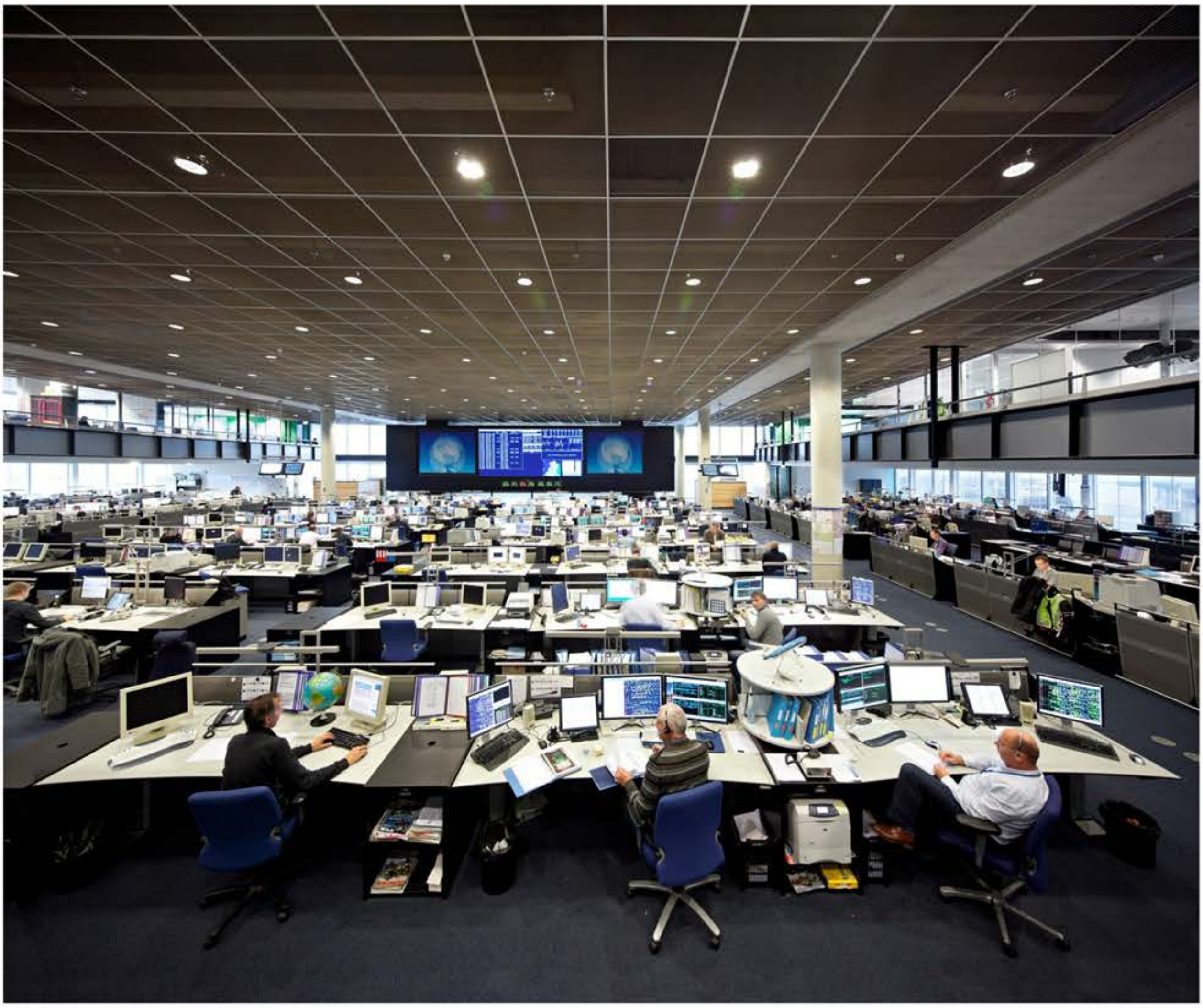
MSc Graduation Thesis

Modelling and Simulation of Decision-making Processes in Airline Operations Control



K. Belhadji

Faculty of Aerospace Engineering

June 21, 2016

Modelling and Simulation of Decision-making Processes in Airline Operations Control

By

K. Belhadji

in partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Thursday July 7, 2016 at 15:00 PM.

