the sequence.
- **Departure routes**: Subsequent flights in a sequence that are flying the same departure route should be separated sufficiently to avoid separation issues after take-off.

New TTOTs are generated by the CPDSP upon every change in the planning parameters. This is an automated process in which the manual adjustments by the outbound planner are taken into account. When the outbound planner provides start-up clearance for a flight, it will enter a revised estimated time of departure (RETD) for a flight. This RETD value will be used by the CPDSP module as a TTOT that cannot be adjusted by the algorithm. Additionally, the outbound planner can change the sequence, by manually entering TTOT values for flights. These manual adjustments will be entered into the sequence firstly by the algorithm establishing the sequence of TTOTs.

The TTOTs resulting from the CPDSP module can be used to derive the start-up delay for a flight. By subtracting the estimated taxi time from the TTOT, the TSAT is derived. The difference between the TSAT and TOBT is the start-up delay of a flight. This start-up delay value should be kept to a minimum, especially for flights that are regulated by Eurocontrol. The reason for this is that Eurocontrol uses projections of future congestion in airspaces, to determine at what time aircraft should depart to prevent this congestion. This is communicated as a calculated take-off time (CTOT). Flights that are assigned a CTOT, must depart in a time window that starts 5 minutes before the CTOT and ends 10 minutes after the CTOT. If this becomes unfeasible due to a high start-up delay, a new CTOT must be requested that results in a high delay. Operators at AAS, therefore aim to avoid this situation.

2.1.5 Runway Reconfiguration Process

Runway reconfiguration is the process of changing the set of active runways. At AAS, the main actors in performing this task are the supervisor in command of Schiphol Approach (APP) and the supervisor in command of Schiphol Tower (TWR). APP is responsible for inbound traffic in the TMA/CTR and TWR is responsible for all traffic in the manoeuvring areas and the CTR. The two supervisors together establish at what time a new runway configuration should be used. A new runway configuration can be required for various reasons. Most often a runway reconfiguration is required due to a change in the demand situation or weather conditions.

Performing a runway reconfiguration safely and efficiently is a highly challenging task. This is confirmed by conversations with a former APP supervisor. It can be illustrated by the following quote: "It is work that can only be done by a person, every reconfiguration is unique and you cannot just base decisions on standard values". For this reason, a runway reconfiguration is carried out by highly experienced and qualified air traffic control operators. To ensure that these operators know the complexities in both the APP system and the airside ground operations, a TWR supervisor will also do shifts as a APP supervisor and vice versa.

During the day there are three operational briefings at 09:15, 14:00 and 20:00 local time. During these briefings the airport operations manager, APP supervisor, and TWR supervisor discuss the planning and required runway configurations for the first three upcoming demand peak periods. These peak periods are defined in a peak schedule. This peak schedule is presented in table 2.3, where the following demand peak configurations can be distinguished:

- **Off-peak**: In an off-peak, a 1+1 configuration is used with one runway for inbound traffic and one runway for outbound traffic.
- **Outbound-peak**: In an outbound-peak, a 1+2 configuration is used with two runways for outbound traffic and one runway for inbound traffic.
- **Inbound-peak**: In an inbound-peak, a 2+1 configuration is used with two runways for inbound traffic and one runway for outbound traffic.
- **Night-peak**: In the night-peak a 1+1 configuration is used in combination with RNAV procedures, resulting in a lower capacity than a 1+1 configuration in an off-peak.

Table 2.3: Demand peak schedule for AAS

Off-peak	**Outbound-peak**	**Inbound-peak**	**Night-peak**
0430-0500 UTC	0500-0520 UTC	0520-0720 UTC	2030-0430 UTC
	0720-0900 UTC	0900-0940 UTC	
	0940-1100 UTC	1100-1150 UTC	
	1150-1300 UTC	1300-1410 UTC	
	1410-1540 UTC	1540-1830 UTC	
	1830-2010 UTC	2010-2030 UTC	

The peak schedule is based on the expected demand pattern and serves as a guideline for selecting the times for using specific demand peak configurations. In the operational briefings, it is discussed if the times for different peak configurations should be adjusted, what runways should be selected, and what the capacity for these configurations will be. The results of this discussion are translated into a document called "capacity forecast". This document is communicated to other stakeholders, such that these stakeholders can adapt their planning.

Decisions made during operational briefings are based on a prediction for the future traffic situation, which still involves a lot of uncertainties. Any changes in the traffic demand situation can be monitored on a traffic demand overview tool. Using this tool it can be decided when the capacity of the current runway configuration will not fit the demand pattern anymore. An example of a traffic demand overview is provided in figure 2.9, where the upper part of the histogram shows the outbound demand and the lower part shows the inbound demand. By using the dashed lines for the capacity of one and two runways, it can be easily observed at what moment in time the capacity of one runway will not be sufficient anymore. This serves as a trigger for initiating an assessment if a runway reconfiguration is required.

In the periods between the operational briefings, the traffic is continuously monitored using the traffic demand overview and delay predictions resulting from the inbound and outbound planning tools. If possible, at least 30 minutes before a foreseen runway reconfiguration, a negotiation is started between the APP supervisor and the TWR supervisor to decide on the precise moment in time for the runway reconfiguration. This negotiation is initiated by the TWR supervisor if an additional runway for outbound traffic is required and by the APP supervisor

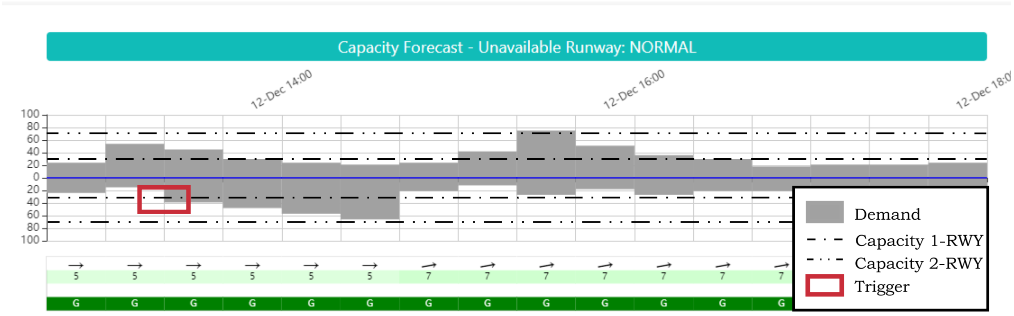


Figure 2.9: Example traffic demand overview [70]

if an additional runway for inbound traffic is required. In this negotiation, both supervisors assess what the best moment in time is to perform the reconfiguration.

This assessment consists of anticipating the future traffic situations in both systems and an evaluation of how a certain moment in time for the reconfiguration would impact the safety and efficiency of the system. After selecting an appropriate moment, this is proposed to the other supervisor. This supervisor will predict how the proposed moment in time will affect the workload, safety and delays in the control areas it is responsible for. Given this information, a trade-off has to be made if this moment in time can be agreed upon or if another time should be proposed. This negotiation normally quickly results in a decision that is mutually acceptable for both supervisors.

There could also be sudden events that require an immediate change in runway reconfiguration, such as the sudden unavailability of a runway. In this case, an ad-hoc decision has to be taken. This is however highly undesirable because this eliminates the option for other air traffic controllers and pilots to adapt to the reconfiguration and increases safety risks. It is therefore in general desired to decide on the moment timely, such that air traffic controllers can be informed and anticipate to the upcoming runway reconfiguration. During the reconfiguration, the APP supervisor and TWR supervisor stay in contact to refine the exact moment in case of unexpected developments in the traffic or weather situation.

To adhere to the fourth runway rule introduced in section 2.1.2, it is desired to avoid using four runways simultaneously when possible. To do so, a second runway for inbound traffic can only be used when one runway is used for outbound traffic and vice versa. It is however not always possible or efficient to deactivate a runway for the opposite traffic direction, before activating a new runway for the desired traffic direction (inbound or outbound). For situations in which the demand is too high and delays would become unacceptable by deactivating a runway for the opposite traffic direction, the runway that should be deactivated can remain active for a short period. This is called a double peak or a 2+2 configuration.

In a double peak, the fourth runway can be used at full capacity or only for a few flights. When it is only used for a few flights, this is called a 2+1+1 configuration. This may be opted for during reconfigurations where some aircraft should not be rerouted because of their position. Another option during reconfiguration is to initially activate an alternative runway, that is not part of the desired configuration. This is beneficial when the same runway that is deactivated for inbound traffic is also activated for outbound traffic. If no alternative runway is initially used here, outbound traffic will have to wait for the last inbound traffic to clear the runway and taxiways. This will result in no departures from that runway for a certain period. An example scenario for such a runway reconfiguration is a switch from southerly to northerly use (see figure 2.10), often required due to wind conditions.

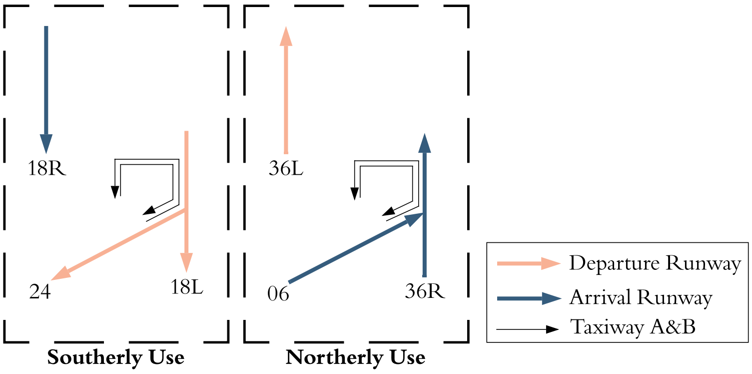


Figure 2.10: Southerly and northerly runway use

Selecting the moment of a runway reconfiguration is not done by selecting a time, but by selecting the first flight that uses the newly activated runway or the last flight that uses the to be deactivated runway. Apart from selecting this flight in such a way that the number of flights on the fourth runway is limited, it is aimed to avoid high delay values for either the inbound or outbound system. As discussed in section 2.1.3, a highly undesirable situation arises when inbound delay values exceed the threshold of four minutes, leading to holding instructions. It is therefore aimed to select the moment of a runway reconfiguration in such a way that inbound delay stays below this threshold.

For the outbound system, an undesirable situation can occur when delay values lead to missing departure slots or overflow causes taxi queues. As discussed in section 2.1.4, departure slots will be missed when a flight departs later than 10 minutes after its CTOT. It is therefore aimed to avoid start-up delays larger than 10 minutes. As can be seen in figure 2.10, the main taxiway system consisting of taxiway Alpha and Bravo is located close to the departure runway ends in a northerly configuration. When queues form at the departure runway ends in a northerly configuration, this will block the main taxiway system. It is therefore especially in a northerly configuration aimed to avoid runway queues caused by an overflow.

Apart from considerations for the manoeuvring area, the situation in the CTR and TMA should be considered as well during a runway reconfiguration. Two examples of a problem that can arise in the CTR/TMA are provided in figure 2.11 and figure 2.12. In figure 2.11, a situation is depicted where a first flight is selected to use the secondary runway 36R. This flight is arriving from ARTIP in front of another flight from ARTIP that is close behind it. Due to the new flight path for the first flight, the flight paths will now cross. This means that selecting this flight will cause a high workload for air traffic control and a delay for the following flight to avoid interference of the flight paths.

In figure 2.12 an example is provided for issues in the CTR/TMA after selecting a first flight to use the new departure runway. By selecting the first flight to use the second departure runway 36C that will depart shortly after a flight that takes-off from 36L and flies to the same departure sector, a conflict in the air will occur. This flight can therefore only selected if it is delayed by a sufficient amount.

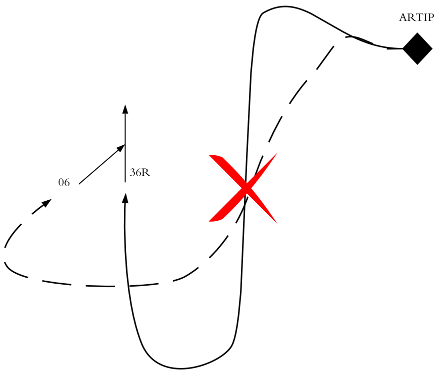


Figure 2.11: Inbound trajectory interference caused by runway reconfiguration

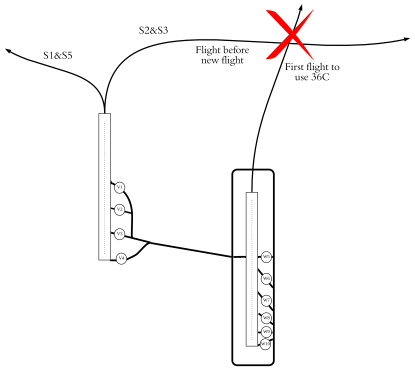


Figure 2.12: Example of departure flight trajectory interaction

2.2 High-Level Research Requirements

The phenomenon of a runway reconfiguration takes place in an environment that is comprised of multiple interacting socio-technical subsystems. By investigating the operational context it is established which aspects are influencing a runway reconfiguration. To gain a better understanding of the phenomenon it is required to address these aspects. These aspects are from here on referred to as high-level research requirements.

From the operational context, it has become clear that a runway reconfiguration features control actions by either of the supervisor operators that affect all air traffic on the airport and in the airspace around the airport. In the process of applying these control inputs the following aspects need to be addressed:

- **Anticipation**: A runway reconfiguration should be selected in advance such that stakeholders can adapt.
- **Satisficing solution**: The reconfiguration should be acceptable for the APP system as well as for the TWR system, this requires decision-making in situations with opposing goals and inconsistent information.
- **Noise abatement**: It should be aimed to avoid using secondary runways as much as possible.
- **Inbound planning aspects**: Issues for inbound planning, such as holding patterns should be considered.

- **Outbound planning aspects**: Issues for outbound planning, such as taxiway queues and missing slots should be considered.
- **Airspace conflicts**: Possible conflicts in airspace around the airport resulting from runway reconfigurations should be taken into account.

3

Literature Overview

Establishing the operational context has provided insights into relevant aspects of the runway reconfiguration phenomenon. To investigate which of these aspects are well understood and which are less well understood, the existing literature related to runway reconfiguration management is reviewed. This literature review allows selecting a research focus, by defining a research gap. In order to establish what methods and techniques can be used to resolve the research gap, an additional literature review is done to identify suitable alternatives.

3.1 State of the Art

To gain insights into the state of the art on the subject of runway reconfiguration management, a review is performed on existing literature related to this topic. This review identifies a variety of approaches to researching this topic. Each of these approaches is evaluated for its strengths and weaknesses. The evaluation focuses on which modelling techniques were used and to what extent these modelling techniques were capable of providing insights into various aspects of the runway reconfiguration phenomenon.

3.1.1 Existing Runway Configuration Management Models

Runway configuration management is an essential part of capacity and demand management. Jacquillat and Odoni [39], provide a comprehensive review of the research advancements and developed models related to capacity and demand management. It identifies a division of runway capacity and demand models into theoretical and empirical models.

Empirical models are based on historical data related to the operations at an airport. Most of the used methods are in the form of a machine learning model, a neural network, or a behavioural choice model. Empirical models can be used to predict runway configurations and are often used for tactical or strategic decisions. A variety of approaches towards modelling runway configuration management using machine learning methods can be identified [71, 60, 5, 46, 9]. These models make use of supervised learning methods in the form of support vector machines (SVM), ensemble learning, and regression models. The predictors that are most often used in models for predicting runway configurations are meteorological conditions, demand conditions, and inertia factors.

An example of using a SVM method for researching runway configuration management is the model of Wang [71]. In this model, meteorological conditions and arrival rates are translated to runway configurations. The study shows an improved accuracy for predictions as compared to regular SVM methods by employing an ensemble learning method to account for missing features and noisy data. Roy et al. [60] omit the step of predicting the runway configuration and focus on predicting the runway capacity, which is often the desired parameter, directly.

In the work of Leege de and Janssen [46], a probabilistic forecast is generated based on meteorological model output statistics and demand conditions. The uncertainty in the model output statistics is accounted for by employing a Monte Carlo simulation. Avery and Balakrishnan [9] take an alternative data-driven approach by using behavioural choice modelling. The prediction in this model is based on a utility function, that has weights according to the maximum likelihood estimate of the predictors. Altinok et al. [5] focuses on determining the key predictors for runway configurations. It is found that wind condition is the sole significant predictor, regardless of the underlying prediction model. It is furthermore identified in this work that capacity prediction introduces the problem of determining the moment of configuration switches. Due to the dynamic nature of this process, this aspect is found to be hard to model accurately using data-driven models. As a proxy solution, the moment in time for a runway reconfiguration is in these empirical models often modelled by using inertia or a switching complexity factor. The reduction in capacity resulting from the reconfiguration is often not modelled or approximated.

Theoretical models related to runway configuration management are most often formulated as an optimisation model that minimises delay. The majority of these models is based on queuing theory combined with a form of linear programming or dynamic programming. The results of theoretical models are often used as support for strategical decision making. Some significant advancements have been achieved using theoretical models [33, 47, 68, 40, 44, 62, 41]. In these models, simulations are performed to predict the delay build up as a result of the demand pattern and runway capacities. The runway configuration and moment for reconfiguration are selected in such a way that the delay is minimised over a certain time horizon.

In the work of Heblij and Wijnen [33], the runway configuration is not only determined by wind conditions and a queuing model, but the noise emissions and third party risks are included in the model as well. Li and Clarke [47] propose a model based on queuing simulations and stochastic dynamic programming. In this model, the ideal time of performing a runway reconfiguration is determined by balancing the queue size and an approximated configuration switch cost. Thorne and Kincaid [68] take an alternative approach by performing a Monte Carlo simulation of an airport model with variations in demand pattern, weather conditions, and runway configurations. To heuristically find the locally optimal runway configuration schedule, a Tabu search is performed.

Jacquillat et al. [40] recognise that delay build-up does not solely depend on capacity and demand. It is recognised that delay is also influenced by sequencing, human factors, and aircraft mix. To account for this, stochastic queuing conditions are introduced in the model. An alternative approach is taken by Janic [41]. Instead of using linear programming, greedy criteria are used in this work to select the capacity allocation based on minimising the outflow of delay for small time intervals. The models of Kamgarpour et al. [44] and Sama et al. [62] make use of graph models instead of queuing models. Kamgarpour et al. [44] model the runway configuration management as an optimal control problem using a directed graph. Control inputs for this model are the runway configuration and aircraft speeds. The runway configuration schedule is determined based on weather conditions, demand pattern, and switching times. Aircraft speeds are thereafter determined by solving a mixed-integer linear programme. Sama et al. [62] introduces an alternative graph model. The optimal runway configuration schedule is optimised by linear programming and heuristic methods for different policies that can be introduced in the alternative graph model.

The literature on runway configuration management shows a variety of advanced models that are valuable for tactical and strategical decision making of operators. The empirical models show good results for the prediction of which runway configuration will be used. Although the timing of a runway reconfiguration is an important influencing factor for the runway capacity, little attention is paid to this aspect in these models. This aspect is often modelled by using an inertia factor and an approximation of the effects. It is furthermore often difficult to explain how a result was established by the model because the model is purely based on historical data and not the rationale behind the decisions made. This causes problems in the light of new operational procedures that were not seen before in the historical data, e.g. new runway selection rules.

Theoretical models found in the literature are focused on advising optimal runway configuration schedules. An optimal solution does however often not reflect how decisions are taken in reality and do not include aspects like interactions, opposing goals and resilience aspects. Another issue found in the existing theoretical models is that the runway configuration switch is often modelled by using a period of reduced capacity or a switching cost. The values for this parameter are often not well known and approximated. It is therefore hypothesised that focusing on this particular aspect and studying the underlying decision models, influencing factors, and emergent phenomena involved in the runway reconfiguration process can provide a significant contribution towards gaining a better understanding of the process and help in improving existing models.

3.1.2 Research Gap

The literature shows a variety of empirical models related to runway configuration and capacity management. These models provide highly useful insights for tactical and strategic decisions of operators. Empirical models are based on historical data and make use of a variety of machine learning methods. The empirical models are mostly focused on prediction and in particular the prediction of runway combinations. Even though the moment of the runway reconfiguration is an important factor in predicting runway combinations and capacity, relatively little attention is paid to this aspect. The dynamic nature of this aspect is often hard to capture in a data-driven model. Therefore, most empirical models approximate this by using an inertia factor or based on historical data.

The empirical approach works well in general but does not capture the dynamics of the system well for every scenario. When a model is based on historical data, this will cause problems in the face of new operational procedures that were not seen before in the empirical data set. It is also often hard to explain where results are coming. This makes these models unsuitable to gain a detailed understanding of the runway reconfiguration process.

The theoretical models in the existing literature are most often in the form of an optimisation model using queuing theory or graph theory. These models are highly useful to aid in decision making, to select a theoretical optimum schedule of runway configurations. Queuing and graph models used in the existing literature, do however take a high-level approach and do not model the interactions in for example the arrival process or the taxiway system accurately. Furthermore, a theoretically optimal solution is rarely achieved in reality. Often a satisficing solution needs to be established that considers opposing goals. Existing theoretical models are not aimed towards satisficing solutions and therefore not suitable to investigate the underlying decision-making models that reach these satisficing solutions. To gain a better understanding of the workings of the runway reconfiguration process, a different approach is deemed required.

The aspect of timing a runway reconfiguration is often modelled by using a switching cost or a period of reduced capacity. The value for this switching cost and the time and capacity for the period of reduced capacity are not well known and an estimated value is often used. A complex operation, like a runway reconfiguration, with a high amount of interactions and uncertainties requires a detailed understanding to model this aspect of timing a runway reconfiguration correctly. What is missing in the literature, is a model that can provide insights into the dynamics and underlying decision-making models for a runway reconfiguration process. Constructing such a model using a bottom-up description is expected to aid in creating a better understanding of the runway reconfiguration process, which in turn can aid in addressing the aspect of timing a runway reconfiguration in existing models.

3.2 Modelling Technique Alternatives

The established research gap has identified the need for alternative modelling techniques capable of representing aspects required to model the timing of a runway reconfiguration. To select these techniques, possible candidates are explored and evaluated for how well they fit the posed high-level research requirements. Using this evaluation a substantiated decision is made on which techniques to use for various aspects in the model at different levels of aggregation.

3.2.1 Modelling and Simulation Techniques

Studying a real-world system can be done by employing a variety of modelling and simulation techniques. Each of these techniques has its advantages, disadvantages, and most appropriate use cases. In order to progress to a technique that is suitable for modelling the runway reconfiguration, it is important to select the main modelling technique that best fits for resolving the defined research gap.

Discrete-Event Modelling

Discrete-event models are appropriate for systems in which the state change only occurs at discrete points in time [59]. The system usually consists of three types of variables. A time variable is used to keep track of the amount of time that has elapsed in the simulation. To register the number of times an event has occurred, a count variable is used. To keep track of the state of the system a system state variable is used. These variables are updated when the next event occurs. There is an event list that stores information on which event will be the first upcoming event. When the event occurs, a snapshot of the system is taken and the relevant output data is collected to keep track of the developments of output parameters. A typical example of a discrete event simulation would be a queuing system where entities arrive with a certain predetermined distribution and leave the system according to a constant value or distribution.

Monte-Carlo Simulation

There exists no single Monte-Carlo method. This is a collective name to describe random sampling and statistical modelling to explore mathematical functions or investigate the operations of complex systems. Even though there exists a large variety of Monte Carlo simulations, in general, the simulations will follow the following pattern[32]:

- Model a system in terms of a series of probability density functions.
- Sample from the probability density functions repeatedly.
- Compute the statistics that are of interest.

The method is highly useful when it is intractable to solve a problem analytically. By simulating the problem for a sufficient number of times it is for example possible to estimate the true mean. This is an effect of the law of large numbers that dictates that the mean of the samples if taken enough samples, approaches the expected value. Furthermore, depending on the allowed standard deviation, it can be determined how many simulations are required. This is a result of the central limit theorem, which dictates that the normalised sum of independent random variables tends towards a Gaussian distribution, with a rate of convergence of $\frac{1}{\sqrt{n}}$, where n is the number of simulations.

Agent-Based Modelling

In an agent-based model, the system is constructed using separate entities in the form of software agents. A software agent is autonomous and is capable of operating as an independent process and performing actions without the intervention of the user [66]. Rather than a global approach, the model is constructed in a bottom-up fashion. By implementing local rules and behaviour for agents, complex systems can be modelled naturally.

Agents can exhibit heterogeneous behaviour and can be modelled with quantitative as well as qualitative relations. The decision-making and cognitive models of software agents can be modelled in great detail using cognitive aspects. The resulting model can be used to model important emerging phenomena resulting from the behaviours and interactions between agents. It is for that reason, that the technique is recognised for being able to model complex socio-technical systems that exhibit emergent phenomena[35].

In general, the benefits of agent-based modelling over other techniques can be described by [14]:

- The ability to capture emergent phenomena of a system.
- The possibility to describe a system naturally.
- The flexibility and adaptive capacities it provides.

Evaluation of Modelling and Simulation techniques

The modelling technique candidates found in literature can be divided up by their specific use. A discrete-event modelling can describe systems with discrete events that have certain distributions for the times of entering the system and being in a certain state in a very natural way. The method does however not provide the ability to model the interaction of heterogeneous behaviour. It is furthermore limited to model only specifically the before described case and does not allow for modelling cognitive structures.

Monte Carlo simulation is a highly flexible way of performing simulations. It can often provide good insights into the properties of problems that cannot be analytically solved. It is used in combination with a variety of models such as Markov chain models, Agent-based models and statistical models. For this simulation technique to be useful it is required, that a model exhibits stochastic behaviour.

Agent-based modelling is recognised for its ability to describe systems in a natural and bottom-up approach. It provides flexible models and the ability to model behavioural aspects. The paradigm is highly suited to study complex socio-technical systems and explore emergent behaviour. A disadvantage of agent-based modelling is that models grow complex very quickly and that it can be hard to completely understand a model.

Modelling the timing aspect of a runway reconfiguration process requires a model that is capable of handling interactions, cognitive structures and adaptive behaviour. Agent-based modelling is renowned for these capabilities and is deemed a more suitable technique than discrete-event modelling for this specific use case. It is therefore decided to employ an agent-based approach for modelling the runway reconfiguration phenomenon.

3.2.2 Cognitive Concepts

The selected agent-based modelling paradigm allows for modelling system behaviour using artificial intelligence mechanisms. The field of artificial intelligence (AI) is described by Luger [49] as follows:

> *"AI is the study of the mechanisms underlying intelligent behaviour through the construction and evaluation of artefacts designed to enact those mechanisms."*

When constructing intelligent artefacts, nature and especially the human mind often serve as an important source of inspiration. A field that explores the nature of the human mind and intelligence by modelling it, is the field of cognitive modelling[17]. Apart from exploring the human mind, cognitive modelling also offers mechanisms that can be exploited to introduce intelligent behaviour in agent architectures by emulating mechanisms employed by humans.

An important aspect for the runway reconfiguration phenomenon, identified in section 2.2, is the aspect of anticipation. Incorporating anticipatory behaviour into a model requires internal mechanisms which facilitate this behaviour. In the light of exploiting the human mind as an inspiration for designing intelligent artefacts, a closer look should be taken at how humans perform such a complex task.

An anticipatory system can be described as a system that has a predictive model of the environment or itself, that allows it to change state at a point in time by the model's predictions for a later point in time [51]. When regarding how a human fulfils the role of an anticipatory system, there exists a large body of evidence that supports the so-called "simulation hypothesis" originating from the work of Hume [37] and Bain [10]. This hypothesis relies on assumptions about how the human brain functions when performing anticipatory behaviour. These assumptions can be condensed into three core assumptions[34]:

- A human can activate motor structures of the brain in a way that resembles activity during an action without actually acting.
- Imagining perceiving something is essentially the same, regardless of whether this perception is generated by the brain or external stimuli.
- A simulated action can elicit perceptual activity resembling the activity that would have occurred if the action had been performed.

Expressing mechanisms of the human mind may be done from a realist or functionalist perspective[45]. When expressing models from a realist perspective, states that exist such as neuron states are considered. Whereas from a functionalist perspective, states do not necessarily exist in reality. The states may only serve to describe a process properly, thereby making models more useful for modelling intelligent artefacts. For the cognitive architecture design providing anticipatory behaviour, it is therefore desired to utilise a formal model that describes the anticipation mechanism employed by the human mind from a functionalist perspective. In the work of Blok et al. [12], such a formal model for anticipatory behaviour in the context of air transport operations is developed. The approach taken in this work differentiates itself from simplified models of anticipation, by following the principles of amongst others the work of Hesslow [34] and other inspirations from behavioural science such as the described simulation hypothesis.

In the work of Blok et al. [12], a layered architecture is introduced for the cognitive design of an anticipatory agent, see figure 3.1. This architecture is based on the work of Blumberg and Galyean [13] and Hoogendoorn and Bovy [36]. It consists of an operational and strategical layer. The operational layer is formed by the perception and action module, whereas the strategical layer is formed by the belief and reasoning module. The perception module facilitates observations and interpretation of these observations and the action module is responsible for preparing and executing actions. Current knowledge in the form of rules or facts is contained in the belief module. This information is used by the reasoning module to generate a prediction, analyse these predictions and make decisions for actions that should be performed.

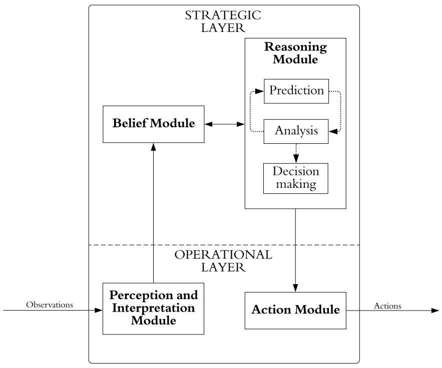


Figure 3.1: Cognitive architecture for anticipation[12]

Using the defined cognitive architecture, cognitive anticipation is performed by an agent according to a fixed set of steps. This fixed set of steps is described as[12]:

1. Observation of the environment to detect cues that trigger a prediction procedure to determine the expected undesirable state
2. Generation of action options, that could potentially avoid reaching the predicted undesirable state
3. Generation of the consequences for the generated action options
4. Valuation of the simulated results for applying the action options
5. Selection of the to be performed the action

A more formal description of this anticipation process can be found in [12].

Another important aspect identified in the high-level research requirements (see section 2.2), is decision-making in situations with opposing goals, incomplete information and inconsistent information. In light of exploiting the human mind to develop intelligent artefacts, a closer look should be taken at how humans handle such complex situations. In these situations, humans often resort to the use of argumentation to make sense of the situation[8]. An argument is regarded as an information element that may allow an agent to justify its stance or to influence another agent's stance[42]. Argumentation can either be performed externally or internally.

External argumentation is done by exchanging arguments in a cooperative or competitive setting to reach a mutually acceptable agreement. In a situation, with inconsistent and incomplete information internal argumentation can provide valuable insights to make sense of the situation. This is done by constructing arguments in favour or opposed to making certain decisions and selecting the argument that prevails. Translating such argumentation capabilities requires an internal agent architecture that is capable of external argumentation interactions and reasoning using internal argumentation.

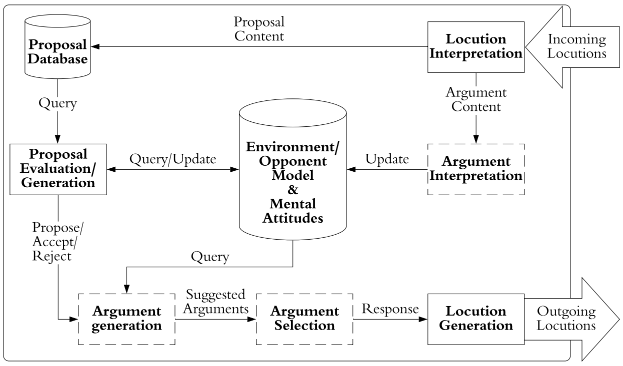


Figure 3.2: Cognitive architecture for ABN agent [58]

In the work of Rahwan et al. [58] an overview is provided of the internal agent mechanism required to perform external argumentation. An overview of the elements required for an argumentation-based negotiation (ABN) agent is depicted in figure 3.2. The internal architecture must consist of modules that are capable of interpreting and generating a locution. A locution is an element of the communication language, that may also be referred to as a speech act or utterance. These are in a negotiation scenario often including terms such as propose, reject, and accept.

In an ABN, incoming locutions do not only consist of proposal (i.e. potential deal or agreement) content but may also possess argument content. A module is therefore required that is capable of interpreting these contents. At the centre of the architecture is the knowledge base of the agent that is updated with incoming proposal and argument content as well as environment observations. This knowledge base is utilised to construct arguments and evaluate/generate proposals by reasoning about the generated arguments. Once a proposal has been established, it is required to select which arguments should accompany this proposal and a locution must be generated containing this proposal and the set of selected arguments.

Reasoning activities required in this architecture, require a computational model of argument. A variety of models have been developed to reflect how humans build, analyse and use arguments in their daily life to make decisions in environments with incomplete and inconsistent information. Whereas these models may vary in structure and implementation a discriminating set of layers can be distilled that can be regarded as the basic building blocks[8]. These layers can be described as:

- Structural layer: This layer defines the actual structure of the arguments used.
- Relational layer: A layer that specifies sub-argument/super-argument relations, attack/support relations and preference relations.
- Dialogical layer: A layer that specifies how arguments are exchanged, often in the form of a dialogue game protocol.
- Assessment layer: This layer defines how it is determined which arguments prevail in a reasoning scenario.
- Rhetorical layer: A layer that defines how strategical considerations are implemented.

Not all of these layers are required for every model of argument, but they serve as a basis for selecting the formal model of argument that will be employed for internal argumentation.

3.2.3 Negotiation Technique

Given that the cognitive concept of argumentation is selected, a closer look is taken at the negotiation techniques that could aid in constructing such an external argumentation model. Negotiation is a mechanism that aims to solve these conflicts and can be seen as a process by which a joint decision is made by two or more actors. The actors first verbalise contradictory demands and then progress towards an agreement by a process of making concessions or searching for new alternatives [57].

The runway reconfiguration phenomenon can be regarded as a scenario in which two agents compete for scarce resources and conflicts about the allocation of these resources are inevitable. In order to mitigate these conflicts some sort of mechanism to come to a consensus on the allocation of resources is required. The term resources here should be taken in the widest sense of the word and can be in the form of time, space, agent resources, et cetera.

An agreement can be seen as a vector in a solution space. Every dimension in the multidimensional space represents an attribute of the agreement vector and can take on a domain of possible values. In this regard, finding an acceptable agreement for all parties involved can be seen as a search through a space of potential agreements[43]. The size of this space of potential agreements can grow rapidly in complex negotiation problems with a large number of attributes and large domain spaces. This is often the case when time is involved. When the agreement space grows to a large size, it is often unfeasible to consider every possible agreement.

Negotiation techniques can be divided up into game-theoretic, heuristic and argumentation-based techniques. The game-theoretic negotiation technique describes the negotiation interaction as a game with agents as players that make moves. Despite the game analogy the theory can be applied to study and engineer a wide variety of negotiation interactions, in particular, those between self-interested automated agents[73]. Game theory is in general concerned with two key problems[43]: The design of negotiation protocols to achieve desirable properties when agent behave strategically and the design of negotiation strategies that reach desirable equilibria under the given resource allocation mechanism.

Heuristic methods can be used for multi-attribute negotiations to get approximate, close to optimal solutions. These methods are often approximations of game-theoretic models that are less formally defined. Agents employ heuristics to make decisions during the negotiation process, that are designed to improve the convergence towards a mutually acceptable agreement. The core advantages of using heuristic techniques over game-theoretic models are[43]:

- Heuristic models for negotiations are based on realistic assumptions, such as bounded computational resources. This makes the heuristic models more widely applicable.
- The relaxation of assumptions in Game theory allows for using less constrained models of rationality for agent architectures.

Techniques ranging from linear programming, machine learning and genetic algorithms can be used for evolving adaptive strategies[58].

Heuristic models are often used in a bargaining scenario. In such a scenario, once a negotiation domain is decided on, the negotiation interaction between two agents will consist of an alternate sequence of offers and counter-offers. In heuristic models, the counter-proposal is generated using tactics that are influenced by for example time or resources. Some commonly used heuristic tactics are time-dependent, resource-dependent or behaviour dependent[29]. In the time and resource-dependent tactics, proposal valuation change as time or resources run out. In behaviour dependent tactics the agent uses the behaviour of opponents, by for example imitating the opponents to avoid being exploited.

Game-theoretic and heuristic approaches provide highly useful insights for a wide variety of negotiation problems. The frameworks do however have their drawbacks. The information that can be provided during the negotiation in locutions is limited to acceptance, withdrawals or counter proposals[43]. An argumentation-based approach can mitigate this limitation by allowing for additional information to be exchanged during interactions, in the form of arguments. These arguments can be used to justify the agent's negotiation stance or to persuade the opponent to change its stance [42]. Furthermore, in contrast to Game-theoretic and heuristic approaches, argumentation-based negotiation allows for the ability of agents to benefit from acquiring and revising their preferences during negotiation[58].

In order to address the aspect of negotiation identified in the high-level research requirements, a negotiation technique has to be selected. Given the time element in selecting a time for the runway reconfiguration, the solution space grows infinitely large. This makes Game-theoretic methods unsuited because every resource allocation is assessed using this method. For a heuristic method, a well-defined resource allocation mechanism is required. When selecting a future moment in time for a runway reconfiguration, it is not possible to have a well-defined resource allocation mechanism, because it is not known what the payoffs will be for a selected action, only a belief can be established. Argumentation-based methods allow for using beliefs and defeasible information. This technique also provides the ability for fast convergence in large solution spaces and is in line with the cognitive concept of argumentation. It is therefore deemed the most appropriate negotiation technique for the runway reconfiguration model.

3.2.4 Reasoning Technique

Reaching mutually acceptable agreements using argumentation-based argumentation, does apart from external elements such as a communication language and a protocol also require internal mechanisms to make decisions. To develop a robust intelligent system, the internal decision-making mechanism must be capable of handling inconsistent and incomplete information. Modelling the human mind using artificial intelligence techniques offers an important source of information for modelling such intelligent artefacts [17]. Computational models of the argument are an excellent example of this. These models emulate the way a human would solve a complex task of deciding in a situation with incomplete and inconsistent information[8].

The argument evaluation and argument generation/selection components of an argumentation-based negotiation agent are usually designed using an argumentation framework. One of the most fundamental frameworks on which most other argumentation frameworks are based is Dung's abstract argumentation framework[25]. This framework can be described by two core elements. These elements are a non-empty set of arguments A and a binary relation that is called "attack relation" R_{att}. In this framework, an argument is an abstract element. Let us say there are two arguments a_1 and a_2 such that $a_1, a_2 \in A$, then a_1 attacks a_2 iff $(a_1, a_2) \in R_{att}$. In Dung's framework, an attack always defeats the attacked arguments, which is an acceptable assumption when dealing with deductive arguments[28].

Argumentation models exist in a wide variety of forms. One discriminating factor for these models is the structure of arguments. For the structure of argument models, discrimination can be made between case-based, rule-based and information-based models. Case-based argumentation techniques are often utilised in domains that have a weak domain theory and allow for an easy way of sampling cases. In a rule-based model, arguments are constructed by simply applying the logic of the language in which the logic formulae in the knowledge base are defined. Such a knowledge base usually consists of facts and rules. Constructing a rule-based model requires a strong domain theory, but results in a model that is independent of empirical data. This makes these models more suitable for dynamic environments. In information-based argumentation techniques, models are built up by continuously comparing expectations with observations.

For argumentation models, both attack and support relations may exist between arguments. To establish which arguments prevail given these relations, an assessment is required. This assessment method is another discriminating factor for argumentation models. Assessment methods can either be a preference-based, value-based or assumption based. A preference-based argumentation framework[6] introduces preference orderings for acceptability criteria. Such a framework can be used if agents have consistent and fully defined preference functions. In Dung's abstract framework it is assumed that arguments are always deductive, while arguments often vary in their persuasiveness.

To account for variations in persuasiveness, the value-based framework[11] was introduced. Arguments are mapped to values in this framework and a binary preference relation decides the preference between values that were promoted from the arguments. In an assumption-based framework[15], the focus lies on how to find arguments and exploit beliefs. This framework allows an agent to formulate theories and validate them, thereby allowing for reasoning using beliefs. The described frameworks are often combined with one of the case-based, rule-based or knowledge-based reasoning techniques[19].

A more recent argumentation framework, that bridges the gap between fully abstract and specific frameworks is the ASPIC+ framework[52]. This framework is built on two main ideas. The first idea is that conflicts between the two arguments are often resolved by explicit preferences. The second is that arguments are built with two types of rules, strict and defeasible rules. These defeasible rules represent information that could be defeated after considering the entire corpus of knowledge. Arguments in ASPIC+ can be attacked in three ways. They can be attacked on:

- Uncertain premise.
- Defeasible inference.
- On the conclusions of defeasible inferences.

ASPIC+ is a framework for argumentation rather than a system, meaning that it does not provide any information on the structure of arguments. Instead, it is aimed at generating abstract argumentation frameworks. Such a framework consists of a directed graph of argument relations for which the extension can be determined using the calculus of opposition. ASPIC+ aims to bridge the gap between fully abstract argumentation frameworks and concrete logics. The fallibility of arguments need not only be located in its premises but can also be located in the inference steps from premises to conclusions. So in ASPIC+ arguments can be constructed using fallible and infallible arguments.

Apart from these abstract argumentation frameworks that do not define the structure of arguments, specific argumentation frameworks that do define the argument structure also exist. Dimopoulos and Kakas [23] propose a framework called Logic Programming without Negation as Failure, that is based on the logic programming language that is supplemented with explicit negation. In this framework, the logic programme consists of a set of rules and a set of partial orderings over these rules. The rules are defined as positive or explicit negative rules and the only inference is that of the modus ponens [23].

In LPwNF framework, an attack occurs when a composite argument, consisting of the rules in the object level argument and rules in the priority argument, derives a conclusion that is the explicit negation of the conclusion of another composite argument and the rules of this composite argument are at least as strong as the rules of the composite argument that is attacked [24]. In order for an argument to succeed, it has to bring along corresponding priority arguments that are at least as strong as those corresponding with counter arguments[24]. New priority rules can be required when the aforementioned priority rules are attacked by opposing priority arguments.

Another proposed specific framework is Defeasible logic programming (DeLP) [31]. This framework combines Logic programming and Defeasible argumentation. The logic programme can consist of strict rules as well as defeasible rules. It is capable of handling incomplete and contradictory information in dynamic domains. When the knowledge representation is queried, a dialectical analysis is executed to determine if an undefeated argument can be found that supports the query. Whether an argument defeats another argument is decided by the generalised specificity criterion. This criterion favours arguments with greater information content and arguments that require less use of rules [65]. If this is not sufficient, ordering constraints can be used.

In decision-making scenarios by the APP supervisor and TWR supervisor it was identified in chapter 2, that reasoning is often nonmonotonic. Nonmonotonic reasoning can be seen as a mechanism for taking back conclusions, that in the light of new information turn out to be wrong and for deriving alternative conclusions instead [7]. Addressing this aspect leaves the LPwNF, DeLP and ASPIC+ frameworks. These frameworks employ the nonmonotonicity which is also observed in human decision-making scenarios.

The APIC+ framework does not provide an implementation and specific argument structure, which is disadvantageous when automating reasoning processes. Both the LPwNF and the DeLP framework do provide a specific argument structure. The DeLP framework allows for a more dynamic implementation as compared to the LPwNF framework. This results from the fact that the outcome is not programmatically defined using preference relations. Decisions follow from assessing the entire corpus of knowledge in a heuristic way. The DeLP method allows for automating and is deemed more in line with the natural description of the cognitive model required for reasoning. This framework therefore selected as the technique for implementing reasoning aspects.

4

Research Objective and Methodology

Using the established high-level research requirements and the identified research gap, a research objective is defined. This research objective leads to various research questions which require answering to achieve the research objective. To describe the research steps required to answer these questions, a research methodology is defined.

4.1 Research Objective

A runway reconfiguration process is a dynamic and complex phenomenon that involves a variety of interactions. In this process, the timing of the runway reconfiguration plays a crucial role in maintaining an efficient and safe operation. Adequate timing requires an interplay between the ground and the arrival system in which both qualitative and quantitative aspects are considered. Such a complex socio-technical system exhibits emergent behaviour. It is therefore highly important to have a thorough understanding of the system when developing models related to runway reconfigurations. The research objective is for this reason stated as:

> *"To gain a better understanding of the principles and underlying decision-making processes, influencing factors, and emergent phenomena involved in the timing of demand-based runway reconfigurations by following an agent-based approach."*

The defined research objective leads to research questions that need to be answered by the research:

1. *What operations, actors and influencing factors can be identified that are related to the timing of the runway reconfiguration process?*

   In order to be able to study and gain an understanding of the runway reconfiguration process, it is required to become familiar with the relevant aspects of the process. This will involve a study of the current operations in which the actors, operational procedures and other influencing factors need to be identified. The relevant sub-questions are:

   - Which actors can be identified in the process and what is their responsibility/role?
   - What operation does every actor perform that is related to the timing of the runway reconfiguration?
   - Which aspects are taken into account for runway reconfigurations that are related to actions performed by operators?
   - What goals do the relevant actors have and are these goals conflicting?
   - What interaction exists between the agents involved in the timing of the runway reconfiguration process?
   - What external factors can be identified that influence the timing of a runway reconfiguration?

2. *What is the current state of the art of the literature related to the runway reconfiguration phenomenon?*

   Identifying the existing corpus of knowledge related to the subject can aid in identifying the relevant aspects of the subject. Additionally, it guides the research approach. The relevant sub-questions are:

   - What models do currently exist that are related to runway reconfigurations?
   - How is the aspect of timing runway reconfigurations handled in existing models?

  - What relevant factors and procedures were identified for runway reconfigurations?
  - What difficulties and challenges were identified in the literature on runway reconfigurations?

3. *What is a suitable method to model the processes involved in the timing of a runway reconfiguration?*

   Existing literature is used to identify which models exist, which modelling techniques have been used, and what alternative techniques might be suitable for modelling the timing of runway reconfigurations. The relevant sub-questions are:

   - What is a suitable modelling technique to model the runway reconfiguration phenomenon?
   - How can anticipatory behaviour be incorporated into the model?
   - How can negotiation and reasoning aspects be best addressed in the model?

4. *How well does the system reflect the actual operations and what differences can be observed?*

   In order to evaluate how well the phenomenon can be modelled, it must be compared with actual operations. The relevant sub-questions are:

   - How does the timing of the runway reconfiguration compare in the model and the actual operations?
   - What differences in decision-making can be observed between the model and the actual operations?
   - How can the differences between the model and the actual operations be explained?

5. *How can the gained understanding of the runway reconfiguration process be used to improve existing models?*

   Researching the aspect of timing is done by modelling on micro-scale, whereas existing systems that are used in practice are often based on more macro-scale models. To utilize the developed model for improving existing models, the lessons learned need to be translated into useful information for macro-scale models.

   - What identified influencing factors were not identified before in existing models and can help in improving the performance of existing models?
   - How can the used modelling technique or parts of the used modelling technique be integrated into existing models?

4.2 Scope

With the definition of the research questions that should be answered by the research, the scope of the research can be defined. This consists of the elements that are included in the research, the elements that are excluded, and the assumptions.

Elements Included in the Scope

- The demand-based type of runway reconfiguration will be modelled.
- A decision-making model that addresses negotiation and reasoning aspects for timing the runway reconfigurations.
- The development of a model that is able to emulate the inbound process of flights from entering the Dutch FIR until entering the apron, including a trajectory predictor and inbound planning module. This is referred to as the inbound model.
- The development of a model that is able to emulate the outbound process of a flight from leaving the apron until take-off, including an outbound planning module. This is referred to as the outbound model.
- A weather model that represents the wind conditions relevant for inbound flights.

Elements Excluded from the Scope and Assumptions

- Interactions before landing and after take-off between inbound and outbound trajectories, will be taken into account in the decision-making model but not be explicitly modelled.
- Weather induced runway reconfigurations are not taken into account.
- Only runway reconfigurations for northerly runway use (see figure 2.10) are considered.
- Taxiway and other ground interactions are not explicitly modelled.

4.3 Methodological Steps

Given the research scope and research questions, it can now be defined which methodological steps are required to carry out the research and answer the research questions. To do so, a research flow consisting of the required steps of the research is defined. This research flow consists of six steps, see figure 4.1.

The first step is aimed at obtaining as much knowledge as possible, that may prove valuable when modelling the phenomenon. This is done by investigating operational procedures, existing literature and the available data. This is followed by step 2 in which the conceptual model is developed and the inbound and outbound domain models are designed. With the established inbound and outbound model, a simulation scenario is developed in step 3 such that the inbound and outbound models can be implemented for this scenario. After these models are implemented, they are validated for the simulation scenario and a module for extracting sensory data from these models are established in step 4.

Using the developed domain models and sensory data from these models, the control model is constructed. For constructing this control model, the DeLP library of the tweety project is used[67]. Once this model is constructed in step 5, the model will be simulated with the simulation scenario established in step 6. The final step in the research consists of drawing conclusions from the research and generalising the gained knowledge such that it can be used to improve existing models.

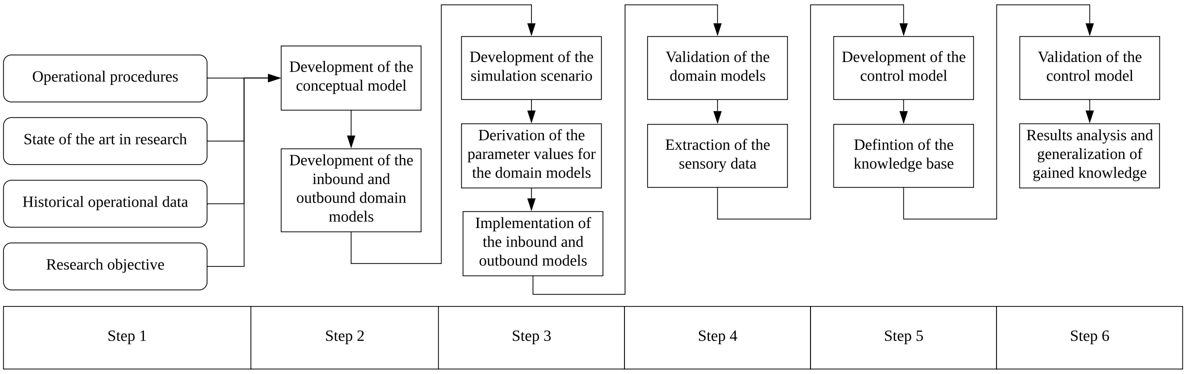


Figure 4.1: Schematic overview of the methodological steps

5

Conceptual Model

To design a model that aims to achieve the posed research objective within the defined scope, a conceptual model is developed. The first step towards establishing this conceptual model is taken by defining a high-level architecture, which provides an overview of the model elements and their interaction at an aggregate level. This high-level architecture comprises of a control model and domain model that interact in an environment. These elements are developed further by incorporating concepts that provide the desired functionality and behaviour for each element.

5.1 Overview

Central actors in the decision-making process leading up to a runway reconfiguration, are the APP supervisor and TWR supervisor. The adaptive and negotiation capabilities of these actors are important aspects to attain satisficing solutions for runway reconfiguration decisions. In accordance with the agent-based modelling paradigm which allows for a natural way of modelling systems, the two central actors are reflected in the model in the form of agents. The combination of these two agents controlling the runway configuration is referred to as the control model. The emphasis for the control model lies on the internal models of the agents and their interaction that forms the decision-making process.

The total agent-based model features an interaction between a control model and a domain model. The domain model represents the air traffic operations at AAS, that are relevant to the runway reconfiguration process (defined in section 4.2). The domain model is comprised of an interaction between flight agents, planner agents, and air traffic control agents. This interaction consists of air traffic control agents providing instructions to inbound and outbound flight agents, according to planning information communicated by planner agents. The domain model is divided up into an inbound and an outbound model which are focussed towards efficiently using the available inbound and outbound runway capacity, respectively. A graphical overview of the model elements is depicted in figure 5.1.

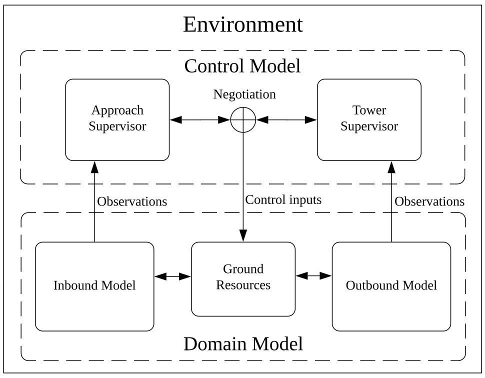


Figure 5.1: High-level architecture of the conceptual model

To control the demand peak configuration and thereby the runway capacity, the control model utilises the domain model as sensory input. This sensory input is obtained by monitoring planning information and by performing forward simulations of the domain model to predict the state evolution of either the outbound or the inbound traffic system. This information is required to facilitate the anticipatory and negotiation behaviour of the supervisor agent, which results in control inputs for the domain model in the form of runway activation and runway deactivation.

5.2 Environment

The domain model and the control model interact in a model environment. This model environment serves as an infrastructure that provides resources for agents and can be used for communication between agents. In this section, the components that form the environment are described. The maintained clock variable and global state in the environment are described, followed by a description of the airspace, aerodrome and meteorological components.

5.2.1 Clock Variable and Global State

To facilitate time updates for all agents in the model a global clock exists, which provides information about the current time. This global clock is continuously updated in a time interval between 4:00 UTC and 20:00 UTC with discrete time steps, which are referred to as ticks. At each tick, every agent in the model is updated with the new current time. During this update, state information is obtained by agents through perception.

In order for the agent population to update their knowledge about the world, using perception capabilities, an information source is required. This information source can either be in the form of direct interaction between agents or by using the environment as an intermediate information store. Because of the reduced complexity, an intermediate information store is used to facilitate most communication between agents. This information store is the form of a global state kept by the environment. This enables an agent to obtain specific knowledge, that is intended to be available to that agent, using perception abilities.

5.2.2 Airspace

In the model environment, the airspace is structurally similar to the airspace dedicated for AAS. This structure consists of multiple separated static three-dimensional volumes and points with a specific spatial location. These volumes are called the UTA airspace, CTA airspace and TMA airspace and are unidirectionally accessible by flight agents that are of the arrival type (see Fig. 5.2). The airspace can only be entered from the arrival source via the UTA and can only be exited by entering the aerodrome object. Entering the TMA is only possible via IAF point and leaving the TMA is only possible through the FAF point. Once a flight agent leaves through the FAF point, it can only enter the aerodrome area (see figure 5.3).

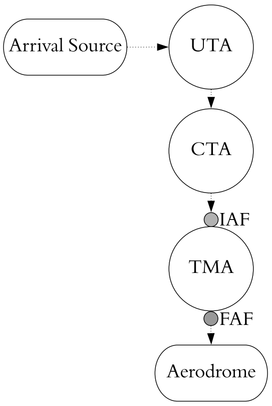


Figure 5.2: Schematic overview of the airspace elements

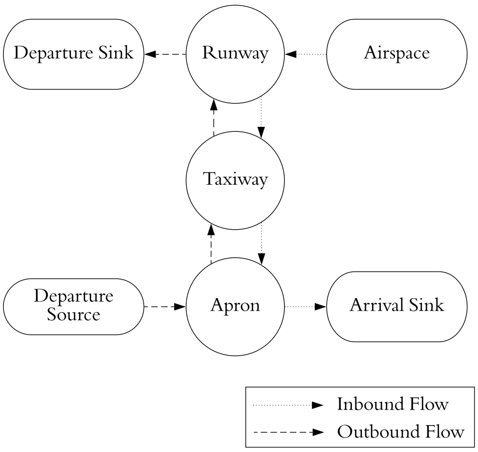


Figure 5.3: Schematic overview of Aerodrome elements

5.2.3 Aerodrome

Structurally the aerodrome consists of runway objects and apron objects that are connected by taxiway objects. This aerodrome component is aimed at resembling the aerodrome of AAS. The apron areas are defined as areas consisting of a group of parking positions. More information on the definition of these apron areas can be found in section 7.2. It is assumed that the apron areas can be used for pushback or entering them by one agent at a time. The taxiway objects are not limited to a defined capacity. Both the taxiway and apron area objects are static and can at any point in time be used by flight agents. A flight agent of the arrival type can use the taxiway element in the direction from the runway area to the apron area and a flight agent of the departure type can use the taxiway object to move from the apron area to the runway area.

Runway objects are the only dynamic objects in the aerodrome part of the environment. The runway object can either be active or inactive. If the runway is active and in arrival mode, it is only active for a flight agent of the arrival type. Likewise, the runway is only accessible for a flight agent of the departure type if the runway is in departure mode. In the model, four runway objects exist that are aimed at resembling runway directions 06, 36R, 36C, and 36L at AAS. Runway objects resembling 06 and 36R can only be used in arrival mode and runway objects resembling 36C and 36L can only be used in departure mode.

5.2.4 Meteorological Elements

Throughout the entire environment, a dynamic element exists in the form of wind and temperature conditions. These conditions are defined for a grid of positions and altitudes. For each of these grid points, the conditions change for discrete time steps. A more detailed description of the grid and meteorological parameters can be found in appendix E.

5.3 Domain Model

The domain model is constituted by the behaviour and actions of agents. In this section, these agents are described. Firstly the agent types are introduced. Thereafter the characteristics, perception capabilities, acting capabilities, and the action selection mechanisms for each agent type are described.

5.3.1 Agent Types

Throughout the entire model, we can distinguish between four types of agents. The domain model includes three of these four agent types. These are the flight agent, air traffic control agent and the planner agent. These agents are together responsible for emulating the inbound and outbound traffic systems at AAS.

Flight Agent

A flight agent emulates the combination of aircraft and pilot. This agent's behaviour is reactive, so inputs are directly translated to actions. A flight agent's perception module is responsible for the perception of environmental elements. For flight agents, these perceivable environment elements consist of the spatial objects and the clock variable. It can therefore at any point in time perceive in which airspace or aerodrome area it is located and what the current time is.

Using this time and position information, a flight agent is capable of updating its position and time properties. This is done by using the action module. Next to updating position and time properties, this action module also facilitates communication. This allows it to receive and update the current trajectory attribute of the agent, using a trajectory update action.

A flight agent has an action selection mechanism that is highly reactive, it is schematically presented in figure 5.4. Its action selection mechanism is activated at every tick update. The first step in the action mechanism is to check whether the agent has reached its current trajectory endpoint or not. If the endpoint of the current trajectory is not reached, the flight agent will update its time according to the current time it perceives using its perception module. This trajectory is defined for the arrival flight agent in section 5.3.2 and for the departure flight agent in section 5.3.3.

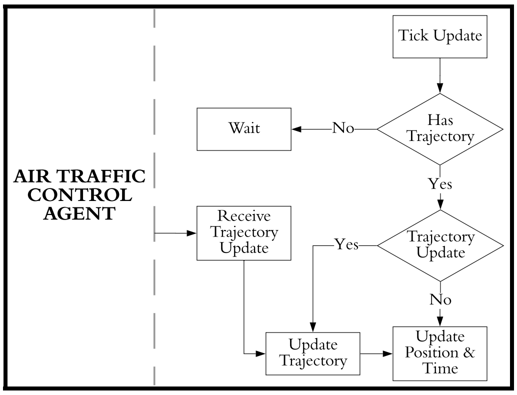


Figure 5.4: Action selection for the flight agent

Next to the time update, the flight agent also checks if the new position is located in a different spatial object of the environment than it was before the tick update. If this is the case, the agent will update its position property. In case of a later current time than the end time of its current trajectory, the agent does not have a current trajectory anymore. This means that the agent cannot update its time and position until it receives a new trajectory. A negative result from the trajectory check results in a wait action of the agent. Once the flight agent receives the new trajectory from the air traffic control agent, the flight agent will update its trajectory property. The new trajectory property then allows the flight agent to update its position and time again.

Air Traffic Control Agent

Air traffic control agents are responsible for providing instructions to the flight agents. The behavioural model of this agent is mostly reactive. The instructions that this agent provides are generated by a rule-based algorithm, that is based on observations of the environment and planning updates.

An air traffic control agent is capable of observing flight agents when they are positioned in an area that is dedicated for observation by the air traffic control agent. Given that the air traffic control agent's functioning is focused on providing instructions for flight agents, the action module of the air traffic control agent is capable of generating and sending these trajectory instructions. Apart from these two actions, the action module also facilitates a planning information request action. The resulting information of this request is the most up-to-date planning information. This information is combined with perceived information from the environment, to generate trajectory instructions.

The environment information that is available for the perception of the air traffic control agent consists of clock variable information, weather conditions and trajectory information. These weather conditions consist of wind and temperature conditions for every half hour of the day. The trajectory information consists of trajectory leg distances and taxi times. More detailed information about this weather and trajectory information can be found in appendix E.

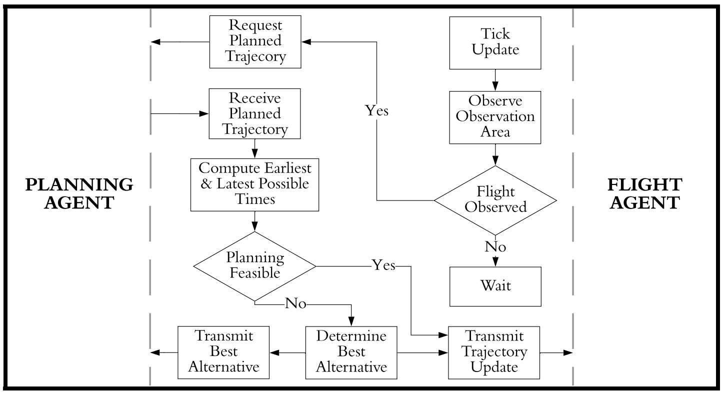


Figure 5.5: Action selection for the ATC agent

The action selection mechanism of an air traffic control agent is presented in figure 5.5. This mechanism is initiated when a flight agent is observed that has a position contained by the observation area of the air traffic control agent. Observation actions are performed at the start of every tick update. If no flight is observed, the air traffic control agent will not act and wait. Once it is established that a planning update is required for the flight agent, the air traffic control agent will request a planning update by communicating this to the planning agent.

Upon receiving the planning update, the air traffic control agent will compute the earliest and latest possible arrival times for the trajectory endpoint. These computations are done by using the flight attributes and environment information that is perceived by the agent, more information can be found in appendix C. After these computations, a check is done to determine whether the planned trajectory time is feasible for the flight. This is only the case when the planned arrival time is less than the latest arrival time and greater than the earliest arrival time. If the planned trajectory is feasible, this trajectory is transmitted to the flight agent. This can be accompanied by temporal displacement instructions to delay the flight.

If an air traffic control agent determines that the flight agent will not be able to reach the trajectory endpoint at the planned time, it will determine the best alternative. This best alternative will either be the earliest or latest arrival, whichever one is closer to the planned time. After the best alternative is determined, this information is both transmitted to the planning agent and transformed into a trajectory update. This trajectory update including temporal displacements is hereafter transmitted to the flight agent.

Planner Agent

A planner agent exhibits a more pro-active behaviour. This agent type is aimed at resembling the planning capabilities of inbound and outbound planning systems that exist in reality at AAS. Planning information is crucial for all other agent types in the system. A planning agent serves as the facilitator of this information.

The action module of this agent features an action that facilitates receiving planning update requests and an action that provides up-to-date planning information to other agents. In order for this agent to be able to generate up-to-date planning information, it requires the perception module to perceive information from multiple information sources. Using its perception module the planning agent perceives the current time, weather conditions, trajectory information, schedule information, and runway configuration information. The runway configuration information consists of the number of active arrival runways and departure runways at that moment in time and the schedule information consists of the schedule attributes defined in section 7.2.

Action selection for the planning agent is focused on providing up-to-date planning information for the air traffic control agents. A schematic representation of the action selection can be found in figure 5.6. Trigger for starting the action selection is a planning update request from an air traffic control agent. Once this request for a planning update is received, the planning agent's subsequent action is to check whether a runway assignment is required. A runway assignment is required if the flight does not have a runway assigned or if the runway that is assigned to the flight is inactive at the planned departure or arrival time. If this is the case, the runway assignment action is activated. More detailed information about how the runway assignment action is implemented can be found in section 5.3.2 and 5.3.3.

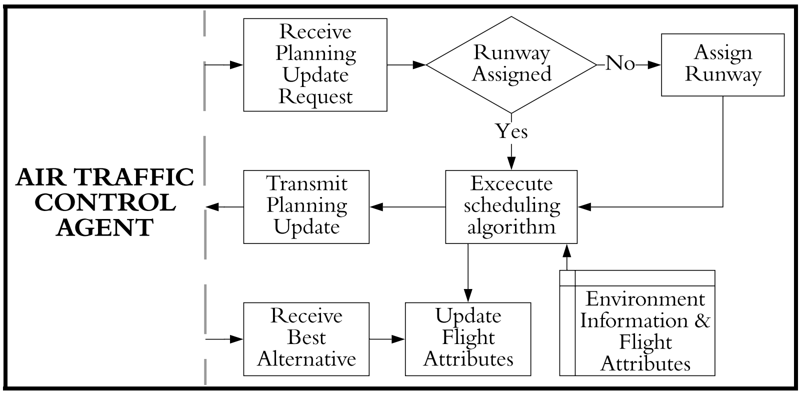


Figure 5.6: Action selection for the planning agent

Once the flight has the correct runway assigned, a scheduling algorithm is activated. This scheduling algorithm uses perceived information about the environment and flight attributes. The way this scheduling algorithm is implemented for both the inbound and outbound model is described in appendix D. The scheduling action results in a planned time for the trajectory update. This planning update is communicated to the air traffic control agent.

If an air traffic control agent determines that the transmitted planning update is not feasible, it will transmit the best alternative to the planning agent. Upon receiving this best alternative, a flow in the action selection mechanism is activated that adds the received information to the flight attribute information of the planning agent. This updated information will thereafter be available to the planning agent for new scheduling actions.

5.3.2 Inbound Domain Model

The inbound model emulates the functioning of the system responsible for controlling inbound flights at AAS. It consists of a collection of entities responsible for controlling an aircraft from upper terminal airspace to the aerodrome. This section describes how the flight, air traffic control and planning agents are implemented to handle arrival flights.

Inbound Model Overview

To maintain a natural way of describing the model, agents in the inbound model are implemented in such a way that they fit the real-world airspace structure of AAS. As identified in section 2.1.1, the airspace structure for arrivals is based around three merging points for entering the TMA, which are referred to as IAFs. The functioning of the inbound model is in line with this structure. Like in the real-world structure an ACC and TMA airspace exists where instructions are provided by air traffic control agents, which base their instructions on the planning information of an inbound planner agent.

In figure 5.7, an overview is provided of the implementation of the inbound model using the agent types described in section 5.3.1. Control of flight agents starts when a flight agent passes the FIR boundary. This control is divided based on the IAF used by the flight agent. For this reason, three air traffic control agents are implemented, that provide instructions to flight agents that are observed at the FIR boundary (in the observation area) and have an assigned IAF that corresponds with the responsibility of the air traffic control agent.

A second instruction in the form of a trajectory is provided, once the flight agent reaches the boundary of the TMA. This instruction is provided by one of the three air traffic control agents in the model that are responsible for instructions at the IAF. The information for generating these instructions originates from the planner agent.

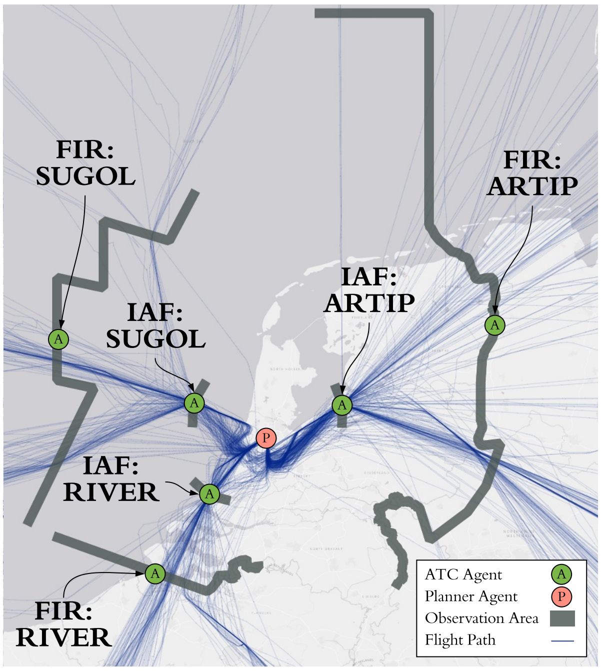


Figure 5.7: Conceptual inbound model overview

The inbound model aims to handle the arrival flight agents as efficiently as possible in terms of delay. This is achieved by optimising the available capacity for arrival runways. The capacity information is made available by the aerodrome interface for the planning agents and can be adapted by the supervisor agents. By using agents at different observation positions in the inbound model, information that is specific for those locations can be maintained. This information will serve as a crucial source of sensory information for the approach supervisor when making runway reconfiguration decisions.

Inbound Trajectory

An arrival flight agent follows a trajectory(see figure 5.8) that starts at the eligibility horizon and ends at the apron area. This eligibility horizon mimics the moment that flights would normally be detected by the MUAC radar. After a flight is detected, the first trajectory segment consists of a time for entering the eligibility horizon and a time and position for entering the FIR. Once the flight agent passes the eligibility horizon, the flight agent will update its dynamic position attribute to $atPosition(flight, atUTA)$. This position holds until the current time is greater than the entry time of the FIR.

When the current time is greater than the entry time of the FIR for the first time, the arrival flight agent enters the next trajectory segment. At this point, the arrival flight will be observed by an ATC agent that is responsible for that observation area. When observed, it will receive a trajectory update that consists of a position and time for reaching the IAF which is referred to as the ETA of the flight agent. After receiving these instructions, the arrival flight agent updates its position to $atPosition(flight, atACC)$.

A similar process occurs if the current time is greater than the trajectory time for reaching the IAF. At this point, a planned time is provided for reaching the runway. Once a flight leaves the IAF it updates its position property to $atPosition(flight, atTMA)$. The received information consists of the runway assigned to the flight agent and a planned time for reaching the runway.

The last segment of the trajectory is entered when the current time is larger than the planned time of reaching the runway, at this point the agent updates its position to $atPosition(flight, atAerodrome)$. The trajectory ends for an arrival flight agent once the current time is greater than the planned time to be at the apron area. At this point the position property is updated to $atPosition(flight, atApronArea)$. Once this position property holds, the arrival flight agent will remove itself from the simulation.

Inbound Air Traffic Control

Air traffic control agents in the inbound model are responsible for providing trajectory instructions. Instructions are provided when the arrival flight agent arrives at the FIR boundary or at the IAF and is observed by an ATC agent

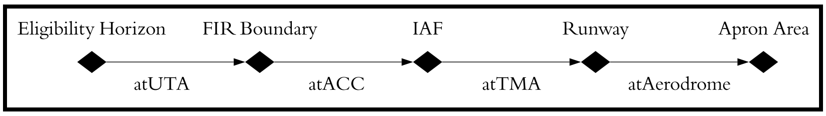


Figure 5.8: Overview of the trajectory legs for an arrival flight agent

responsible for that observation area. When observing a flight agent at the FIR boundary, the air traffic control agent determines the required new trajectory for the flight agent. This trajectory is defined by the designated IAF of the flight agent and the FIR entry angle towards the IAF. In case of an arrival flight arriving at the IAF, the air traffic control agent selects a new trajectory based on this IAF and the assigned runway of the arrival flight agent.

For the selected trajectory, the air traffic controller will determine the latest and earliest possible time of arrival at the end of the trajectory segment. This time is determined by a trajectory prediction computation using the method described in appendix C. The nominal paths used for this computation are depicted in figure 5.9. For an ACC segment, the nominal path is defined as the direct path from FIR entry angle to the IAF point. It is assumed that a continuous descent operation is performed on this leg.

For the TMA segment, the nominal path is chosen to be a path that corresponds with the median travelled distance for the IAF-runway combination. For both the ACC and TMA segments, the maximum distance is retrieved as the observed 95th percentile of distances in historic flight data. The minimum distance for the ACC leg is equal to the nominal distance because this is the direct path. For the TMA path, the minimum distance is taken as the 5th percentile of observed distances in historic flight data. The way that nominal paths for every FIR boundary-IAF and IAF-runway segment are derived, is explained in appendix E.

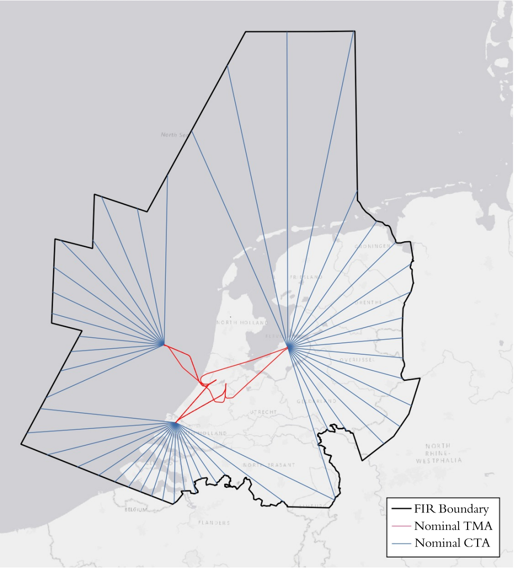


Figure 5.9: Overview of the nominal paths used for trajectory prediction

For the nominal speed profile in the ACC segment, it is assumed that the indicated airspeed at FIR entry is maintained up until 10 nautical miles distance before the IAF. At this point, if the indicated airspeed is above 250 knots it is reduced to 250 knots. For speeds below 250 knots at 10 nautical miles before the IAF point, it is assumed that this speed is maintained. These speeds are considered to be aircraft type independent. The reason for this assumption is that speeds in the ACC airspace are most often chosen based on a cost index, which is airline and scenario dependent [1].

In contrast to the assumed nominal speeds in the ACC segments, the nominal speeds in the TMA segment are assumed to be aircraft type dependent. Because of the observed indicated airspeed differences at different altitudes for all types, the speeds are defined per aircraft type for two altitude ranges. These ranges are based on the observed speed profiles selected to be 10000-6000 feet and 6000-1310 feet.

In the interaction with the planning agent, after requesting a planning update, the ATC agent will receive up-to-date planning information. This planning update is used by the air traffic control agent to provide required instructions to the flight agent. If the planned time that is returned by the planning agent lies within the earliest and latest arrival time, this time is chosen as the instruction for the arrival flight agent. If however, the planned time does not lie within the latest and earliest arrival time, the closest of the two maxima is selected as the time instruction.

An air traffic control agent responsible for instructions at the IAF can also provide a holding instruction. For this instruction, the latest arrival time is determined using the minimal speed and maximum distance. By comparing this latest time and the planned time, it is determined how much additional delay needs to be absorbed by flying a holding pattern. If a holding instruction is provided, the arrival flight agent's holding time is adjusted accordingly.

Inbound Planning

Inbound planning is aimed at making optimal use of the available runway resources. The first step in this procedure is to assign a runway to the flight agent. In the case of one active arrival runway, this process is straightforward because every flight is assigned the same runway. For the situation in which two arrival runways are available, arrival flight agents are distributed among the active arrival runways. In both scenarios, the runways act as a queue (see figures 5.10 and 5.11).

The flights are scheduled to arrive at times based on a first arrive first serve (FAFS) principle. The exact implementation of the flight distribution over runways and the scheduling algorithm can be found in appendix D. After this scheduling action has been performed, the schedule in combination with the received nominal times is used to determine delays for arrival flight agents. These delay values are updated in the flight agent delay property by the planner agent.

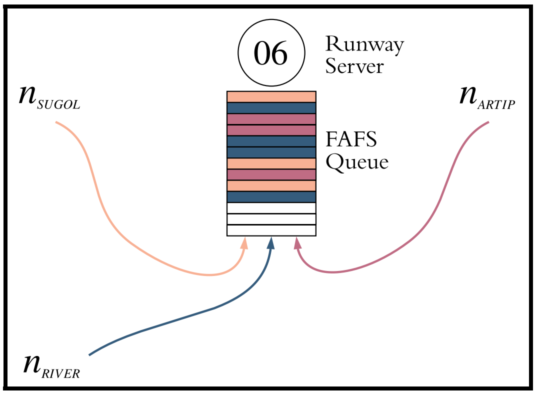


Figure 5.10: The FAFS queue with one active inbound runway

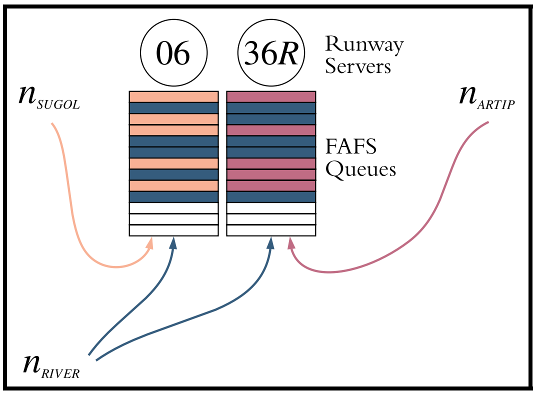


Figure 5.11: The FAFS queue with two active inbound runways

5.3.3 Outbound Model

The outbound model is aimed to resemble the real-world outbound system at AAS, albeit a simplified version of this system. The term outbound system is used to describe all entities and systems that are responsible for the control of outbound traffic at AAS. This section describes the model implementation for the outbound system using the flight agent, air traffic control agent, and planning agent types.

Outbound Model Overview

For the implementation of the outbound model, the elements that make up the model are limited to the ones that are relevant for the outbound planning. Structurally this results in the overview provided in figure 5.12. The model does not take into account the departure trajectory after take-off. This results in a trajectory for a departure flight agent that starts at the apron area and ends at the runway.

The only runway ends that can be used are those for runway 36C and 36L. A runway end act as the entry to the departure runways. To make sure that flights are separated according to separation limits, an ATC agent is implemented that acts as a runway controller at every runway end. The apron areas in the outbound model are an aggregation of the parking positions used by the flight agents and can be used by one flight agent at a time. Aggregation is done based on the parking positions that interfere when pushback is provided to a flight in one of these parking positions. Once a flight agent enters an apron area, it is blocked for a short period depending on the ICAO category of the flight agent.

At each of the apron areas, an ATC agent is present that observes this apron area. Both departure flight agents that are spawned here or arrival flight agents that arrive here may be observed. The ATC agents are responsible for providing instructions to the flight agents in the form of trajectory updates. Planning information for this trajectory updates originates from an implemented outbound planning agent.

For the arrival flight agent, an apron area serves as a sink, while for a departure flight agent an apron area acts as the starting point of the trajectory to the runway end. Depending on the runway configuration, either one or

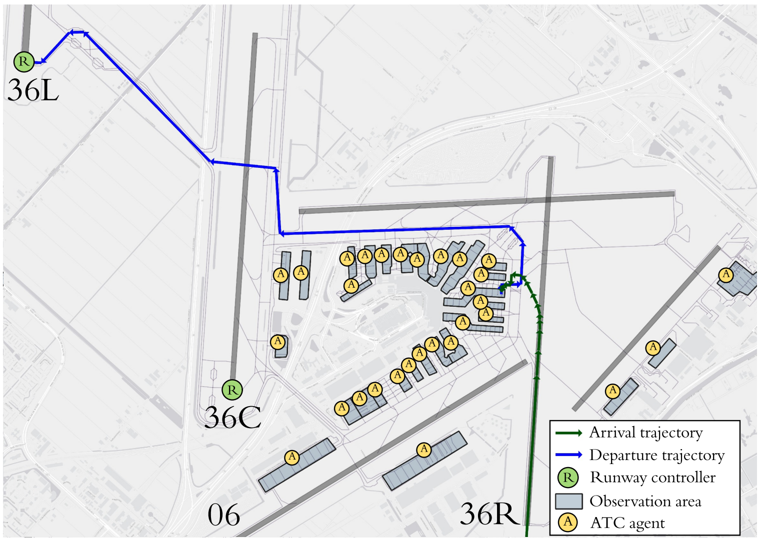


Figure 5.12: Overview of the conceptual outbound model

two runway ends can be active and reachable from the apron areas. This runway configuration is controlled by the supervisor agents.

Outbound Trajectory

The outbound trajectory consists of only two points with an assigned time for each of these points. This is schematically presented in figure 5.13. A departure flight agent receives a trajectory that consists of a position for the scheduled apron area of the agent and a position that corresponds with the assigned runway end of the flight agent. The time of the trajectory point at the runway end is the planned departure time of the flight agent and the time for the first node of the trajectory corresponds with the planned departure time minus the taxi time. The initial node can only be left when the apron area is not blocked. Once the departure flight agent leaves this apron the flight agent will have the position attribute $atPosition(flight, atTaxiway)$.

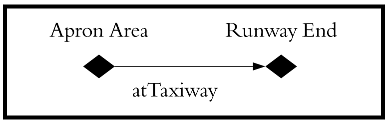


Figure 5.13: The trajectory of a departure flight agent

Outbound Air Traffic Control

In the outbound air traffic control implementation, a set of air traffic control agents exists that are each responsible for one apron area. These air traffic control agents will observe a departure flight agent when it is spawned in the observation area corresponding with the apron area that they are responsible for. When a flight is observed by an apron ATC agent, a planning update will be requested to the outbound planning agent for the observed flight agent.

The air traffic control agent at the apron area uses the received information from the planner agent, in the form of the planned departure time and the assigned runway, to determine the trajectory times for the departure flight agent. The time for leaving the start point of the trajectory, i.e. the apron area, is determined using the estimated taxi time between the apron area and the assigned runway. This is done by subtracting the estimated taxi time from the planned departure time. If the apron area is blocked at the time that the flight agent should leave it, the time for both nodes of the trajectory is translated by the time that the apron is occupied. The resulting times for the trajectory points are then transmitted to the departure flight agent. An update of these times is also provided to the outbound planning agent.

The process of adjusting the trajectory values for the departure flight is repeated as long as the departure flight agent has not left the apron area. Once the departure flight leaves the apron area, the trajectory times cannot be updated by the apron ATC agents anymore. For departure flight agents that have left the apron area, only the runway controller ATC agents can adjust the time for the last node of the departure trajectory to make sure that separation minima are maintained at the runway. Once the current time is larger than the endpoint of the trajectory for a departure flight agent, the flight agent has departed and will remove itself from the simulation.

Outbound Planning

The outbound planning is performed by the outbound planning agent. This agent observes the simulation scenario schedule and enters departure flight agents in the planning at 40 minutes before the departure flight agent is scheduled to leave the apron area. Outbound planning is done in a first arrive serve manner using the inbound times and a computed taxi time property. The first step in this planning is the runway assignment. This runway assignment depends on the runway configuration. In the situation with one active departure runway, all departure flight agents are assigned this runway. In the case of two active departure runways, departure flight agents with departure sector one, four and five are assigned runway 36L and departure flight agents with departure sectors two and three are assigned runway 36C. These departure sectors are depicted in figure 5.15

Once the runway property is established, the taxi time can be determined based on the apron area and runway combination. For each of these combinations, the estimated time is determined which results from the different taxi distances between these points. Using this information in combination with the scheduled time for leaving the apron area, a planning mechanism is performed to determine the planned departure time. The algorithm used in this mechanism is described in more detail in appendix D. The combined system with queues that can arise at the runway ends and the apron areas can be schematically presented by the system in figure 5.14.

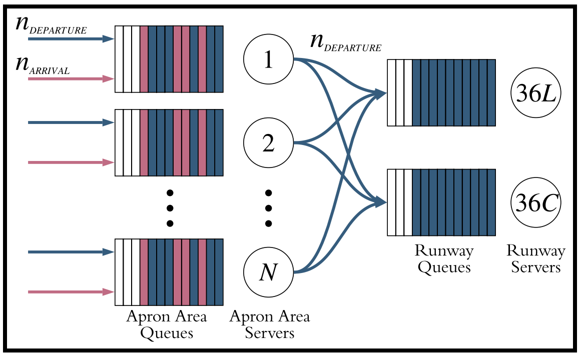


Figure 5.14: Departure planning queues

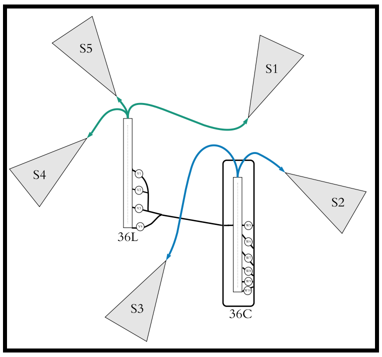


Figure 5.15: Overview of the relation of departure runways and sectors

5.4 Control Model

The control model is formed by the APP supervisor and TWR supervisor agent. These agents emulate the function of the APP supervisor and TWR supervisor air traffic controllers at AAS. The functioning of the control model is aimed at controlling the runway configuration such that it best fits the combination of arrival and departure demand. In this section, the concept for the control model is described. This is done by describing the required external elements and internal architecture for the supervisor agents.

5.4.1 Overview

Central actors in the decision-making process leading up to a runway reconfiguration, are the APP and TWR supervisor. The adaptive and negotiation capacities of these actors are crucial for attaining a satisficing solution that is mutually acceptable, as was identified in the literature review. For this reason, the emphasis of the control model is put on these aspects.

A supervisor agent's internal model should comprise an of an architecture that facilitates the desired cognitive processes and intelligent behaviour. This intelligent behaviour is captured by the cognitive concepts, described in section 3.2.2. The internal model of the supervisor agent is therefore designed based on a combination of cognitive architectures for an anticipatory agent and an ABN agent, that originate from the work of Blok et al. [12](see Fig. 3.1) and Rahwan et al. [58](see Fig. 3.2) respectively. This internal model design is implemented for both the APP supervisor agent and TWR supervisor agent, together forming the control model of the total agent-based model. The conceptual elements for this internal model design, combining anticipatory and negotiation capabilities are schematically presented in an overview of the conceptual model in figure 5.16.

Architecturally the control model is divided up into an operational and strategical layer analogous to the cognitive anticipation architecture in the work of Blok et al. [12]. The operational layer is formed by an action module and a perception module. In this operational layer, the action module is responsible for the preparation and execution of the action that influences the domain model. Observations from the domain model and perception of incoming communications are facilitated by the perception module. This obtained information is transferred to the strategical layer by updating the knowledge base, after interpreting the information.

The belief module and reasoning module together form the strategical layer of the cognitive architecture of the supervisor agent. At any point in time, the knowledge about the world that is available to a supervisor agent is contained in the belief module of the agent. This may be in the form of facts about the current state or a future state of the world, but also in the form of rules that contain epistemic as well as subjective considerations. Updates of the belief module may result from observations performed in the operational layer, from the reasoning steps performed by the agent itself or from the argument content uttered by the other supervisor agent. This knowledge in the belief module is available as input for the reasoning process of the agent.

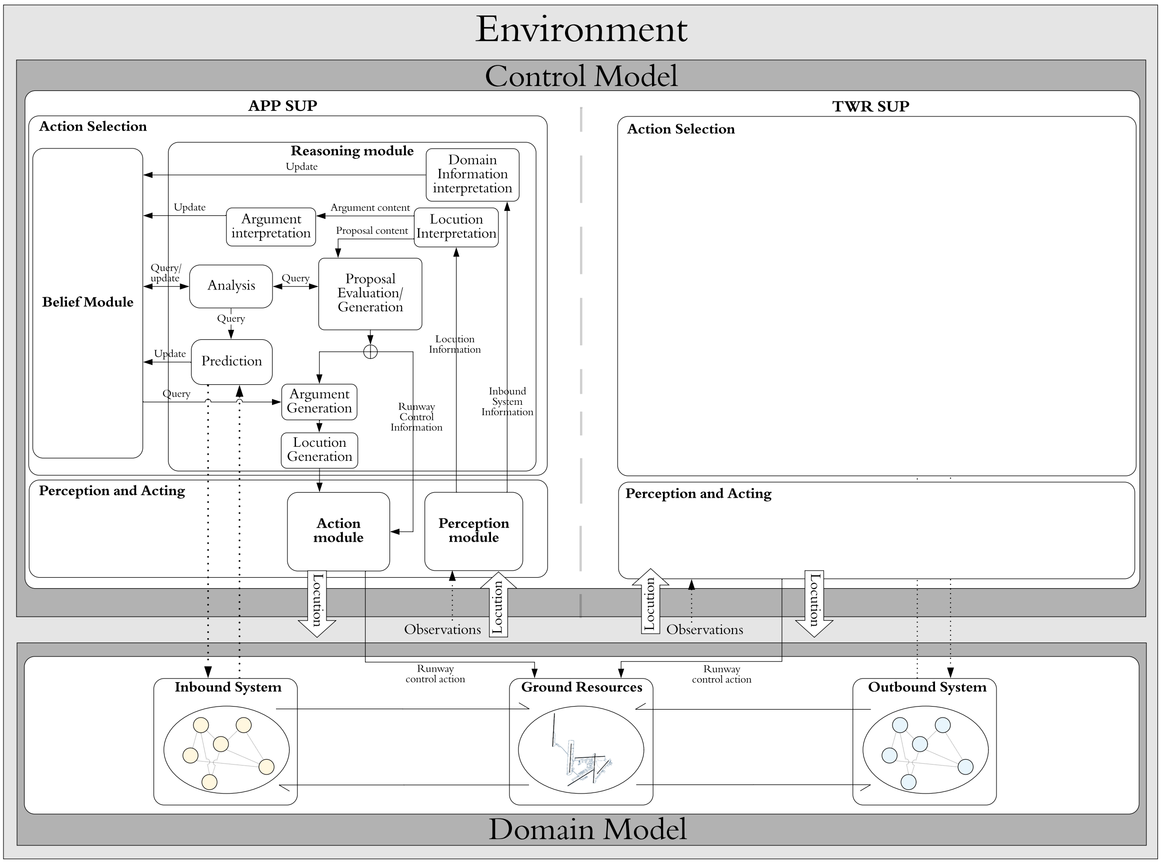


Figure 5.16: Overview of the conceptual model elements of the agent-based model

In the reasoning module, the information of the knowledge base is used during the inference steps to come to a decision. Rather than deciding on the action to perform, the agent is deciding on what locution to utter to the other supervisor to reach a mutually acceptable agreement. In order to perform a negotiation interaction, the minimal set of elements that is required is defined in the schematic overview in figure 3.2. These are[58]:

- An element dedicated to interpreting the incoming locutions.
- An element capable of evaluating incoming proposals.
- An element that facilitates the evaluation of proposals.
- An element that enables an agent to utter locutions.