Supervisor:	Prof. dr. ir. H.A.P. (Henk) Blom
Thesis committee:	Prof. dr. ir. H.A.P. (Henk) Blom Dr. M.A.(Mihaela) Mitici Dr. M. (Milan) Janic Ir. Hans Mulder (T.J. Mulder)

Acknowledgement

This thesis is written as part of the MSc programme of Aerospace Engineering of the Delft University of Technology. It has been a long road, but I'm more than happy that I can make a contribution to the research goals of the Air Transport and Operations section.

First I would like to express my gratitude to my supervisor prof. dr. ir. Henk Blom for supervising me during this thesis and for providing valuable feedback towards my work. I would like to thank Dr. Alexei Sharpanskykh for introducing me to the modelling software and for sending me helpful literature.
Even though it was not possible to finish this thesis with my initial supervisor Soufiane Bouarfa, I still would like to thank him for the effort and the time he has put into this project. Special thanks to Arjen Blom from KLM and Thomas Omondi from Kenya Airways for providing me valuable information that helped me to write this thesis.

Special thanks to my family and friends for their patience, encouragement and support!

Delft, 21th June 2016 Kamal Belhadji

Summary

Studies have estimated that irregular airline operations can cost between 2% and 3% of the airline its annual revenue and a loss of passenger goodwill (Castro et al., 2012). This amounts to €521M- €780M annually for an airline like Air France KLM (Air France-KLM, 2015). To handle these irregular operations airlines have an Airline Operations Control department that is responsible for ".. the planning and coordination of the disruption management process to achieve network punctuality and customer service while utilizing assets effectively and minimising cost" (Bruce, 2011). During disruption management controllers monitor the progress of operations, identify problems, make decisions and implement solutions (Kohl et al., 2007). Due to the complexity of the airline operating environment, controllers are confronted with many operational uncertainties. Coupled with an inadequate information supply and time constraints, this could create hazardous situations what might result in extreme economic consequences for the airline (Feigh, 2008). Furthermore, airlines have become more concerned with optimizing operational schedule by being reserved in adding robustness into their schedule i.e. slacks, buffers and standby resources (Clarke, 1998, Kohl et al., 2007). This results in the operational schedule being more prone to disruptions and limits the possibilities for recovery, which adds more pressure on Airline Operations Control.

Controllers at Airline Operations Control rarely have time to explain their reason for decision-making (Bruce, 2011). Additionally, multiple decision-makers are involved during disruption management resulting in more difficulties to evaluate decision-making processes. By modelling and simulation of these decision-making processes it should enable more insights into the effect of robust scheduling and operational uncertainties on disruption management.

To effectively evaluate the decision-making process, an ontological correspondence of the model with the real-world is required. The systematic methodology of Nikolic and Ghorbani (2011) promotes ontological correspondence and consists of five steps: (1) system analysis, (2) model design, (3) detailed design, (4) software implementation and (5) model evaluation.

The first step of the modelling process is the analysis and identification of the socio-technical system. Due to endless possibilities and complexities in this system, a frame will be chosen in the form of two scenarios. These two scenarios are expert validated scenario and have been used for the decision-making study of Bruce (2011). In that study, controllers commented on real-life scenarios by expressing their thoughts regarding the uncertainties they face, scheduling parameters they are interested in and the decision consideration they make. Through the use of a think-aloud protocol, this resulted in a wealth of qualitative data which can be used for the identification of the socio-technical system in this study. Next to the data of Bruce (2011), data has been gathered through meetings with two industry experts and amended with literature. The socio-technical system identification resulted in an inventory of actors, their relationships, specifications of their actions, the environment they are operating in, the uncertainties they face and the objectives they are trying striving to achieve.

The second step of the modelling process is the model design. The two scenarios used in the system identification will be combined into one scenario that initiates the decision-making process for the model. This scenario relates to an aircraft mechanical failure that takes place at outstation. The purpose of the model is that the Operations Controller, Aircraft Controller, Crew Controller, Stations Operations Controller and Flight Dispatch will interact with each other and with the environment during the decision-making process to eventually choose a strategy to recover from this disruption.