Apart from these elements, in an argumentation-based negotiation scenario, additional elements are required. To interpret incoming arguments an agent requires an argument interpretation module and to generate arguments that are uttered in the interaction the agent also needs an argument generation component [58]. All these elements are therefore reflected in the internal architecture of the supervisor agent.

To control the demand peak configuration, the control model interacts with the domain model. The domain model is used to predict the state evolution of either outbound or inbound traffic system. This information will be analysed by the supervisor agent and used for proposal evaluation and proposal generation actions. When an agreement is reached, the supervisor agent will use its action module to perform control inputs to the domain model in the form of runway activation or runway deactivation.

5.4.2 External Supervisor Agent Elements

In order to be able to perform actions and the argumentation-based negotiations, certain cognitive elements are required. Apart from this internal architecture of the supervisor agent, external elements are required as well. These external elements consist of the runway control actions, the monitoring action and the protocol. For each of these elements, it is described how they work in this section.

Runway Activation/Deactivation

A runway activation or deactivation action can be performed by a supervisor agent once a mutually acceptable agreement has been reached. Given that no more than two runways are allowed per direction, a runway activation can only be performed when one runway is currently active for either inbound or outbound. A deactivation action can only happen when two runways are active in a traffic direction. It is assumed that only the secondary runways, described in section 2.1.2, can be deactivated. In the runway set used in the model, this is runway 36R for outbound traffic and runway 36C for inbound traffic.

It is assumed that this action always succeeds and no checks have to be performed, which deviates from the real-world situation. In the real-world situation, often runway checks have to be performed. Once a decision is reached on the moment of the runway reconfiguration, the runway status is changed and transmitted to the planning agent. For a runway activation, the runway is activated in the planning system at the moment that the flight option is selected for the reconfiguration. Every flight that is scheduled to arrive later than this activation point can be rescheduled to the newly activated runway. When a decision is made on the deactivation moment for a runway, all flights that are scheduled to arrive later than the last flight using the to be deactivated runway are rescheduled to the remaining active runway.

Monitoring

To timely detect the possibility of reaching an undesirable state, requires monitoring information about the current state of the system. Monitoring is aimed at providing this functionality, such that the anticipatory and negotiation behaviours can be activated. Reaching an undesirable state for the model would mean that a situation occurs in which the operations are not functioning within their desired operational performance bounds.

Desired operational performance can be expressed in terms of performance indicators. The most important performance indicator throughout the system is the delay of either arrival or departure flight agents. For arrival flight agents, this delay is absorbed in the flight legs in ACC and TMA airspace. This can be done by speed or vectoring instructions or instructing a holding pattern. While applying speed and vectoring instructions is regarded as being a part of normal operations, instructing a holding pattern is regarded as an undesirable operation, as explained in section 2.1.3. For this reason, four minutes of delay for an arrival flight agent is considered as the operational performance bound for the inbound system.

In the outbound system, the operational performance is regarded as within the desirable bounds when the assigned departure slots can be maintained, as explained in section 2.1.4. Assigning new slots causes high delays and should, therefore, be avoided. It is therefore desired to keep the start-up delay below 10 minutes. If this is not possible any more new slots will have to be requested for these flights which would result in unacceptable high delays for these flights.

In order to avoid a state that lies outside the desired performance bounds, the schedule is adapted towards an expected runway configuration throughout the day. The peak schedule(see section 2.1.2) is regarded as a guideline for when a certain demand peak configuration should be used. In the monitoring function of the supervisor, the peak schedule acts as the first information source for detecting a signal triggering anticipatory and negotiation behaviour. When a peak is being observed in the peak schedule, that is within 40 minutes and is different from the current observed demand peak configuration it is considered as a cue to activate the anticipatory and negotiation behaviour.

Due to the uncertain nature of the demand pattern throughout the day, the required demand peak configurations for any point in time can significantly differ from the defined peak schedule. It is therefore not sufficient to only monitor the peak schedule. To identify an upcoming undesirable state it is therefore required to monitor the information provided by the planner agent as well. This information consists of delay values. This delay information is used to determine when a runway reconfiguration is expected to be required, by identifying the flights that have delays lying outside of the desired performance bounds.

A detected signal for an upcoming demand peak is regarded as a strong signal, whereas a delay signal observed in planning information is referred to as a weak signal. The distinction between a weak and a strong signal does not arise from the future state, but the system observing the signal. A weak signal is weak because of a lack of certainty hindering the system to relate a reference to an event with a reference to a cognitive mapping [20]. If a weak signal is detected in the monitoring activities, a supervisor agent is not certain about the effect of this signal because it is based on uncertain planning information. For a weak signal, a forward simulation is therefore performed to generate more information about the event related to this signal in the form of a prediction of the effect. A signal of an undesirable state perceived in this forward simulation is regarded as a strong signal.

If a delay signal is observed it is checked if the delay is not resulting from only one flight that has a delay value above the threshold of four minutes. If this were to be the case, the delay is likely not for capacity reasons. If it were to result from capacity reasons, a delay buildup pattern would be expected. To ensure that the delay signal is a signal for a delay build-up resulting from insufficient capacity, the slope of the cumulative delay of the subsequent flights is checked. If this slope is higher than zero, the delay signal is considered as a valid cue. If either a cue signal of an upcoming peak is observed or a valid delay cue is observed in the upcoming 40 minutes the anticipatory behaviour of the supervisor agent is triggered.

Negotiation Protocol

To define which of the two supervisor agents is allowed to perform what communicative act at any point in the interaction, a protocol is implemented. This protocol serves as the initial element in the action selection mechanism of the supervisor agent, to determine which actions are allowed. The protocol is aimed at structuring the interaction between the supervisor agents. It is essential to know at every point in the interaction which role is played by every supervisor agent. This can either be the initiator or the participant. Which role is taken on, results from the protocol based on the current demand peak configuration and the upcoming demand peak configuration. Details of this protocol are specified in appendix B in the form of dialogue games, a formalism introduced in the work of Mcburney et al. [50].

The protocol dictates that the supervisor agent requiring an additional runway according to the peak schedule is always the initiator. The reason for this is that the urgency for performing a reconfiguration for the supervisor that requires more capacity in the new peak configuration is higher than for the supervisor deactivating. This sense of urgency results from the fact that it is a larger problem to deal with delay at the start of a peak than at the end of a peak. This is illustrated in figure 5.17. If the delay is encountered at the start of peak like in scenario 1 in the figure, it is much harder to compensate for this delay because the demand is rising. At the end of the peak, this is easier because demand is decreasing and fewer capacity problems are encountered. Furthermore, it can be observed that the amount of flights affected by delay at the start of a peak is much higher than the number of flights affected by delay at the end of a peak.

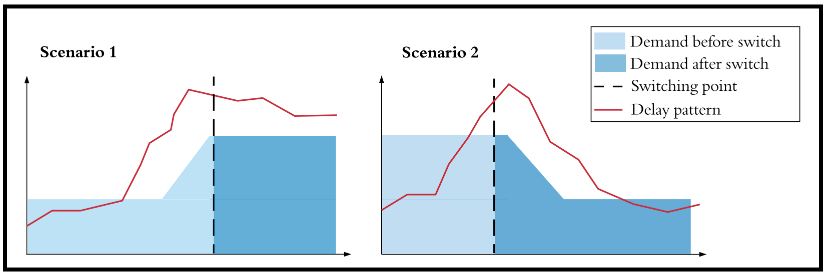


Figure 5.17: Sense of urgency for the initiator

5.4.3 Internal Supervisor Agent Elements

In the overview of the control model, the internal elements of the supervisor agent are introduced (see figure 5.16). A conceptual description for these internal agent elements is provided in this section. This consists of the elements required for interpreting incoming information, proposal generation and proposal evaluation.

Option Generation

Given the allowed locution resulting from the protocol, the type of locution that can be uttered is established. For this type of locution, it is however still to be decided what proposal content is inserted into the locution. This proposal content consists of flight names. These flight names will be the options for deactivating or activating the runways depending on the locution content. In order to make a well-informed decision for selecting these options, requires considering multiple options.

As defined in the runway activation/deactivation action in section 5.4.2, there is only one option for the to be activated or deactivated runway. The option generation is therefore limited to generating options for which flight must be selected as the first flight to use the activated runway or the last flight to use the to be deactivated runway. Option generation for this free variable is aimed at finding a minimal set containing viable candidates. In order to achieve this, it is desired to discard any options that can be considered to be unattractive.

For generating activation options, the detected cue provides an indication of the moment in time at which the system is expected to be in an undesirable state. If this is due to insufficient capacity and a runway activation is required, avoiding this undesirable state is only possible by activating a runway before this moment. This means that for any flight option that is scheduled to utilise the runway after the time of the cue signal, it can be established with a fairly high degree of certainty that this option will not be suitable to avoid the undesired state of the system. This assumption is strengthened by the fact that it is highly undesirable to have a delay at the start of a new peak. These options are therefore disregarded in the initial option selection.

Disregarding options that are scheduled to use the runway capacity at a later point in time than the cue signal still leaves all flight options before that point in time. It is desired to avoid delay by letting aircraft use a second runway. If a flight cannot be assigned this runway, due to its departure sector or IAF, it will not be able to use a newly activated runway. Flights that cannot be assigned to the to be activated runway are therefore also be disregarded as activation options.

This leaves only the flight options that are scheduled before the cue signal and can be assigned to the newly activated runway. When absorbing delay, flights are dependent on the previous flight because a flight can only use the runway after the previous flight in the planning has cleared the runway. When considering the delay build-up, it is therefore only relevant to look at the options that are causing the delay build-up. This will be the flights that form a closely spaced group in the planning.

A graphical example of this option generation for a runway activation is presented in figure 5.18. In the lower part of the bar chart, a cue can be observed in the form of an upcoming demand peak represented by the blue line. The flight names in the chart are positioned at the time that they are scheduled. A bold flight name is used to indicate a flight that can be scheduled on the to be activated runway and a normal font is indicating a flight that cannot use the to be activated runway. The group of options that should be selected in this example is the indicated dashed box.

In the upper part of the chart, an option selection scenario for the activation of an arrival runway is depicted. In this scenario, the cue is in the form of a delay cue around 5:05 UTC. The options that are now selected are the options that can use the newly activated runway and are closely spaced thereby forming a group before the cue signal. In this scenario, the options KLM80Z and KLM34Z are therefore chosen as candidates for using the to be activated runway as the first flight.

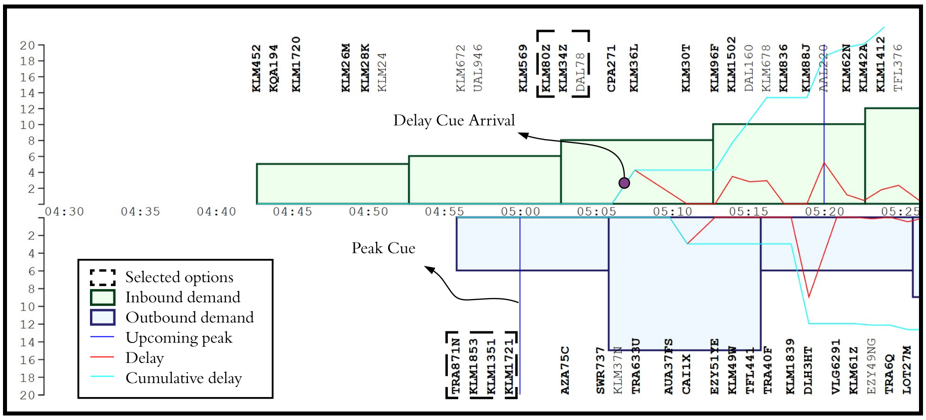


Figure 5.18: Option generation example for activating a runway

Selecting options for deactivating a runway is done following a different procedure. This is done by first evaluating the results of selecting the option that avoids using a fourth runway. If an arrival runway is activated at a certain moment, a departure runway should be deactivated immediately after this moment and vice versa. The initial option to deactivate a runway is therefore the first flight that is scheduled to land/depart after the moment of activating a runway in the other traffic direction. The effect of this initial deactivation option can be evaluated using a forward simulation of the domain model. If the initial deactivation option is expected to not result in violating the desired operational performance bounds, this option can be selected. However, if this is not the case an alternative deactivation option should be selected. An example simulated result of the deactivation of a runway for the option that avoids using a fourth runway is depicted in figure 5.19. It can be observed from the figure that the simulated delay pattern resulting from this deactivation action shows delay values that exceed the acceptable delay threshold of four minutes in the two shaded areas. This initial deactivation option is therefore expected to be an undesirable option and an alternative later option for deactivation should be selected.

The simulated result of the initial deactivation option can be used for selecting alternative options. The flight names in the figure are positioned horizontally at their scheduled arrival time and are indicated in bold if the flight would be scheduled to a different runway than the previous flight when two runways were to be active. If different runways can be used by subsequent flights, this means that the capacity of two runways can be used. The only

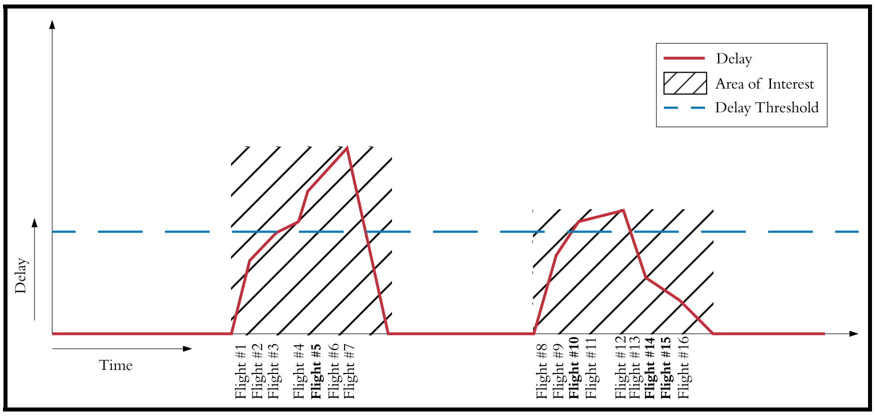


Figure 5.19: Alternative selection

way to avoid the delay build-up in the shaded areas is to deactivate at a moment after the capacity of two runways is used, for example by the Flight#4 and Flight#5 that can use a different runway when two are active. Flight#6, Flight#11, and Flight#16 are therefore suitable options because they are scheduled after flights that can make use of the capacity of a second runway. These flights are therefore selected as suitable alternative deactivation options.

Option Selection and Proposal Generation

Consequences of the generated action options are simulated by a forward simulation of the domain model. This process is aimed at resembling a mental simulation that a human operator would perform in reality to assess the result of performing a certain action. During the forward simulation, the considered action is performed in the model and resulting parameters for executing this action are derived. This is done in such a way that the model that is running in real-time (of the simulation) is not affected.

After running a forward simulation, the results of this forward simulation are interpreted by the supervisor agent. The information retrieved from the forward simulation is considered sensory information and will be in the form of facts. These facts are combined with expert knowledge rules to form the knowledge base of the supervisor agent. Using an argumentation procedure that will be described in more detail in section 6.6.3, one of the generated action options is selected as the best option. This option will together with the allowed locution, derived from the protocol, form the proposal that is uttered by the supervisor agent.

Argument selection and locution Generation

Once it is decided what the best option is, a locution must be generated. This locution does not only contain the proposal information but also the information about the considered options generated in the argumentation procedure. This information can be interpreted by the other supervisor agent and used for the reasoning process of this supervisor agent. The information that will be transferred is only information that is relevant to the opponent supervisor agent and provides only information about action options that are desirable for the supervisor agent providing it.

Argument and Proposal Evaluation

Once a locution comes in, the first step for the supervisor agent is to interpret the argument and proposal content. Based on the proposal content, the supervisor agent performs the semantic analysis to determine the required actions for this proposal. In case of an acceptance locution, the supervisor will have to perform a runway control action. When a deactivation proposal for the system of the supervisor agent is received, the supervisor agent will as a first step determine what the effects are for the deactivation action. The resulting information is used in the evaluation of the proposal.

In case of a proposal with two options, the supervisor agent will assess what the effect of the combination of options is. Both the information about the effect in terms of delay and in terms of fourth runway use are added to the knowledge base of the supervisor agent. With all the required information collected for assessing the proposal, the supervisor agent will perform an argumentation procedure to evaluate the proposal. If the proposal can be accepted the supervisor agent will utter a locution indicating accepting the received proposal. If the proposal cannot be accepted, the supervisor agent will start preparing a counter-proposal.

5.5 Interactions

The perception and acting capabilities of the agents in the model result in an interaction between the different entities in the model. This section describes which interactions occur in the model. The interactions among agents are described in section 5.5.1 and the interactions between the agents and the environment are described in section 5.5.2.

5.5.1 Interaction Among Agents

In the model, interactions happen between the different agent types and among agents of the same type. These interactions are schematically presented in figure 5.20. Interactions between the different agent types are for complexity reasons modelled to go through the environment. This means that agents communicate information to the environment that keeps a global state. This updated global state of the environment, in turn, acts as an information source that all agent types can use to update their internal state. Organising communications in this manner omits the necessity for a protocol and reception confirmations.

The only agent type for which interaction is done directly, without using the environment, is the supervisor agent type. For this agent type communication will always be between two parties, namely the APP supervisor agent and TWR supervisor agent. This simple interaction structure does not require any intricate protocol. It is therefore decided to make use of a simple protocol that is similar to a propose interaction protocol by FIPA [30].

The negotiation model is aimed at providing an agreement between the two supervisor agents at a high rate of convergence. Potential for a high convergence is increased if both parties in the negotiation are well informed about their own as well as their opponent's mental attitude. This requires an interaction that shows a high level of expressiveness. Replacing a reject locution by a counter-propose locution increases this expressiveness and provides information in the form of proposal content. Given this desired property and the fact that it is aimed to avoid no-agreement results from the interaction between supervisor agents, it is decided to replace the reject-proposal communicative act by a counter-proposal communicative act. This results in the protocol of the form presented in figure 5.21. A description of how this protocol is implemented can be found in appendix B.

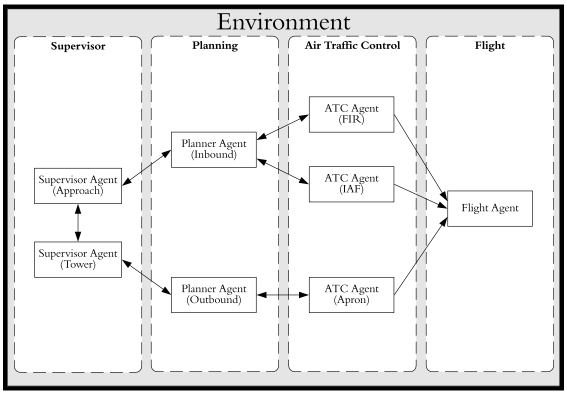


Figure 5.20: Agent interaction

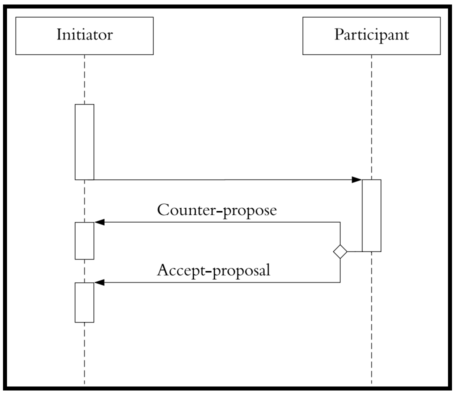


Figure 5.21: Supervisor agent protocol

5.5.2 Interaction Between Agents and Environment

The environment acts as a central hub that facilitates perception for agents. It holds a global state from which specific information can be obtained by the perception functions of agents, at the start of every tick update. This information is the result of updates that are done by agent types at the end of every tick update. This interaction between the agent types and the environment provides an indirect way for communication between different agent types.

The supervisor agent and the air traffic control agent are the only agent types for which interactions with the environment are not limited to perception. A supervisor agent is also capable of changing the status property of a runway object from active to inactive and the other way around. Changing this status is only possible for a supervisor agent if the runway object belongs to the system that the supervisor agent is responsible for. Air traffic control agents responsible for apron areas are capable of updating the status property of an apron area. This status can either be blocked or free.

6

Model Description

The conceptual description in chapter 5, has introduced the elements of the model conceptually. To describe the model, the conceptual elements are translated into a more formal description. For this description, apart from natural language, a set of formal languages is used that will be introduced. After these languages are introduced, the environment and agent types are described using these languages.

6.1 Formal Languages

For the description of the model, natural language description is supplemented by formal description. Because formal languages are suited to describe specific types of processes, a combination of formal languages is used to describe different elements. To describe the dynamics of the model in an executable format, the LEADSTO language is used[18]. This language allows for modelling the dynamics of the model by using dynamic properties, which feature temporal relations leading to simulation traces (sequences of states). Procedures that are carried out in a predefined sequence, are described using an algorithmic description. The more specific defeasible reasoning elements are described in the Defeasible Logic Programming language (DeLP)[31] and the protocol is described using a Dialogue Games formalism[50]. This section introduces these formalisms.

6.1.1 LEADSTO Language

The LEADSTO language is used for describing direct temporal dependencies between two states[18]. It focuses on combining qualitative relations and quantitative relations. The relations can therefore be used to define the dynamics of a cognitive process. The LEADSTO language provides the capability to describe an internal process of an agent using mental states. Such a mental state is a property defined from a functionalists perspective, meaning that these properties do not necessarily exist in reality, but may merely serve as instruments to describe a process[45].

Model dynamics are described by state properties that evolve. These state properties are defined using an ontology, that maintains the set of properties that do or do not hold at a certain point in time. These states can be written in an executable format, by defining unique future states for current states. A time-indexed sequence of these states is referred to as a simulation trace. Simulation of a model defined using the executable format will result in different traces for different simulation scenarios.

An ontology that formalises the state properties is defined in a first-order logic format. It is constructed by defining a finite set of sorts, constants contained by these sorts, relations over sorts, and functions over sorts[16]. For the ontology, a distinction is made between internal, external, input and output state properties. The input and output ontology together define the state properties relevant for the interaction of agents. The internal ontology specifies internal states and the external ontology specifies the states of the external world.

LEADSTO facilitates modelling direct temporal relationships between states. The format used to specify these relationships is defined as $\alpha \twoheadrightarrow_{e,f,g,h} \beta$, which is referred to as a dynamic property. In this definition, if state property α holds for a time duration g then after a delay between e and f state property β holds for a time duration h. This format allows to define specifically timed state property evolutions and can also be used to define general state evolutions, when defined as $\alpha \twoheadrightarrow_{0,0,1,1} \beta$.

6.1.2 Algorithmic Description

When a process is a defined sequence of states, this process can be described procedurally using an algorithmic description. The algorithmic description is done using a combination of flowcharts and pseudo-code. Flowcharts are used to provide an overview of the sequence and relations of processes that need to be performed for a procedure.

The individual processes that require further elaboration are described using a combination of pseudocode and natural language. In these descriptions, mathematical symbols, as well as programmatic constructs such as for loops and while loops, are used to describe the process.

6.1.3 Defeasible Logic Programming Language

The Defeasible Logic Programming(DeLP) formalism is defined in a language that facilitates expressing three disjoint sets of information[31]. These are a set of facts, a set of strict rules, and a set of defeasible rules. Facts are defined as ground atoms or negated ground atoms. Whereas an atom $p(t_1, \dots, t_n)$ is a predicate symbol with several free terms corresponding with the arity of the predicate symbol, a ground atom is an atom that only has terms that are fixed. These ground atoms and negated ground atoms are referred to as literals, which encompass strong negation if the predicate symbol is preceded by the symbol $\sim$.

The other two disjoint sets of information describe strict rules and defeasible rules respectively. A strict rule consists of a body $\{L_1, \dots, L_n\}$ which is a non-empty set of literals and the head L_0, that are combined by a $\leftarrow$ symbol. This syntax corresponds to the definition of basic rules in the work of Lifschitz [48]. The fact that these rules are called strict rules in the DeLP formalism stems from the desire to differentiate these rules with non-defeasible information from the rules with defeasible information. These rules with non-defeasible information syntactically only differ from strict rules in the meta-relation symbol that is used. Where the $\leftarrow$ is used to represent the strict relation for strict rules, the $\prec$ symbol is used to represent a weak relation for defeasible rules. The information contained in the defeasible rules can be interpreted as tentative information that can be used if nothing can be posed against it.

All information contained in the three disjoint sets is combined into a defeasible logic programme $\mathcal{P}$. This logic programme will be used to compute query outcomes using an argumentation procedure, described in section 6.6.3. In the DeLP programme, a distinction is made between a subset of facts and strict rules and a subset of defeasible rules. The subset of facts and strict rules is denoted by Π and the subset of defeasible rules is denoted by Δ. The defeasible logic programme $\mathcal{P}$ can therefore be defined as (Π, Δ).

6.1.4 Dialogue Game Protocol

For the description of the protocol used for the negotiation between supervisor agents, the Dialogue Game formalism is used[50]. The protocol is used by agents to identify the appropriate speech act. It defines constraints on which utterances are permitted and defines which sub-tasks must be performed by an agent to engage in the negotiation dialogue. To avoid ambiguity and have a finally verifiable protocol, all rules must be defined syntactically without any semantic element.

The Dialogue Game formalism in the work of Mcburney et al. [50] provides an example format that allows defining rules, pre-conditions, post-conditions and what information and commitment stores are required for the locution. An example of this format is provided below[50]:

Locution: willing_to_sell($P_{Yj}, \mathcal{T}, P_{Sk}, V$), for $Y \in \{A, S\}$, $\mathcal{T}$ a set of participants which includes both P_{Yj} and P_{Sk}, where P_{Sk} is a seller participant and V is a set of sales options.
Preconditions: Some participant P_{Xi} must have previously uttered a locution **seek_info**($P_{Xi}, \mathcal{S}, p$) to a participant P_{Yj}, where $P_{Yj} \in \mathcal{S}$, and the set of sales options V in the **willing_to_sell(.)** locution must satisfy constraint p.
Meaning: The speaker, a seller or advisor P_{Yj}, indicates to the audience $\mathcal{T}$ a willingness by a seller participant P_{Sk} to supply a finite and possibly empty set $V = \{\tilde{a}, \tilde{b}, ...\}$ of purchase-options to any buyer participant in the set $\mathcal{T}$. Each of the sales options tendered in the set V must satisfy constraint p uttered as part of the prior **seek_info(.)** locution.
Response: None required.
Information Store Updates: For each $\tilde{a} \in V$, the 3-tuple $(\mathcal{T}, P_{Sk}, \tilde{a})$ is inserted into $IS(P_{Yj})$, the Information Store for participant P_{Yj}.
Commitment Store Updates: No effects.

6.2 Global Clock

In the model, a global clock variable exists that can be perceived by every agent in the model. This clock variable provides the global current time. The global current time is updated in discrete time steps, referred to as ticks. Tick updates are a trigger for every agent to update their knowledge about the world, using its perception and interpretation modules. This may activate internal action selection mechanisms that use the newly obtained information.

An interpretation of a tick update exists for an agent if it does not know the time value for the current tick or if the perceived global time is different from the believed time value for the current tick. If this state holds, the agent will update its belief for the current tick and interpret this as a tick update. This is described more formally in LEADSTO as:

$$\begin{aligned}&\mathit{external}(\mathbf{A}{:}\mathbf{Agent})|\mathit{global\_clock}(\mathbf{current\_time},\ \mathbf{t})\ \&\ \mathit{input}(\mathbf{A}{:}\mathbf{Agent})|\mathit{observes}(\mathbf{A}{:}\mathbf{Agent},\ \mathit{global\_clock}(\mathbf{current\_time},\ \mathbf{t}))\\ &\&\ \mathit{internal}(\mathbf{A}{:}\mathbf{Agent})|\neg \mathit{has\_tick\_time}(\mathbf{A}{:}\mathbf{Agent},\ \mathbf{Time}{:}\mathbf{Time\_value}) \vee [\mathit{has\_tick\_time}(\mathbf{A}{:}\mathbf{Agent},\ \mathbf{t}_{tick}) \wedge \mathbf{t} > \mathbf{t}_{tick}]\\ &\twoheadrightarrow_{0,0,1,\mathbf{t}_{tick\_interval}} \mathit{internal}(\mathbf{A}{:}\mathbf{Agent})|\mathit{has\_tick\_time}(\mathbf{A}{:}\mathbf{Agent},\ \mathbf{t})\end{aligned}$$

$$\begin{aligned}&\mathit{external}(\mathbf{A}{:}\mathbf{Agent})|\mathit{global\_clock}(\mathbf{current\_time},\ \mathbf{t})\ \&\ \mathit{input}(\mathbf{A}{:}\mathbf{Agent})|\mathit{observes}(\mathbf{A}{:}\mathbf{Agent},\ \mathit{global\_clock}(\mathbf{current\_time},\ \mathbf{t}))\\ &\&\ \mathit{internal}(\mathbf{A}{:}\mathbf{Agent})|\neg \mathit{has\_tick\_time}(\mathbf{A}{:}\mathbf{Agent},\ \mathbf{Time}{:}\mathbf{Time\_value}) \vee [\mathit{has\_tick\_time}(\mathbf{A}{:}\mathbf{Agent},\ \mathbf{t}_{tick}) \wedge \mathbf{t} > \mathbf{t}_{tick}]\\ &\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}{:}\mathbf{Agent})|\mathit{int}(\mathbf{A}{:}\mathbf{Agent},\ \mathbf{tick\_update})\end{aligned}$$

6.3 Flight Agent

A flight agent exhibits purely reactive behaviour. The actions of this agent are limited to updating its time interpretation, position and trajectory. This section describes the characteristics, perceptions and actions of the flight agent.

6.3.1 Characteristics

For a flight agent, the characteristics are dependent on whether the flight agent is of the arrival type or the departure type. The features that a flight agent possesses are therefore described using features for an arrival flight agent and features for a departure flight agent. The features of a flight do not merely consist of physical aspects of the agent. Most features of a flight agent are related to the planning and control of the flight agent.

The static features of a flight agent result from the parameters in the input traffic schedule for the simulation scenario. These are described for arrival flight agents as well as for departure flight agents in section 7.2.1. Apart from features that persist throughout the lifetime of an agent instance, the flight agent also possesses attributes that are updated throughout its lifetime in the simulation. These dynamic attributes are:

Arrival flight agent

- Planning status
- Assigned runway
- Position
- Time for arriving at the IAF
- Time for arriving at the runway
- Time for arriving at the apron
- Arrival delay

Departure flight agent

- Planning status
- Assigned runway
- Position
- Time for going off-block
- Time for arriving at the runway
- Start-up delay

6.3.2 Perception Capabilities

A flight agent utilises its perception module to perceive elements of the model environment. This information is used by the agent to update its mental attitudes and perform actions based on these mental attitudes. The information elements perceived by the flight agent are:

- **Global clock**: A global clock variable that is, part of the environment, facilitates the perception of the global current time. This enables the flight agent to update its internal model with the current time at every tick.
- **Instructions**: A flight agent can perceive communications by the air traffic control agent. These communications provide instructions to the flight agent.

6.3.3 Actions and Cognitive Processes

The actions of a flight agent are initiated by receiving a trajectory. If a flight agent receives a trajectory update from an air traffic control agent, it will have a current trajectory. This trajectory, $\mathit{trajectory}(\mathbf{A}{:}\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle)$, consists of two position-time tuples. It will furthermore initiate an action to move along this trajectory. This is defined in excecutable format using LEADSTO as:

$$\mathit{external}(\mathbf{A}:\mathbf{Flight})|\mathit{communicated}(\mathbf{B}:\mathbf{Atc},\ \mathbf{A}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{A}:\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle))$$

$$\&\ \mathit{input}(\mathbf{A}:\mathbf{Flight})|\mathit{receives}(\mathbf{A}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{A}:\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle))$$

$$\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}:\mathbf{Flight})|\mathit{has\_trajectory}(\mathbf{A}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{A}:\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle))$$

$$\&\ \mathit{output}(\mathbf{A}:\mathbf{Flight})|\mathit{to\_be\_performed}(\mathbf{A}:\mathbf{Flight},\ \mathit{go\_to\_from}(\mathbf{pos}_{end},\ \mathbf{pos}_{start}))$$

The trajectory property will remain valid for the flight agent as long as the tick time is less than the end time of the trajectory. Once this condition is violated the trajectory property does not hold anymore and the flight is at the end position of the trajectory:

$$\mathit{internal}(\mathbf{A}:\mathbf{Flight})|\mathit{has\_trajectory}(\mathbf{A}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{A}:\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle))$$

$$\wedge\ \mathit{has\_tick\_time}(\mathbf{A}:\mathbf{Flight},\ \mathbf{t}_{tick})\ \wedge\ \mathbf{t}_{tick} < \mathbf{t}_{end}$$

$$\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}:\mathbf{Flight})|\mathit{has\_trajectory}(\mathbf{A}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{A}:\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle))$$

$$\mathit{internal}(\mathbf{A}:\mathbf{Flight})|\mathit{has\_trajectory}(\mathbf{A}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{A}:\mathbf{Flight},\ \langle \mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle \mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle))$$

$$\wedge\ \mathit{has\_tick\_time}(\mathbf{A}:\mathbf{Flight},\ \mathbf{t}_{tick})\ \wedge\ \mathbf{t}_{tick} \geq \mathbf{t}_{end}$$

$$\twoheadrightarrow_{0,0,1,1} \mathit{external}(\mathbf{A}:\mathbf{Flight})|\mathit{is\_at\_position}(\mathbf{A}:\mathbf{Flight},\ \mathbf{pos}_{end})$$

6.4 Planner Agent

A planner agent serves as a source of information for air traffic control agents. It is reactive and provides scheduling information upon request by air traffic controller agents. Because of the difference in planning methods, one dedicated planner agent exists in the model that is responsible for planning arrival flights and one dedicated planner agent exists that is responsible for planning departure flights.

6.4.1 Characteristics

The physical characteristics are not relevant for the planner agent. The only relevant feature of the planner agent is its responsibility. A planner agent is either responsible for flight agents that are of the type inbound or outbound. This is denoted by $\mathit{responsible\_for}(\mathbf{A}:\mathbf{Planner},\ \mathbf{Flight\_type}:[\mathbf{Inbound},\ \mathbf{Outbound}])$. This feature is used by air traffic control agents to communicate with the appropriate planner agent.

6.4.2 Perception Capabilities

- **Schedule:** A planner agent can observe the flight schedule. This information is required for the planner agent to be able to include flights in the planning once these flights are deemed eligible for planning. More details on this process can be found in appendix D.
- **Planning update request:** Once a planning update is requested by an air traffic control agent, a planner agent can perceive this request.

6.4.3 Actions and Cognitive Processes

The planner agent has two responsibilities. Firstly, it is responsible for including a flight in the planning when a flight is eligible for planning. Secondly, a planner agent will provide planning updates when requested by air traffic control agents. The first action is initiated when a planner agent observes a flight that is eligible for planning and falls under the responsibility of the planner agent. If this is the case, the flight will be included in the schedule by the planner agent:

$$\mathit{external}(\mathbf{A}:\mathbf{Planner})|\mathit{responsible\_for}(\mathbf{A}:\mathbf{Planner},\ \mathit{flight\_type}(\mathbf{B}:\mathbf{Flight},\ \mathbf{flight\_type}))$$

$$\&\mathit{input}(\mathbf{A}:\mathbf{Planner})|\mathit{observes}(\mathbf{A}:\mathbf{Planner},\ \mathit{eligible\_for\_planning}(\mathbf{schedule},\ \mathbf{B}:\mathbf{Flight}))$$

$$\twoheadrightarrow_{0,0,1,1} \mathit{output}(\mathbf{A}:\mathbf{Planner})|\mathit{performed}(\mathbf{A}:\mathbf{Planner},\ \mathit{pre\_planning}(\mathbf{B}:\mathbf{Flight}))$$

The eligibility check and the scheduling action are described in more detail in appendix D. The provision of flight updates by the planner agent is initiated by receiving a planning request. Once this planning request is received, the planner agent will perform a planning update action. Newly acquired planning information resulting from the planning update action for a specific flight agent will subsequently be communicated to the air traffic control agent, that requested this information. Additionally, this information is shared with the supervisor agent that supervises the system affected by the planning update. This is more formally described as:

$$\mathit{external}(\mathbf{A}:\mathbf{Planner})|\mathit{communicated}(\mathbf{B}:\mathbf{Atc},\ \mathbf{A}:\mathbf{Planner},\ \mathit{planning\_update\_request}(\mathbf{C}:\mathbf{Flight}))$$

$$\wedge\ \mathit{input}(\mathbf{A}:\mathbf{Planner}))|\mathit{receives}(\mathbf{A}:\mathbf{Planner},\ \mathit{planning\_update\_request}(\mathbf{B}:\mathbf{Flight}))$$

$$\twoheadrightarrow_{0,0,1,1}\mathit{output}(\mathbf{A}:\mathbf{Planner})|\mathit{performed}(\mathbf{A}:\mathbf{Planner},\ \mathit{planning\_update}(\mathbf{B}:\mathbf{Flight}))$$

$$\mathit{output}(\mathbf{A}:\mathbf{Planner})|\mathit{performed}(\mathbf{A}:\mathbf{Planner},\ \mathit{planning\_update}(\mathbf{B}:\mathbf{Flight}))$$

$$\twoheadrightarrow_{0,0,1,1}\mathit{output}(\mathbf{A}:\mathbf{Planner})|\mathit{communicated}(\mathbf{A}:\mathbf{Planner},\ \mathbf{B}:\mathbf{Atc},\ \mathit{planning\_update\_for}(\mathbf{C}:\mathbf{Flight}))$$

$$\wedge\ \mathit{communicated}(\mathbf{A}:\mathbf{Planner},\ \mathbf{D}:\mathbf{Sup},\ \mathbf{planning\_update}))$$

6.5 Air Traffic Control Agent

The air traffic control agent is responsible for providing trajectory instructions to the flight agents that it observes at specific positions. Internal mechanisms of the air traffic control agent differ depending on the observation position of the agent. This section describes these mechanisms as well as the characteristics and perception capabilities of the agent.

6.5.1 Characteristics

The relevant characteristics for an air traffic control agent are limited to its observation position. This feature is denoted by $\mathit{observation\_position\_of}(\mathbf{A}:\mathbf{Atc},\ \mathbf{Position}:\mathbf{Pos})$. For an overview of the observation positions of the different air traffic control agent, the reader is referred to section 5.3.2.

6.5.2 Perception Capabilities

- **Flight agent**: An air traffic control agent can observe a flight agent when this flight agent has a position that is contained by the observation positions of this agent.
- **Planning update**: Planning updates by the planner agent, containing planning information, can be perceived by the air traffic control agent.

6.5.3 Actions and Cognitive Processes

The action mechanism of an air traffic control agent is triggered by the observation of a flight agent. An air traffic control agent has its dedicated observation positions. Once a flight agent reaches one of these observation positions, the air traffic control agent will perform a planning update request. This planning update request is communicated to the planner agent responsible for this flight.

$$\mathit{external}(\mathbf{A}:\mathbf{Atc})|\mathit{is\_at\_position}(\mathbf{B}:\mathbf{Flight},\ \mathbf{pos})\ \wedge\ \mathit{observation\_position\_of}(\mathbf{A}:\mathbf{Atc},\ \mathbf{pos})$$

$$\twoheadrightarrow_{0,0,1,1}\mathit{input}(\mathbf{A}:\mathbf{Atc})|\mathit{observed}(\mathbf{A}:\mathbf{Atc},\ \mathbf{B}:\mathbf{Flight})$$

$$\mathit{input}(\mathbf{A}:\mathbf{Atc})|\mathit{observes}(\mathbf{A}:\mathbf{Atc},\ \mathbf{B}:\mathbf{Flight})\ \&\ \mathit{internal}(\mathbf{A}:\mathbf{Atc})|\mathit{int}(\mathbf{A}:\mathbf{Atc},\ \mathbf{tick\_update})$$

$$\twoheadrightarrow_{0,0,1,1}\mathit{output}(\mathbf{A}:\mathbf{Atc})|\mathit{performed}(\mathbf{A}:\mathbf{Atc},\ \mathit{request}(\mathit{planning\_update\_for}(\mathbf{B}:\mathbf{Flight})))$$

$$\&\ \mathit{output}(\mathbf{A}:\mathbf{Atc})|\mathit{communicated}(\mathbf{A}:\mathbf{Atc},\ \mathbf{C}:\mathbf{Planner},\ \mathit{planning\_update\_request}(\mathbf{B}:\mathbf{Flight}))$$

Using the up-to-date planning information for the flight, the air traffic control agent will determine the trajectory update required for the flight. The trajectory update procedure differs depending on the observation positions of an air traffic control agent. These update procedures are described algorithmically in action mechanism 1.

$$\mathit{output}(\mathbf{A}:\mathbf{Atc})|\mathit{communicated}(\mathbf{B}:\mathbf{Planner},\ \mathbf{A}:\mathbf{Atc},\ \mathit{planning\_update\_for}(\mathbf{C}:\mathbf{Flight}))$$

$$\&\ \mathit{internal}(\mathbf{A}:\mathbf{Atc})|\mathit{receives}(\mathbf{A}:\mathbf{Atc},\ \mathit{planning\_update\_for}(\mathbf{B}:\mathbf{Flight}))$$

$$\twoheadrightarrow_{0,0,1,1}\mathit{output}(\mathbf{A}:\mathbf{Atc})|\mathit{performed}(\mathbf{A}:\mathbf{Atc},\ \mathit{update\_trajectory}(\mathbf{B}:\mathbf{Flight},\ \mathit{trajectory}(\mathbf{B}:\mathbf{Flight},\ \langle\mathbf{pos}_{start},\ \mathbf{t}_{start}\rangle,\langle\mathbf{pos}_{end},\ \mathbf{t}_{end}\rangle)))$$

For an air traffic control agent that observes positions along the FIR boundary, the update procedure is determined by comparing the EAT received in the planning update and the earliest and latest possible arrival times at the IAF. If it is deemed possible to arrive at the IAF, at the planned EAT time it will receive a trajectory with an end time that corresponds to this EAT. The planned EAT will never be earlier than the earliest possible time because this is constrained in the planning algorithm of the planner agent. It could, however, be that a flight has to be delayed to arrive at a later point in time than the latest arrival time at the IAF. If this is the case, the trajectory will be updated using the latest possible time at the IAF.

For an air traffic control agent with an observation position at a holding stack, a similar procedure is followed. In this case, the instruction is dependent on the earliest and latest arrival time at the runway. If the flight is scheduled to arrive at the runway at a later point in time than the received slot in the planning update, it will receive a holding time. To effectuate this holding time, the time for the first trajectory point will be set to the sum of the arrival time at the IAF and the holding time.

In case of an air traffic control agent with an observation position at the apron area, the update procedure is determined by whether the apron area is blocked or not. An apron area can be blocked by a flight agent of that is of type arrival as well as by a flight agent of type departure. If an apron area is blocked the second position-time combination will be set to the current position and the next tick time. For a situation in which the apron is not blocked the position-time combination of the trajectory will be set to the position of the runway and a time that is equal to the sum of the current time and the taxi time between the apron area and the runway.

```
Action Mechanism 1: $update\_trajectory(\mathbf{B}:\mathbf{Flight}, trajectory(\mathbf{B}:\mathbf{Flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle))$
FIR boundary procedure:
Compute earliest and latest arrival times at the IAF, $earliest\_at\_IAF$ and $latest\_at\_IAF$, using trajectory prediction (see appendix C)
if $earliest\_at\_IAF \leq EAT \leq latest\_at\_IAF$ then
    $trajectory(\mathbf{flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle) \leftarrow trajectory(\mathbf{flight}, \langle \mathbf{pos}_{FIR}, \mathbf{t}\rangle, \langle \mathbf{pos}_{IAF}, \mathbf{EAT}\rangle)$
else
    $trajectory(\mathbf{flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle) \leftarrow trajectory(\mathbf{flight}, \langle \mathbf{pos}_{FIR}, \mathbf{t}\rangle, \langle \mathbf{pos}_{IAF}, \mathbf{latest\_at\_IAF}\rangle)$
end if
Holding stack procedure:
Compute earliest and latest arrival times at the runway, $earliest\_at\_RWY$ and $latest\_at\_RWY$, using trajectory prediction and determine the taxi time (see appendix C)
if $earliest\_at\_RWY \leq slot \leq latest\_at\_RWY$ then
    $time\_at\_RWY \leftarrow slot$
    $time\_at\_apron \leftarrow time\_at\_RWY + taxi\_time$
    $trajectory(\mathbf{flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle) \leftarrow trajectory(\mathbf{flight}, \langle \mathbf{pos}_{IAF}, \mathbf{t}\rangle, \langle \mathbf{pos}_{apron}, \mathbf{time\_at\_apron}\rangle)$
else
    $holding\_time \leftarrow slot - latest\_at\_RWY$
    $time\_at\_RWY \leftarrow slot$
    $time\_at\_apron \leftarrow time\_at\_RWY + taxi\_time$
    $trajectory(\mathbf{flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle) \leftarrow trajectory(\mathbf{flight},$
            $\langle \mathbf{pos}_{IAF}, \mathbf{t} + \mathbf{holding\_time}\rangle, \langle \mathbf{pos}_{apron}, \mathbf{time\_at\_apron}\rangle)$
end if
Apron procedure:
Determine the taxi time ($taxi\_time$) for the flight (see appendix E)
if $apron\_status \neq blocked$ then
    $trajectory(\mathbf{flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle) \leftarrow trajectory(\mathbf{flight}, \langle \mathbf{pos}_{apron}, \mathbf{t}\rangle, \langle \mathbf{pos}_{rwy}, \mathbf{t} + \mathbf{taxi\_time}\rangle)$
else
    $trajectory(\mathbf{flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle) \leftarrow trajectory(\mathbf{flight}, \langle \mathbf{pos}_{apron}, \mathbf{t}\rangle, \langle \mathbf{pos}_{apron}, \mathbf{t} + \mathbf{t}_{tick}\rangle)$
end if
```

After performing the trajectory update for a flight agent, a new trajectory is available for that agent. This trajectory update is then communicated to the flight agent such that it can update its position accordingly. The trajectory information determined by the air traffic control agent is communicated to the planner agent as well. This allows the planner agent to use this information in future planning updates.

$$
\begin{aligned}
& output(\mathbf{A}:\mathbf{Atc})|performed(\mathbf{A}:\mathbf{Atc}, update\_trajectory(\mathbf{B}:\mathbf{Flight}, trajectory(\mathbf{B}:\mathbf{Flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle))) \\
& \twoheadrightarrow_{0,0,1,1} output(\mathbf{A}:\mathbf{Atc})|communicated(\mathbf{A}:\mathbf{Atc}, \mathbf{B}:\mathbf{Flight}, trajectory(\mathbf{B}:\mathbf{Flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle)) \\
& \wedge\ communicated(\mathbf{A}:\mathbf{Atc}, \mathbf{C}:\mathbf{Planner}, trajectory(\mathbf{B}:\mathbf{Flight}, \langle \mathbf{pos}_{start}, \mathbf{t}_{start}\rangle, \langle \mathbf{pos}_{end}, \mathbf{t}_{end}\rangle))
\end{aligned}
$$

6.6 Supervisor Agent

The supervisor agent supervises either the inbound or outbound system. This agent uses an anticipation mechanism, a negotiation mechanism, and reasoning mechanisms to determine when to activate or deactivate a runway for the system that it supervises. This section describes the characteristics, perception capabilities, actions, and cognitive processes that together form the supervisor agent.

6.6.1 Characteristics

For a supervisor agent, the responsibility feature is the only relevant characteristic. This responsibility dictates which system the supervisor agent is responsible for and is denoted by $responsible\_for(\mathbf{A}:\mathbf{Sup},\mathbf{System}:[\mathbf{Inbound},\mathbf{Outbound}])$. This responsibility is used to determine which system must be observed and which runway control and negotiation actions are permitted for the supervisor agent.

6.6.2 Perception Capabilities

- **Global clock**: A supervisor agent can observe the global clock to obtain an update for the global current time property. Using this current time information, the agent can interpret whether the current time property of the tick must be updated or not.

- **Planning**: The planner agent will update the global state that the environment holds with the updated planning information at the end of every tick. This planning information can be perceived by the supervisor agent. From this planning information, a supervisor agent can perceive the delays for flights in the planning horizon. This information can lead to a mental state in which the supervisor is aware of a weak signal for an upcoming undesirable state for the system that it supervises.

- **Protocol**: The supervisor agent can observe the protocol to determine which locutions are allowed and what information must be updated to commence the negotiation interaction or participate in the negotiation interaction.

- **Runway configuration**: The supervisor agent can observe the status of all runways. This information can be interpreted as the current configuration. Current configuration information is used to determine the allowed locution in the protocol.

6.6.3 Actions and Cognitive Processes

The actions and cognitive processes are aimed towards finding a satisficing solution for the selection of a runway reconfiguration. To reach this satisficing solution the supervisor agent has anticipatory capabilities, negotiation capabilities and reasoning capabilities. The internal mechanisms facilitating these capabilities are described in this section.

Anticipatory Mechanism

The anticipatory capabilities are facilitated by the cognitive architecture for an anticipatory agent introduced in section 3.2.2, originating from the work of Blok et al. [12]. This introduces the five steps required for anticipatory behaviour. These steps are adjusted for the supervisor agent to:

1. Monitoring the environment and detecting weak and strong signals for an expected undesirable state of the system. Both a weak and a strong signal act as a trigger for predicting the effects of the observed signal. If a weak signal leads to an observed undesirable state in the prediction results, it is considered a strong signal.

2. If a speech act is allowed by the protocol, generating applicable action options to avoid an expected undesired state that is detected as a strong signal. These action options are in the form of a flight option for either activating or deactivating a runway.

3. Generation of the consequences of activating or deactivating a runway for a certain flight option. This is done using a forward simulation of the domain model that the supervisor agent is responsible for, i.e. the inbound or outbound model.

4. Valuation of the options for activation or deactivation of a runway, by interpreting the consequences observed in the forward simulation.

5. Selecting which option to propose to the opponent supervisor agent, using an internal reasoning mechanism.

A supervisor agent's anticipatory mechanism is activated by an interpretation of a tick update. Given a tick update, the supervisor agent will check whether a monitoring action is required or not. This is only the case when the supervisor agent has not performed a monitoring process or if a monitoring interval has passed. If a monitoring procedure is required, this procedure will be performed and the last monitoring time is updated and will hold for the monitor interval time:

$$\mathit{external}(\mathbf{A}:\mathbf{Sup})|\mathit{current\_time}(\mathbf{global\_clock}, \mathbf{t}) \wedge \mathit{monitor\_interval}(\mathbf{global\_clock}, \mathbf{t}_{interval})$$
$$\& \; \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathit{int}(\mathbf{tick\_update}) \wedge [\neg \mathit{last\_monitor\_time}(\mathbf{t}_{monitor}) \vee \mathbf{t} - \mathbf{t}_{monitor} \geq \mathbf{t}_{interval}]$$
$$\twoheadrightarrow_{0,0,1,\mathbf{t}_{interval}} \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathit{last\_monitor\_time}(\mathbf{t})$$

$$\mathit{external}(\mathbf{A}:\mathbf{Sup})|\mathit{current\_time}(\mathbf{global\_clock}, \mathbf{t}) \wedge \mathit{monitor\_interval}(\mathbf{global\_clock}, \mathbf{t}_{interval})$$
$$\& \; \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathit{int}(\mathbf{tick\_update}) \wedge [\neg \mathit{last\_monitor\_time}(\mathbf{t}_{monitor}) \vee \mathbf{t} - \mathbf{t}_{monitor} \geq \mathbf{t}_{interval}]$$
$$\twoheadrightarrow_{0,0,1,1} \mathit{output}(\mathbf{A}:\mathbf{Sup})|\mathit{performed}(\mathit{monitor}(\mathbf{planning}, \mathbf{System}{:}[\mathbf{Arr, Dep}], \mathbf{t}, \mathbf{t} + \mathbf{t}_{horizon}))$$

During the monitoring process, the current planning for the system that the supervisor agent is responsible for is observed and interpreted. Provided that a different demand peak configuration is scheduled than the current demand peak configuration in the monitoring time horizon, the supervisor agent detects a strong signal of an undesirable state coming up.

If the supervisor agent observes an individual flight delay in the planning that is higher than the threshold d, the supervisor agent will regard this as a weak signal for an upcoming undesirable state. For an internal state which dictates that a weak signal of an undesirable state holds, the supervisor agent will perform a mental simulation to predict whether the weak signal will lead to an actual undesirable state or not. This mental simulation is described by the combination of functions $\mathit{leadsto}(\mathbf{Sensory\_state}{:}\mathbf{S}, \mathbf{Response\_state}{:}\mathbf{R})$ and $\mathit{leadsto}(\mathbf{Response\_state}{:}\mathbf{R}, \mathbf{Sensory\_state}{:}\mathbf{S})$, originating from the work of Blok et al. [12]. These functions are based on Hesslow's simulated behavior and perception chains [34]. The process can be described for the supervisor agent as:

$$\mathit{internal}(\mathbf{A}:\mathbf{Sup})|\neg \mathit{has\_strong\_signal}(\mathbf{undesirable\_state}, \mathbf{t}, \mathbf{t} + \mathbf{t}_{horizon}) \wedge \mathit{weak\_signal}(\mathbf{undesirable\_state}, \mathbf{t}, \mathbf{t} + \mathbf{t}_{horizon})$$
$$\wedge \; \mathit{leadsto}(\mathit{weak\_signal}(\mathbf{undesirable\_state}, \mathbf{t}, \mathbf{t} + \mathbf{t}_{horizon}), \mathbf{Response\_state}{:}\mathbf{R})$$
$$\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathbf{Reponse\_state}{:}\mathbf{R}$$

$$\mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathbf{Response\_state}{:}\mathbf{R} \wedge \mathit{leadsto}(\mathbf{Response\_state}{:}\mathbf{R}, \mathbf{Sensory\_state}{:}\mathbf{S})$$
$$\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathbf{Sensory\_state}{:}\mathbf{S}$$

$$\mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathbf{Sensory\_state}{:}\mathbf{S} \wedge \mathit{leadsto}(\mathbf{Sensory\_state}{:}\mathbf{S}, \mathbf{Response\_state}{:}\mathbf{R})$$
$$\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathbf{Response\_state}{:}\mathbf{R}$$

$$\mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathbf{Response\_state}{:}\mathbf{R} \wedge \mathit{undesirable\_state}(\mathbf{Response\_state}{:}\mathbf{R})$$
$$\twoheadrightarrow_{0,0,1,1} \mathit{internal}(\mathbf{A}:\mathbf{Sup})|\mathit{has\_strong\_signal}(\mathbf{undesirable\_state}, \mathbf{t}, \mathbf{t} + \mathbf{t}_{horizon})$$

If a supervisor agent has detected a strong signal for an undesirable state is coming up, it will have the goal to avoid this undesirable state. If it is allowed by the protocol (see appendix B) to utter a locution, the supervisor agent will generate applicable action options. This is done using the procedure described in section 5.4.3.

After generating options for the proposal content, the supervisor agent will generate the consequences for the action option. This consequence generation process is performed by a mental simulation procedure similar to the procedure described for evaluating cues. The only difference is that in the mental simulation, it is now simulated what the effect of an action option will be. The consequences of this action option, observed in the mental simulation are translated to a set of facts referred to as literals. Using these consequences of the forward simulation in the form of literals, it is reasoned which option should be selected to be entered into the proposal. This selection is done using the reasoning mechanism of the supervisor agent.

Reasoning Mechanism

The reasoning mechanism of the supervisor agent is described by example. This mechanism is fully based on the DeLP model and generalized specificity criterion, originating from the work of Garcia and Simari [31] and Stolzenburg et al. [65], respectively. The mechanism is aimed at reasoning what action option to select and this reasoning procedure is started by constructing a DeLP programme.

In the reasoning mechanism, a DeLP programme is used to compute the answer to a query predicate, which in the case of the supervisor agent will be a query on which action option to select or if a proposal should be accepted. The programme is constructed by defining non-defeasible and/or defeasible information that can be used for this purpose. Defining the non-feasible information, comes down to defining the strict rules and literals that apply to the current reasoning scenario. Combined with the defeasible information in the form of defeasible rules, this results in a DeLP programme. An example programme, $\mathcal{P}_{EXAMPLE}(\Pi, \Delta)$, can be described as follows:

$$\Pi = \left\{ \begin{array}{l} \sim propose(X) \leftarrow newRunwayUnavailable(X) \\ highIndividualDelay(klm123) \\ lowTotalDelay(klm123) \end{array} \right\}$$

$$\Delta = \left\{ \begin{array}{l} propose(X) \mathrel{-\!\!\prec} \sim undesirableOption(X) \\ \sim propose(X) \mathrel{-\!\!\prec} undesirableOption(X) \\ \sim undesirableOption(X) \mathrel{-\!\!\prec} lowIndividualDelay(X) \\ undesirableOption(X) \mathrel{-\!\!\prec} highIndividualDelay(X) \\ \sim undesirableOption(X) \mathrel{-\!\!\prec} lowTotalDelay(X) \\ \sim undesirableOption(X) \mathrel{-\!\!\prec} lowTotalDelay(X), highIndividualDelay(X) \end{array} \right\}$$

The non-defeasible part ,Π, of this $\mathcal{P}_{EXAMPLE}$ consists of one strict rule and two positive literals. The strict rule is represented in the form of a schematic rule, making it independent of the reasoning scenario. To distinguish free variables from fixed terms in schematic rules, they are denoted using an upper-case letter. Although this is a convenient way of expressing rules, schematic rules must be ground (have only fixed terms) to be used in the logic programming computations. This example rule, being a non-defeasible rule, dictates that a flight option for the moment of a runway reconfiguration can never be proposed if the "new runway" of a to be used configuration is unavailable. This is an example of a rule that adds expert knowledge to the reasoning mechanism. For the supervisor agents all rules adding expert knowledge to the reasoning mechansim are defined in appendix A.

Information in the form of literals in the set Π originates from sensory information. For the definition of the strict and defeasible rules, that relate literals to conclusions in the reasoning mechanism, it is desired to use literals that have a higher level of expressiveness than simple metrics. Rather than a simple metric like $delay(klm123, 1)$, denoting an individual maximum delay of 1 minute when performing a reconfiguration using flight klm123, a more expressive literal like $lowDelay(klm123)$ is desired. This desire stems from the fact that using less expressive literals like $delay(klm123, 1)$, would make it hard to interpret rules.

Generating a more expressive literal like $lowDelay(klm123)$, could be achieved by adding new strict schematic rules in combinations with numerical comparisons like $lowDelay(X) \leftarrow delay(X, d), d < 4$. Another common practice is to make use of a literal reification design pattern [56]. Literal reification allows literals to participate in numerous relations and enables an increased level of expressiveness. In the reification procedure, reified literals are defined as instances of a type. The only downside to this practice is the decreased performance introduced by this method. As the normal DeLP framework does not provide the ability to introduce numerical comparisons and performance is not the main focus, it is decided to use a design reification pattern to increase the expressiveness of the literals used. This reification pattern is defined in table 6.1. Notice that for explainability purposes, terms other than in the reification pattern are used in the reasoning example.

Table 6.1: Reification pattern

Ontological entity	Literal values
NoHoldingPatterns	**maxIndividualDelay**$(\leq 4min)$
HoldingPatterns	**maxIndividualDelay**$(> 4min)$
LowTotalDelay	$sum(\mathbf{delay}_i, \ldots, \mathbf{delay}_n) \leq 30min$
HighTotalDelay	$sum(\mathbf{delay}_i, \ldots, \mathbf{delay}_n) > 30min$
AcceptableFourthRunwayTime	$(\mathbf{t}_{deactivation_i} - \mathbf{t}_{activation_i}) \leq 30min$
UnacceptableFourthRunwayTime	$(\mathbf{t}_{deactivation_i} - \mathbf{t}_{activation_i}) > 30min$
BreakingBunch	**IAF_PreviousFlight**$(ARTIP) \wedge$ **IAF_ActionOption**$(ARTIP)$
NotBreakingBunch	$\neg$(**IAF_PreviousFlight**$(ARTIP) \wedge$ **IAF_ActionOption**$(ARTIP)$)
SlotsMissed	**maxIndividualDelay**$(> 10min)$
NoSlotsMissed	**maxIndividualDelay**$(\leq 10min)$
BlockingPreviousFlight	**Sector_PreviousFlight** $=$ **Sector_ActionOption**
NotBlockingPreviousFlight	**Sector_PreviousFlight** $\neq$ **Sector_ActionOption**
HigherTotalDelay	$\mathbf{totalDelay}_i > \mathbf{totalDelay}_j$
LowerTotalDelay	$\mathbf{totalDelay}_i < \mathbf{totalDelay}_j$

The subset Δ consists of the defeasible part of the DeLP programme. Defeasible rules in this subset are like the strict rules also defined as schematic rules in the logic programme. In order for all the rules to be used in the reasoning mechanism, both strict and defeasible rules should be ground. Using the set of schematic rules R and the fixed terms existing in literals, the set of ground instances for this rule $Ground(R)$ can be derived. Using these ground instances for each schematic rule, the ground DeLP programme can be derived by the following relation[48]:

$$Ground(\mathcal{P}) = \bigcup_{R \in \mathcal{P}} Ground(R)$$

Using this relation, the ground programme for $\mathcal{P}_{EXAMPLE}$ can be derived as:

$$Ground(\mathcal{P}_{EXAMPLE}) = \left\{ \begin{array}{l} \sim propose(klm123) \leftarrow newRunwayUnavailable(klm123) \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} lowIndividualDelay(klm123) \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} lowTotalDelay(klm123) \\ \sim propose(klm123) \mathrel{-\!\!\prec} undesirableOption(klm123) \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} highIndividualDelay(klm123), lowTotalDelay(klm123) \end{array} \right\}$$

In order to establish the answer to a predicate, a check is done if a defeasible derivation L can be established from $\mathcal{P}$, denoted by $\mathcal{P} \mid\!\sim L$. Such a defeasible derivation L consists of a sequence of literals. Each literal L_i is either in this sequence because it is a fact in Π or because a rule exists in $\mathcal{P}$ that has a head L_i and a body with literals contained in the sequence at a position before L_i. For the query predicate $propose(klm123)$, the defeasible derivations $derivation_1$ and $derivation_2$ can be established from $\mathcal{P}_{EXAMPLE}$.

$$derivation_1 = \left[\begin{array}{l} propose(klm123) \mathrel{-\!\!\prec} \sim undesirableOption(klm123), \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} highIndividualDelay(klm123), lowTotalDelay(klm123), \\ lowTotalDelay(klm123), \\ highIndividualDelay(klm123) \end{array} \right]$$

$$derivation_2 = \left[\begin{array}{l} propose(klm123) \mathrel{-\!\!\prec} \sim undesirableOption(klm123), \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} lowTotalDelay(klm123), \\ lowTotalDelay(klm123) \end{array} \right]$$

Although contradictory information may exist in the defeasible part of $\mathcal{P}$, the non-defeasible part Δ must be non-contradictory such that no derivation for a complimentary pair of literals from Δ exists. Furthermore, for each of the established derivations, a consistency check is performed that examines if all rules that exist in a defeasible derivation $\mathcal{P} \mid\!\sim L$ are consistent with the strict part Π of the DeLP programme. Only when this consistency check is passed, a defeasible derivation is accepted.

The notion of an argument structure $\langle \mathcal{A}, h \rangle$ is defined as being a minimal non-contradictory set of defeasible rules $\mathcal{A}$, that is the result of a defeasible derivation for a literal h [63]. Apart from an existing defeasible derivation from $\mathcal{A}$ to h and $\mathcal{A}$ being consistent with Π, it should therefore also be checked if $\mathcal{A}$ is minimal. In order to check if $\mathcal{A}$ is minimal, a sub-argument check is performed. The sub-argument checks if there exists an argument structure $\langle \mathcal{B}, h \rangle$ that satisfies the other conditions and satisfies $\mathcal{B} \subseteq \mathcal{A}$. If this is the case, then $\mathcal{A}$ is not minimal and $\langle \mathcal{A}, h \rangle$ is not a valid argument structure.

After performing the outlined checks for the example derivations $derivation_1$ and $derivation_2$, $\mathcal{A}_1$ and $\mathcal{A}_2$ can be established as valid argument structures for the predicate in the query $propose(klm123)$. To determine whether this argument structure supports the answer for the query, it is required to check if other arguments can be established that disagree with it. An argument is in disagreement with another argument when the literals of that make up the head of the arguments are in disagreement. Two literals h and h_1 are in disagreement if the set $\Pi \cup \{h, h_1\}$ is contradictory.

$$\mathcal{A}_1 = \left\{ \begin{array}{l} propose(klm123) \mathrel{-\!\!\prec} \sim undesirableOption(klm123), \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} highIndividualDelay(klm123), lowTotalDelay(klm123) \end{array} \right\}$$

$$\mathcal{A}_2 = \left\{ \begin{array}{l} propose(klm123) \mathrel{-\!\!\prec} \sim undesirableOption(klm123), \\ \sim undesirableOption(klm123) \mathrel{-\!\!\prec} lowTotalDelay(klm123) \end{array} \right\}$$

To check whether counter-arguments can be found that could defeat the established arguments, requires searching for attack opportunities. Attack opportunities are literals that are in disagreement with any of the sub-arguments of the established arguments. For the example arguments $\mathcal{A}_1$ and $\mathcal{A}_2$, the attack opportunities are made up by the set $\{\sim propose(klm123), undesirableOption(klm123)\}$. These literals in the attack opportunity set are then used to generate the argument structures, following the same procedure that was used for generating the arguments for the predicate of the query. For the example DeLP programme this leads to the arguments $\mathcal{A}_3$ and $\mathcal{A}_4$.

$$\mathcal{A}_3 = \left\{ \begin{array}{l} \sim propose(klm123) \prec undesirableOption(klm123), \\ undesirableOption(klm123) \prec highIndividualDelay(klm123) \end{array} \right\}$$

$$\mathcal{A}_4 = \left\{ \; undesirableOption(klm123) \prec highIndividualDelay(klm123) \; \right\}$$

Given the arguments that can be constructed that support the query predicate and the counter-arguments that can be constructed for them, a warranting procedure is required to establish if an argument exists that supports the query and is not defeated. An argument $\langle \mathcal{A}_1, h_1 \rangle$ that supports the query predicate can be attacked directly by an argument $\langle \mathcal{A}_2, h_2 \rangle$ for literal h_2 which is contradicting with $\Pi \cup \{h_1\}$. An attack could also be indirect when one of the sub-arguments of $\langle \mathcal{A}_1, h_1 \rangle$ is contradicting with $\Pi \cup \{h_2\}$. To decide which argument prevails in such a situation, a criterion is required.

The criterion used is the generalized specifity criterion which states that an argument $\langle \mathcal{A}_1, h_1 \rangle$ is preferred to $\langle \mathcal{A}_2, h_2 \rangle$ ($\langle \mathcal{A}_1, h_1 \rangle \succ \langle \mathcal{A}_2, h_2 \rangle$) if the following conditions are true[65]:

1. $\forall\ H \subseteq \mathcal{F}$: if $\Pi_G \cup H \cup \mathcal{A}_1 \mathrel{|\!\!\sim} h_1$ and $\Pi_G \cup H \not\mathrel{|\!\!\sim} h_1$, then $\Pi_G \cup H \cup \mathcal{A}_2 \mathrel{|\!\!\sim} h_2$, and
2. $\exists\ H' \subseteq \mathcal{F}$ such that $\Pi_G \cup H' \cup \mathcal{A}_2 \mathrel{|\!\!\sim} h_2$ and $\Pi_G \cup H' \not\mathrel{|\!\!\sim} h_2$ and $\Pi_G \cup H' \cup \mathcal{A}_1 \not\mathrel{|\!\!\sim} h_1$

In these conditions $\mathcal{F}$ is the set of all literals that can be derived from a DeLP programme, Π_G is the set of strict rules in Π and H is the set of literals that activates an argument. The term activates, comes from the fact that with $\Pi_G \cup \mathcal{A}_1$ it is not possible to have a defeasible derivation because there are no facts. If H is added, facts are added and a defeasible derivation becomes possible.

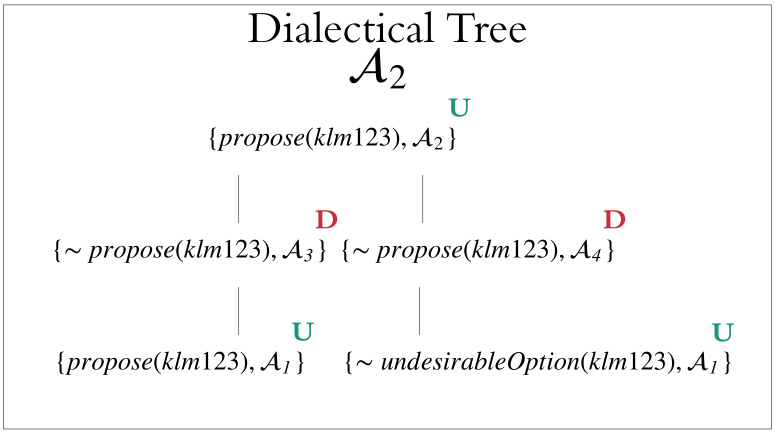


Figure 6.1: Warranting procedure

The first condition in the generalized specifity criterion dictates that every activation set that activates $\langle \mathcal{A}_1, h_1 \rangle$ must also activate $\langle \mathcal{A}_2, h_2 \rangle$. In the second condition it is stated that there must exist an activation set H' that activates $\langle \mathcal{A}_2, h_2 \rangle$, but does not activate $\langle \mathcal{A}_1, h_1 \rangle$. In the example $\mathcal{A}_3$ is defeated by $\mathcal{A}_1$, because every activation set in $\mathcal{A}_1$, $H = \{highIndividualDelay(klm123), lowTotalDelay(klm123)\}$, also activates $\mathcal{A}_3$ and the activation set $H' = highIndividualDelay(klm123)$ does not activate $\mathcal{A}_1$. Intuitively this comes down to the fact that arguments that have more information or are more direct, are preferred. By more direct it is meant that less rules are used to reach a conclusion.

After searching for defeaters and applying the generalized specificity criterion, a dialectical tree can be established with arguments and its attackers. An example for argument $\mathcal{A}_2$ is provided in figure 6.1. In this example, it can be observed that $\mathcal{A}_2$ is attacked by $\mathcal{A}_3$ and $\mathcal{A}_4$. When not considering the attackers of $\mathcal{A}_3$ and $\mathcal{A}_4$, $\mathcal{A}_2$ would be defeated. However, because $\mathcal{A}_3$ and $\mathcal{A}_4$ are defeated by $\mathcal{A}_1$, both attackers of $\mathcal{A}_2$ are also defeated. Using the warranting procedure it can now be established that $\mathcal{A}_2$ is not defeated because all defeaters of $\mathcal{A}_2$ are in turn defeated themselves. This means that the predicate $propose(klm123)$ is warranted by argument $\mathcal{A}_2$ and KLM123 should thus be proposed.

Negotiation Mechanism

The negotiation mechanism is formed by the negotiation protocol, semantic actions, and the reasoning mechanism of the agent. The protocol describes which locutions may be uttered and what information must be updated to participate in the negotiation. This information is in the form of rules that represent expert knowledge. Which supervisor agent takes on the role of initiator in the negotiation interaction is based on the current runway configuration and the upcoming runway configuration in the peak schedule. The exact pre-conditions under which a certain locution is allowed and which rules must be added to the knowledge base is defined in the protocol in appendix B. Depending on the locution that is allowed and the responsibility of the supervisor agent, certain expert knowledge is required. This required expert knowledge is captured by the schematic rules that are added. These rules are described for every locution in appendix A.

An example of an allowed locution resulting from the protocol is depicted in figure 6.2. This locution consists of a proposal and a placeholder for possible meta-information about the proposal. A proposal is formed by a proposal type and proposal content. The proposal content will consist of selected flight options for a runway control action. This proposal type will be used for semantic analysis to determine what actions should be performed to generate the proposal.

$$Location = \{proposal : \overbrace{ProposeArrActivationDepDeactivation}^{\text{Proposal type}}\overbrace{(ArrFlight, DepFlight)}^{\text{Proposal content}},\ meta\_information : [\ldots]\}$$

Figure 6.2: Structure of a locution

The steps for initiating a negotiation are described by the five steps defined in the anticipatory mechanism of the supervisor agent. These steps result in an utterance in the form of a locution. The proposal type in this locution results from the protocol and the proposal content results from the option generation and option selection actions. The generated literals during the reasoning steps in the option selection are added as meta-information in the locution. Upon receiving the uttered locution the negotiation mechanism can be described by a set of steps inspired by the internal mechanism for a ABN agent originating from the work of Rahwan et al. [58] (see section 3.2.2):

1. Interpret the incoming locution by adding the proposal content and the literals in meta-information to the knowledge base. Both the proposal content and literals resulting from reasoning steps of the opponent supervisor agent can now be used in the subsequent reasoning and analysis steps.

2. Consult the protocol on which locutions may be generated under the current conditions. This will always be a locution indicating accepting the proposal or a locution indicating a counter-proposal.

3. Perform a semantic analysis on the proposal to determine which actions are required to obtain additional information about the proposal required for evaluation.

4. Perform the semantic actions and add obtained information from these actions to the knowledge base.

5. Reason if the proposal can be accepted or if a counter-proposal is required by using the reasoning mechanism and all information contained in the knowledge base.

6. If the reasoning mechanism dictates that the proposal must be accepted, select the proposal content and allowed locution type for accepting the proposal. If the reasoning mechanism dictates that the proposal must be rejected, follow steps 2-5 of the anticipatory mechanism to select the proposal content for the counter-proposal. In this process, the meta-information received from the opponent supervisor agent can now be used to select a suitable option for both supervisor agents, this is effectuated by the schematic rules used in the reasoning step. If the supervisor has already generated action options during the negotiation interaction, step 2 is omitted and already generated actions will be used.

7. Construct the locution using the selected proposal type and content. Meta-information resulting from reasoning steps only needs to be added when the locution contains a counter-proposal.

8. Communicate the locution. If the communicated locution indicates accepting the proposal, perform the runway control action implied (if any) by the proposal type for the system under the responsibility of the supervisor.

The semantic actions described in step 4 and resulting from the semantic analysis in step can be described as:

- **Evaluate deactivation information**: This action will be performed if the locution type implies a deactivation action for the system that the supervisor agent is responsible for. In this evaluation, a forward simulation is performed to predict the result of deactivating the initial option. This information is used to evaluate the proposal and as information for generating alternative deactivation options. This process is introduced in section 5.4.3.

- **Evaluate option combination**: If the interpreted proposal content consists of two options, the supervisor agent will assess the option combination to evaluate the effect of applying this combination of options. These options will always be an activation option and a deactivation option. By performing a forward simulation, expected consequences for the option combination is assessed in terms of the parameters defined in table 6.1. This information is used in subsequent reasoning steps.

The negotiation interaction ends when a locution is uttered that indicates accepting the received proposal. For the supervisor agent, the reasoning rules are constructed in such a way that when no alternative options remain the proposal will be accepted. This ensures that the negotiation will not finish without an agreement.

7

Simulation Scenario

To explore the workings of the developed model, a simulation scenario is defined. This scenario is aimed at emulating the real-world conditions at AAS. Both the criteria for selecting the scenario and the scenario parameters are described in this chapter.

7.1 Selection of Simulation Scenario

For the simulation scenario, one day of traffic at AAS in 2018 is selected. The selection of this day for the scenario results from multiple reasons. Modelling the runway reconfiguration phenomenon is focused towards demand-based runway reconfigurations, it is therefore not of interest to have a wide variety of runway combinations. The selected day in 2018, shows northerly use with a high demand pattern. This is a highly common runway use scenario, in which the runway directions 06 and 36R are used for arrivals and the runway directions 36L and 36C are used for departures.

The runway combinations in northerly use, are limited to the combinations presented in figure 7.1. For modelling purposes, it is favourable to have this limited set of combinations. Apart from the limited set of runway combinations, the selected day shows consistent northerly wind directions, a maximum wind speed of 3 Beaufort, and no precipitation. As the focus of the model is not towards weather conditions, having moderate weather conditions is another desired property for the scenario. Lastly, the scenario consists of a day with 1528 movements, this is significantly higher than the average of 1368 movements in 2018. Given that the model is focused on demand-based reconfigurations, selecting this busy day with high demand fluctuations results in a suitable scenario to explore the model's workings.

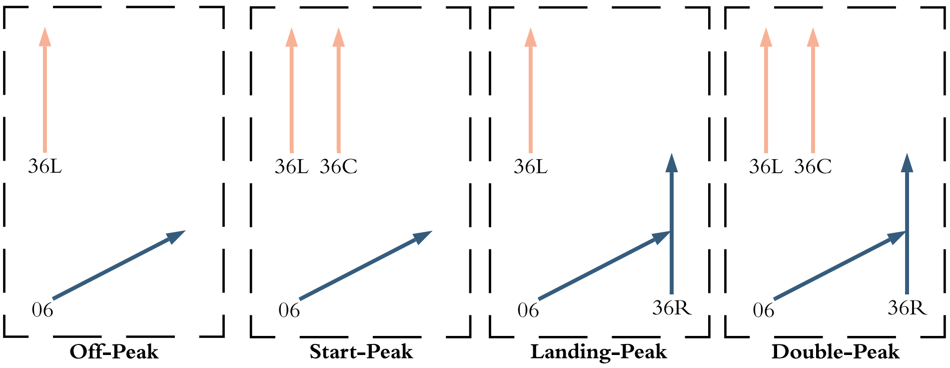


Figure 7.1: Runway combinations for northerly runway use at AAS

A scenario day starts when the first flight arrives that does not use a RNAV procedure. This is around 4:30 UTC because the daytime operation starts around this time. The scenario day ends after 20:00 UTC when the night operation commences. This time frame for the scenario is chosen because it eliminates the need for modelling routes for night flights and the use of runway combinations that must be used in the night. The day always starts with an off-peak, that is followed by an alternating pattern of start and landing peaks until the end of the scenario day. The alternation between start and landing peaks can be accompanied by an intermediate double peak.

7.2 Scenario Parameters

In order to run the simulation model, a flight schedule is required. The parameters for this flight schedule are derived from ADS-B data, that contains trajectory parameters and flight parameters for the flights of the selected scenario day. This ADS-B data is translated into schedules that contain the required parameters. The schedule parameters combined with weather parameters define the scenario for the simulation.

7.2.1 Scenario Schedule

Deriving the schedule parameters for the simulation day is done using an ADS-B data set, which when general aviation and helicopters are disregarded, covers 97 % of the movements of the selected day. This is deemed to be sufficient coverage to create a realistic scenario. The ADS-B data set contains the following elements:

- Timestamp in Unix time
- Callsign
- Registration
- Position as longitude/latitude point
- Ground speed in knots
- Indicated airspeed (IAS)
- Direction in degrees from the north

Using the position parameter and timestamps, the flight trajectory for arrival flights is recreated, see figure 7.2. This flight trajectory is used to determine the intersection with the boundary of the FIR and the Eligibility horizon. The intersection with the FIR boundary in combination with the ADS-B data parameters provides the entry angle towards the IAF, the entry altitude, the entry IAS, and the FIR entry time. The intersection with the eligibility horizon, which is an approximation of the MUAC coverage, is used to retrieve the time that arrival flights should enter the inbound planning. Next to these timestamps, it is determined for every flight which IAF is used, by using the recreated trajectory. The last parameter derived from the flight trajectory is the apron area a flight uses. This is found by using the spatial intersection of the apron areas, as defined in figure 7.3, and the flight trajectory.

For the departure flight schedule, a similar procedure is performed. A flight trajectory is firstly constructed from the ADS-B data, see figure 7.3. This trajectory is used to determine which apron area is used, the time the apron area is left (the off-block time), and which of the departure sectors is used. The approximated off-block time is used to determine the first time the departure flight should enter the planning by subtracting 40 minutes, which is referred to as the pre-planning horizon. The entirety of these processing steps for arrival and departure flights results in a schedule with the following parameters:

Arrival:

- Flight name
- Aircraft type
- Wake turbulence category
- IAF
- Entry time FIR
- IAS at FIR
- Altitude at FIR
- Entry position FIR
- Eligibility horizon time
- Apron area

Departure:

- Flight name
- Wake turbulence category
- Departure sector
- Off-block time
- Pre-planning horizon time
- Apron area

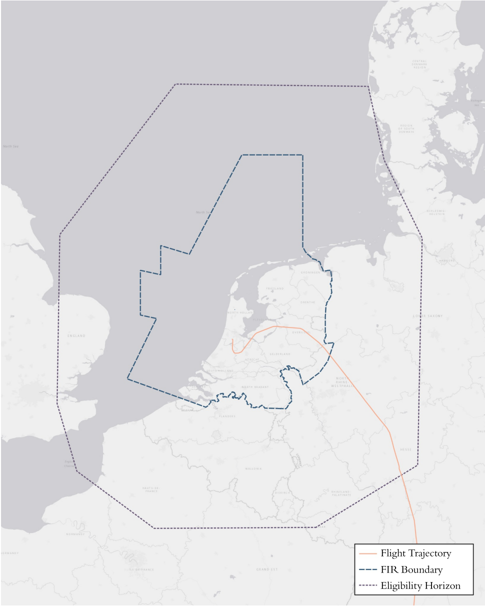


Figure 7.2: Schematic overview of the intersection points of an arrival trajectory

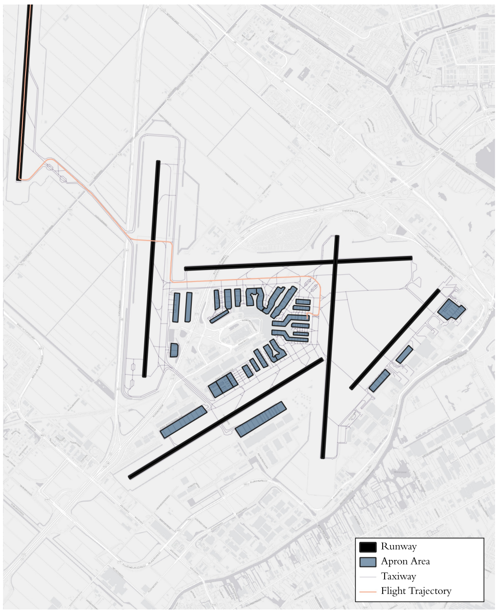


Figure 7.3: Schematic overview of Aerodrome elements and an example departure trajectory

7.2.2 Weather Scenario

Weather conditions for the scenario day are derived from Mode-S data that is provided by the Dutch meteorological institute. This data is obtained by flights that were flying on the day. Although the data does not have perfect coverage, the data is concentrated around the flight paths (see Fig. 7.4). The area around the flight paths is the most relevant area for the arrival model, making this the most credible data source for meteorological conditions at altitude. The values and method for deriving the values can be found in appendix E.

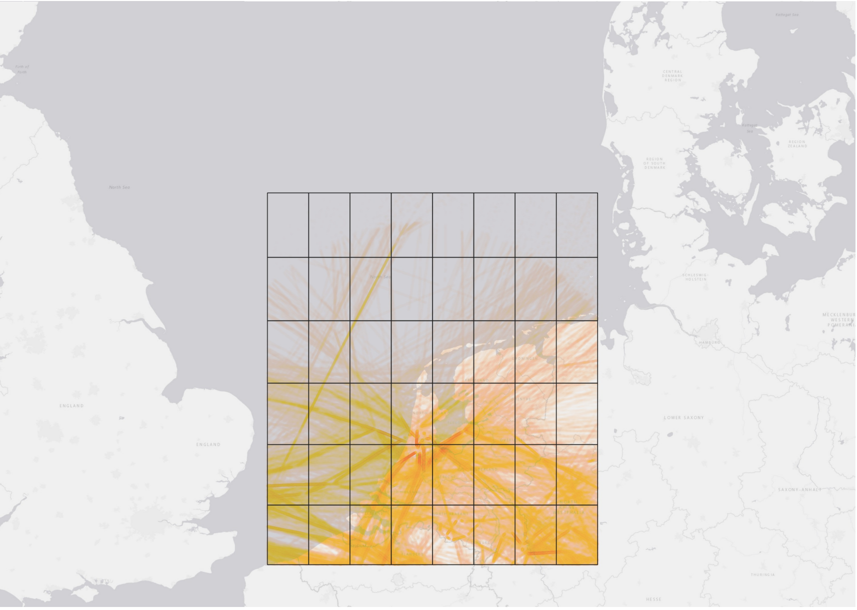


Figure 7.4: Distribution of the Mode-s meteorological data

7.2.3 Trajectory Distances and Taxi Times

The trajectory distances used for the arrival flight trajectories as well as the taxi times used for both arrival flights and departure flights are derived using ADS-B data. Both, the taxi times and trajectory distances are estimated by using statistical values observed in the historic data. The values and procedures for establishing these values are described in appendix E.

7.2.4 Aircraft Speeds

To determine aircraft speeds from the ADS-B data, the indicated airspeed values contained in the messages are used. A division is made for between speeds for the ACC part of the trajectory and speeds for the TMA part of the trajectory. For both parts, the values, as well as the methods for establishing them, are described in appendix E.

7.2.5 Start and Landing Intervals

The start intervals and landing intervals are important parameters influencing the capacity of the inbound and outbound model. For these separation intervals, values based on ICAO categories are normally defined as distances, as described in section 2.1.2. Because time values are required for planning computations, distance values are not suitable for the model. For this reason, time values are used.

The start and landing interval together with the aircraft mix determine the capacity of the inbound and outbound model. To set the interval values such that the capacity in the model reflects the actual capacity, a closer look is taken at the deviations for arrival times and departure times between the model output and the realisation. A mismatch in arrival and departure times using the same runway configurations indicates a mismatch in capacity between the model and actual operations. The interval values are therefore adapted to minimise this mismatch, thereby resembling the runway capacity in actual operations. This has led to the values for landing intervals and start intervals, presented in table 7.1 and table 7.2 respectively. The resulting mismatch in terms of the difference in landing and departure times using these intervals is presented in section 8.2.

Table 7.1: Values used for landing interval

In-front	Behind	Interval (sec)
H	H	104
H	M	140
M	H	114
M	M	80
S	M	180
S	H	180

Table 7.2: Values used for start interval

In-front	Behind	Interval (sec)
H	H	81
H	M	128
M	H	110
M	M	68
S	M	180
S	H	120

8

Validation and Results

To investigate the working of the model defined in chapter 6, the model is operationalised for the simulation scenario defined in chapter 7. As a first step to assessing the validity of the model an internal validation is performed, that focuses on the correct functioning of the model. To investigate how well the model corresponds with the runway reconfiguration phenomenon, the simulation results are compared with the runway configurations used in reality.

8.1 Internal Validation

Internal validation is aimed at verifying the correctness of the workings of the model. To do this, the desired working of the causal mechanisms are checked. This is done for all relevant subsystems in the inbound model, the outbound model, and the control model.

8.1.1 Inbound Model

The inbound model is constituted by multiple submodules that are required to function correctly such that the entire inbound model functions correctly. For each of these components, a description is provided on how the module is verified.

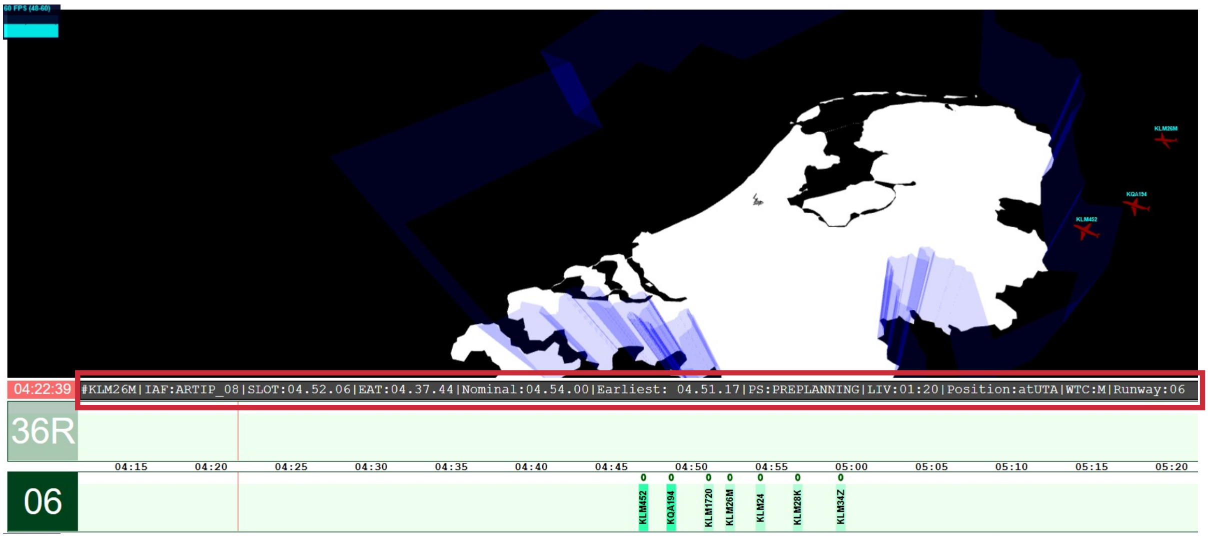
Trajectory Prediction Module

Trajectory prediction is a crucial element of the inbound model to determine the correct arrival times. In order for this module to function correctly, the formulas defined in appendix C need to be applied correctly. With the input speeds in calibrated airspeed, the conversion to true airspeed and the addition of windspeed elements is required to determine the ground speed required for trajectory time computation. To ensure correct computations for the speed conversion, the results of the computation in the model are compared to computations using an online tool [72]. For the same input conditions, the resulting values are verified to correspond with values produced by the trajectory prediction module.