By using the socio-technical system analysis, a model outline is designed in which uncertainties, objectives and the decision outcome per resource are presented for this scenario. The outcome of the decision-making process is a combination of decision outcomes per resources i.e. what will happen regarding aircraft, crew and passengers. However, during model design numerous combinations of decision-outcomes have been found to be invalid, therefore, truth-tables are used to eliminate all invalid combinations. This resulted in twenty valid recovery strategies that declare implications on aircraft, crew and passengers. After the model outline, the environment the controllers are operating in is conceptualized using conditions and parameters. Time-based uncertainties are parameterized, while the non-time based uncertainties are conceptualized into Boolean valued conditions. Furthermore, the tasks of the controllers are abstracted from the action identification of the socio-technical system analysis. A model structure is designed to provide a general view of the different phases of the decision-making process. The next step is to identify for each phase the tasks that are done and the

conditions and parameters used. These controller tasks are translated into processes that represent outgoing or incoming interactions of the controllers and the environment. The resulting decision-making processes start at the initiation by the scenario and ends with a choice in recovery strategy.

The model will be formalised in the detailed design phase. To make sure that there is some ontological correspondence of the decision-making process with the real-world actors, LEADSTO will be used. This software can express both qualitative and quantitative aspects and is used to simulate organization dynamics (Sharpanskykh and Treur, 2010). LEADSTO is an executable sub-language of Trace Temporal Language (TTL). TTL uses ordered sort predicate logic that can specify dynamics over time. Description of the behaviour of the system component is done using ontologies that are specified by sorts, constants, variables, functions and predicates.

The software implementation is done by coding in a plain text editor and loaded into the LEADSTO simulation tool. The simulation result is a specification of all the states and state properties referred to as a trace. Verification of these traces is done using the cross-functional flow charts that are made during model design.

The last step is model evaluation. The amount of combinations of the conditions that can be evaluated is considerable, for this reason, a case by case approach is considered. Four cases are defined, three cases in which the conditions are favourable to repair the aircraft in the mechanical failure scenario and one case in which the conditions are unfavourable. For each case, parameters are chosen and conditions that represent recovery opportunities like rebooking of passengers or availability of reserve resources.The simulations results from the LEADSTO-tool have been imported into a spreadsheet for the purpose of data-filtering and cost calculations. With a sensitivity analysis it is shown what the effect is of certain uncertainties if these are not overcome and what its effect are on decision-making process in terms of tasks that are done by the controllers.

By modelling and simulation, conclusions can be drawn regarding robust scheduling, operational uncertainties and its effect on decision-making process and decision outcome.

This model shows that slacks and buffers in a robust schedule promote a degree of self-recovery, which results in controllers being less dependent on availability of recovery opportunities. This translates into fewer tasks to be performed during the decision-making process and being less prone to uncertainties. However, in certain cases, transit buffer time seemed to be insensitive leading to unnecessary tasks to be performed by the controllers. Having planned standby resources in a robust schedule showed to have great satisfactory results in terms of delivering high customer service level, but could result in higher direct costs. Robust scheduling during the tactical phase serves as a prelude to disruption management during the tactical phase and should be taken into account when evaluating decision-making processes. Humans play a major role in disruption management when it is characterized with a high degree in uncertainty. Analysis of the results has shown that a lack of information to overcome an uncertainty could result in extra (unnecessary) tasks to be performed by other controllers or it could result in (unnecessary) deployment of costly recovery strategies. This study also shows that the relative importance of the (sometimes) competing objectives play a major role in decision outcome and influences the tasks performed during the decision-making process.