A common mistake in working with wind conditions is using the inversed direction. To ensure that the wind component vector addition is performed correctly, a closer look is taken at the results of adding the wind component. Given that during the chosen simulation day the wind consistently originates from the north. Flights that arrive from the RIVER IAF and are therefore flying northward, should therefore on average see higher true airspeeds than ground speeds. By confirming this aspect, it was verified that the wind component addition is performed correctly in the model.

Verification of ATC Instructions

To verify the correct working of the instructions of ATC agents, the graphical user interface depicted in appendix G is used. The three-dimensional visualisation of the arrival flight agents allows for inspecting when a flight should receive instruction. This instruction can be observed by hovering above the flight in the planning, see figure 8.1. Once an instruction is provided, the attributes for the arrival flight that are displayed in the red container are updated. When room exists between flights in the planning displayed in the green bottom part of figure 8.1 and the earliest arrival time is earlier than the nominal time in the observed flight attributes, an ATC agent should speed up the flight agent. The reverse is true for flights that are observed to be delayed in the planning. By verifying that the correct flight paths are flown and that instructions are consistent with expectation, it is verified that the instructions of the ATC agents function correctly for the inbound model.



```
#KLM26M|IAF:ARTIP_08|SLOT:04.52.06|EAT:04.37.44|Nominal:04.54.00|Earliest: 04.51.17|PS:PREPLANNING|LIV:01:20|Position:atUTA|WTC:M|Runway:06
```

Figure 8.1: Interface used for verifying the ATC instructions

Inbound Planning Verification

To verify the inbound planning module, a visualisation of the planning is used. In the visualisation in figure 8.2, all flights in the planning are represented with coloured rectangles at the position of their current slot. From this visualisation, it can be verified that flights are spaced according to the landing interval that corresponds with the wake categories for the flights. Furthermore, it can be verified that the correct runway is assigned and that the correct order of flights is used.

The colour of the rectangle in figure 8.2 represents the planning status of the flight. This is important to verify that flights with a planned status are not reordered anymore, as opposed to the pre-planned flights. By continuously monitoring the dynamic visualisation, the functioning of the inbound planning module is verified.

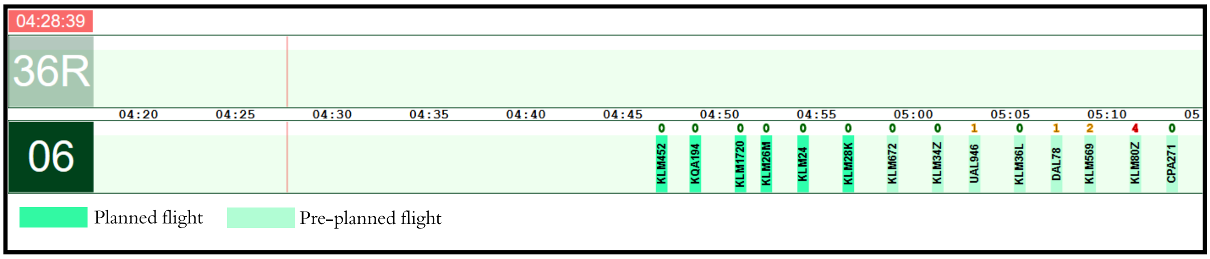


Figure 8.2: Visualisation used for verifying the inbound planning

8.1.2 Outbound Model

For the outbound model to function correctly, the correct instructions must be provided by the apron and runway controller ATC agents. These instructions are based on the outbound planning, which should therefore also be verified. Both these verification steps are done by visual inspection.

Verification of ATC Instructions

The ATC instructions consist of instructions at the apron for going off-block and instructions at the runway to ensure the desired separation. To check whether the right instructions are provided or not, a similar planning visualisation is used for the outbound model as for the inbound model. This visualisation is depicted in figure 8.3. From this visualisation, it is immediately clear when a flight goes off-block. By checking the planned departure time, in combination with the taxi time, it can be verified that flights receive off-block instructions at the correct time. It is furthermore verified, that the correct minimum separation is used at the runway by checking the start intervals in the visualisation.

Outbound Planning Verification

For the outbound planning verification, it is checked if the correct order is maintained, using the graphical interface displayed in figure 8.3. From this interface, it can be verified that the correct departure slots are assigned by

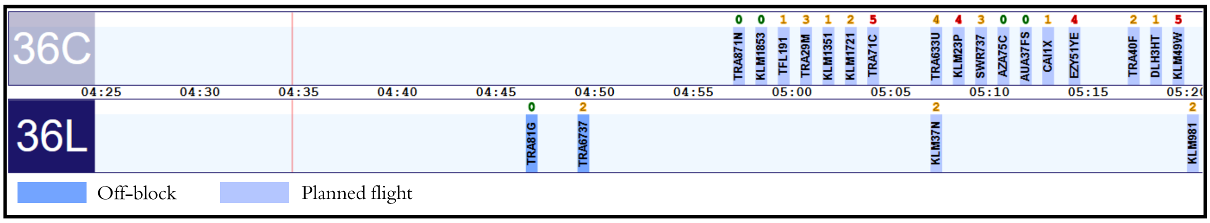


Figure 8.3: Visualisation used for verifying the outbound planning

checking the positions of the flights in the visualisation. It was furthermore verified that no instructions are provided and the sequence is maintained after the flight is off-block.

8.1.3 Control Model

The correct functioning of the control model is dependent on the correct functioning of the behaviour of the supervisor agent defined in section 6.6 in combination with an external element such as the sensory input and the negotiation protocol. To verify the working of the control model, the different aspects of the defined behaviour in combination with external elements are assessed.

Monitoring Action

The monitoring activities of the supervisor agent are aimed at recognising either weak or strong signals. A strong signal is observed when an upcoming peak is planned to occur within 40 minutes. This peak moment is visualised in the graphical user interface, with a vertical line at the moment of the peak in the peak schedule (see figure 8.4). Apart from the upcoming peak in the peak schedule, the delay values for all flights in the flight planning are visualised by the number of delay minutes above the flight name and a colour corresponding to the severity of the delay (see figure 8.5). A red value, corresponding to unacceptable delay, can be observed here as a weak signal. By checking if the model observes the weak and strong signals for the monitor horizon (40 minutes), it is verified that the monitoring function works correctly.

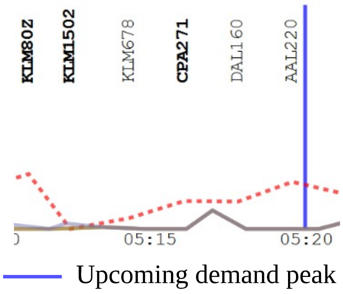


Figure 8.4: Visualisation of the upcoming demand peak

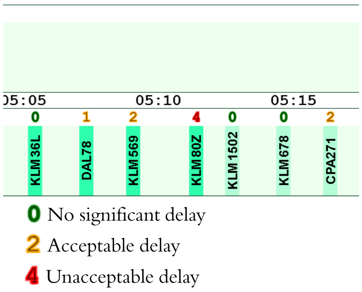


Figure 8.5: Visualisation of the delay values in the planning

Forward Simulation

Forward simulation of the model is a crucial aspect for sensory input used in the negotiation scenarios. It is therefore important to assess that the forward simulation functions correctly. Given that no random variables are used, a scenario that is simulated using the forward simulation module should return the same values as the real-time model for the same scenario. In order to verify the forward simulation model, the results of the forward simulation runs are compared with the results of the real-time model using the same initial state and scenario. By comparing the results and assuring that no discrepancies are present, the forward simulation model is verified.

Option Generation

To ensure that the right options are generated, this process is inspected visually. The option generation procedure is explained by example in section 5.4.3. To visually inspect if the correct options are selected, a visualisation of the expected times of arrival or departure for flights in combination with the observed cue is used. This visualisation is depicted in figure 8.4.

The horizontal position of the flight names corresponds with the moment in time that the flight is expected to land. It can be observed that these flights are either presented in bold or normal font. The bold font indicates that these flights can be scheduled for the to be activated runway. The cue in this figure is the vertical blue line indicating an upcoming different demand peak in the peak schedule. Using these visual elements it can be assessed if the group of bold flights before the cue is selected. By verifying that these options are reflected in the option selection procedure in the window depicted in figure 8.6, it is verified that the correct options are selected.

Argumentation

To verify the argumentation module is implemented correctly, the option and proposal evaluations are visualised. Using this visualisation it can be verified that the correct options are selected and the correct conclusions are drawn by both the APP and TWR supervisor agent. The visualisation is depicted in figure 8.6.

In the left upper part of the containers, the considered options for the performed queries are displayed in the form of buttons. By pressing the button, the warranted conclusion for this option is revealed. To make sure the correct argumentation line is used to arrive at the warranted conclusion, the arguments used are displayed when hovering over the warranted conclusion. In the example in figure 8.6, by clicking the button for option KLM1001, it can be observed why the combination of arrival activation flight KLM34Z and departure deactivation flight KLM1001 is not selected in the displayed argumentation line. This argumentation line should be interpreted as:

1. A better performance alternative exists in the form of flight BAW423 because this option results in a lower total delay and when selecting this flight, no slots will be missed.

2. KLM1001 is an eligible option because no slots are missed if this option is selected.

3. Given that KLM1001 is an eligible option but a better alternative is available in terms of performance in the form of flight option BAW423, option KLM1001 should not be selected to propose.

4. The conclusion that the combination of arrival activation flight KLM34Z and departure deactivation flight KLM1001 should not be proposed is warranted.

By continuously evaluating if the right conclusions are drawn, the argumentation module is verified.

```
!ProposeArrActivationDepDeactivation(klm34z,kl
m1001)-<EligibleOption(klm1001),PerformanceAlt
ernative(klm1001).,
EligibleOption(klm1001)-<NoSlotsMissed(klm100
1).,
PerformanceAlternative(klm1001)-<TotalDelayHig
her(klm1001,baw423),NoSlotsMissed(baw423).
```

Figure 8.6: Visualisation used for verifying the argumentation module

Dynamics of the Negotiation Interaction

The dynamics of the negotiation interaction are verified by continuously displaying which action is performed by each of the supervisor agents, during the simulation. Additionally, the states that hold at any point in time for the APP supervisor agent, the TWR supervisor agent, and the domain model are visualised as a simulation trace. This is done for specific negotiation procedures, to check if the correct sequence of states (trace) occurs. An example of a trace visualisation is provided in appendix F.

In figures F.1 and F.2 the states that hold or do not hold are depicted for the APP supervisor agent and the TWR supervisor agent, respectively. A state that holds is indicated as a dark blue bar, whereas a state that does not hold is indicated as a light blue bar. Using the same time axis as the figures for the supervisor agent traces, the observation information from the domain model for the supervisor agent is depicted in figure F.3. By analysing the order of the states that hold at different points in time, it is verified that the steps in the action mechanism of the supervisor agent are performed in the correct order.

8.2 Simulation Results

Whereas the internal validation provides insights into the correct functioning of the model, it does not provide any insights into whether the model corresponds well with the real-world phenomenon or not. In this section, a closer look is taken into this aspect, using external validation. This is done for the domain models as well as for the control model.

8.2.1 Validity of the Domain Model

To assess how well the different submodels correspond to the actual operations, the results from the simulation of the scenario are compared with the actual operations on the selected scenario day. To assess this for the inbound and outbound model, the times for different points of the arrival and departure trajectory are compared. For the control model, the times for runway reconfigurations are compared with the actual times of the runway reconfigurations.

Validity of the Inbound Model

For the inbound model, a validation run is performed using the times observed in the historic ADS-B data and the times observed in the simulation output. Apart from the times of arrival at the runway threshold, the times of arriving at the IAF, as well as the times for arriving at the FAF, are compared. To do so, the runway schedule observed in the realisation is used as an input for the model. The resulting deviation between actual times and simulation times is presented in table 8.1. For the three selected points of the arrival trajectory, the mean value (μ) and the standard deviation (σ) is provided.

Table 8.1: Sample mean and standard deviation for deviation of arrival times

	Time deviation at runway		Time deviation at FAF		Time deviation at IAF	
Runway	μ[sec]	σ[sec]	μ[sec]	σ[sec]	μ[sec]	σ[sec]
06	83	56	82	55	88	81
36R	62	50	63	52	63	67

The largest time deviations are observed for the times of arriving at the IAF. The reason for this larger deviation is that the time for the segment in the CTA is harder to model than the segment in the TMA. In the CTA segment, the difference in behaviour for flights from different airlines plays an important role. In the model, it is assumed that the indicated airspeed remains constant for the CTA segment. In reality, there can be several reasons, apart from ACC instructions, for changing this speed during this segment. Especially the cost index that airliners employ to determine the optimal speed in this segment, results in unpredictable speeds. This aspect is introduced in section 5.3.2.

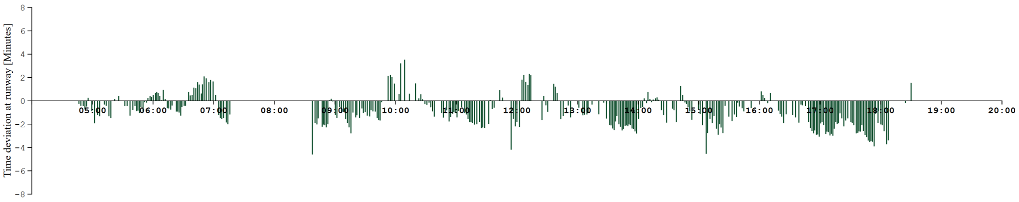


Figure 8.7: Deviation between actual arrival times and simulation arrival times for runway 06

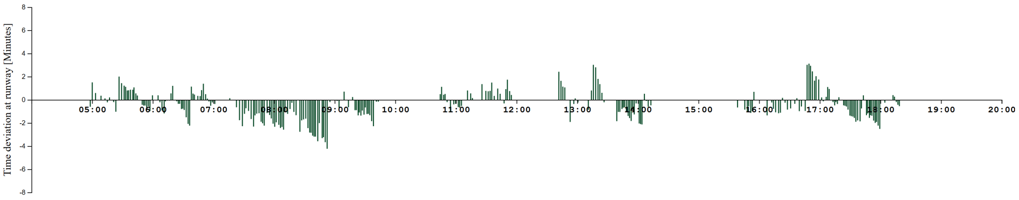


Figure 8.8: Deviation between actual arrival times and simulation arrival times for runway 36R

Time deviations for arriving at the FAF and the runway, are smaller compared to the deviations for arriving at the IAF. In these segments, the times are more restricted by the instructions provided by the ATC agents. In figures 8.7 and 8.8, the deviation from the actual times at the runway are displayed. It can be observed that the build-up of positive time deviations, as well as the build-up of negative time deviations, exist. A build-up of positive time deviations indicates that the runway capacity in the model was lower than in reality, whereas a build-up of negative time deviations indicates a higher runway capacity than in reality. Given that the deviations are distributed over positive and negative values reasonably equally, it is assessed that capacity parameters used for the entire day in the model are not exclusively too high or too low and resemble the real-world capacity well.

The time deviation figures in table 8.1 show a larger deviation for runway 06 than for runway 36R. Given that similar values for the time deviations are observed for both the time at the runway and the time at the IAF, the deviations are not the result of an incorrect estimate for the flight times of the final approach segment and the segment in the TMA. These deviations must, therefore, be the result of either incorrectly assumed speeds in the CTA or the runway capacity. For both runways, the same capacity is assumed. In reality, however, differences in capacity exist for different runways in different configurations. From the mean deviation figures, it can be concluded that the selected capacity parameters (land intervals) are a better fit for the actual capacity for runway 36R than for runway 06.

When interpreting these figures it should be considered that a displacement of one flight causes displacement for every flight that is scheduled closely behind it. The mismatch between actual and simulation results, therefore, builds rapidly if more flights have a small deviation in time. When taking this into account and looking at the purpose of determining the moment of a runway reconfiguration, an average mismatch in the order of one to two minutes is deemed acceptable.

Validity of the Outbound Model

For the outbound model, the correspondence with the actual operations is determined for the time that a flight goes off-block and the time that a flight departs. The time deviations with the actual operations are presented in table 8.2. The mean time deviation is larger for runway 36L than for 36C. This can be explained by the fact that in the model, it is assumed that taxi times are constant regardless of the runway configuration. In reality departure flights often cross runway 36C when this runway is inactive, which significantly reduces the taxi time to runway 36L and causes deviation between the model and the realisation. The deviations are, however, for both runways deemed to be acceptable for the purpose of determining the time of runway reconfigurations.

Table 8.2: Sample mean and standard deviation for deviation of departure times

	Time deviation at runway		Time deviation off-block	
Runway	μ[sec]	σ[sec]	μ[sec]	σ[sec]
36L	90	70	148	155
36C	80	50	106	106

The buildups of deviation between the actual operations and the simulation over the day are depicted in figures 8.9 and 8.10. In both figures, build-ups of positive as well as negative time deviations can be observed. This indicates that the capacity values used, emulate the overall actual capacity reasonably well.

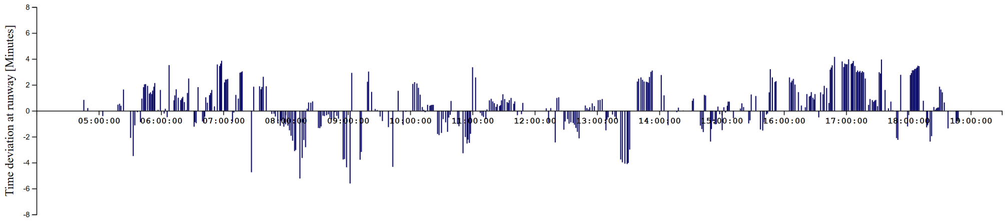


Figure 8.9: Deviation of departure times with realistation for runway 36L

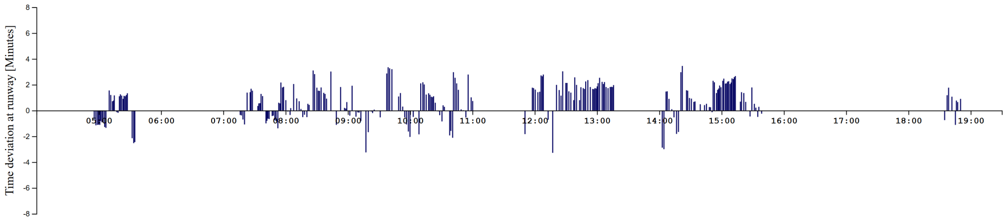


Figure 8.10: Deviation of departure times with realistation for runway 36C

The larger deviations observed for the off-block times can be explained by the fact that apron interactions are not modelled very precisely. In the model, it assumed that an apron area is a collection of parking positions that can be entered or left. In reality, a complex structure exists for certain aprons which causes interference for flights requiring pushback.

Validity of the Control Model

To assess how well the runway reconfiguration decisions generated by the model reflect the actual operations, the runway activation and deactivation times are compared. To do this, the flight selected as the first flight using the new runway configuration and the corresponding time for this flight is compared. The results of this comparison are displayed in table 8.3 and table 8.4 for arrival runway activations/deactivations and departure runway activations/deactivations, respectively. These results are established by 11 negotiation interactions between the APP supervisor and the TWR supervisor. Even though the actual selected flight does not always correspond, only small deviations for the selected time are observed with a maximum of 14 minutes. On average, the selected time for runway activation/deactivation by the model only deviates 4 minutes from the actual operations.

Table 8.3: Comparison of the actual and simulation times for inbound runway reconfiguration

	Actual		**Simulation**	
Interaction	Flight	Time	Flight	Time
2	KLM34Z	04:58	KLM34Z	04:58
3	KLM434	07:16	EZY36WM	07:15
4	TRA132K	08:38	TRA132K	08:36
5	KLM36T	09:44	EZY43CU	09:41
6	KLM1766	10:44	VLG6292	10:50
7	KLM1254	11:57	KLM598	11:49
8	TRA39V	12:42	KLM76L	12:56
9	TRA56M	14:14	TRA56M	14:10
10	KLM16V	15:37	THY9EH	15:31
11	KLM18X	18:25	ADR498	18:30

Table 8.4: Comparison of the actual and simulation times for outbound runway reconfiguration

	Actual		**Simulation**	
Interaction	Flight	Time	Flight	Time
1	TRA871N	04:56	TRA871N	04:55
2	KLM1001	05:35	KLM1001	05:34
3	KLM59K	07:15	KLM59K	07:15
6	KLM1425	11:03	KLM765	11:03
7	KLM91L	11:49	AUI5HE	11:48
8	SAS54P	13:17	SAS54P	13:22
9	KLM1631	14:03	KLM1459	14:08
10	EIN60Y	15:40	KLM1633	15:37
11	AZA94M	18:35	SWR70T	18:25

In figure 8.11, a visualisation shows how well the runway configuration pattern generated by the model fits the actual runway configuration pattern. In the upper part of this figure, above the timeline, the number of active arrival runways is displayed. The bottom part shows the number of active departure runways. A line represents the simulation model output and the filled area shows how many runways were used during the day in the actual operations. From this visualisation, it can be observed that the pattern is highly similar.

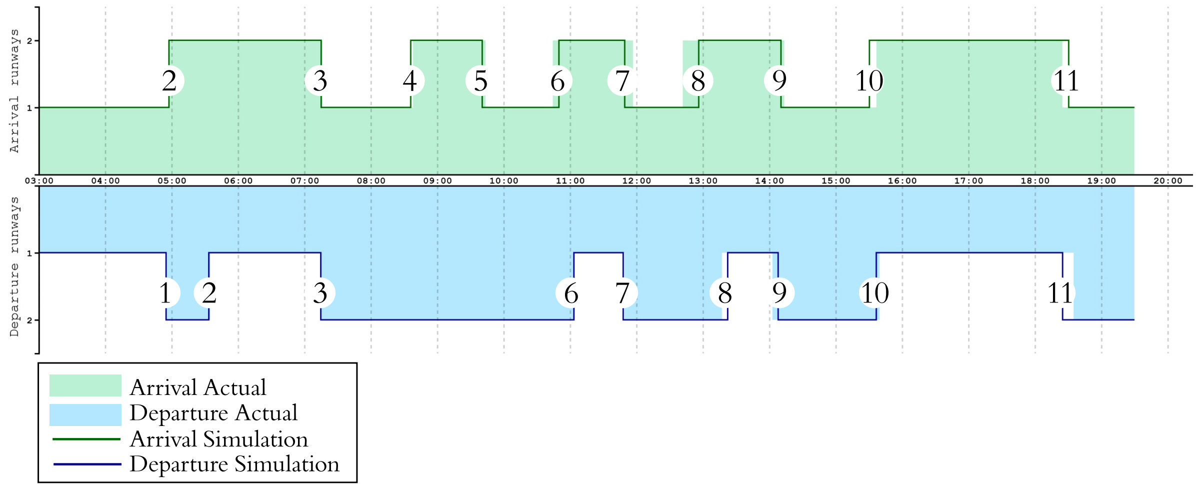


Figure 8.11: Comparison of the actual time and simulation time for runway reconfigurations

To further explain the deviations between the runway configuration pattern resulting from the simulation and the actual runway configuration pattern, the individual negotiation interactions are elaborated upon below. These are numbered from 1 to 11 and are reflected in table 8.3, table 8.4, and figure 8.11.

1. In the first negotiation interaction, flight TRA871N is selected by the model as the first flight for departure runway activation. This corresponds with the actual operations. This flight is selected because it is the first flight in a group before the upcoming outbound peak at 05:00 UTC. Selecting this first flight allows all subsequent flights in the group to depart earlier.
2. For negotiation interaction 2, an inbound peak is scheduled for 05:20 UTC. It is however observed by the APP supervisor that a delay exceeding four minutes is expected to occur around 05:00 UTC. To avoid this delay KLM34Z is proposed as the option for activating an additional arrival runway. For the outbound system, it is not desirable to deactivate the second runway because it was just activated. This leads to a negotiation interaction, that results in a later option in the form of KLM1001 around 05:35 UTC. The same flight options were also selected in the actual operations.
3. In negotiation interaction 3, it is required to determine the activation and deactivation options for the outbound peak at 07:20 UTC. No earlier delay signals for exceeding desired operational boundaries are observed. The group of outbound flights scheduled before 07:20 UTC is therefore used to select an option. KLM59K is selected as the best option in this group using the rules defined in appendix A. This departure runway activation option corresponds with the one selected in the actual operations. The time that flight KLM59K is planned to use the newly activated runway is earlier in the simulation than in reality. The selected initial deactivation flight that avoids the use of a fourth runway, is therefore selected one flight earlier than in reality. Flight EZY36WM is selected instead of KLM434. Selecting this option is expected to not lead to violating the desired operational performance bounds and is therefore accepted as the deactivation option by the APP supervisor.
4. An upcoming delay expected to lead to an undesirable state for the inbound system acts as a signal to start negotiation interaction 4. To avoid this delay, the APP supervisor proposes option TRA132K as the option for activating an additional arrival runway. This option is selected in the actual operations as well. After this option is proposed to the TWR supervisor, it is assessed if it is possible to use the initial deactivation option for the outbound system that avoids the use of a fourth runway. This is expected to result in too much delay for the outbound system and is therefore not accepted. When investigating alternative later deactivation options, no suitable option can be found that results in maintaining the desired outbound performance bounds. No flight option can therefore be selected for a departure runway deactivation, resulting in a 2+2 configuration. This corresponds with the actual operations.
5. For negotiation interaction 5, only an arrival runway deactivation is required because a transition from a double peak to an outbound peak is desired according to the peak schedule at 09:40 UTC. To select an option for this, the first arrival flight option after 09:40 UTC is selected. In the simulation, this is flight option EZY43CU while in the actual operations this was flight KLM36T that uses the runway three minutes later.
6. In negotiation interaction 6, a decision for the time of a transition from an outbound to an inbound peak is required around 11:00 UTC according to the peak schedule. In the simulation, option VLG6292 is selected as the activation option for the inbound system. This option is selected because it is considered the first option in the group before 11:00 UTC. In the actual operations, one flight option earlier is selected. This may be explained by the fact that this option is still considered as being part of the group in reality while it is not considered part of the group in the model. After this option is proposed by the APP supervisor, the TWR supervisor assesses if the initial deactivation can be used without violating the desired operational performance bounds of the outbound system. This is deemed impossible and a later deactivation option KLM765 is therefore selected. This is not the same option as the one used in reality, but the time that this option uses the runway is the same as the time of the option selected in the actual operations. The different selected flight option for departure deactivation can be attributed to a deviation in times for the actual outbound planning and the outbound planning in the model.
7. In negotiation interaction 7, the activation and deactivation options for transitioning to an outbound peak need to be established. For the departure runway activation, the first option before the outbound peak in the peak schedule at 11:50 UTC is selected. In the model, this is option AUI5HE according to the outbound planning. In the actual operations, this is flight KLM91L. For the arrival runway deactivation, the model assesses that it is possible to use the initial deactivation option KLM598. In the actual operations, it is observed that a short period of 2+2 is used and that flight KLM1254 is used, that is 8 minutes later than the option selected in the model. This deviation can be attributed to different capacity values in the actual operations and the model. Capacity is sufficient for an earlier deactivation in the model, while in the actual operations this was not the case.

8. Negotiation scenario 8 is required to transition to an inbound peak. For the selection of the arrival activation option, a deviation in time of 14 minutes is observed. The selected option in the model, KLM76L, is selected because it is deemed the first eligible option of the group before the upcoming inbound peak in the peak schedule at 13:00 UTC. In the actual operations, earlier flights may have been deemed to also be part of the group that can land earlier as the result of the arrival runway activation. If demand is high and a high number of closely spaced aircraft are scheduled before the desired reconfiguration moment, this can lead to a significant time deviation. This could explain why the earlier option TRA39V is selected. For the outbound system, it is not possible to select a deactivation option that avoids the use of a fourth runway without violating the desired operational performance bounds. The negotiation therefore leads to a later option in the form of option SAS54P. The same option is selected in the actual operations.

9. For negotiation scenario 9, the activation and deactivation options must be selected to transition to an outbound peak. For the departure activation, an option is selected that uses the runway 5 minutes later than the selected option in the actual operations. This is deemed to be the effect of a deviation in the outbound planning times. It can be observed in negotiation scenario 8 that the time for the same flight option SAS54 is 5 minutes later in the model than in reality, indicating a deviation of departure times between the actual outbound planning and the outbound planning in the model. This deviation is expected to have caused selecting a later option for the departure activation. For the arrival deactivation, the same option is selected as the one selected in reality.

10. In negotiation scenario 10, the flight selected for arrival activation is THY9EH. This is one flight later in the sequence than flight option KLM16V which is selected in reality. The reason for this is that the model assesses KLM16V to break the bunch because THY9EH and KLM16V both arrive from the ARTIP IAF. In the actual operations, this may have been assessed to not cause any problems, because of the spacing of the flights. For the departure deactivation, one flight earlier is selected than option EIN60Y selected in reality. A deviation of 3 minutes is observed here.

11. In negotiation scenario 11, SWR70T is selected as the departure activation option which is 10 minutes earlier than option AZA94M which was selected in reality. The reconfiguration is required in this situation because of an upcoming outbound peak at 18:30 UTC. The model will never select an option later than this point, while in the actual operations this was opted for in this case. The departure runway was deactivated in the actual operations before the activation of an arrival runway leading to a 1+1 configuration for 10 minutes, which is normally not used during the. A reason that could explain this. would be the temporary unavailability of the arrival runway due to a runway check.

It can be concluded from the highly similar pattern that the model is capable of emulating the decisions made in reality for the simulation day, with some small deviations. To determine how well the model reflects actual decision for a larger range of scenarios a more thorough validation is required. One of the main benefits of the selected modelling method is that expert knowledge can be added by including more defeasible or strict rules without rendering already used rules invalid. Facilitating a wider range of simulation scenarios, therefore comes down to expanding the knowledge base of supervisor agents.

8.2.2 Reasoning Procedures

To provide more insights into the reasoning procedures used in the negotiation interactions, a more detailed description of some interactions is described in this section. This is done for negotiation interaction 1 and 2, that were introduced in section 8.2.1.

Negotiation Interaction 1

The first negotiation scenario starts when both supervisors monitor their system at 04:27 UTC. According to the peak schedule, the first upcoming peak will be the outbound peak at 05:00 UTC. Given that the supervisors monitor for the upcoming 40 minutes, the new demand peak is planned to occur within the monitoring horizon. For the monitoring activity, the supervisor agents will inspect if a weak or a strong signal exists for an undesirable state of the system.

The flight planning for both systems is graphically presented in figure 8.12. Inbound planning information is presented in the upper part of the figure in green. At the instance of monitoring, only runway 06 is active. Each flight in the planning is presented in the form of a bar with the callsign of the flight, at a position that corresponds with the time in the planning of the flight. Above each flight bar, the minutes of delay in the planning(which is subject to change) is presented. For the current monitoring action, no delay higher than four minutes can be observed for the inbound system. This means that no weak signal is observed by the APP supervisor agent in the

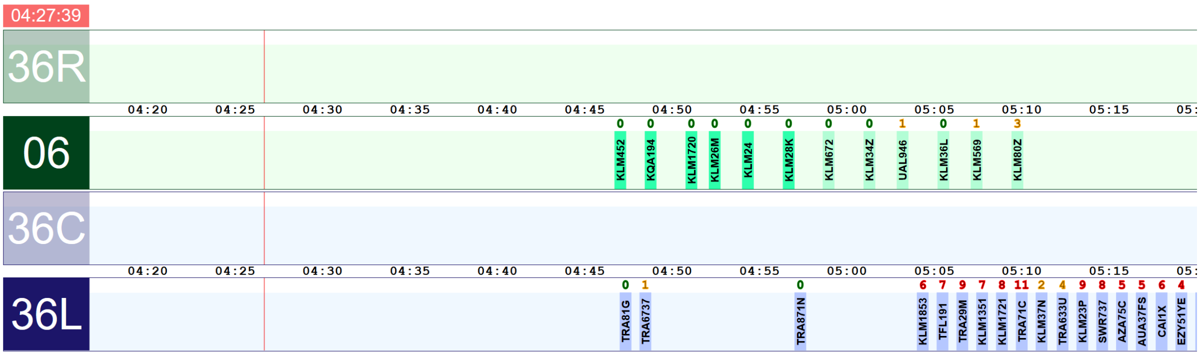


Figure 8.12: Planning information for the first decision scenario

monitor horizon. However, a strong signal is observed in the form of an upcoming outbound peak at 05:00 UTC in the peak schedule which is within the monitor horizon of 40 minutes.

The planning for the outbound system is depicted in blue, in the bottom section of figure 8.12. As indicated by the faded out 36C, only runway 36L is active. In the planning, it can be seen that within the monitoring horizon, before 05:07 UTC, delays are observed that exceed the threshold of 4 minutes. This is interpreted as a weak signal by the TWR supervisor agent. A weak signal normally gives rise to performing a prediction, to assess if the weak signal results in an undesirable state. If this is the case it is regarded as a strong signal. In this case, however, a strong signal is already observed by the TWR supervisor resulting from the upcoming new demand peak within 40 minutes. It is therefore not required to investigate if the weak signal leads to a strong signal.

Given that both the APP supervisor and the TWR supervisor have a strong signal for an upcoming undesirable state, both agents will assess what they should do to avoid this undesirable state. This is done by following the protocol defined in B. Following the protocol, the APP supervisor has to wait and the TWR supervisor is allowed to utter the locution $ProposeDepActivation(S,\delta)$. Given that it is allowed to utter this locution, the TWR supervisor agent has the goal to generate a proposal using this locution.

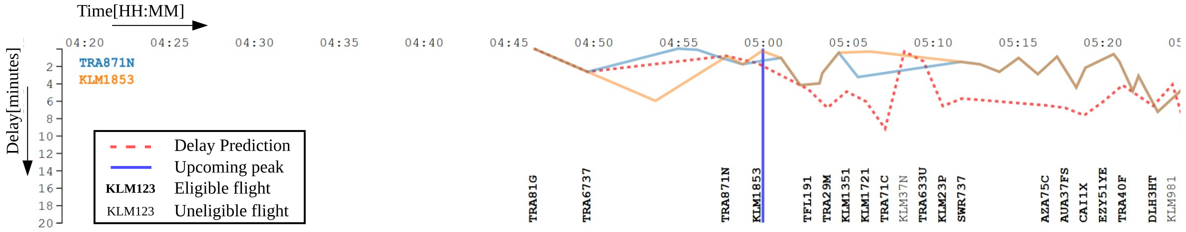


Figure 8.13: Prediction of the state evolution of the outbound system

In order to fulfil the goal of generating a proposal, the first step is to generate options for this proposal. This option generation is done based on a prediction. The prediction is depicted in figure 8.13. The dashed red line represents the delay values observed in the mental forward simulation. Apart from these delay values, the predicted times for the flights being at the runway are displayed in the bottom portion of the chart, either in normal font or bold font. Flights displayed in bold font, are flights that are destined for the newly activated runway(36C) in a two-runway active scenario.

In the option generation step, the aim is to find the minimal set of potential candidates that could be used to activate a runway, thereby potentially avoiding the undesirable state. This is only possible if a new runway is activated before the undesirable state occurs, so all potential options should be planned before the time of the undesirable state. In this case, the undesirable state is expected to occur at 05:00 UTC, indicated by the vertical blue line in figure 8.13. The options that are scheduled to depart before this point in time are TRA81G, TRA6737, TRA871N and KLM1853. One of these four options will have to be selected to be the first flight that uses the newly activated runway. This is only possible when the flight is destined to use the newly activated runway in a two-runway active scenario. In this case, all four flights can use the to be activated runway and all four could be selected.

The aim of activating a new runway is to avoid delay build-up. This is most efficient if an entire group of flights can be shifted back, such that for all these flights delay is mitigated. This is only possible of the flights are closely spaced in a group and have to wait on each other. In this case, only flight TRA871N and KLM1853 are closely spaced before the expected point of the undesirable state. It is therefore likely that flight TRA871N cannot be shifted back, because it will already be at its earliest possible departure time. The flights TRA871N and KLM18153 are therefore selected as the group of flights before the undesirable state, that is eligible for the runway activation action.

Given that the TWR supervisor now has two options for the runway activation, it will have to select one of the two. To do so, the options are evaluated. This is done by performing the activation action for each flight in a mental forward simulation. The results in terms of delay of this forward simulation are displayed in figure 8.13. The observations from the forward simulation for both options are interpreted resulting in facts for both options:

$$\left\{\begin{array}{ll} NoSlotsMissed(tra871n) & NotBlockingPreviousFlight(tra871n) \\ HighTotalDelay(tra871n) & NoSlotsMissed(klm1853) \\ NotBlockingPreviousFlight(klm1853) & HighTotalDelay(klm1853) \\ MaxDelayHigher(klm1853, tra871n) & TotalDelayHigher(klm1853, tra871n) \\ TimeLater(tra871n, klm1853) & \end{array}\right\}$$

Using this set of literals a decision is made on which of the two options to select for the runway activation proposal. This is done by querying $ProposeDepActivation(klm1853)$ and $ProposeDepActivation(tra871n)$, against the knowledge base containing the literals and the rules defined in appendix A. After performing a defeasible reasoning procedure, the answer for TRA871N is yes and for KLM18153 is no according to the following reasoning lines:

$$\left[\begin{array}{l} \{ProposeDepActivation(tra871n) \mathrel{-\!\!\prec} EligibleOption(tra871n)\}, \\ \{EligibleOption(tra871n) \mathrel{-\!\!\prec} NoSlotsMissed(tra871n)\}, \\ \{NoSlotsMissed(tra871n) \mathrel{-\!\!\prec} MaxDelay(5),\ 5 < 10\} \end{array}\right]$$

$$\left[\begin{array}{l} \{\sim ProposeDepActivation(klm1853) \mathrel{-\!\!\prec} EligibleOption(klm1853),\ PerformanceAlternative(klm1853)\}, \\ \{PerformanceAlternative(klm1853) \mathrel{-\!\!\prec} NotBlockingPreviousFlight(tra871), \\ NoSlotsMissed(tra871n), TotalDelayHigher(klm1853,\ tra871n)\}, \\ \{EligibleOption(klm1853) \mathrel{-\!\!\prec} NoSlotsMissed(klm1853)\}, \\ \{NoSlotsMissed(klm1853) \mathrel{-\!\!\prec} MaxDelay(6),\ 6 < 10\} \end{array}\right]$$

From the reasoning lines it can be observed, that TRA871N is selected because it is a better alternative in terms of delay performance when compared to flight KLM1853. This is the case because flight KLM1853 results in less total delay than flight TRA871N. This fact could be contradicted by the fact that flight TRA871N would block the previous flight, leading to more delay and worse performance. This is not the case here.

Given that the option is now selected, this option will be inserted as proposal content into the allowed locution. This leads to the locution $ProposeDepActivation(tra871n)$, which is uttered to the APP supervisor agent. The literals constructed in the reasoning procedure for eligible options are included in the locution to provide meta-information about the locution that can potentially be used by the APP supervisor agent to evaluate the proposal.

Upon receiving the uttered proposal by the TWR supervisor agent, the APP supervisor agent interprets the proposal and argument content contained by the locution. By interpreting the proposal content the APP supervisor agent semantically infers, that enough information is available and a reasoning action should be performed. In this reasoning action, the APP supervisor agent queries whether it should accept the incoming proposal with query $AcceptDepActivation(tra817n)$. Given that no deactivation for the inbound system is required, the APP supervisor agent will normally always accept this. The only reason for not accepting this action would be if no personnel is available to facilitate the new configuration. Since no information is available on this aspect, the APP supervisor agent decides to accept the proposed departure runway activation. This is expressed to the TWR supervisor agent by uttering the locution $AcceptDepActivation(tra871n)$. Upon receiving this locution, the TWR supervisor agent will activate the departure runway with flight TRA871N as the first flight to use this runway.

Negotiation Interaction 2

The second interaction is initiated by a monitoring action at 04:37 UTC. During this monitoring action, the APP supervisor agent observes a weak signal in the form of a delay higher than four minutes in the planning for flight KLM80Z (see figure 8.14). The TWR supervisor observes a weak signal as well. Given that for the TWR supervisor, two departure runways are already activated in the planning, no action may be performed by the TWR supervisor. For the APP supervisor, however, only one runway is active and the weak signal is inspected to determine whether a strong signal results from it and action should be performed.

In order to inspect the weak signal, the APP supervisor does a forward simulation to assess the delay build-up. Given that this delay build-up does not result from only one flight but is a delay build-up pattern for multiple flights, this is regarded as a strong signal. The delay pattern for the prediction generated in the forward simulation

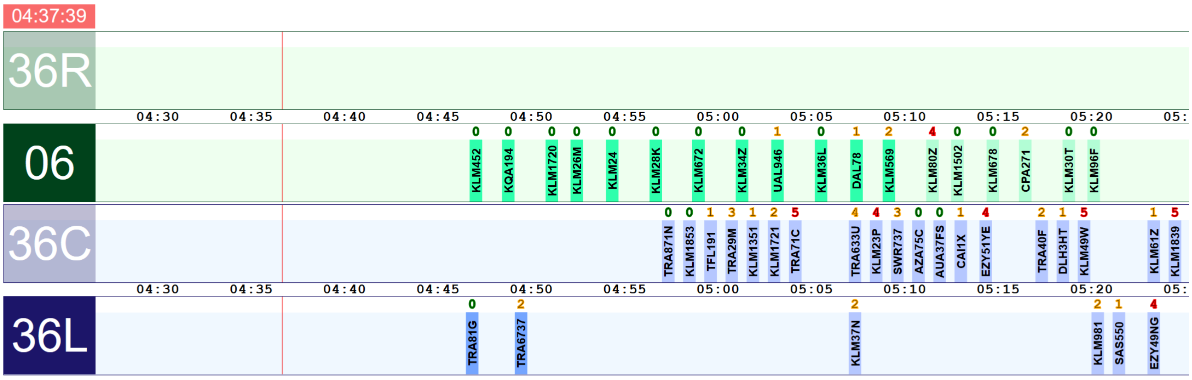


Figure 8.14: Planning information for negotiation interaction 2

is displayed as a dashed red line in figure 8.15. From this line, it is observed that an undesirable state is predicted to happen from flight KLM80Z and onwards.

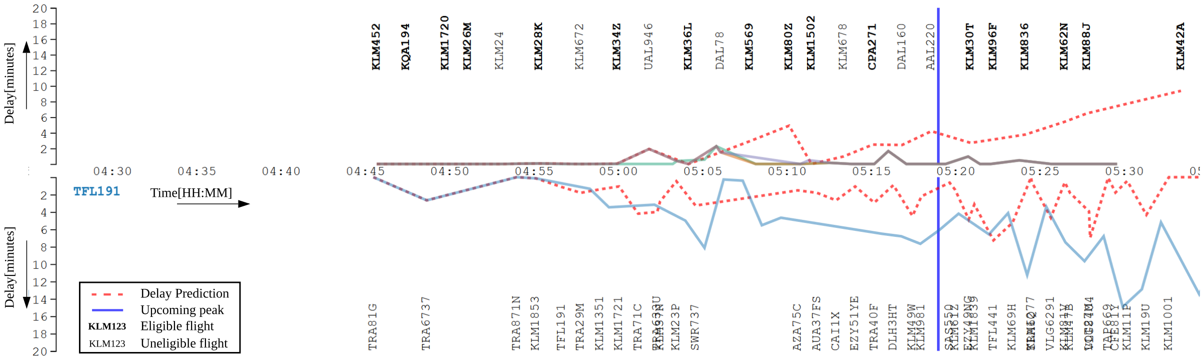


Figure 8.15: Prediction of the state evolution of the inbound and outbound system for negotiation interaction 2

In order to avoid this undesirable state, the APP supervisor starts looking for options to activate an additional arrival runway. These options should be flights that are expected to land before the time of the occurrence of the undesirable state. The group of flights that are closely spaced before this point and could resolve the delay build-up is determined to be the group that is expected to land later than flight KLM28K. In this group, only three flights could use the newly activated runway if it were activated given their designated IAFs. These flights are KLM34Z, KLM36L and KLM569. This is regarded as the option set for activating an additional arrival runway.