Contents

List of abbreviations & symbols

Abbreviations

AOC	Airline Operations Control
OC	Operations Control
SG	Supporting Group
AC	Aircraft Controller
CC	Crew Controller
FD	Flight Dispatch
SC	Station operations Controller
ENV	Environment
AOG	Aircraft On Ground
MX	Maintenance
PAX	Passengers
TPAX	Transit passengers
CXN	Connection
RS	Recovery Strategy
DO	Decision Outcome
NDM	Naturalistic Decision-making Model
GO	General Objective
SO	Scenario Objective
ABM	Agent-based Modelling
RP	Role Property
TP	Transfer Property
EIP	Enviroment Interaction property
EP	Environment Property

Symbols

a	Technical diagnosis adequateness	[Boolean]
b	Spare part availability	[Boolean]
c	Weather pattern favourability	[Boolean]
d	Hangar space availability	[Boolean]
e	Organizing connection possibility	[Boolean]
f	Positioning possibility	[Boolean]
g	Reserve crew availability	[Boolean]
h	Rebooking possibility	[Boolean]
i	Reserve aircraft availability	[Boolean]
p	condition sequences for aircraft under unscheduled maintenance	[-]
q	condition sequences for aircraft on ground	
r_t	Repair time	[min.]
c_t	Crew duty slack time	[min.]
d_t	Positioning time	[min.]
k_t	Ferry time	[min.]
b_t	Rebooking time	[min.]
p_t	Transit-buffer time	[min.]
d_f	Passenger delay	[min.]
C	Operating Costs	[€]
D	Direct Costs	[€]
Q	Quality Costs	[€]
F	Flight Costs	[€]

R	Crew Costs	[€]
P	Passenger Cost	[€]
β	Weight coefficient of quality costs	[-]
γ	Coefficient of importance of connection	[-]
l_p	total passengers in pleasure profile	[-]
t_p	transit passengers in pleasure profile(economy)	[-]
l_b	total passengers in business profile	[-]
t_b	transit passengers in business profile(economy)	[-]

List of Figures

List of Tables

1. Introduction

This chapter discusses the initiation, problem definition and motivation of this thesis (section 1.1 and 1.2). By using this problem definition, a research objective and framework is formulated and described in section 1.3. This section also includes research questions to steer this study towards the research objective. Section 1.4 describes the research strategy and section 1.5 provides an overview of the structure of this report.

1.1 Background

Airlines are subject to external events that disrupt their day-to day operations (Shavell, 2001). These disruptions lead to additional costs for crew, fuel, aircraft, maintenance and passenger good will (Belobaba et al., 2009, Peterson et al., 2013).

The main cause of these high costs is due to propagation of disruption into the entire airline planning. Coupled with current threat to profitability, it is imperative that airlines desire to manage these disruptions effectively to control the costs of irregular operations (Belobaba et al., 2009). Furthermore, due to high costs, airlines are reserved in planning slacks and standby resources into their airline operational schedule adding pressure to disruption management.

Because of the severity of the implications of irregular operations, airlines take proactive steps to mitigate the losses suffered (Shavell, 2001). Each airline has an Airline Operations Control (AOC) department which is responsible for effective disruption management (Clarke, 1998). Studies have shown that improved recovery process for these disruptions can result in significant cost reductions (Castro et al., 2012).

One of the research areas of the Air Transport Operations profile of the Aerospace Engineering faculty at the Delft University of Technology is the understanding and development of more effective aerospace and airline operations support (van der Zaken, 2015). This thesis is part of the master’s programme curriculum and its aim is to provide new insights into AOC decision-making processes of airline operations.

1.2 Motivation and Problem definition

The chief function of an AOC is to mirror actual airline operations with the current operational schedule (Bruce, 2011). In the case of a disruptive event, various entities within AOC make decisions on a collaborative or individual basis to recover from this disruption. This decision- making process usually involves multiple decision makers as well as stakeholders and can lead to unexpected outcomes. Additionally, the problems the AOC face can be highly complex and decisions are made under severe time constraints and economic pressure (Igbo, 2013). Concluding, decision-making in the airline operating environment is critical, so any improvements to the decision-making process are likely to result in more effective ways in which AOCs can manage disruptions (Castro et al., 2012).

By interaction, AOC decision-makers collect information to gain situation awareness to provide adequate decision alternatives (Bruce, 2011). However, decision-makers are confronted with many operational uncertainties in the airline operating environment. These uncertainties coupled with an incomplete, ambiguous or unreliable distributed information supply, could create hazardous situations that might result in extreme economic consequences for the airline (Feigh, 2008). Furthermore, due to time constraints, decision-makers have limited time to collect information what might increase the chance of poor decision outcome (Klein, 1999).