To decide which of the selected options to propose, the APP supervisor estimates the effect of performing each of the action options by mental forward simulation. The results of this forward simulation for each option are displayed in the coloured lines in the upper half of figure 8.15. It can be observed that all of the three options are expected to fulfil the goal of preventing the undesirable state of four minutes or more delay. Selecting the best of the three is therefore done based on the performance comparison information. After the literal reification, the following information is available for the options:

$$\left\{\begin{array}{ll}
NoHoldingPatterns(klm34z) & NotBreakingBunch(klm34z) \\
LowTotalDelay(klm34z) & NoHoldingPatterns(klm36l) \\
NoHoldingPatterns(klm569) & NotBreakingBunch(klm569) \\
BreakingBunch(klm36l) & LowTotalDelay(klm569) \\
TotalDelayHigher(klm569,klm34z) & TotalDelayHigher(klm36l,klm34z) \\
TimeLater(klm569,klm34z) & TotalDelayHigher(klm569,klm36l) \\
TimeLater(klm36l,klm569) &
\end{array}\right\}$$

This information is used in the defeasible reasoning procedure to generate the following argumentation lines for the three options:

$$\left[\begin{array}{l}
\{\mathit{ProposeArrActivationDepDeactivation}(\mathit{klm}34z,\mathit{depFlight}) \mathrel{-\!\!\prec} \mathit{EligibleOption}(\mathit{tra}871n)\},\\
\{\mathit{EligibleOption}(\mathit{klm}34z) \mathrel{-\!\!\prec} \mathit{NoHoldingPatterns}(\mathit{klm}34z)\},\\
\{\mathit{NoHoldingPatterns}(\mathit{klm}34z) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2),\ 2 < 4\}
\end{array}\right]$$

$$\left[\begin{array}{l}
\{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{klm}36l,\mathit{depFlight}) \mathrel{-\!\!\prec} \mathit{EligibleOption}(\mathit{klm}36l),\\
\mathit{ConflictingOption}(\mathit{klm}36l)\},\\
\{\mathit{ConflictingOption}(\mathit{klm}36l) \mathrel{-\!\!\prec} \mathit{BreakingBunch}(\mathit{klm}36l),\mathit{NotBreakingBunch}(\mathit{klm}569),\\
\mathit{NoHoldingPatterns}(\mathit{klm}569)\},\\
\{\mathit{EligibleOption}(\mathit{klm}36l) \mathrel{-\!\!\prec} \mathit{NoHoldingPatterns}(\mathit{klm}36l)\}\\
\{\mathit{NoHoldingPatterns}(\mathit{klm}36l) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2),\ 2 < 4\}\\
\{\mathit{NotBreakingBunch}(\mathit{klm}569) \mathrel{-\!\!\prec} \mathit{StackNextFlight}(\mathit{sugol}),\mathit{sugol} \neq \mathit{artip}\}
\end{array}\right]$$

$$\left[\begin{array}{l}
\{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{klm}569,\mathit{depFlight}) \mathrel{-\!\!\prec} \mathit{EligibleOption}(\mathit{klm}569),\\
\mathit{PerformanceAlternative}(\mathit{klm}569)\},\\
\{\mathit{PerformanceAlternative}(\mathit{klm}569) \mathrel{-\!\!\prec} \mathit{NotBreakingBunch}(\mathit{klm}34z),\mathit{NoHoldingPatterns}(\mathit{klm}34z),\\
\mathit{TotalDelayHigher}(\mathit{klm}569,\mathit{klm}34z)\},\\
\{\mathit{EligibleOption}(\mathit{klm}569) \mathrel{-\!\!\prec} \mathit{NoHoldingPatterns}(\mathit{klm}569)\}\\
\{\mathit{NoHoldingPatterns}(\mathit{klm}569) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2),\ 2 < 4\}\\
\{\mathit{NoHoldingPatterns}(\mathit{klm}34z) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2),\ 2 < 4\}
\end{array}\right]$$

From these argumentation lines, it is observed that KLM34Z is the best option. Flight KLM34Z is reasoned to be an eligible option for the arrival activation and no argument can be established from the knowledge base of the supervisor agent that defeats this argument. For flight KLM36L, the argument can also be made that it is an eligible flight option. For this option, however, an argument can be constructed that successfully attacks it. The defeater for this argument stems from the fact that opting for this flight will break the bunch of flights from the ARTIP arrival stack. This phenomenon is described in section 2.1.5.For flight KLM569, no breaking the bunch will occur if this flight is selected. For this flight option, the total delay is higher than for flight KLM34Z, making it a worse alternative. The APP supervisor, therefore, opts to select flight KLM34Z.

With flight KLM34Z selected as the best option for activating the arrival runway, the APP supervisor will utter the locution $\mathit{ProposeArrActivationDepDeactivation}(\mathit{klm}34z, \mathit{DepFlight})$. This locution indicates a proposal for an arrival runway activation and a departure runway deactivation. It is proposed that this should be done with flight KLM34Z as the first arrival flight that uses the newly activated runway and the departure flight option for the to be deactivated runway is left as a variable. This is done because the APP supervisor does not know the predicted take-off times for the outbound model and does therefore not know what a suitable option for the departure runway activation is.

Upon receiving this locution, the TWR supervisor performs a semantic analysis to determine which information should be acquired to form an evaluation of the proposal. The TWR supervisor recognises that this proposal implies a runway deactivation for the outbound system. To evaluate the effects of this action, the TWR supervisor will first determine which flight should be selected to avoid a 2+2 scenario. To do so the meta-information from the received locution is used. This contains the proposed time of the arrival runway activation. To prevent a flight from using the fourth runway, the first flight that is planned to depart later than the proposed activation time for the arrival runway should be selected. In this case, flight TFL191 is the first flight that is planned to take-off later than the expected landing time of flight KLM34Z. To assess what the impact will be of deactivating departure runway 36C before flight TFL191 has taken off, a forward simulation is performed. The resulting delay values from this forward simulation are displayed in the bottom part of figure 8.15. As can be observed by looking at the blue line in the bottom part of this figure, it is expected that performing this action will result in unacceptable delays that are higher than ten minutes. The TWR supervisor interprets this information as leading to missed slots for outbound flights. This leads to not accepting option TFL191 for deactivation according to the following undefeated argumentation line:

$$\left[\begin{array}{l}
\{\sim\mathit{AcceptArrActivationDepDeactivation}(\mathit{klm}34z,\mathit{tfl}191) \mathrel{-\!\!\prec} \mathit{SlotsMissed}(\mathit{tfl}191)\},\\
\{\mathit{SlotsMissed}(\mathit{tfl}191) \mathrel{-\!\!\prec} \mathit{MaxDelay}(18),\ 18>10\}
\end{array}\right]$$

After recognising that the initial option TFL191 is not a viable option, the TWR supervisor agent will generate alternatives. To do so, the information from the mental forward simulation of deactivating for option TFL191 is

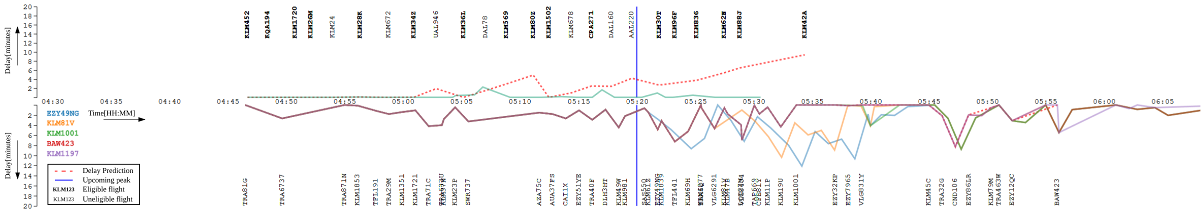


Figure 8.16: Results of mental simulation for deactivating a departure runway before flight TFL191

used. From this simulation, it can be derived where delay build-up exists which can be prevented by deactivating the runway after flights that cause this delay build-up. It only makes sense to select flights that are planned after flights that use the secondary runway and therefore use the capacity of the second runway. The options are searched for in a broad timespan up until the next scheduled peak after 5:20 UTC, which is 7:20 UTC. The options that are selected can partly be seen as the flight names depicted in the bottom part of figure 8.15. Flight names that are overlapping indicate moments for which both runways are used, flights after these flights are therefore good alternatives because the double runway capacity is used before these options. In this case, five appropriate options can be found which are presented in a coloured font in the bottom part of figure 8.16.

To decide which of the five generated options should be selected, a forward simulation is performed to evaluate the consequences of the five options. The resulting delay values from these mental forward simulations are presented in figure 8.16. It is clear from these simulations that opting for early options EZY49NG or KLM81V will result in unacceptable delays, leaving only KLM1001, BAW423 and KLM1197 as viable candidates. For these candidates, the information resulting from the forward simulation is interpreted and a reasoning procedure is performed to decide which option to select. This leads to the following argumentation lines for the three options:

$$\left[\begin{array}{l}
\{\sim ProposeArrActivationDepDeactivation(klm34z,klm1001) \mathrel{-\!\!\prec} EligibleOption(klm1001), PerformanceAlternative(klm1001)\},\\
\{EligibleOption(klm1001) \mathrel{-\!\!\prec} NoSlotsMissed(klm1001)\},\\
\{PerformanceAlternative(klm1001) \mathrel{-\!\!\prec} TotalDelayHigher(klm1001,baw423), NoSlotsMissed(baw423)\},\\
\{NoSlotsMissed(klm1001) \mathrel{-\!\!\prec} MaxDelay(9),\ 9<10\},\\
\{TotalDelayHigher(klm1001,baw423) \mathrel{-\!\!\prec} TotalDelay(klm1001,186),\ TotalDelay(baw423,178),\ 186>178\}\\
\{NoSlotsMissed(baw423) \mathrel{-\!\!\prec} MaxDelay(9),\ 9<10\}
\end{array}\right]$$

$$\left[\begin{array}{l}
\{\sim ProposeArrActivationDepDeactivation(klm34z,baw423) \mathrel{-\!\!\prec} EligibleOption(baw423), PerformanceAlternative(baw423)\},\\
\{EligibleOption(baw423) \mathrel{-\!\!\prec} NoSlotsMissed(baw423)\},\\
\{PerformanceAlternative(baw423) \mathrel{-\!\!\prec} TotalDelayHigher(baw423,klm1197), NoSlotsMissed(klm1197)\},\\
\{NoSlotsMissed(baw423) \mathrel{-\!\!\prec} MaxDelay(9),\ 9<10\},\\
\{TotalDelayHigher(klm1001,baw423) \mathrel{-\!\!\prec} TotalDelay(baw423,178),\ TotalDelay(klm1197,178),\ 178>169\}\\
\{NoSlotsMissed(klm1197) \mathrel{-\!\!\prec} MaxDelay(8),\ 8<10\}
\end{array}\right]$$

$$\left[\begin{array}{l}
\{ProposeArrActivationDepDeactivation(klm34z,klm1197) \mathrel{-\!\!\prec} EligibleOption(klm1197)\},\\
\{EligibleOption(klm1197) \mathrel{-\!\!\prec} NoSlotsMissed(klm1197)\},\\
\{NoSlotsMissed(klm1197) \mathrel{-\!\!\prec} MaxDelay(8),\ 8<10\}
\end{array}\right]$$

From the argumentation lines, it can be concluded that flight options KLM1001 and BAW423 are discarded because a better performance alternative exists for these two options. The TWR supervisor agent, therefore, selects KLM1197 as the best option to propose. This leads to the locution for a deactivation with flight KLM1197 as the deactivation option: $ProposeArrActivationDepDeactivation(klm34z, klm1197)$. This locution is communicated to the APP supervisor agent.

Upon receiving the locution, the APP supervisor agent will determine which actions to perform to acquire the required information for evaluating the proposal. The APP supervisor agent recognises, that the proposal is ground with an option for the arrival activation and an option for the departure activation. It will therefore first assess if flight KLM34Z is still eligible given the combination of the activation deactivation action. In figure 8.16, it can be seen in the upper part that the delay pattern remains the same for the forward simulation of the combination of actions. Flight option KLM34Z, therefore, remains eligible. Because the combination of the options is now known, the APP supervisor can determine what the fourth runway time will be. This information is added to already available information about the combination of options in its knowledge base. Using this knowledge base, the combination of options is evaluated. This leads to the following argumentation line:

$$\left[\begin{array}{l}
\{\sim \mathit{AcceptArrActivationDepDeactivation}(klm34z, klm1197) \mathrel{-\!\!\prec} \mathit{UnacceptableFourthRunwayTime}(klm34z),\\
\mathit{BetterDepartureAlternative}(klm1197), \mathit{EligibleOption}(klm34z)\},\\
\{\mathit{BetterDepartureAlternative}(klm1197) \mathrel{-\!\!\prec} \mathit{TimeLater}(klm1197, baw423)\},\\
\{\mathit{EligibleOption}(klm34z) \mathrel{-\!\!\prec} \mathit{NoHoldingPatterns}(klm34z)\},\\
\{\mathit{NoHoldingPatterns}(klm34z) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2), 2 < 4\}
\end{array}\right]$$

From the information interpreted from the incoming locution and the derived fourth runway time, the APP supervisor agent reasons that a better departure option alternative exists. In this case, KLM1977 is a later alternative than BAW423, which is also an eligible option. So through the information contained in the locution, the APP supervisor agent now knows that an earlier deactivation option exists that is deemed eligible for deactivation by the TWR supervisor agent. Because the APP supervisor prefers this earlier departure deactivation option that prevents fourth runway use, the APP supervisor agent rejects the proposal.

Given that it is not desired to generate delay at the start of a new demand peak, the APP supervisor will not consider later options than the options it currently knows of (KLM34Z, KLM36L and KLM569). For these options, it is assessed if it is possible to solve the excessive fourth runway time. Because this is not the case, the APP supervisor agent sticks to KLM34Z as the option for activating the arrival runway. Using the meta-information in the locution received from the TWR supervisor agent, the APP supervisor starts a reasoning procedure to determine which alternative departure option to select. This leads to the following argumentation lines:

$$\left[\begin{array}{l}
\{\sim \mathit{ProposeArrActivationDepDeactivation}(klm34z, baw423) \mathrel{-\!\!\prec} \mathit{EligibleOption}(klm34z), \mathit{TimeLater}(baw423, klm1001)\}.\\
\{\mathit{EligibleOption}(klm34z) \mathrel{-\!\!\prec} \mathit{NoHoldingPatterns}(klm34z)\},\\
\{\mathit{NoHoldingPatterns}(klm34z) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2), 2 < 4\}
\end{array}\right]$$

$$\left[\begin{array}{l}
\{\mathit{ProposeArrActivationDepDeactivation}(klm34z, klm1001) \mathrel{-\!\!\prec} \mathit{EligibleOption}(klm34z)\}.\\
\{\mathit{EligibleOption}(klm34z) \mathrel{-\!\!\prec} \mathit{NoHoldingPatterns}(klm34z)\},\\
\{\mathit{NoHoldingPatterns}(klm34z) \mathrel{-\!\!\prec} \mathit{MaxDelay}(2), 2 < 4\}
\end{array}\right]$$

From these argumentation lines, it is concluded that flight KLM1001 was selected as the option for the departure runway activation by the APP supervisor. The reasoning line dictates that flight BAW234 is a later option than KLM1001 and will perform worse in terms of fourth runway time. Option KLM1001 is therefore selected and the APP supervisor agent utters the locution $\mathit{ProposeArrActivationDepDeactivation}(klm34z, klm1001)$ to the TWR supervisor.

Upon receiving this locution, the TWR supervisor agent assesses the consequences for this combination of options. This leads to the following argumentation line:

$$\left[\begin{array}{l}
\{\mathit{AcceptArrActivationDepDeactivation}(klm34z, klm1001) \mathrel{-\!\!\prec} \mathit{EligibleOption}(klm34z)\}.\\
\{\mathit{EligibleOption}(klm1001) \mathrel{-\!\!\prec} \mathit{NoSlotsMissed}(klm1001)\},\\
\{\mathit{NoSlotsMissed}(klm1001) \mathrel{-\!\!\prec} \mathit{MaxDelay}(9), 9 < 10\}
\end{array}\right]$$

It is clear from this argumenation line, that flight KLM1001 remains a viable option for the departure deactivation moment. No other argument can be constructed that defeats this argument. This means that the TWR supervisor agent accepts the combination of the activation and deactivation options. It therefore utters the locution $\mathit{AcceptArrActivationDepDeactivation}(klm34z, klm1001)$, to the APP supervisor agent. From semantic analysis of this locution the TWR supervisor recognises that the departure runway must be deactivated in the planning system after this flight and communicates this with the outbound planner agent. Upon receiving the acceptance locution the APP supervisor recognises that the arrival runway is allowed to be activated with KLM34Z as the first flight and communicates this with the inbound planner agent.

9

Conclusions and Recommendations

After developing the model and analysing the results of the simulation scenario, the main findings can be summarized. They can be found in this chapter, combined with an overview of the contributions of the research and recommendations for follow-up research.

9.1 Main Findings

The research objective defined in section 4.1 and the following research questions are aimed to be answered by the findings of the research. These findings are presented in this section for the modelling aspects, aspects related to the runway reconfiguration phenomenon and general findings that could be utilised by existing models.

Modelling Method

In the development of the runway reconfiguration model, the emphasis was put on a natural way of description and explainable results. In line with this aim, an agent-based approach is followed. This has resulted in a way of modelling that provides flexibility for incorporating aspects in the model that aid in understanding the phenomenon that is researched. When compared to following a high-level approach, such as queuing theory, more detail for the behaviour of individual components can be introduced. This leads to more detailed insights and allows to capture all aspects relevant for deciding on the moment of a runway configuration. It furthermore allows to easily introduce new aspects to the model, in the form of adding new agents or new behaviours.

Opting for argumentation techniques for the control model has resulted in a model that is easily interpreted and highly adaptable. Being able to track all argumentation lines, makes it easier to understand and adapt the behaviour of the model. In a defeasible logic reasoning approach, this comes down to simply adding additional rules to the knowledge base of a supervisor agent. Given that the conclusion of an argumentation procedure results from evaluating the entire corpus of knowledge and explicit negation is allowed to exist, adding new rules does not invalidate the existing set of rules forming the knowledge base. Apart from this adaptability, the resemblance with the reasoning methods employed by the human mind makes results easier to accept. Causes for unexpected solutions can easily be found by analysing the argumentation line leading to an unexpected solution.

Employing an argumentation-based negotiation method, has resulted in a highly efficient negotiation mechanism. Especially for a complex system, such as the inbound-outbound system at AAS, running forward simulations is highly expensive in terms of computing power leading to the desire for an efficient negotiation mechanism. By transferring information about motives for proposals, the exact points that are preventing an agreement can be addressed in a way that corresponds with the goals of an agent.

From the simulation results, it is observed that the selected modelling method resembles the actual runway reconfiguration decisions well. On average, only a deviation in time of four minutes is observed between the timing of runway reconfigurations by the model and reconfigurations in the actual operation. This result can be improved by incorporating more expert knowledge into the model in the form of new rules.

Runway Reconfiguration Phenomenon

The low-level way of modelling has resulted in exploring the runway reconfiguration phenomenon in great detail. Apart from the more high-level aspects of the reconfiguration phenomenon, the research has aided in understanding the dynamics and decision-making aspects, by modelling it. These gained insights aid in answering the research questions posed in section 4.1.

The main actors in the runway reconfiguration process are the APP supervisor and the TWR supervisor. A decision by these two actors, for the runway reconfiguration moment, arises from a negotiation interaction. Most influential information in this interaction is the information resulting from the planning tools for both the inbound and the outbound system.

From both the expert knowledge and the analysis of the simulation scenario, it has become clear that the best moment for starting a negotiation interaction, is between 20 and 30 minutes before the expected moment that this runway reconfiguration is required. It is aimed to follow the peak schedule, when possible. A current runway configuration that is different from the upcoming demand peak configuration in the peak schedule should be considered a signal to start an interaction for deciding on the runway reconfiguration timing. This results from the fact that the schedules are created in such a way that the resulting demand follows the runway capacity for the demand peak configurations in the peak schedule over the day. The peak schedule, therefore, serves as a good guideline for determining the moment of a runway reconfiguration. This does however not mean that the peak schedule can be followed in all situations.

Traffic realisations often deviate significantly from the pre-defined schedule. In these scenarios, the capacity distribution resulting from the peak schedule will not fit the demand pattern anymore. In this case, another method for determining the approximate moment for adapting the runway reconfiguration is required. From expert knowledge, it has become clear that this is done based on signals from the planning systems and experience. These signals are in the form of expected delays for flights. The experience of a supervisor operator is used to assess if the observed signal will result in an undesired state that cannot be resolved. This is reflected in the model by an anticipation mechanism, in which delay signals are observed as weak signals and are assessed by a forward simulation of the domain model.

For the inbound system, an expected arrival delay of larger than four minutes is determined to be the threshold for a weak signal. For the outbound model, this threshold is determined to be at five minutes of start-up delay. These values are confirmed by expert knowledge. The reasoning behind the value of four minutes of arrival delay results from the fact that more than four minutes of required arrival delay in the FIR can only be resolved by instructing a holding pattern. This causes instability in the planning and is therefore highly undesirable. In the outbound model start-up delays of more than 10 minutes, would cause undesirable situations because new slots must be requested in this case. This causes high delays and destabilises the outbound systems. A delay of more than 5 minutes is regarded as a signal for this situation to arise.

For a detected weak signal, it is first assessed of this weak signal results from a single flight or a delay build-up pattern. If the flight that acts as the signal is not part of a delay build-up, the reason for the delay is not a deficit of runway capacity and a runway reconfiguration is not required. This can be the case if start-up delay results from departure flights regulated by Eurocontrol or arrival delay that results from a bunged set of arrival flights. This assessment is in reality done based on experience and is in the model assessed by a forward simulation for which the resulting delay slopes are assessed.

If it is assessed that either a different demand peak configuration is upcoming in the next 40 minutes or it is established that an observed weak signal leads to an undesirable state, the negotiation interaction should be initiated. From expert knowledge, it is determined that this interaction is always initiated by the supervisor operator that requires an additional runway for capacity. This operator will then assess the options for the to be performed action. This action is determined by the current demand peak configuration and the upcoming demand peak configuration in the schedule. Given that it is aimed to follow the peak schedule, this will most often result in activation of an arrival runway and deactivation of a departure runway or the opposite.

For a given activation or deactivation action, it is assessed by the supervisor operator which flight should be selected, as the first flight to use the new configuration. Selecting a suitable option for the activation of a runway is of higher importance than selecting a suitable option for deactivating a runway. This results from the fact that delay at the start of a peak is much more undesirable than delay at the end of a peak which can be easier resolved. The option selection for activating a runway is therefore limited to options that are expected to avoid this delay at the start of a peak.

When selecting a flight option that is taken as the first flight to use a newly activated runway, only the candidates that are forming a group that causes the delay build-up should be considered. This limits the group of flights that can be selected for an activation action, to flights that are part of the delay build up in the group before the expected moment of the undesirable state. In practice, this comes down to a limited set of flights.

Whereas the set of flights that is eligible as options for an activation action is limited to only a few flights, the set of flights that is eligible for a deactivation action is initially limited to one flight. In case of a combination of an activation action and deactivation action, it is aimed to prevent the use of a fourth runway. To do so, the first flight that is planned to use the fourth runway later than the activation flight must be selected as the deactivation option. Only if this is not possible, due to this resulting in unacceptable delays, an alternative later option can be selected for deactivation.

The option that should be selected if the initial option is not possible, results in reality from the experience of the supervisor operator to select later options that are suitable. In the model, this is reflected by assessing what the result would be of deactivating the initial option. Delay that is simulated to occur from an initial deactivation action can only be prevented if a later flight is selected as an option, such that the second runway capacity can be utilised for a longer period. The suitable options are therefore limited to later options that are planned after flights that can utilise the second runway capacity. Delay is more acceptable for the system for which a runway is deactivated. For this reason, a broader range of options for the deactivation is selected. This range is limited by the flights that are planned after the initial option and before the occurrence of subsequent peak after the upcoming demand peak.

Once options are selected different aspects were found to be important in determining which option or combination of options to select for a proposal. To be able to compare options, first information about the consequences of these options must be collected. This is done in the model by simulating performing the selected action for the selected option. Initially, a supervisor has to come up with a solution that is best for the system that it is responsible for. To do so, the delay consequences of different options are compared. Normally it is desired to select the option that performs best in terms of individual delay. It could, however, be that the option that results in the lowest maximum individual delay results in a lower total delay. In this case, the flight with a lower total delay is preferred. Another aspect that should be considered is if an arrival flight breaks the bunch at ARTIP or conflicts in the air after take-off occur. If this is the case, the performance benefits of an option are cancelled. Other aspects to consider are unavailability of personnel and the runway or emergencies such as fuel problems or queuing situations in the taxiway system that invalidate options from being viable. To select an option, all these aspects must be considered in a non-monotonic way and such that it is not possible to select no option.

After selecting the best option this should be proposed to the opponent supervisor, which should in turn generates information about this option and assesses it. After both supervisor actors have selected an option. The resulting combination of options is assessed based on the fourth runway time. From expert knowledge, it is found that a combination of options is aimed to result in a fourth runway time that is less than 30 minutes. During the simulation scenario, it is observed that this is mainly done by shifting the deactivation option backwards in time. Another option would be to shift the activation option forward, but because this causes delay at the start of a peak it is regarded as a better solution to select an earlier deactivation option. If no option can be found that fits the 30 minutes for the fourth runway time and does not result in an undesirable state of the system, no deactivation action will take place and a 2+2 configuration will be used for longer than 30 minutes until a runway can be deactivated.

From the simulation scenario, it is observed that mutually acceptable agreements are reached swiftly, with no more than three interactions. This is efficiency is mainly due to the information exchange during the argumentation-based negotiation. This aspect is confirmed to be true in reality as well, by an operational expert. The way the model uses argumentation is reflected in real life where information is exchanged. Given the fact that supervisor operators are required to be capable of operating as an approach as well as tower supervisor, the operators are capable of interpreting information from the opponent. The fast negotiation interactions make the decision-making interaction highly dynamic and eliminate the need for time-based tactics during the negotiation.

9.2 Contribution

The main aim of this research was to find a way to model the decision-making interaction for the timing of a runway reconfiguration and gain knowledge about the phenomenon in the process. As a result of this research, a step has been taken towards formalising this decision making interaction. Developing a model that combines anticipation capabilities as well as internal and external argumentation capabilities is to the best knowledge of the author something that has not been done before for an air transportation scenario. Developing the model by trying to construct intelligent artefacts that emulate these cognitive capabilities has resulted in two major contributions. Firstly it has aided in exploring the process of decision-making for a runway reconfiguration in detail and understanding what aspects are involved. Secondly, it has resulted in a formal model for performing the timing of a runway reconfiguration that provides explainable solutions.

Especially in the air transport industry, acceptability for innovations and automated systems is due to high safety requirements and the complexity of operations in general low. By developing a model in such a way that the description is very low level and natural, acceptability is deemed to be higher. This because it is easier to understand how the model works when it is more relatable to the workings of the actual system. When comparing solutions from machine learning models, deep learning models and optimisation models with the solutions resulting from argumentation procedures used in the developed model, more interpretable and traceable results are observed. A solution can be traced back to the information in the argumentation line that is responsible for rendering this solution. Given that the way the solution is established is in line with how a human reason in scenarios with incomplete information, makes the results even more relatable.

Apart from the control model, that combines argumentation-based negotiation and defeasible logic programming methods, the research has yielded an inbound-outbound model for AAS. These models should be regarded as proxy models, given that these models are constructed with various assumptions and simplifications. The models are however designed following an agent-based approach which makes it easy to introduce new elements that increase the sophistication of the models. Additionally, the models do possess sufficient sophistication for providing delay estimations for both systems.

The findings that have resulted from modelling the timing of the runway reconfiguration phenomenon can be generalised for existing meso models and macro models related to runway configuration management. In the literature research described in section 3.1.1, it was found that the timing aspect of runway configuration management is usually addressed by introducing inertia factors, cost factors or based on historical data. Additionally, the reconfiguration is often based on the weather rather than demand conditions. From the research, it is clear that a decision for the timing of a runway reconfiguration is mainly based on delay values of both systems in combination with a predefined peak planning. For the existing macro models, the exact flight that will be selected for reconfiguration is not relevant. It is however relevant that the times for reconfiguration are selected correctly. To do so it is hypothesized that the existing models could benefit from combining the existing factors with delay estimates for both systems, that are used to derive satisficing solutions for the moment of configuration changes. Especially for models that are focussed at the tactical time horizon, this is expected to aid in improving the accuracy for the moment of runway reconfigurations.

9.3 Recommendations for Further Research

This research on the timing of a runway reconfiguration should be considered as a first step towards developing a better understanding of this aspect. The complexity of the system underlying the runway reconfiguration phenomenon and the limited scope of this research leaves various aspects that can be improved upon or supplemented to the research line. In this regard, the bottom-up approach of developing the model yields a highly modular is relatively easy to expand upon. This could be by adding new quantitative aspects as well as qualitative aspects.

Domain Model

The domain model consisting of an inbound-outbound system is a simplified version of the real-world system. This is exemplified by the assumption that no taxiway interactions occur in the ground system. Aspects like taxiway interactions, availability of air traffic control operators, more sophisticated trajectory prediction and more detailed inbound traffic control models could aid in the completeness of the sensory information. Even though the model is capable of handling incomplete information elegantly, improved information quality will benefit decision-making.

Apart from information quality, another way of improving upon the current working of the model would be by integrating the working of the domain model with the decision-making in the control model. In the current model, the domain model merely acts as a source of sensory information and the only interaction that occurs is the control of the runway configuration. When a supervisor agent receives meta-information about the reasons that made the opponent supervisor agent reject the proposal, a solution provided by the domain model to mitigate these reasons could potentially lead to better agreements. Given the explainable nature of the arguments used in the interactions, this could be implemented without adjusting the main elements of the control model.

Stochastic Behaviour

In the current model, decisions are based upon deterministic values. Especially for predictions used in the model, uncertainty plays an important role and should be incorporated using random variables. A crucial aspect for the delay buildup in the inbound outbound system is the arrival and departure times. By incorporating stochastic behaviour for the arrival times and turnaround times in the model, these effects could be better assessed. The DeLP framework used in the current model does not allow for explicit possibilistic information. To introduce possibilistic information in the current defeasible logic reasoning mechanism a reification step could be used. Alternatively, an extension of the DeLP framework called P-DeLP could be introduced [21]. This framework extends the DeLP framework by allowing the introduction of uncertainty and fuzzy logic at the object-language level.

Dynamic Data

During the negotiation interactions for deciding on the moment of a runway reconfiguration, sensory data is in reality continuously updated. In the model, this aspect is currently not included. It is assumed that the observed state of the system remains valid throughout the negotiation interaction. Including updates of the information throughout the negotiations will improve the decision making, given that information could arise that has a significant effect on the desirability of certain agreements. This dynamic aspect works well with the argumentation procedures. The heuristic and non-monotonicity of these procedures allow for handling new information without requiring a new negotiation interaction for the updated knowledge of the world.

A

Schematic Rules

ID	Loc.	DeLP Notation	Meaning
SR1	L1	$ProposeArrActivation(ArrFlight) \prec EligibleOption(ArrFlight)$	If an arrival flight is an eligible option for activation, an arrival runway activation should be proposed such that an arrival flight is the first arrival on the activated runway
SR2	L2	$ProposeDepActivation(DepFlight) \prec EligibleOption(DepFlight)$	If a departure flight is an eligible option for activation, a departure runway activation should be proposed such that a departure flight is the first departure on the activated runway
SR3	L3	$ProposeArrDeactivation(ArrFlight) \prec EligibleOption(ArrFlight)$	If an arrival flight is an eligible option for deactivation, an arrival runway deactivation should be proposed such that an arrival flight lands when the secondary runway is deactivated
SR4	L4	$ProposeDepDeactivation(DepFlight) \prec EligibleOption(DepFlight)$	If the departure flight is an eligible option for deactivation, a departure runway deactivation should be proposed such that a departure flight starts when the secondary runway is deactivated
SR5	L5	$ProposeArrActivationDepDeactivation(ArrFlight, DepFlight) \prec EligibleOption(ArrFlight)$	If the arrival flight is an eligible option for the activation/deactivation combination, this activation/deactivation should be proposed
SR6	L6	$ProposeArrDeactivationDepActivation(ArrFlight, DepFlight) \prec EligibleOption(ArrFlight)$	If the arrival flight is an eligible option for the arrival deactivation/ departure activation combination, this arrival deactivation/ departure activation combination should be proposed

ID	Loc.	DeLP Notation	Meaning
SR7	L7	$AcceptArrActivation(ArrFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight)$	If the arrival flight is an eligible option for the runway activation, this activation option should be accepted
SR8	L8	$AcceptDepActivation(DepFlight) \mathrel{-\!\!\prec} EligibleOption(DepFlight)$	If the departure flight is an eligible option for the runway activation, this activation option should be accepted
SR9	L9	$AcceptArrDeactivation(ArrFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight)$	If the arrival flight is an eligible option for the runway deactivation, this deactivation option should be accepted
SR10	L10	$AcceptDepDeactivation(DepFlight) \mathrel{-\!\!\prec} EligibleOption(DepFlight)$	If the departure flight is an eligible option for the runway deactivation, this deactivation option should be accepted
SR11	L11	$AcceptArrActivationDepDeactivation(ArrFlight, DepFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight)$	If the arrival flight is an eligible option for the arrival activation/ departure deactivation combination, this arrival activation/ departure deactivation option should be accepted
SR12	L12	$AcceptArrDeactivationDepActivation(ArrFlight, DepFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight)$	If the arrival flight is an eligible option for the arrival deactivation/ departure activation combination, this arrival-deactivation/departure-activation option should be accepted
SR13	L1,L3	$\sim ProposeArrActivation(ArrFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight), ConflictingOption(ArrFlight)$	Do not propose an arrival flight option if it is eligible for proposal but breaks the bunch while another option does not
SR14	L5	$\sim ProposeArrActivationDepDeactivation(ArrFlight, Depflight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight), ConflictingOption(ArrFlight)$	Do not propose a combination of an arrival flight option and a departure flight option, if teh arrival flight option is eligible for proposal but breaks the bunch while another option does not
SR15	L1,L3, L5,L6, L9	$\sim EligibleOption(ArrFlight) \mathrel{-\!\!\prec} NoHoldingPatterns(ArrFlight)$	When no holding patterns are expected as a result of the arrival flight option for reconfiguration, the arrival flight is an eligble option for proposal
SR16	L1,L3, L5,L6	$ConflictingOption(ArrFlight) \mathrel{-\!\!\prec} NoHoldingPatterns(ArrAlternative), BreakingBunch(ArrFlight), NotBreakingBunch(ArrAlternative)$	An arrival flight option must be considered as conflicting when an alternative option exists that does not break the bunch and does not result in holding patterns
SR17	L1,L5	$EligibleOption(ArrFlight) \mathrel{-\!\!\prec} HoldingPatterns(ArrFlight), LowTotalDelay(ArrFlight)$	If a selected flight option results in a holding pattern, but the total delay from selecting this option is low it is an eligible option for proposal

ID	Loc.	DeLP Notation	Meaning
SR18	L2,L5	$\sim ProposeDepActivation(DepFlight) \mathrel{-\!\!\prec}$ $EligibleOption(DepFlight), BlockingAlternative(DepFlight)$	If the departure flight is an eligible option, but is blocking a previous flight departing in the same direction and an alternative exists, this departure option must not be proposed
SR19	L2,L4 ,L5,L6	$\sim ProposeDepActivation(DepFlight) \mathrel{-\!\!\prec}$ $EligibleOption(DepFlight), PerformanceAlternative(DepFlight)$	If the departure flight is an eligible option, but a better alternative exists in terms of delay, this option should not be proposed
SR20	L2,L4 ,L5,L6	$PerformanceAlternative(DepFlight) \mathrel{-\!\!\prec}$ $TotalDelayHigher(DepFlight, DepAlternative),$ $NotBlockingPreviousFlight(DepAlternative),$ $NoSlotsMissed(DepAlternative)$	A performance alternative for a departure flight exists if this alternative is not blocking a previous flight and no missed slots resulting from selecting this alternative
SR21	L2,L5 ,L6	$BlockingAlternative(DepFlight) \mathrel{-\!\!\prec}$ $BlockingPreviousFlight(DepFlight),$ $NotBlockingPreviousFlight(DepAlternative),$ $NoSlotsMissed(DepAlternative)$	An alternative exist for a blocking option, if the alternative option results in no slots missed and the alternative is not blocking a previous departure flight
SR22	L2,L4 ,L5,L6	$EligibleOption(DepFlight) \mathrel{-\!\!\prec}$ $SlotsMissed(DepFlight), LowTotalDelay(DepFlight)$	A departure for which slots are missed, but a low total delay results should be regarded an eligible option
SR23	L2,L4 ,L5,L6 ,L10	$EligibleOption(DepFlight) \mathrel{-\!\!\prec} NoSlotsMissed(DepFlight)$	A departure flight is an eligible option for reconfiguration if no assigned slots are missed as a result of selecting this option
SR24	L2,L4 ,L5,L6	$\sim EligibleOption(DepFlight) \mathrel{-\!\!\prec} GroundQueues(DepFlight)$	A departure flight is not an eligible option if selecting this option for a runway reconfiguration results in ground queues
SR25	L2,L4 ,L5,L6	$EligibleOption(DepFlight) \mathrel{-\!\!\prec}$ $GroundQueues(DepFlight), NorthernConfiguration(DepFlight)$	A departure flight is an eligible option if selecting this option for a runway reconfiguration results in ground queues, but a northern runway configuration is used
SR26	L7	$AcceptArrActivation(ArrFlight) \mathrel{-\!\!\prec}$ $ProposeArrActivation(ArrFlight)$	If an arrival runway activation is proposed, this should normally be accepted
SR27	L8	$AcceptDepActivation(DepFlight) \mathrel{-\!\!\prec}$ $ProposeDepActivation(DepFlight)$	If a departure runway activation is proposed, this should normally be accepted
SR28	L7	$AcceptArrActivation(ArrFlight) \leftarrow$ $NoControllersAvailable(ArrFlight)$	If no air traffic controllers are available, the arrival runway activation should not be accepted
SR29	L8	$AcceptDepActivation(DepFlight) \leftarrow$ $NoControllersAvailable(DepFlight)$	If no air traffic controllers are available, the departure runway activation should not be accepted
SR30	L11	$\sim AcceptArrActivationDepDeactivation(ArrFlight, DepFlight) \mathrel{-\!\!\prec}$ $EligibleOption(ArrFlight),$ $BetterDepartureAlternative(DepFlight),$ $UnacceptableFourthRunwayTime(ArrFlight)$	Do not accept the combination of an arrival runway activation and a departure runway deactivation, if this leads to an unacceptable fourth runway time and a better alternative is available for the deaprture deactivation option

ID	Loc.	DeLP Notation	Meaning
SR31	L12	$\sim AcceptArrDeactivationDepActivation(ArrFlight, DepFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight),\ BetterArrivalAlternative(ArrFlight),\ UnacceptableFourthRunwayTime(DepFlight)$	Do not accept the combination of an arrival runway deactivation and a departure runway activation, if this leads to an unacceptable fourth runway time and a better alternative is available for the arrival deactivation option
SR32	L11	$\sim BetterDepartureAlternative(DepFlight) \leftarrow Rejected(Alternative)$	If a departure alternative is rejected in the negotiation interaction it should not be considered a better alternative for the departure flight
SR33	L12	$\sim BetterArrivalAlternative(ArrFlight) \leftarrow Rejected(Alternative)$	If an arrival alternative is rejected in the negotiation interaction it should not be considered a better alternative for the arrival flight
SR34	L11	$\sim BetterDepartureAlternative(DepFlight) \leftarrow TimeLater(DepFlight, Alternative)$	If a departure flight option is a later option than an alternative, a better alternative exists in terms of fourth runway time
SR35	L12	$\sim BetterArrivalAlternative(ArrFlight) \leftarrow TimeLater(ArrFlight, Alternative)$	If an arrival flight option is a later option than an alternative, a better alternative exists in terms of fourth runway time

B

Protocol

Protocol

L1: The **ProposeArrActivation()** locution:

Locution: ProposeArrActivation(S, α) where $\alpha \in AF$.
Preconditions:
A supervisor agent S is allowed to utter locution **L1** for the duration of the communication act, when the state property $allowed\_locution(S, locution\_L1)$ holds. This external state property follows from the dynamic property:

$$
\begin{aligned}
& external(S)|\; responsibility(S, approach) \wedge current\_configuration(off\_peak) \\
& \wedge\; upcoming\_peak(inbound\_peak) \\
& \twoheadrightarrow_{0,0,1,t_{communication\_act}}\; external(S)|allowed\_locution(S, locution\_L1)
\end{aligned}
$$

Meaning: A **ProposeArrActivation** locution expresses the desire of a supervisor agent S to activate an additional arrival runway. The term α denotes the first arrival flight option to use the newly activated runway and AF denotes the non-empty set of generated arrival flight options. This locution implies that no departure runway needs to be deactivated for this reconfiguration.
Response: Upon receiving locution **L1**, this must be responded with locution **L7**.
Proposal store update: No proposal store update required.
Knowledge base update: For locution **L1** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L1** for the approach supervisor agent, denoted by $R_{S_{APP},L1}$, is:

$$
R_{S_{APP},L1} = \left\{ \begin{array}{l}
\{ProposeArrActivation(ArrFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight)\} \\
\{\sim ProposeArrActivation(ArrFlight) \mathrel{-\!\!\prec} \sim EligibleOption(ArrFlight)\} \\
\{\sim ProposeArrActivation(ArrFlight) \mathrel{-\!\!\prec} EligibleOption(ArrFlight),\; ConflictingOption(ArrFlight)\} \\
\{EligibleOption(ArrFlight) \mathrel{-\!\!\prec} NoHoldingPatterns(ArrFlight)\} \\
\{ConflictingOption(ArrFlight) \mathrel{-\!\!\prec} NoHoldingPatterns(ArrAlternative), \\
BreakingBunch(ArrFlight),\; NotBreakingBunch(ArrAlternative)\} \\
\{EligibleOption(ArrFlight) \mathrel{-\!\!\prec} HoldingPatterns(ArrFlight), LowTotalDelay(ArrFlight)\}
\end{array} \right\}
$$

L2: The **ProposeDepActivation()** locution:

Locution: ProposeDepActivation(S, δ) where $\delta \in DF$.
Preconditions:
A supervisor agent S is allowed to utter locution **L2** for the duration of the communication act, when the state property $allowed\_locution(S, locution\_L2)$ holds. This external state property follows from the dynamic property:

$$
\begin{aligned}
& external(S)|\; responsibility(S, tower) \wedge current\_configuration(off\_peak) \\
& \wedge\; upcoming\_peak(outbound\_peak) \\
& \twoheadrightarrow_{0,0,1,t_{communication\_act}}\; external(S)|allowed\_locution(S, locution\_L2)
\end{aligned}
$$

Meaning: A **ProposeDepActivation** locution expresses the desire of a supervisor agent S to activate an additional departure runway. The term δ denotes the first departure flight option to use the newly activated departure runway and DF denotes the non-empty set of generated departure flight options. This locution implies that no arrival runway needs to be deactivated for this reconfiguration.
Response: Upon receiving locution **L2**, this must be responded with locution **L8**.
Proposal store update: No proposal store update required.
Knowledge base update: For locution **L2** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L2** for the tower supervisor agent, denoted by $R_{S_{TWR},L2}$, is:

$$R_{L2,tower} = \left\{ \begin{array}{l}
\{\mathit{ProposeDepActivation}(\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight})\} \\
\{\sim\mathit{ProposeDepActivation}(\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight}), \mathit{BlockingAlternative}(\mathit{DepFlight})\} \\
\{\sim\mathit{ProposeDepActivation}(\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight}), \mathit{PerformanceAlternative}(\mathit{DepFlight})\} \\
\{\sim\mathit{ProposeDepActivation}(\mathit{DepFlight}) \prec \sim\mathit{EligibleOption}(\mathit{DepFlight})\} \\
\{\mathit{PerformanceAlternative}(\mathit{DepFlight}) \prec \mathit{TotalDelayHigher}(\mathit{DepFlight}, \mathit{DepAlternative}), \\
\mathit{NotBlockingPreviousFlight}(\mathit{DepAlternative}), \mathit{NoSlotsMissed}(\mathit{DepAlternative})\} \\
\{\mathit{BlockingAlternative}(\mathit{DepFlight}) \prec \mathit{BlockingPreviousFlight}(\mathit{DepFlight}), \\
\mathit{NotBlockingPreviousFlight}(\mathit{DepAlternative}), \mathit{NoSlotsMissed}(\mathit{DepAlternative})\} \\
\{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{SlotsMissed}(\mathit{DepFlight}), \mathit{LowTotalDelay}(\mathit{DepFlight})\} \\
\{\sim\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{SlotsMissed}(\mathit{DepFlight})\} \\
\{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{NoSlotsMissed}(\mathit{DepFlight})\} \\
\{\sim\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{GroundQueues}(\mathit{DepFlight})\} \\
\{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{GroundQueues}(\mathit{DepFlight}), \mathit{NorthernConfiguration}(\mathit{DepFlight})\}
\end{array} \right\}$$

L3: The **ProposeArrDeactivation()** locution:

Locution: **ProposeArrDeactivation**(S, α) where $\alpha \in \{AF, \cdot\}$.
Preconditions:
A supervisor agent S is allowed to utter locution **L3** for the duration of the communication act, when the state property $allowed\_locution(S, locution\_L3)$ holds. This external state property follows from the following dynamic properties:

$$\begin{array}{l}
external(S)|\ responsibility(S, tower) \wedge current\_configuration(double\_peak) \\
\wedge\ upcoming\_peak(outbound\_peak) \\
\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_locution(S, locution\_L3)
\end{array}$$

$$\begin{array}{l}
external(S)|\ responsibility(S, approach) \wedge received\_locution(S, locution\_L3) \\
\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_locution(S, locution\_L3)
\end{array}$$

Meaning: A **ProposeArrDeactivation** locution expresses the desire of a supervisor agent S to de-activate an active arrival runway. The term α denotes the last flight that uses the secondary landing runway. The variable AF denotes the non-empty set of generated arrival flight options and $\cdot$ denotes a generic value for the flight option. A generic value will be chosen if the supervisor agent S has no knowledge about the system, inbound or outbound, for which a runway is proposed to be deactivated.
Response: Upon receiving locution **L3**, this must be responded with locution **L3** or **L9**.
Proposal store update: When uttering this locution, the proposal store must be updated with the literal $proposed(S, \alpha)$.
Knowledge base update: For locution **L3** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L3** for the tower and approach supervisor agent, denoted by $R_{S_{TWR},L3}$ and $R_{S_{APP},L3}$, is:

$$R_{S_{TWR},L3} = \left\{ \ \{\mathit{ProposeArrDeactivation}(\mathit{ArrFlight}) \prec \mathit{EligibleOption}(\mathit{ArrFlight})\} \ \right\}$$

$$R_{S_{APP},L3} = \left\{ \begin{array}{l}
\{\mathit{ProposeArrDeactivation}(\mathit{ArrFlight}) \prec \mathit{EligibleOption}(\mathit{ArrFlight})\} \\
\{\sim\mathit{ProposeArrDeactivation}(\mathit{ArrFlight}) \prec \mathit{EligibleOption}(\mathit{ArrFlight}), \mathit{ConflictingOption}(\mathit{ArrFlight})\} \\
\{\mathit{EligibleOption}(\mathit{ArrFlight}) \prec \mathit{NoHoldingPatterns}(\mathit{ArrFlight})\} \\
\{\mathit{ConflictingOption}(\mathit{ArrFlight}) \prec \mathit{NoHoldingPatterns}(\mathit{ArrAlternative}), \\
\mathit{BreakingBunch}(\mathit{ArrFlight}),\ \mathit{NotBreakingBunch}(\mathit{ArrAlternative})\}
\end{array} \right\}$$

L4: The **ProposeDepDeactivation()** locution:

Locution: ProposeDepDeactivation(S, δ) where $\delta \in \{DF, \cdot\}$.

Preconditions:

A supervisor agent S is allowed to utter locution **L4** for the duration of the communication act, when the state property $allowed\_location(S, locution\_L4)$ holds. This external state property follows from the following dynamic properties:

$$external(S)|\ responsibility(S, approach)\ \wedge\ current\_configuration(double\_peak)$$
$$\wedge\ upcoming\_peak(inbound\_peak)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_location(S, locution\_L4)$$

$$external(S)|\ responsibility(S, tower)\ \wedge\ received\_locution(S, locution\_L4)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_location(S, locution\_L4)$$

Meaning: A **ProposeDepDeactivation** locution expresses the desire of a supervisor agent S to de-activate a active departure runway. The term δ denotes the last flight that uses the secondary landing runway. The variable DF denotes the non-empty set of generated departure flight options and $\cdot$ denotes a generic value for the flight option. A generic value will be chosen if the supervisor agent S has no knowledge about the system, inbound or outbound, for which a runway is proposed to be deactivated.

Response: Upon receiving locution **L4**, this must be responded with locution **L4** or **L10**.

Proposal store update: When uttering this locution, the proposal store must be updated with the literal $proposed(S, \delta)$.

Knowledge base update: For locution **L4** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L4** for the tower and approach supervisor agent, denoted by $R_{S_{TWR},L4}$ and $R_{S_{APP},L4}$, is:

$$R_{S_{TWR},L4} = \left\{ \begin{array}{l} \{\sim ProposeDepDeactivation(DepFlight) \prec EligibleOption(DepFlight), \\ PerformanceAlternative(DepFlight)\} \\ \{\sim ProposeDepDeactivation(DepFlight) \prec \sim EligibleOption(DepFlight)\} \\ \{PerformanceAlternative(DepFlight) \prec TotalDelayHigher(DepFlight, DepAlternative), \\ NoSlotsMissed(DepAlternative)\} \\ \{EligibleOption(DepFlight) \prec SlotsMissed(DepFlight), LowTotalDelay(DepFlight)\} \\ \{\sim EligibleOption(DepFlight) \prec SlotsMissed(DepFlight)\} \\ \{EligibleOption(DepFlight) \prec NoSlotsMissed(DepFlight)\} \\ \{\sim EligibleOption(DepFlight) \prec GroundQueues(DepFlight)\} \\ \{EligibleOption(DepFlight) \prec GroundQueues(DepFlight), NorthernConfiguration(DepFlight)\} \end{array} \right\}$$
$$R_{S_{APP},L4} = \left\{\ \{ProposeDepDeactivation(DepFlight) \prec EligibleOption(DepFlight)\}\ \right\}$$

L5: The **ProposeArrActivationDepDeactivation()** locution:

Locution: ProposeArrActivationDepDeactivation(S, α, δ) where $\alpha \in AF$ and $\delta \in \{DF, \cdot\}$.

Preconditions:

A supervisor agent S is allowed to utter locution **L5** for the duration of the communication act, when the state property $allowed\_location(S, locution\_L5)$ holds.This external state property follows from the following dynamic properties:

$$external(S)|\ responsibility(S, approach)\ \wedge\ current\_configuration(outbound\_peak)$$
$$\wedge\ upcoming\_peak(inbound\_peak)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_location(S, locution\_L5)$$

$$external(S)|responsibility(S, tower)\ \wedge\ received\_locution(S, locution\_L5)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_location(S, locution\_L5)$$

Meaning: A **ProposeArrActivationDepDeactivation** locution expresses the desire of a supervisor agent S to activate an additional arrival runway and to deactivate an active departure runway. The term α denotes the first arrival flight to use the newly activated landing runway and AF denotes the non-empty set of generated arrival flight options. The term δ denotes the last departure flight that uses the secondary departure runway. This can either be the generic value, $\cdot$, if no knowledge about departure options is available to the supervisor agent or a flight in the non-empty set of departure flight options DF.

Response: Upon receiving locution **L5**, this must be responded with locution **L5** or **L11**.

Proposal store update: When uttering locution **L5**, the proposal store must be updated with the literal $proposed(S,\alpha,\delta)$.

Knowledge base update: For locution **L5** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L5** for the tower and approach supervisor agent, denoted by $R_{S_{TWR},L5}$ and $R_{S_{APP},L5}$, is:

$$R_{S_{APP},L5} = \left\{ \begin{array}{l} \{\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{ArrFlight})\} \\ \{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \sim\mathit{EligibleOption}(\mathit{ArrFlight})\} \\ \{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{ArrFlight}), \\ \mathit{ConflictingOption}(\mathit{ArrFlight})\} \\ \{\mathit{EligibleOption}(\mathit{ArrFlight}) \prec \mathit{NoHoldingPatterns}(\mathit{ArrFlight})\} \\ \{\mathit{ConflictingOption}(\mathit{ArrFlight}) \prec \mathit{NoHoldingPatterns}(\mathit{ArrAlternative}), \\ \mathit{BreakingBunch}(\mathit{ArrFlight}),\ \mathit{NotBreakingBunch}(\mathit{ArrAlternative})\} \\ \{\mathit{EligibleOption}(\mathit{ArrFlight}) \prec \mathit{HoldingPatterns}(\mathit{ArrFlight}),\mathit{LowTotalDelay}(\mathit{ArrFlight})\} \end{array} \right\}$$

$$R_{S_{TWR},L5} = \left\{ \begin{array}{l} \{\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight})\} \\ \{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight}), \\ \mathit{BlockingAlternative}(\mathit{DepFlight})\} \\ \{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight}), \\ \mathit{PerformanceAlternative}(\mathit{DepFlight})\} \\ \{\sim\mathit{ProposeArrActivationDepDeactivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \sim\mathit{EligibleOption}(\mathit{DepFlight})\} \\ \{\mathit{PerformanceAlternative}(\mathit{DepFlight}) \prec \mathit{TotalDelayHigher}(\mathit{DepFlight},\mathit{DepAlternative}), \\ \mathit{NotBlockingPreviousFlight}(\mathit{DepAlternative}),\mathit{NoSlotsMissed}(\mathit{DepAlternative})\} \\ \{\mathit{BlockingAlternative}(\mathit{DepFlight}) \prec \mathit{BlockingPreviousFlight}(\mathit{DepFlight}), \\ \mathit{NotBlockingPreviousFlight}(\mathit{DepAlternative}),\mathit{NoSlotsMissed}(\mathit{DepAlternative})\} \\ \{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{SlotsMissed}(\mathit{DepFlight}),\mathit{LowTotalDelay}(\mathit{DepFlight})\} \\ \{\sim\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{SlotsMissed}(\mathit{DepFlight})\} \\ \{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{NoSlotsMissed}(\mathit{DepFlight})\} \\ \{\sim\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{GroundQueues}(\mathit{DepFlight})\} \\ \{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{GroundQueues}(\mathit{DepFlight}),\mathit{NorthernConfiguration}(\mathit{DepFlight})\} \end{array} \right\}$$

L6: The **ProposeArrDeactivationDepActivation()** locution:

Locution: ProposeArrDeactivationDepActivation(S,δ) where $\alpha \in \{AF,\cdot\}$ and $\delta \in DF$.

Preconditions:

A supervisor agent S is allowed to utter locution L6 for the duration of the communication act, when the state property $allowed\_locution(S,locution\_L6)$ holds. This state property follows from the following dynamic property:

$$\begin{array}{l} external(S)|\ responsibility(S,tower)\ \wedge\ current\_configuration(inbound\_peak) \\ \wedge\ upcoming\_peak(outbound\_peak) \\ \twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_locution(S,locution\_L6) \end{array}$$

$$\begin{array}{l} external(S)|responsibility(S,approach)\ \wedge\ received\_locution(S,locution\_L6) \\ \twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_locution(S,locution\_L6) \end{array}$$

Meaning: A **ProposeArrDeactivationDepActivation** locution expresses the desire of a supervisor agent S to deactivate an active arrival runway and to activate an additional departure runway. The term δ denotes the first departure flight that uses the newly activated departure runway and DF denotes the non-empty set of departure flight options. The term α denotes the last arrival flight that uses the secondary arrival runway. This can either be the generic value, $\cdot$, if the supervisor agent has no knowledge about arrival flight options or an arrival flight option in the non-empty set of arrival flight options AF.

Response: Upon receiving locution **L6**, this must be responded with locution **L6** or **L12**.

Proposal store update: When uttering locution **L6**, the proposal store must be updated with the literal $proposed(S,\alpha,\delta)$.

Knowledge base update: For locution **L6** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for L1, is dependent on the responsibility of supervisor agent S. The set of schematic rules that must be added for **L6** for the tower and approach supervisor agent, denoted by $R_{S_{TWR},L6}$ and $R_{S_{APP},L6}$, is:

$$R_{S_{APP},L6} = \left\{ \begin{array}{l}
\{\mathit{ProposeArrDeactivationDepActivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{ArrFlight})\} \\
\{\sim\mathit{ProposeArrDeactivationDepActivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \sim\mathit{EligibleOption}(\mathit{ArrFlight})\} \\
\{\mathit{EligibleOption}(\mathit{ArrFlight}) \prec \mathit{NoHoldingPatterns}(\mathit{ArrFlight})\} \\
\{\mathit{ConflictingOption}(\mathit{ArrFlight}) \prec \mathit{NoHoldingPatterns}(\mathit{ArrAlternative}), \\
\mathit{BreakingBunch}(\mathit{ArrFlight}),\ \mathit{NotBreakingBunch}(\mathit{ArrAlternative})\} \\
\{\mathit{EligibleOption}(\mathit{ArrFlight}) \prec \mathit{HoldingPatterns}(\mathit{ArrFlight}),\mathit{LowTotalDelay}(\mathit{ArrFlight})\}
\end{array} \right\}$$

$$R_{S_{TWR},L6} = \left\{ \begin{array}{l}
\{\mathit{ProposeArrDeactivationDepActivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight})\} \\
\{\sim\mathit{ProposeArrDeactivationDepActivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight}), \\
\mathit{BlockingAlternative}(\mathit{DepFlight})\} \\
\{\sim\mathit{ProposeArrDeactivationDepActivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \mathit{EligibleOption}(\mathit{DepFlight}), \\
\mathit{PerformanceAlternative}(\mathit{DepFlight})\} \\
\{\sim\mathit{ProposeArrDeactivationDepActivation}(\mathit{ArrFlight},\mathit{DepFlight}) \prec \sim\mathit{EligibleOption}(\mathit{DepFlight})\} \\
\{\mathit{PerformanceAlternative}(\mathit{DepFlight}) \prec \mathit{TotalDelayHigher}(\mathit{DepFlight},\mathit{DepAlternative}), \\
\mathit{NotBlockingPreviousFlight}(\mathit{DepAlternative}),\mathit{NoSlotsMissed}(\mathit{DepAlternative})\} \\
\{\mathit{BlockingAlternative}(\mathit{DepFlight}) \prec \mathit{BlockingPreviousFlight}(\mathit{DepFlight}), \\
\mathit{NotBlockingPreviousFlight}(\mathit{DepAlternative}),\mathit{NoSlotsMissed}(\mathit{DepAlternative})\} \\
\{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{SlotsMissed}(\mathit{DepFlight}),\mathit{LowTotalDelay}(\mathit{DepFlight})\} \\
\{\sim\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{SlotsMissed}(\mathit{DepFlight})\} \\
\{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{NoSlotsMissed}(\mathit{DepFlight})\} \\
\{\sim\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{GroundQueues}(\mathit{DepFlight})\} \\
\{\mathit{EligibleOption}(\mathit{DepFlight}) \prec \mathit{GroundQueues}(\mathit{DepFlight}),\mathit{NorthernConfiguration}(\mathit{DepFlight})\}
\end{array} \right\}$$

L7: The **AcceptArrActivation()** locution:

Locution: **AcceptArrActivation**(S,α) where $\alpha \in \{AF,\emptyset\}$.

Preconditions:
A supervisor agent S is allowed to utter locution **L7** for the duration of the communication act, when the state property $allowed\_locution(S,locution\_L7)$ holds. This state property follows from the following dynamic property:

$$\begin{array}{l}
external(S)|received\_locution(S,locution\_L1) \\
\twoheadrightarrow_{0,0,1,t_{communication\_act}}\ external(S)|allowed\_locution(S,locution\_L7)
\end{array}$$

Meaning: An **AcceptArrActivation** locution expresses a commitment to accept the proposed arrival runway activation. The term α denotes the last flight that uses the secondary landing runway and AF denotes the non-empty set of generated arrival flight options. If no option can be accepted, an empty set $\emptyset$ can be selected for the α term.

Response: No response required.

Proposal store update: No proposal store update required.

Knowledge base update: For locution **L7** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L7** for the tower supervisor agent, denoted by $R_{S_{TWR},L7}$, is:

$$R_{S_{APP},L7} = \left\{ \ \{AcceptArrActivation(ArrFlight) \prec EligibleOption(ArrFlight)\} \ \right\}$$

$$R_{S_{TWR},L7} = \left\{ \begin{array}{l} \{AcceptArrActivation(ArrFlight) \prec ProposeArrActivation(ArrFlight)\} \\ \{\sim AcceptArrActivation(ArrFlight) \leftarrow NoControllersAvailable(ArrFlight)\} \end{array} \right\}$$

L8: The **AcceptDepActivation()** locution:

Locution: AcceptDepActivation(S, δ) where $\delta \in \{DF, \emptyset\}$.
Preconditions:
A supervisor agent S is allowed to utter locution **L8** for the duration of the communication act, when the state property $allowed\_locution(S, locution\_L8)$ holds. This state property follows from the following dynamic property:

$$external(S)|received\_locution(S, locution\_L2)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}} \quad external(S)|allowed\_locution(S, locution\_L8)$$

Meaning: An **AcceptDepActivation** locution expresses a commitment to accept the proposed departure runway activation. The term δ denotes the last flight that uses the secondary departure runway and DF denotes the non-empty set of generated departure flight options. If no option can be accepted, an empty set $\emptyset$ can be selected for the δ term.
Response: No response required.
Proposal store update: No proposal store update required.
Knowledge base update: For locution **L8** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L8** for the approach supervisor agent, denoted by $R_{S_{APP},L8}$, is:

$$R_{S_{APP},L8} = \left\{ \begin{array}{l} \{AcceptDepActivation(DepFlight) \prec ProposeDepActivation(DepFlight)\} \\ \{\sim AcceptDepActivation(DepFlight) \leftarrow NoControllersAvailable(DepFlight)\} \end{array} \right\}$$

$$R_{S_{TWR},L8} = \left\{ \ \{AcceptDepActivation(DepFlight) \prec EligibleOption(DepFlight)\} \ \right\}$$

L9: The **AcceptArrDeactivation(.)** locution:

Locution: AcceptArrDeactivation(S, α) where $\alpha \in \{AF, \emptyset\}$.
Preconditions:
A supervisor agent S is allowed to utter locution **L9** for the duration of the communication act, when the state property $allowed\_locution(S, locution\_L9)$ holds. This state property follows from the following dynamic property:

$$external(S)|received\_locution(S, locution\_L3)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}} \quad external(S)|allowed\_locution(S, locution\_L9)$$

Meaning: An **AcceptArrDeactivation** locution expresses the commitment of a supervisor agent S to accept the deactivation of an arrival runway. The term α denotes the last flight that uses the secondary landing runway and AF denotes the non-empty set of generated arrival flight options. If no suitable option can be found to accept, an empty set $\emptyset$ can be selected for the α term.
Response: No response required.
Proposal store update: No proposal store update required.
Knowledge base update: For locution **L9** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L9** for the tower and approach supervisor agent, denoted by $R_{S_{TWR},L9}$ and $R_{S_{APP},L9}$, is:

$$R_{S_{APP},L9} = \left\{ \begin{array}{l} \{AcceptArrDeactivation(ArrFlight) \prec EligibleOption(ArrFlight)\} \\ \{EligibleOption(ArrFlight) \prec NoHoldingPatterns(ArrFlight)\} \end{array} \right\}$$

$$R_{S_{TWR},L9} = \left\{ \ \{AcceptArrDeactivation(ArrFlight) \prec EligibleOption(ArrFlight)\} \ \right\}$$

L10: The **AcceptDepDeactivation(.)** locution:

Locution: AcceptDepDeactivation(S,δ) where $\delta \in \{DF,\emptyset\}$.
Preconditions:
A supervisor agent S is allowed to utter locution **L10** for the duration of the communication act when the state property $allowed\_locution(S,locution\_L10)$ holds. This state property follows from the following dynamic property:

$$external(S)|received\_locution(S,locution\_L4)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}} \quad external(S)|allowed\_locution(S,locution\_L10)$$

Meaning: An **AcceptDepDeactivation** locution expresses the commitment of a supervisor agent S to accept the deactivation of a departure runway. The term δ denotes the last flight that uses the secondary departure runway and DF denotes the non-empty set of generated departure flight options. If no suitable option can be found to accept, an empty set $\emptyset$ can be selected for the δ term.
Response: No response required.
Proposal store update: No proposal store update required.
Knowledge base update: For locution **L10** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L10** for the tower and approach supervisor agent, denoted by $R_{S_{TWR},L10}$ and $R_{S_{APP},L10}$, is:

$$R_{S_{APP},L10} = \left\{ \; \{AcceptDepDeactivation(DepFlight) \prec EligibleOption(DepFlight)\} \; \right\}$$
$$R_{S_{TWR},L10} = \left\{ \begin{array}{l} \{AcceptDepDeactivation(DepFlight) \prec EligibleOption(DepFlight)\} \\ \{EligibleOption(DepFlight) \prec NoSlotsMissed(DepFlight)\} \end{array} \right\}$$

L11: The **AcceptArrActivationDepDeactivation(.)** locution:

Locution: AcceptArrActivationDepDeactivation(S,α,δ) where $\alpha \in AF$ and $\delta \in \{DF,\emptyset\}$.
Preconditions:
A supervisor agent S is allowed to utter locution **L11** for the duration of the communication act when the state property $allowed\_locution(S,locution\_L11)$ holds. This state property follows from the following dynamic property:

$$external(S)|received\_locution(S,locution\_L5)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}} \quad external(S)|allowed\_locution(S,locution\_L11)$$

Meaning: An **AcceptArrActivationDepDeactivation** locution expresses the commitment of a supervisor agent for accepting an arrival runway activation and departure runway deactivation. The term α denotes the first flight that uses the newly activated arrival runway. The variable AF denotes the set of arrival flight options. The term δ denotes the last departure flight that uses the secondary departure runway. For this term, a flight can be selected from the set of departure flight options DF or an empty set $\emptyset$ if no option can be accepted.
Response: No response required.
Proposal store update: No proposal store update required.
Knowledge base update: For locution **L11** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L11** for the tower supervisor agent, denoted by $R_{S_{TWR},L11}$ is:

$$R_{S_{TWR},L11} = \left\{ \begin{array}{l} \{AcceptArrActivationDepDeactivation(ArrFlight,DepFlight) \prec EligibleOption(ArrFlight)\} \\ \{\sim AcceptArrActivationDepDeactivation(ArrFlight,DepFlight) \prec EligibleOption(ArrFlight), \\ BetterDepartureAlternative(DepFlight),UnacceptableFourthRunwayTime(ArrFlight)\} \\ \{BetterDepartureAlternative(DepFlight) \prec TimeLater(DepFlight,Alternative)\} \\ \{\sim BetterDepartureAlternative(DepFlight) \leftarrow Rejected(Alternative)\} \end{array} \right\}$$

L12: The **AcceptArrDeactivationDepActivation(.)** locution:

Locution: AcceptArrDeactivationDepActivation(S,α,δ) where $\alpha \in \{AF,\emptyset\}$ and $\delta \in DF$.

Preconditions:

A supervisor agent S is allowed to utter locution **L12** for the duration of the communication act when the state property $allowed\_locution(S, locution\_L12)$ holds. This state property follows from the following dynamic property:

$$external(S)|received\_locution(S, locution\_L6)$$
$$\twoheadrightarrow_{0,0,1,t_{communication\_act}} external(S)|allowed\_locution(S, locution\_L12)$$

Meaning: An **AcceptArrDeactivationDepActivation** locution expresses the commitment of a supervisor agent for accepting a departure runway activation and arrival runway deactivation. The term δ denotes the first flight that uses the newly activated departure runway. The variable DF denotes the set of departure flight options. The term α denotes the last arrival flight that uses the secondary arrival runway. For this term, a flight can be selected from the set of arrival flight options AF or an empty set $\emptyset$ if no option can be accepted.

Response: No response required.

Proposal store update: No proposal store update required.

Knowledge base update: For locution **L12** the knowledge base of supervisor agent S is updated with a set of schematic rules R, where $R \subseteq \mathcal{P}$. The set of schematic rules that must be added for **L12** for the approach supervisor agent, denoted by $R_{S_{APP},L12}$, is:

$$R_{S_{APP},L12} = \left\{ \begin{array}{l} \{AcceptArrDeactivationDepActivation(ArrFlight,DepFlight) \prec EligibleOption(ArrFlight)\} \\ \{\sim AcceptArrDeactivationDepActivation(ArrFlight,DepFlight) \prec EligibleOption(ArrFlight), \\ BetterArrivalAlternative(ArrFlight), UnacceptableFourthRunwayTime(DepFlight)\} \\ \{BetterArrivalAlternative(ArrFlight) \prec TimeLater(ArrFlight,Alternative)\} \\ \{\sim BetterArrivalAlternative(ArrFlight) \leftarrow Rejected(Alternative)\} \end{array} \right\}$$

C

Trajectory Prediction Model

For the trajectory prediction module, the process is started by defining the path that needs to be travelled and the speed at the start of the path. Using this path and the initial height, the altitude pattern can be determined for a continuous descent approach. Given the altitudes and assumed indicated airspeed values, the true airspeed can be determined using the equations C.1 and C.2 [27].

C.1 FIR boundary- IAF segment

Trajectories for the segment in ACC airspace are determined by the position of crossing the FIR boundary and the destined holding stack for the flight. Entry positions for the FIR boundary are discretised based on the angle towards the destined holding stack. For every 10 degrees of bearing a path is determined. A flight entering the FIR will be assigned the path that has the closest start position. This path consists of a sequence of points that are spaced at a distance of one nautical mile along the path, except for the last point which is spaced at a distance that remains after dividing the path into nautical mile segments.

Using the total distance of the segment and the altitude difference between the entry altitude and the altitude at the holding stack, the required slope for a continuous descent is determined. This slope is used to determine the altitude at every point of the segment. The altitude and location data for every point along the track facilitates retrieving the weather conditions for every point. These conditions consist of wind and temperature conditions, resulting from the weather model (see appendix E).

For the nominal ACC time prediction, it is assumed that the indicated airspeed at entering the Dutch FIR will be maintained. If this indicated airspeed is greater than the desired speed at the IAF, which is set to 250 kts, the speed is set to 250 kts for the last 10 NM of the segment. For an indicated airspeed less than 250 kts at entering the FIR, it is assumed that this indicated airspeed is maintained up until the IAF. Using this indicated airspeed pattern, the altitude and weather conditions, the indicated airspeed is converted to true airspeed using equation C.1. Using the wind conditions, this true airspeed is converted to ground speed. Now that the ground speed is determined for every point along the path, the time for every point along the path is determined by dividing the distance between two points by the average ground speed of the two points. This process is schematically depicted in figure C.1.

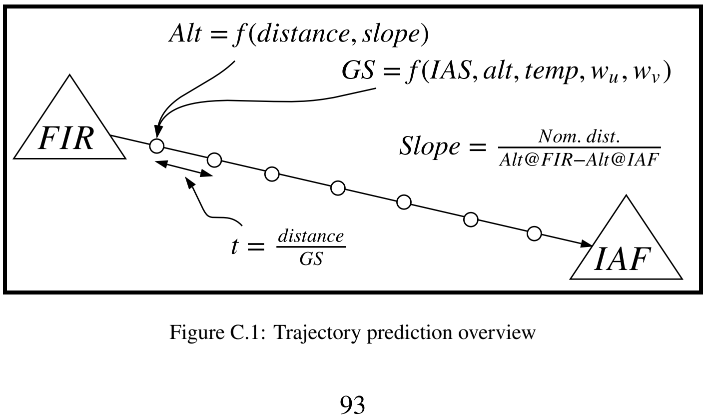


Figure C.1: Trajectory prediction overview

$$V_{TAS} = \left[\frac{2}{\mu}\frac{p}{\rho}\left\{\left(1+\frac{p_0}{p}\left[\left(1+\frac{\mu}{2}\frac{\rho_0}{p_0}V_{CAS}^2\right)^{\frac{1}{\mu}}-1\right]\right)^{\mu}-1\right\}\right]^{\frac{1}{2}} \tag{C.1}$$

$$V_{CAS} = \left[\frac{2}{\mu}\frac{p_0}{\rho_0}\left\{\left(1+\frac{p}{p_0}\left[\left(1+\frac{\mu}{2}\frac{\rho}{p}V_{TAS}^2\right)^{\frac{1}{\mu}}-1\right]\right)^{\mu}-1\right\}\right]^{\frac{1}{2}} \tag{C.2}$$

C.2 IAF - Runway segment

For the trajectory segment between the IAF and the runway, the used path is determined by using a path that corresponds with the average flown distance for the combination of IAF and runway. This is determined with the ADS-B analysis described in appendix E. This path is divided into equally spaced points for every half of a nautical mile path distance. To determine the time that a flight will be at every point along this path, the same procedure as for the ACC segment is used. The used indicated airspeeds for this segment result from the ADS-B data analysis described in appendix E. A different speed is used for two altitude ranges and the final approach.

D

Scheduling Algorithms

D.1 Arrival Scheduling

As defined in the model in section 6.4, a scheduling update action is performed by the planner agent. This scheduling action is activated by observation of an eligible flight that is not in the planning yet or by a planning update request from an air traffic control agent. If the planning mechanism is activated by the observation of a new flight that must be entered in the planning, the flight is firstly assigned a runway. After a flight has an assigned runway a sequence of processes is performed to schedule the flight. This same sequence of processes is initiated if a planning update is requested by an air traffic control agent. This mechanism is presented in the form of a flowchart in figure D.1.

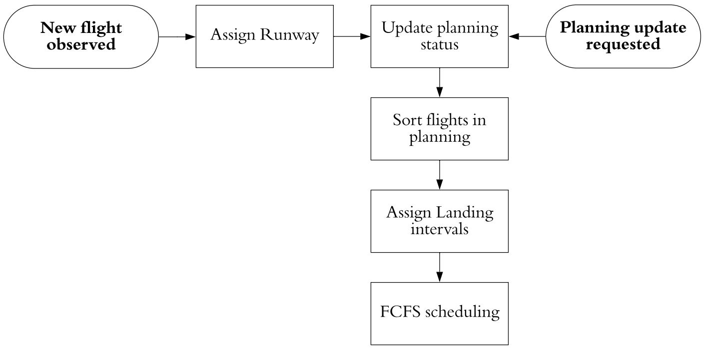


Figure D.1: Process flow for the scheduling update

D.1.1 Entering a Flight in the Pre-Planning Phase

A flight is entered in the pre-planning phase once a flight in the flight schedule is deemed eligible for planning. An arrival flight is deemed eligible for planning once it enters the filtering area. This filtering area is defined as the approximate area that is covered by the radar that surveils flights in the upper airspace. Once a flight enters this area the flight is assigned a runway. After it has been assigned a runway, the flight is entered in the planning and a scheduling update is performed.

D.1.2 Runway Assignment

In order to be able to enter a flight in the planning, the flight must be assigned a runway. The runway assignment procedure is based around the current active runway configuration and the paths that need to be travelled by the flights. In the simulation scenario, two possible runway configurations exist for arrival runways. For the first configuration, only one runway is active, this will always be runway 06. For this configuration arrivals from all directions will use this runway and a single queue is used for the scheduling mechanism, see figure D.3.

In the second possible runway configuration, two runways are active. These runways are 06 and 36R. In this case, scheduling will be done for two runway queues, see figure D.3. This means that flights need to be distributed over the two queues. This is done based on the arrival direction of flights. Because of the spatial properties of the runways and the IAF from which flights approach the runways, certain runway-arrival stack combinations are more desirable in terms of the path distance. For this reason, flights approaching from the SUGOL arrival stack are assigned the 06 runway and flights approaching from ARTIP are assigned runway 36R. For flights approaching from RIVER, the difference in flight distance for the two runways is much smaller. Flights from RIVER are therefore distributed over the two runway queues. This distribution is done according to the distribution observed in the simulation scenario.

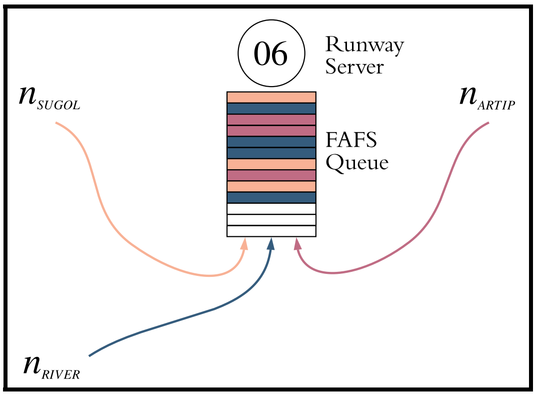


Figure D.2: The FAFS queue in a one-runway active scenario

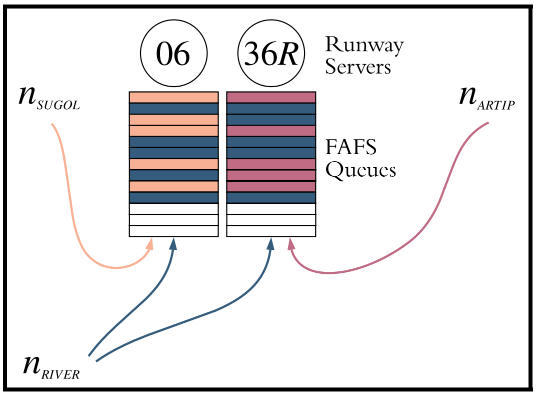


Figure D.3: The FAFS queue in a two-runway active scenario

D.1.3 Update the Planning Status

The planning status of a flight can either be planned or pre-planned. If a flight enters the planning it will initially always have the pre-planned status. At the first moment that a flight is expected to arrive at the IAF in 14 minutes or less the status of this flight is changed to planned and the flight is inserted into the planned flights set. If flights are scheduled at a slot earlier than the slot of the flight that gets the planned status, the status of these flights is also changed to planned and they are inserted into the set while maintaining the sequence they were in.

D.1.4 Sort Flights in the Planning

To sort the flights that exist in the current planning, first the set is divided into two subsets with planned and pre-planned flights. The subset of pre-planned flights is ordered by the expected nominal times at the runway for these flights. This sequence is subject to change by new flights entering the pre-planning phase. The other subset with planned flights is insertion-ordered based on the planning status update. To form the set of flights that serves as the input for assigning landing intervals, the two subsets are appended in such a way that every planned flight is at an earlier position than a pre-planned flight.

D.1.5 Assign Landing Intervals

The landing interval is assigned based on the wake categories of the current and subsequent flights. This combination is mapped to a time interval that corresponds with this combination. In the case where the planning does not contain an earlier flight, the landing interval is set to zero.

D.1.6 FCFS Scheduling

After the preparation steps in the scheduling update procedure, the flights are scheduled on a first come first serve basis. This first come first serve schedule mechanism is described in action mechanism 2. The input for this mechanism is a sorted set of flight tuples F. This set is sorted according to the action described in section D.1.4 and contains the assigned landing interval liv.

To schedule the flights a loop is performed over F. In every iteration, the slot variable is set by taking the maximum of the earliest time a flight can be at the runway $earliest_i$ and the sum of the previous slot and the landing interval between the current and previous flight. The slot variable is stored in an array such that the slot for the desired flight can be retrieved. Furthermore, the delay is calculated for every flight.

Action Mechanism 2: FCFS scheduling algorithm

```
Input:
    Set of sorted flight tuples $F = \{f_1, f_2, \dots, f_n\}$ where $f = \langle nominal_i, earliest_i, liv_i \rangle$
    Array $S[1], S[2], \dots, S[end(F)]$ that holds dummy values for the slots of every flight
    Array $D[1], D[2], \dots, D[end(F)]$ that holds dummy values for the delay of every flight
$slot \leftarrow 0$
for $i \leftarrow 1, end(F)$ do
    $slot \leftarrow max(slot + liv\_i,\ earliest\_i)$
    $S[i] \leftarrow slot$
    if $slot > nominal_i$ then
        $D[i] \leftarrow slot - nominal_i$
    else
        $D[i] \leftarrow 0$
    end if
end for
```

D.2 Departure Scheduling

The departure scheduling mechanism is triggered by the observation of a flight that needs to be entered in the planning or by the request for a planning update by an air traffic control agent. The mechanism performed to update the planning is depicted in figure D.4.

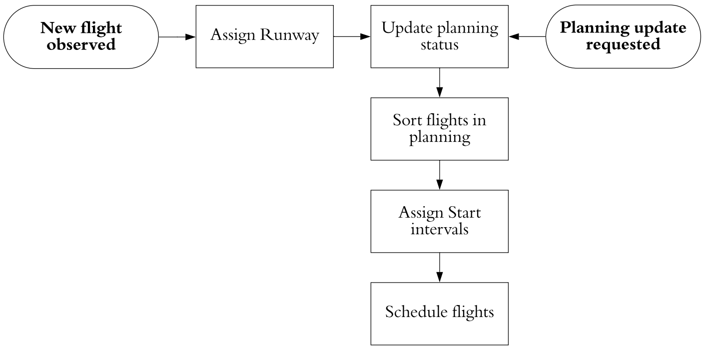


Figure D.4: Process flow for the departure scheduling update

D.2.1 Entering a Flight in the Planning

Departure flights are entered in the planning if the flights are deemed eligible for planning by the planner agent that is responsible for the departure planning. A flight is deemed eligible if the target off block time of a flight is within 40 minutes. Once this is the case, the planner agent will enter the flight in the planning. A flight that enters the planning is firstly assigned a runway, which activates an update for the planning mechanism.

D.2.2 Runway Assignment

For the departure system, the runways 36L and 36C are used in the simulation scenario. These can either be both active or only one of them. In case of a runway configuration where only one runway is active, all flights are scheduled on this runway. When two runways are active, the departure flights are distributed over the two runways. All flights that depart in the direction of sector one (S1), sector four (S4), and sector five (S5) are scheduled to depart from runway 36L. Flights using sector two (S2) and sector three (S3) will be scheduled on runway 36C. This distribution is depicted in figure D.5.

D.2.3 Sorting the Flights Before Planning

The first step in the scheduling update is to sort the flights. To do this set of flights in the planning is sorted according to the order observed in the ADS-B data. For the outbound planning, the scheduling mechanism is highly dependent on manual inputs. It is therefore decided, to not model this part and use the order observed in the ADS-B data.

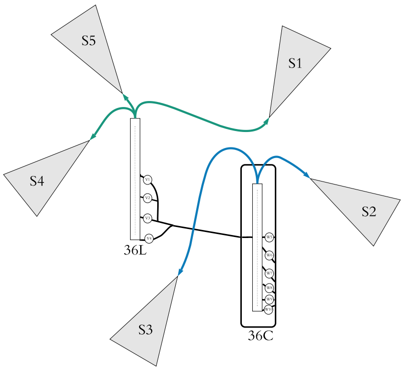


Figure D.5: Distribution of flights for two active departure runways

D.2.4 Assigning a Start Interval

Once the flights are sorted and the order is known, every flight gets an assigned start interval. This start interval is assigned based on the wake category of the flight itself and the wakecategory of the first flight that is sequenced before this flight. The combination of the two wake categories is mapped to a time value dictating the required start interval for the wake category combination.

D.2.5 Scheduling of the Flights

After the flights have been sorted and a start interval is assigned, the flights are assigned a departure slot. This assignment is done in a first come first serve fashion based on the earliest possible TTOT of a flight. This earliest possible TTOT is calculated as the sum of the TOBT and the taxi time. If a flight gets a slot assigned that is later than the earliest possible TTOT, the start up delay for this flight is calculated as the slot minus the eTTOT. This algorithm is presented in action mechanism 3.

Action Mechanism 3: Departure scheduling algorithm

```
Input:
    Set of sorted flight tuples $F = \{f_1, f_2, \ldots, f_n\}$ where $f = \langle earliest\_TTOT_i, stiv_i \rangle$
    Array $S[1], S[2], \ldots, S[end(F)]$ that holds dummy values for the slots of every flight
    Array $D[1], D[2], \ldots, D[end(F)]$ that holds dummy values for the delay of every flight
$slot \leftarrow 0$
for $i \leftarrow 1, end(F)$ do
    $slot \leftarrow max(slot + stiv\_i,\ earliest\_TTOT\_i)$
    $S[i] \leftarrow slot$
    if $slot > TOBT_i + taxi\_time$ then
        $D[i] \leftarrow slot - TOBT_i + taxi\_time$
    else
        $D[i] \leftarrow 0$
    end if
end for
```

E

Data Analysis

To derive parameters for the inbound and the outbound model, data analysis has been performed. This data analysis is based on the track information contained in ADS-B data. This data can be used to determine positions and speeds for historic flight trajectories.

E.1 Input Data

Input data used to derive parameters for the simulation model originates from two sources. For deriving, distances and speed parameters ADS-B data is analysed. This data is extracted from a well-maintained database, that is owned by To70 aviation consultants. From this database, all arrival tracks for the period between April and December 2018 are extracted. This data set consists of approximately 289 million track points for 194337 arrival flights. The geographic coverage of the track data is portrayed in figure E.1. This coverage is deemed sufficient to derive speed and distance parameters for the entirety of EHAA.

To derive parameters for the weather conditions, data from the Dutch meteorological institute KNMI is used. This data consists of processed mode-s surveillance data for the simulation day. The information contained in the surveillance messages is converted into wind and temperature conditions at grid points distributed across an area of 250 nautical miles around the airport. For more information about this data source, see [64]. The coverage of this data source is depicted in figure E.2. Especially for locations along the flight paths, this data is deemed to have sufficient coverage to render accurate weather conditions for the simulation day.

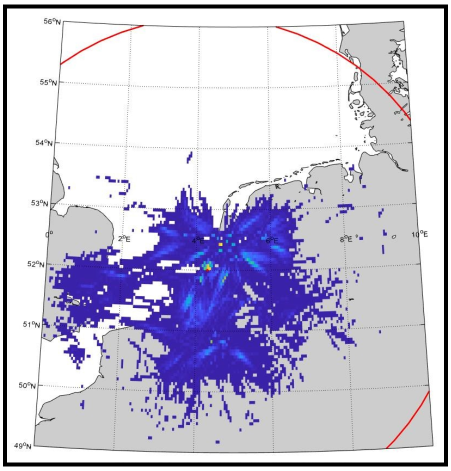


Figure E.1: Heatmap of the ADS-B track data

Figure E.2: Density plot with grid for the mode-s data

E.2 Trajectory and Distance Determination

In order to be able to predict trajectory times, it is required to determine nominal, minimum and maximum distances for flight paths between different various waypoints. Next to the distance, it is required to determine the geographical layout of the trajectories, such that weather conditions can be coupled to the points along these trajectories. For the inbound model, it is required to have information for the trajectories between the FIR boundary and the IAF as well as information for trajectories between the IAF and FAF.

In order to extract distance and trajectory parameters for the segment between the FIR boundary and the IAF, trajectories are split at the IAF. The remaining trajectories that lie between the IAF and the FIR boundary(see figureE.3) are divided based on the IAF that is crossed and the bearing of the point of crossing the FIR boundary towards the IAF. For every IAF - crossing angle combination it is determined what distances are flown. By taking the median of these distances it was observed that this median distance corresponds approximately to the distance of the direct path. The direct path between the FIR crossing and the IAF is therefore regarded as the nominal path for this segment (see figure E.4). This path is then automatically also the minimal distance for this segment, given there exists no shorter route than the direct route. For the maximum distance, it is assumed that the 95th percentile of distances for a specific combination can be attained in every scenario. This is therefore regarded as the maximum distance. Given that it is not possible to know what the exact instructed path will look like, it is assumed that the same geographic points are used for the path, with a distance between these points that is adjusted such that the sum is equal to the desired maximum distance.

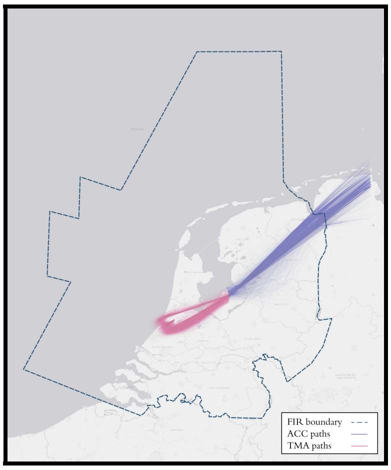


Figure E.3: Visualisation of splitting ADS-B trajectories for analysis

For the distance and trajectory parameters between the IAF and the runway, the same approach is followed. The trajectories after the IAF split are divided up into runway-IAF combinations. For every combination, it is determined what the median distance flown for this combination is. Using this median distance, the nominal path is chosen as a path that has distances that match the median distance and lies resembles the most flown path for this combination (see figure E.5). This trajectory is subsequently divided up into segments of a half nautical mile, resulting in a set of points that can be used to couple weather conditions and perform a trajectory prediction.

Figure E.4: Examples of ACC flight paths

Figure E.5: Examples of TMA flight paths

E.3 Speed Parameters

Like the distance parameters, the speed parameters are also determined for different segments. The speeds are determined for the FIR boundary-IAF segment, IAF-FAF segment and the final approach segment. For the ACC segment, the nominal speed used in the model is determined based on the initial speed. The minimum clean speed and the maximum speed are determined by data analysis. For this analysis, the trajectory data for all track points that lie in the ACC airspace is reduced to the indicated airspeeds for these track points. By aggregating this data per aircraft type and taking the 5th and 95th percentile, the maximum and minimum speed are determined for every aircraft type in ACC airspace.

For the TMA segment, the speed profiles like the one portrayed in figure E.6, show a different slope for different altitude ranges. It is therefore opted to use speed parameters for two altitude ranges. Based on the speed profiles for all aircraft type it is decided to use a speed value for the altitude range between 10000 and 6000 feet and a different value for the range between 6000 and 1330 feet. For all aircraft types and the two altitude ranges it is determined what mode is observed for the indicated airspeed. This value is regarded as the nominal speed for an aircraft type for a specific altitude range in the TMA area. For the maximum and minimum speed, the observed 5th and 95th percentile in the data are used.

For the final approach speed, it is assumed that only one speed can be used per aircraft type. To determine the final approach speed for every aircraft type, all indicated airspeed data for each aircraft type between 500 and 1330 feet altitude is used. By taking the mode of the indicated airspeed values, the final approach speed for every aircraft type is determined.

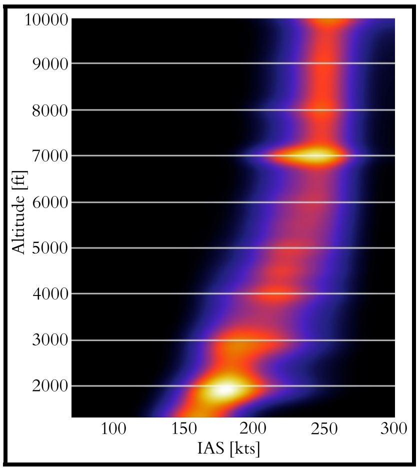


Figure E.6: Speed profile for Boeing 737-800

E.4 Taxi Times

For the taxi times, the historic time values between runways and aprons observed in the ADS-B data are used. These values are divided up into values for arrival and departure flights. The values are found based on the spatial intersection of trajectories the runway-apron combination. Of these values, the observed median is taken as the estimated taxi time that is used in the model.

E.5 Weather Parameters

Converting the mode-s weather data into a usable format for the model is done by aggregating the data using a three-dimensional grid. The grid is in 2D defined between 2 and 8 degrees longitude and between 51 and 55 degrees latitude This area is divided into bounding boxes of $\frac{3}{4}$ longitude and $\frac{2}{3}$ latitude. Vertically the 2D bounding boxes are divided up into altitude layers of 2500 feet ranging from 0 to 45000 feet.

For every of the defined three 3D bounding boxes, the data that is located inside these bounding boxes is aggregated for every half hour of the day. By taking the median value for the temperature and wind conditions, a sparse grid with these values is created. Given that most flight paths will fly through areas that have a high data densities, the weather data will be accurate for these positions. To obtain data for the bounding boxes that do not have sufficient data values, the values for the closest bounding box that has values are taken. The term closest is taken as the bounding box that firstly has the smallest deviation in time and subsequently the closest distance in space. By assigning an index to every of these 4D bounding boxes containing the position in time and space, these can be coupled to trajectory points that have a certain position and are reached in a certain half-hour of the day.

E.6 Data Tables

In this section, all tables resulting from the ADS-B analysis steps are presented. For some of the data elements, it is not feasible to present the entire set. For these elements, only a sample of the set is presented.

Table E.1: ACC trajectory data for flights to ARTIP

bearing	distance	angle	id_weather
ARTIP_00	277	180	[44 36 28 20]
ARTIP_01	284	192	[45 37 29 20]
ARTIP_02	151	204	[37 29 21 20]
ARTIP_03	136	215	[30 29 21 20]
ARTIP_04	133	226	[30 29 21 20]
ARTIP_05	131	236	[30 22 21 20]
ARTIP_06	119	246	[22 21 20]
ARTIP_07	104	255	[22 21 20]
ARTIP_08	101	265	[22 21 20]
ARTIP_09	102	275	[22 21 20]
ARTIP_10	119	285	[14 13 21 20]
ARTIP_11	120	295	[14 13 21 20]
ARTIP_12	121	305	[14 13 21 20]
ARTIP_13	120	315	[14 13 21 20]
ARTIP_14	119	324	[6 13 20]
ARTIP_15	30	334	[13 20]
ARTIP_16	139	343	[5 13 12 20]
ARTIP_17	142	351	[5 13 12 20]
ARTIP_18	140	360	[4 12 20]
ARTIP_19	140	8	[4 12 20]
ARTIP_20	121	17	[4 12 20]
ARTIP_21	124	26	[3 11 12 20]
ARTIP_22	154	35	[3 11 12 20]
ARTIP_23	180	45	[2 10 11 12 20]
ARTIP_24	198	55	[1 2 10 11 12 20]
ARTIP_25	229	65	[0 1 9 10 11 12 20]
ARTIP_26	227	75	[8 9 10 11 19 20]
ARTIP_27	198	85	[8 9 17 18 19 20]
ARTIP_28	189	95	[17 18 19 20]
ARTIP_29	213	105	[16 17 18 19 20]
ARTIP_30	225	115	[24 25 26 18 19 20]
ARTIP_31	207	124	[25 26 27 19 20]
ARTIP_32	172	134	[26 27 19 20]
ARTIP_33	184	145	[34 26 27 19 20]
ARTIP_34	208	156	[35 27 28 20]
ARTIP_35	253	168	[43 35 36 28 20]

Table E.2: ACC trajectory data for flights to RIVER

bearing	distance	angle	id_weather
RIVER_00	237	181	[34 26 18 10]
RIVER_01	351	193	[44 36 35 27 19 11 10]
RIVER_02	376	204	[45 37 36 28 27 19 11 10]
RIVER_03	275	215	[37 29 28 20 19 11 10]
RIVER_04	253	226	[30 29 21 20 19 11 10]
RIVER_05	249	236	[30 22 21 20 12 11 10]
RIVER_06	218	246	[22 21 12 11 10]
RIVER_07	207	256	[22 14 13 12 11 10]
RIVER_08	208	266	[14 13 12 11 10]
RIVER_09	185	276	[14 13 12 11 10]
RIVER_10	146	286	[5 4 12 11 10]
RIVER_11	155	296	[5 4 3 11 10]
RIVER_12	126	306	[4 3 11 10]
RIVER_13	80	316	[3 11 10]
RIVER_14	65	326	[3 11 10]
RIVER_15	54	335	[3 11 10]
RIVER_16	64	344	[3 2 10]
RIVER_17	61	352	[3 2 10]
RIVER_18	71	1	[2 10]
RIVER_19	69	10	[2 10]
RIVER_20	68	18	[2 10]
RIVER_21	68	27	[2 10]
RIVER_22	70	36	[2 10]
RIVER_23	74	46	[1 2 10]
RIVER_24	82	56	[1 2 10]
RIVER_25	95	66	[1 9 10]
RIVER_26	116	76	[0 1 9 10]
RIVER_27	131	86	[8 9 10]
RIVER_28	120	96	[8 9 10]
RIVER_29	113	106	[8 9 10]
RIVER_30	111	115	[8 9 10]
RIVER_31	111	125	[17 9 10]
RIVER_32	154	135	[16 17 9 10]
RIVER_33	190	146	[24 25 17 18 10]
RIVER_34	186	157	[25 17 18 10]
RIVER_35	192	169	[26 18 10]

Table E.3: ACC trajectory data for flights to SUGOL

bearing	distance	angle	id_weather
SUGOL_00	149	181	[34 26 18]
SUGOL_01	257	193	[43 35 27 26 18]
SUGOL_02	302	205	[45 44 36 35 27 19 18]
SUGOL_03	289	216	[45 37 36 28 27 19 18]
SUGOL_04	234	226	[37 29 28 27 19 18]
SUGOL_05	212	237	[30 29 28 20 19 18]
SUGOL_06	223	246	[30 29 21 20 19 18]
SUGOL_07	226	256	[22 21 20 19 18]
SUGOL_08	210	266	[22 21 20 19 18]
SUGOL_09	211	276	[14 22 21 20 19 18]
SUGOL_10	223	286	[14 13 12 11 19 18]
SUGOL_11	209	296	[14 13 12 11 19 18]
SUGOL_12	186	306	[5 13 12 11 19 18]
SUGOL_13	200	316	[5 4 12 11 18]
SUGOL_14	171	325	[4 3 11 10 18]
SUGOL_15	138	335	[3 11 10 18]
SUGOL_16	122	344	[3 11 10 18]
SUGOL_17	129	353	[2 10 18]
SUGOL_18	136	1	[2 10 18]
SUGOL_19	130	10	[2 10 18]
SUGOL_20	128	18	[1 2 10 18]
SUGOL_21	129	27	[1 9 10 18]
SUGOL_22	132	37	[1 9 10 18]
SUGOL_23	140	46	[0 8 9 10 18]
SUGOL_24	149	56	[8 9 10 18]
SUGOL_25	114	66	[8 9 17 18]
SUGOL_26	94	76	[8 9 17 18]
SUGOL_27	82	86	[17 18]
SUGOL_28	75	96	[17 18]
SUGOL_29	102	106	[16 17 18]
SUGOL_30	108	115	[16 17 18]
SUGOL_31	118	125	[24 16 17 18]
SUGOL_32	129	135	[24 25 17 18]
SUGOL_33	111	146	[25 17 18]
SUGOL_34	141	157	[25 26 18]
SUGOL_35	122	169	[26 18]

Table E.4: TMA trajectory data

trajectoryName	distance	angle	id_weather
ARTIP06	101	210.7816204	[20 19 11]
ARTIP36C	100	205.4626526	[20 19 11]
ARTIP36R	83	199.0300219	[20 12 11]
RIVER06	52	43.75318347	[10 11]
RIVER36C	63	47.22132469	[10 11]
RIVER36R	58	50.97255138	[10 11]
SUGOL06	66	121.6912721	[19 18 11]
SUGOL36C	99	113.6437726	[18 19 11]
SUGOL36R	91	111.9753945	[18 10 11]

Table E.5: Aircraft speeds per type part 1

Type	min. Clean	max. ACC	nom. Upper TMA	min. Upper TMA	max. Upper TMA	nom. Lower TMA	min Lower TMA	max Lower TMA	final
CL60	249	351	231	228	292	229	172	279	126
A124	242	280	241	218	250	192	147	231	130
A139	244	300	248	219	271	140	121	167	140
A20N	220	312	250	219	279	202	159	251	137
A21N	224	310	250	220	279	213	160	253	140
A306	250	323	250	220	269	202	153	250	139
A310	244	318	250	220	258	201	160	250	152
A318	239	339	250	219	254	215	162	251	134
A319	221	321	250	218	281	210	159	252	136
A320	222	321	250	218	278	209	159	251	141
A321	226	322	249	219	268	208	159	251	146
A332	230	313	250	219	270	203	157	251	142
A333	230	316	250	219	279	206	158	251	142
A343	248	320	250	219	279	212	158	251	144
A345	249	300	250	248	252	212	157	251	139
A359	248	320	250	219	281	200	151	250	140
A388	230	320	250	219	270	194	155	250	140
A545	248	315	249	201	276	217	160	261	129
AS50	225	299	249	223	253	213	164	250	154
ASTR	286	319	246	237	284	221	169	261	173
AT46	193	239	238	231	242	175	168	202	124
AT72	242	304	248	234	260	214	162	255	119
AT75	249	310	250	219	288	197	157	250	138
AT76	196	239	233	210	239	201	131	236	127
ATP	193	300	219	178	250	196	159	235	138
B190	162	219	223	201	227	211	150	227	129
B350	206	258	244	165	260	203	144	255	139
B36T	228	301	220	215	224	208	164	225	137
B38M	247	321	250	217	287	208	160	255	152
B427	266	309	250	200	286	200	132	253	132
B429	250	294	250	220	279	219	160	251	139
B462	304	316	294	245	300	247	197	254	169
B463	247	356	250	222	278	205	159	251	139
B733	220	304	249	218	274	203	159	250	141
B734	241	311	247	217	272	207	159	250	145
B735	220	308	249	216	270	201	158	250	135
B736	210	308	249	217	280	211	160	251	135
B737	220	321	249	219	268	203	157	251	138
B738	223	321	249	219	269	205	161	250	150
B739	220	321	249	218	259	202	161	250	154
B73M	248	252	233	226	246	170	159	232	155
B741	248	291	249	236	254	220	144	252	134
B742	244	305	247	217	263	212	150	251	147
B744	247	326	249	221	276	202	163	250	158
B748	246	313	249	220	271	200	166	250	162
B752	245	326	248	219	277	210	157	251	134
B753	246	313	248	220	282	204	161	251	148
B762	249	311	250	227	281	217	143	278	142
B763	228	320	249	219	275	210	160	250	146
B764	247	328	248	219	278	210	160	250	149
B772	234	316	249	219	273	207	160	250	145
B773	234	312	250	225	282	201	160	251	152
B77F	230	298	231	218	252	200	83	229	149
B77L	241	316	249	220	275	199	157	250	150
B77W	240	320	250	222	273	203	163	250	152
B788	230	321	250	219	281	212	162	251	148
B789	237	333	249	221	272	202	162	250	153
BCS1	249	304	249	210	253	200	144	248	136
BCS3	234	301	250	219	267	202	155	250	140
BE20	205	304	248	193	255	209	152	251	141
BE30	199	254	246	225	259	222	157	256	156
BE40	249	312	249	210	270	211	157	251	141
BE4W	197	307	247	200	258	201	159	249	127
BE58	158	167	164	162	186	169	157	187	185
BE9L	161	299	249	153	252	184	141	255	132
C17	250	316	248	209	253	220	162	251	135
C172	248	317	239	212	253	206	162	250	154
C195	269	301	250	219	260	181	157	231	141
C206	221	300	250	224	273	209	153	251	143
C25A	208	277	248	193	263	210	153	253	135
C25B	232	300	249	219	270	215	150	253	137
C25C	252	303	250	222	279	212	157	253	144
C25M	222	262	240	211	259	229	172	253	147
C414	220	311	250	220	289	210	161	251	136
C510	217	249	242	210	248	217	152	246	135
C525	237	270	250	221	263	212	143	257	134
C550	231	269	243	189	254	215	160	249	150
C551	222	258	243	215	254	211	153	248	146
C560	243	291	247	219	261	213	155	258	147
C56X	248	303	248	215	267	212	156	252	138
C650	235	319	245	210	267	212	167	251	151
C680	219	301	249	219	266	219	158	251	134
C68A	248	295	250	219	252	217	160	251	131
C750	260	348	254	246	260	225	165	254	165
CL30	219	304	249	215	280	211	162	253	140
CL35	244	305	248	217	267	212	157	251	133
CL60	239	312	249	214	280	207	157	251	137

Table E.6: Aircraft speeds per type part 2

Type	min. Clean	max. ACC	nom. Upper TMA	min. Upper TMA	max. Upper TMA	nom. Lower TMA	min Lower TMA	max Lower TMA	final
CRJ2	249	308	250	220	266	202	156	251	140
CRJ7	229	304	249	218	285	216	158	258	144
CRJ9	229	304	249	218	285	216	158	258	144
CRJX	226	311	250	219	278	209	159	253	146
D328	212	320	250	218	299	194	155	252	148
DA40	211	300	249	201	268	182	155	251	141
DA42	220	320	250	219	274	205	159	251	137
DH8D	197	268	229	210	242	194	153	230	130
E120	219	311	249	219	281	198	156	250	139
E135	219	311	249	219	281	198	156	250	139
E145	250	295	250	207	252	204	159	251	146
E170	220	302	250	219	268	208	160	251	141
E190	220	302	250	218	281	208	159	252	139
E195	220	301	250	219	277	206	160	251	147
E35L	250	316	246	218	285	214	150	264	139
E3TF	233	298	225	200	251	212	152	244	162
E50P	242	269	253	240	265	201	150	255	133
E545	221	311	241	216	253	211	156	250	158
E550	250	312	250	224	291	222	160	254	144
E559	251	311	251	239	256	217	164	233	138
E55P	243	313	247	219	259	210	148	251	129
E75L	220	303	250	219	267	205	160	251	140
E75S	220	303	250	219	267	205	160	251	140
F100	220	310	250	219	268	203	160	251	140
F2TH	249	360	250	219	282	212	159	254	135
F70	220	309	250	219	279	198	158	250	143
F86	250	320	249	200	252	216	164	250	158
F900	221	360	250	219	280	216	155	256	139
F9EX	251	332	251	249	268	214	138	256	129
FA10	281	315	251	215	286	193	118	220	116
FA50	266	346	249	224	273	203	133	259	126
FA7X	249	325	250	220	276	216	141	252	124
FA8X	202	330	250	203	277	209	158	251	125
G109	299	324	247	220	295	183	157	209	151
G150	304	319	270	264	293	218	187	255	177
G280	249	320	251	219	282	202	159	257	134
GALX	245	302	249	219	263	199	160	234	141
GFL6	250	321	249	219	276	204	159	236	159
GL5T	218	320	250	210	285	200	142	251	130
GLEX	249	322	250	217	273	203	151	251	131
GLF4	245	319	248	217	269	212	152	251	144
GLF5	240	321	249	210	262	209	150	251	133
GLF6	250	321	250	220	280	207	160	250	133
H25B	248	306	249	210	272	214	160	250	136
HDJT	206	390	237	175	267	198	146	263	131
IL96	232	249	220	202	244	188	157	238	146
L39	207	225	211	200	226	195	153	220	133
L410	263	275	250	243	255	213	173	246	152
LJ35	240	331	250	218	289	207	160	269	145
LJ45	217	321	249	220	281	219	161	256	140
LJ55	242	271	246	236	270	218	177	241	165
LJ60	244	290	248	217	254	207	158	250	146
LJ75	272	323	249	219	288	205	155	256	137
MD11	219	247	251	230	298	193	142	241	142
MD90	220	306	250	200	283	200	150	251	139
P180	236	259	253	242	259	238	158	257	151
P32R	231	269	235	210	252	205	172	230	122
P46T	176	207	183	177	224	183	138	238	147
P68	250	292	250	220	272	207	159	252	137
PA46	220	311	250	248	299	216	157	252	140
PAY1	224	300	249	212	259	200	160	250	140
PC12	185	235	219	176	236	198	132	234	119
RJ85	249	281	249	220	256	218	159	249	146
S22T	152	161	156	139	164	159	139	171	126
SF24	242	304	251	220	277	204	156	254	145
SR20	248	297	250	152	252	147	110	230	123
SR22	154	303	151	141	177	149	102	170	138
SU95	248	296	249	217	253	219	148	251	142
TAMP	234	312	242	215	270	188	147	249	131
TBM7	188	209	173	165	182	162	139	168	126
TBM8	198	259	214	189	256	188	131	250	124
TBM9	201	247	224	199	251	194	149	237	128

Table E.7: Inbound taxi times for runway 06

Runway	Apron Area	Taxi time [min]
06	Aapron1	2
06	Aapron2	2
06	Aapron3	2
06	Aapron4	3
06	Bapron1	3
06	Capron1	3
06	Capron2	3
06	Dapron1	5
06	Dapron2	5
06	Dapron3	4
06	Dapron4	5
06	Dapron5	4
06	Eapron1	7
06	Eapron2	6
06	Eapron3	6
06	Fapron1	7
06	Fapron2	9
06	Gapron1	8
06	Gapron2	9
06	Gapron3	9
06	Hapron	9
06	Japron2	10
06	Kapron	10
06	Rapron	7
06	Sapron	4

Table E.8: Inbound taxi times for runway 36R

Runway	Apron Area	Taxi time [min]
36R	Aapron1	4
36R	Aapron2	4
36R	Aapron3	4
36R	Aapron4	4
36R	Bapron1	4
36R	Capron1	4
36R	Capron2	4
36R	Dapron1	3
36R	Dapron2	3
36R	Dapron3	3
36R	Dapron4	4
36R	Dapron5	4
36R	Eapron1	5
36R	Eapron2	5
36R	Eapron3	4
36R	Fapron1	6
36R	Fapron2	5
36R	Gapron1	6
36R	Gapron2	7
36R	Gapron3	7
36R	Hapron	6
36R	Kapron	5
36R	Rapron	9
36R	Sapron	10

Table E.9: Outbound taxi times for runway 36L

Runway	Apron Area	Taxi time [min]
36L	Aapron1	14
36L	Aapron2	14
36L	Aapron3	15
36L	Aapron4	14
36L	Bapron1	16
36L	Capron1	17
36L	Capron2	17
36L	Dapron1	16
36L	Dapron2	16
36L	Dapron3	17
36L	Dapron4	17
36L	Dapron5	17
36L	Eapron1	21
36L	Eapron2	22
36L	Eapron3	24
36L	Fapron1	15
36L	Fapron2	15
36L	Gapron1	12
36L	Gapron2	12
36L	Gapron3	12
36L	Hapron	17
36L	Japron2	17
36L	Kapron	9
36L	Rapron	22
36L	Sapron	21

Table E.10: Outbound taxi times for runway 36C

Runway	Apron Area	Taxi time [min]
36C	Aapron1	8
36C	Aapron2	8
36C	Aapron3	10
36C	Aapron4	8
36C	Bapron1	9
36C	Capron1	12
36C	Capron2	12
36C	Dapron1	13
36C	Dapron2	13
36C	Dapron3	12
36C	Dapron4	12
36C	Dapron5	12
36C	Eapron1	15
36C	Eapron2	16
36C	Eapron3	14
36C	Fapron1	15
36C	Fapron2	15
36C	Gapron1	13
36C	Gapron2	13
36C	Gapron3	14
36C	Hapron	12
36C	Japron2	15
36C	Rapron	14
36C	Sapron	9

Table E.11: Weather data sample

Time	Grid ID	Alt (ft)	Wind_u (kts)	Wind_v (kts)	Temp (K)
2018-08-30 03:00:00	0	1250	10.1	1.1	284.5
2018-08-30 03:00:00	0	3750	-8.2	6.4	280.5
2018-08-30 03:00:00	0	6250	3.9	7.3	274.7
2018-08-30 03:00:00	0	8750	8.2	-1.6	273.9
2018-08-30 03:00:00	0	11250	3.5	-7.4	269.7
2018-08-30 03:00:00	0	13750	-1.9	10.3	265.8
2018-08-30 03:00:00	0	16250	8.8	4.1	260.2
2018-08-30 03:00:00	0	18750	-3.3	9.4	254.6
2018-08-30 03:00:00	0	21250	3.2	-12.6	248.0
2018-08-30 03:00:00	0	23750	11.9	7.8	241.2
2018-08-30 03:00:00	0	26250	5.7	13.3	235.6
2018-08-30 03:00:00	0	28750	-2.7	14.2	231.7
2018-08-30 03:00:00	0	31250	2.6	12.4	227.2
2018-08-30 03:00:00	0	33750	-10.1	-1.3	222.8
2018-08-30 03:00:00	0	36250	13.8	9.0	223.4
2018-08-30 03:00:00	0	38750	14.9	-2.7	224.8
2018-08-30 03:00:00	0	41250	14.9	-2.7	224.8
2018-08-30 03:00:00	1	1250	10.1	1.1	284.5
2018-08-30 03:00:00	1	3750	-8.2	6.4	280.5
2018-08-30 03:00:00	1	6250	3.9	7.3	274.7
2018-08-30 03:00:00	1	8750	8.2	-1.6	273.9
2018-08-30 03:00:00	1	11250	3.5	-7.4	269.7
2018-08-30 03:00:00	1	13750	-1.9	10.3	265.8
2018-08-30 03:00:00	1	16250	8.8	4.1	260.2
2018-08-30 03:00:00	1	18750	-3.3	9.4	254.6
2018-08-30 03:00:00	1	21250	3.2	-12.6	248.0
2018-08-30 03:00:00	1	23750	12.2	7.8	241.2
2018-08-30 03:00:00	1	26250	-13.9	3.2	235.6
2018-08-30 03:00:00	1	28750	-2.5	-14.0	232.1
2018-08-30 03:00:00	1	31250	2.6	12.4	227.2

F

Example of Simulation Trace

This appendix shows an example of a simulation trace visualisation. In figures F.1 and F.2 a simulation trace is presented for the APP supervisor agent and the TWR supervisor agent, respectively. Figure F.3, shows the external state observations available to the supervisor agents.

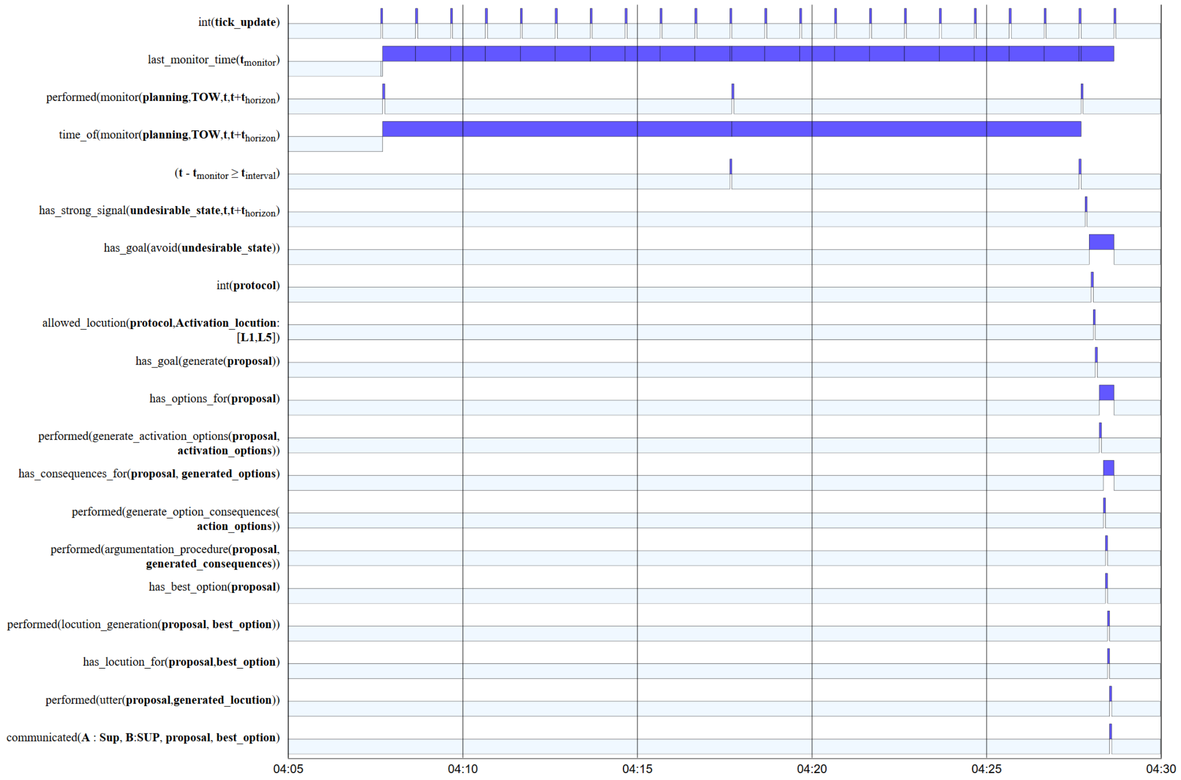


Figure F.1: Visualisation of simulation trace for the APP supervisor agent

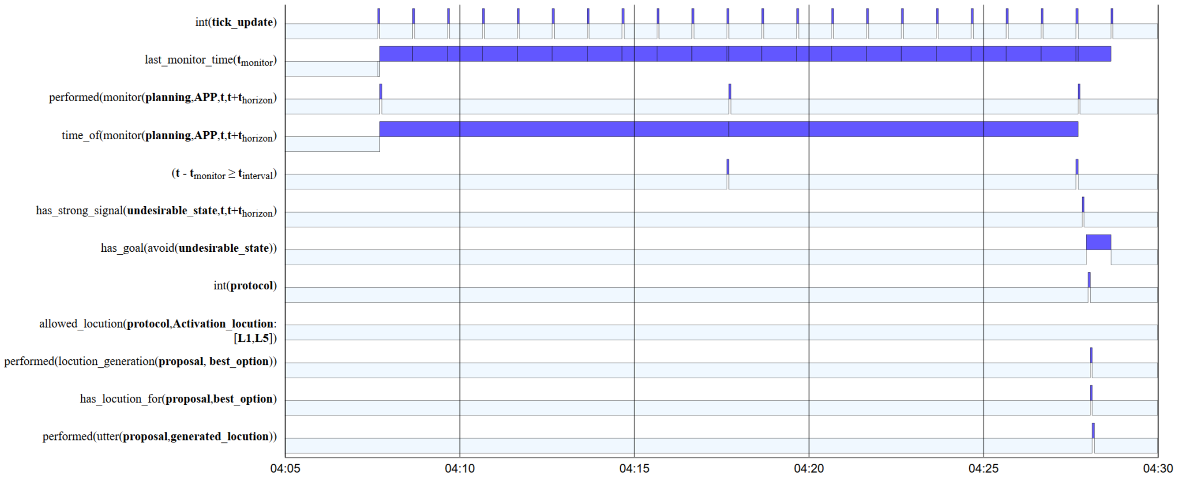


Figure F.2: Visualisation of simulation trace for the TWR supervisor agent

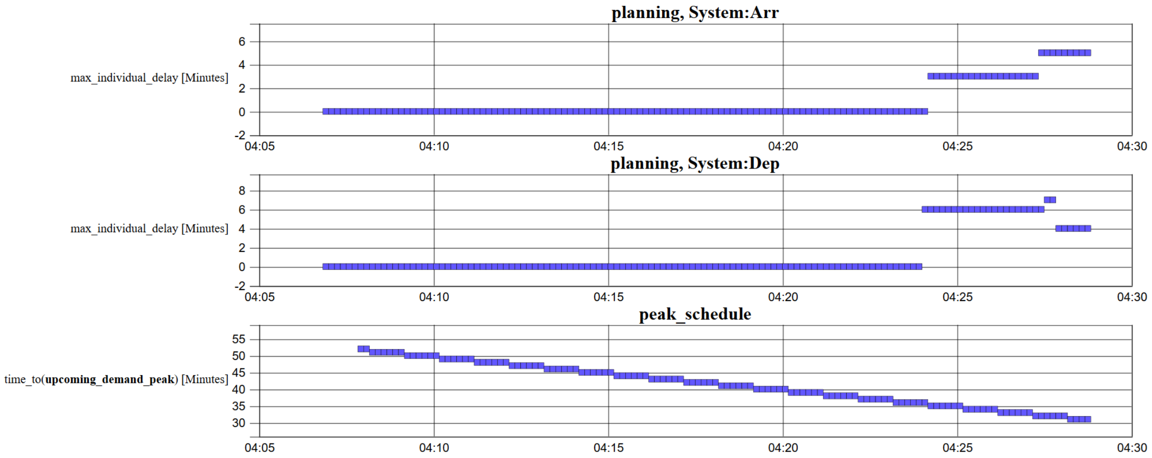
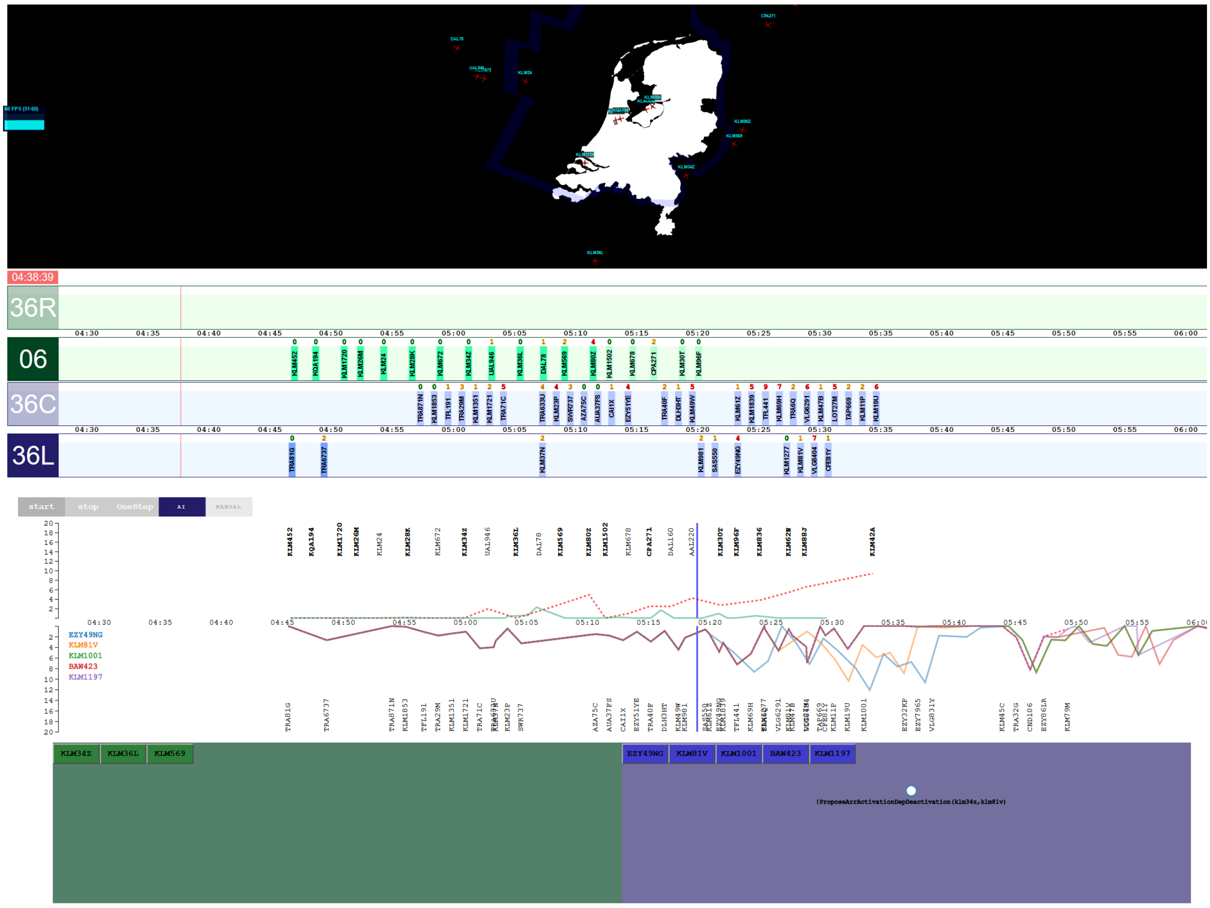


Figure F.3: Visualisation of state of observation values for the supervisor agents

G

Graphical User Interface

This appendix contains an image of the graphical user interface used to verify the model. This is depicted in figure G.1.



!ProposeArrActivationDepDeactivation(klm34z,klm81v)

Figure G.1: Graphical user interface

Bibliography

[1] Achenbach, A. and Spinler, S. Prescriptive analytics in airline operations: Arrival time prediction and cost index optimization for short-haul flights. *Operations Research Perspectives*, 5(1):265 – 279, 2018.

[2] Adecs and To70. Milieueffectrapportage "nieuw normen- en handhavingstelsel schiphol". Technical report, Schiphol Group, June 2016.

[3] Alders, H. Advies alders middelange termijn, October 2018. URL `http://www.alderstafellelystad.nl/uploads/1/4/1/3/14138220/advies-alders-middellange_termijn.pdf`. Accessed on: 2019-01-12.

[4] Alexandre, J., Odoni, A. R., and Webster, O. D. Dynamic control of runway configurations and of arrival and departure service rates. *Transportation science*, 51(1):155–176, 2017.

[5] Altinok, A., Kiran, R., and Bue, B. Modelling key predictors of airport runway configurations using learning algorithms. In *2018 Aviation Technology, Integration and Operations Conference*, 2018.

[6] Amgoud, L. and Cayrol, C. On the acceptability of arguments in preference-based argumentation. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[7] Antoniou, G. *Nonmonotonic Reasoning*. The MIT Press, 1996.

[8] Atkinson, K., Baroni, P., Giacomin, M., Hunter, A., Prakken, H., Reed, C., Simari, G. R., Thimm, M., and Villata, S. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, October 2017.

[9] Avery, J. and Balakrishnan, H. Predicting airport runway configuration. In *Thirteenth USA/Europe Air Traffic Management Research and Development Seminar*, 2016.

[10] Bain, A. *The Senses and the Intellect*. Longmans, Green Co., 1868.

[11] Bench-Capon, T. and Dunne, P. Value based argumentation frameworks. In *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning*, 2002.

[12] Blok, A., Sharpanskykh, A., and Vert, M. Formal and computational modeling of anticipation mechanisms of resilience in the complex sociotechnical air transport system. *Complex Adaptive Systems Modeling*, 6(1): 1–30, 2018.

[13] Blumberg, B. M. and Galyean, T. A. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of the 22nd annual conference on computer graphics and interactive techniques*, 1995.

[14] Bonabeau, E. Agent-based modeling: methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences of the United States of America*, 2002.

[15] Bondarenko, A., Toni, F., and Kowalski, R. An assumption-based framework for non-monotonic reasoning. In *Proceedings of the 2nd International Workshop on Logic Programming and Non-Monotonic Reasoning*, 1993.

[16] Bosse, T., Jonker, C. M., Meij van der, L., Sharpanskykh, A., and Treur, J. A temporal trace language for the formal analysis of dynamic properties. Technical report, Vrije Universiteit Amsterdam, 2006.

[17] Bosse, T. *Analysis of the Dynamics of Cognitive Processes*. PhD thesis, Vrije Universiteit Amsterdam, 2005.

[18] Bosse, T., Jonker, C., and Treur, J. A language and environment for analysis of dynamics by simulation. *International Journal on Artificial Intelligence Tools*, 16(3):435–464, 2007.

[19] Carrera, A. and Iglesias, C. A. A systematic review of argumentation techniques for multi-agent systems research. *Artificial Intelligence Review*, 44(4):509–535, 2015.

[20] Cevolini, M. The strongness of weak signals: self-reference and paradox in anticipatory systems. *European Journal of Futures Research*, 4(1):1–13, 2016.

[21] Chesñevar, C. I., Simari, G. R., Alsinet, T., and Godo, L. A logic programming framework for possibilistic argumentation with vague knowledge. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004.

[22] Clarke, J. P., Ren, L., Mcclain, E., Schleicher, D., Timar, S., Saraf, A., Crip, D., Gutterudand, R., Laroza, R., Thompson, T., Cross, C., and Lewis, T. Evaluating concepts for operations in metroplex terminal area airspace. *Journal of Aircraft*, 49(3):758–773, 2012.

[23] Dimopoulos, Y. and Kakas, A. C. Logic programming without negation as failure. In *Proceedings of the 1995 International Logic Programming Symposium*, 1995.

[24] Dimopoulos, Y. and Moraitis, P. *Negotiation and Argumentation in Multi-Agent Systems: Fundamentals, Theories, Systems and Applications*. Bentham Science Publishers, 2014.

[25] Dung, P. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[26] Dutch Ministry of Infrastructure and the Environment and the Dutch Ministry of Defence. Airspace vision preliminary policy document. Technical report, Rijksoverheid Nederland, 4 2011.

[27] Eurocontrol. User manual for the base of aircraft data. Technical report, Eurocontrol Experimental Center, July 2004.

[28] Fangueiro, P. A. and Maio, O. *An extensible argumentation model for ontology matching negotiation*. PhD thesis, University of Trás-os-Montes and Alto Douro, 2012.

[29] Faratin, P., Sierra, C., and Jennings, N. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[30] FIPA. Fipa propose interaction protocol specification, December 2012. URL `http://www.fipa.org/specs/fipa00036/SC00036H.html`. Accessed on: 2019-05-10.

[31] Garcia, A. J. and Simari, G. R. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(2):95–138, January 2004.

[32] Harrison, R. L. Introduction to monte carlo simulation. In *AIP Conference Proceedings*, 2010.

[33] Heblij, S. J. and Wijnen, R. A. A. Development of a runway allocation optimisation model for airport strategic planning. Technical report, NLR and Delft University of Technology, 2008.

[34] Hesslow, G. Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6 (6):242–247, 2002.

[35] Holland, J. H. *Emergence: From Chaos to Order*. Oxford University Press, 1998.

[36] Hoogendoorn, S. and Bovy, P. H. L. Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological*, 38(2):169–190, 2004.

[37] Hume, D. *A Treatise of Human Nature*. Oxford University Press, 1739.

[38] IATA. Iata forecast predicts 8.2 billion air travelers in 2037, October 2018. URL `https://www.iata.org/pressroom/pr/Pages/2018-10-24-02.aspx`. Accessed on: 2019-01-10.

[39] Jacquillat, A. and Odoni, A. R. A roadmap toward airport demand and capacity management. *Transportation Research Part A Policy and Practice*, 114(PA):168–185, 2018.

[40] Jacquillat, A., Odoni, A. R., and Webster, M. D. Dynamic control of runway configurations and of arrival and departure service rates. *Transportation science*, 51(1):155–176, 2017.

[41] Janic, M. A heuristic algorithm for the allocation of airport runway system capacity. *Transportation Planning and Technology*, 30(5):501–520, 2007.

[42] Jennings, N. R., Parsons, S., Noriega, P., and Sierra, C. On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems*, Boston, 1998.

[43] Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., and Wooldridge, M. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

[44] Kamgarpour, M., Zhang, W., and Tomlin, C. J. Modeling and optimization of terminal airspace and aircraft arrival subject to weather uncertainties. In *AIAA Guidance, Navigation, and Control Conference*, 2011.

[45] Kim, J. *Philosophy of Mind*. Westview Press, 1996.

[46] Leege de, A. and Janssen, C. Probabilistic runway and capacity forecasting using machine learning to support decision making. In *6th SESAR Innovation Days*, 2016.

[47] Li, L. and Clarke, J. P. A stochastic model of runway configuration planning. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[48] Lifschitz, V. Foundations of logic programming. In *Principles of Knowledge Representation,*, 1996.

[49] Luger, G. F. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley, 2002.

[50] Mcburney, P., Eijk van, R. M., Parsons, S., and Amgoud, L. A dialogue game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

[51] Minch, E. A review of: “anticipatory systems”, by Robert Rosen. *International Journal of General Systems*, 12(4):405–409, 1986.

[52] Modgil, S. and Prakken, H. The aspic + framework for structured argumentation: A tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[53] NH Nieuws. Recordaantal klachten over overlast schiphol, July 2018. URL `https://www.nhnieuws.nl/nieuws/228739/Recordaantal-klachten-over-overlast-Schiphol`. Accessed on: 2019-08-07.

[54] OVV. Grenzen in zicht veilige afhandeling vliegverkeer schiphol, April 2017. URL `https://www.onderzoeksraad.nl/nl/page/6981/grenzen-in-zicht-veilige-afhandeling-vliegverkeer-schiphol`. Accessed on: 2019-07-22.

[55] OVV. Veiligheid vliegverkeer schiphol. Technical report, Onderzoeksraad Voor Veiligheid, April 2017.

[56] Peterson, E. Reified literals: A best practice candidate design pattern for increased expressivity in the intelligence community. In *STIDS 2010*, 2010.

[57] Pruitt, D. G. *Negotiation Behavior*. Academic Press, 1981.

[58] Rahwan, I., Ramchurn, S. D., Jennings, N. R., Mcburney, P., Parsons, S., and Sonenberg, L. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2004.

[59] Ross, S. M. *Simulation (5th Edition)*. Elsevier, 2013.

[60] Roy, S., Dhal, R., Taylor, C., Xue, M., and Wan, Y. Generating representative weather-impact futures for strategic traffic flow management: Airport capacity prediction. In *Proceedings of the 93rd American Meteorological Society Annual Meeting*, 2013.

[61] Sajet, D. Recorddrukte op schiphol deze zomer: 220.000 passagiers per dag, July 2018. URL `https://www.nhnieuws.nl/nieuws/227472/recorddrukte-op-schiphol-deze-zomer-220000-passagiers-per-dag`. Accessed on: 2019-08-07.

[62] Sama, M., D’Ariano, A., Corman, F., and Pacciarelli, D. Coordination of scheduling decisions in the management of ariport airspace and taxiway operations. *Transportation Research Procedia*, 23(PA):246–262, 2017.

[63] Simari, G. and Loui, R. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53(2-3):125–157, 1992.

[64] Sondij, J. and Haan de, S. Aircraft as a meteorological sensor, September 2013. URL `http://mode-s.knmi.nl/documents/KNMI_Mode-S_EHS_September_2013.pdf`. Accessed on: 2019-09-10.

[65] Stolzenburg, F., Garcia, A., Chesnevar, C. I., and Simari, G. R. Computing generalized specificity. *Journal of Non-Classical Logics*, 13(1):87–113, 2003.

[66] The AgentBuilder Team. Agentbuilder, 2018. URL `http://www.agentbuilder.com/ Documentation/whyAgents.html`. Accessed on: 2019-11-01.

[67] Thimm, M. The tweety library collection for logical aspects of artificial intelligence and knowledge representation. *Künstliche Intelligenz*, 31(1):93–97, 2017.

[68] Thorne, J. and Kincaid, R. Heuristic search for tactical runway configuration management. In *12th AIAA Aviation Technology, Integration and Operations*, 2012.

[69] Tielrooij, M., Vanwelsenaere, A., and Dijkstra, F. Extending aman radar horizon. Technical report, Knowledge & Development Centre Mainport Schiphol, February 2018.

[70] To70 and KDC. Afos dashboard interface, dec 2018. URL `https://afos.to70.nl`. Accessed on: 2019-04-20.

[71] Wang, Y. Prediction of weather impacted airport capacity using ensemble learning. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011.

[72] Wood, A. Airspeed conversions (cas/eas/tas/mach), 2017. URL `https://aerotoolbox.net/airspeed-conversions`. Accessed on: 2019-03-10.

[73] Zlotkin, G. and Rosenschein, J. S. Mechanism design for automated negotiation, and its application to task oriented domains. *Artificial Intelligence*, 86(2):195–244, 1996.