Disruption managment can be viewed from a strategic and tactical perspective. The latter relates to proceedings of the AOC, while the former is about the proceedings in the pre-AOC phase i.e. the design of the operational schedule. The strategic phase is an important part of airline operational handling. In this phase standby resources and slacks are planned to promote flexibility during disruption management (Kohl et al., 2007). This is also reffered to as adding ‘robustness’ to a schedule and a lack of this robustness could impact the decision-making process by limiting the choice in recovery strategy (Ball et al., 2006). However, it is not clear what implications robustness has on certain AOC objectives.

Numerous research studies have been conducted concerning airline disruption management. The majority of these studies focus on the development of algorithms to reconstruct aircraft or crew schedules in case of disruptions by using operations research methodologies (Clausen et al., 2009). These algorithms are used to design decision-support tools that help decision-makers opting for a recovery strategy.
Even though the relevance of these studies regarding decision support tools is recognized by the industry, they do not provide a total solution for disruption management. These studies focus predominantly on one of the main resources and do not take into account an integrated approach taken all different aspects of the airline operating environment into account. The human has to be involved in disruption management since they are very quickly in assimilating information (Mathaisel, 1996). Humans communicate with systems and with each other to determine whether a particular event has to be acted upon or not. Humans also make decisions based on information and judgements that do not exist in computer systems (Kohl et al., 2007). However, very little research has been done about the role of the human in the AOC decision-making process. Bouarfa et al. (2016) explored the effect of different coordination policies between controller and concluded that these policies could impact AOC objectives significantly. In the research study of Bruce (2011) it was explored what the impact of situation awareness, experience and expertise is on decision considerations.

In the study of Castro et al. (2014) a multi-agent system is designed with the purpose to reduce recovery costs and decision-making time. This is done by automating certain decision-making processes using airline statistical data of occurrence of disruptions and probability of the actions of the controllers. This intelligent approach could make AOC decision-making more efficient but Castro et al. (2014) also states that “repetitive tasks are better performed by software agents and tasks with a high degree of uncertainty are better performed by humans”.

Since humans and technical systems interact with each other at an AOC, it can be regarded as a socio-technical system. To understand any system thoroughly, it has to be modelled and analysed. A model is defined by Siegfried (2014) as “an idealized simplifying and with respect to certain aspects similar representation of a system. The purpose of a model is to allow a better study of specific properties than using the original system“. When a reliable model of the organization is made, it is possible to play with it, change some parameters and measure how the performance of the model varies in response to these changes (Bonabeau, 2002). This gives the modeller the opportunity to obtain global behaviour of a system by perceiving how individual participants behave (Borshchev and Filippov, 2004).

In an interview conducted for this study, a flight operations director of a large Dutch airline identified that there is a need for evaluation of AOC processes to understand root causes of sub-optimal disruption management (Appendix A). An example was provided of the difference in used (standby) reserve resources among shifts and the difficulty of evaluating these anomalies due to involvement of multiple decision-makers. Another aspect that makes it difficult to evaluate decision-making processes is the freedom that is provided to controllers in choosing certain recovery strategies. Peters (2006) & Feigh (2008) describe in their study that controllers can be confronted with ill-defined, shifting and sometimes competing objectives. Bruce (2011) also highlighted in his study that “controller rarely have time during airline disruption to explain their reasons for decision-making. This means that there is considerable difficulty of examining decision-making process for disruptions that occur in real-life”.

The need to understand AOC decision-making processes is driven by the growing complexity of the AOC operating environment in which decisions often need to be made rapidly with sometimes incomplete or inaccurate information. Since there are difficulties in evaluating these decision-making processes in real-life, a model should provide more insights into behaviour and sensitivities of the AOC operating environment.

1.3 Conceptual research design

1.3.1 Research objective

From the problem definition and motivation it was clear that it is interesting to study the human controller’s behaviour, uncertainties that are present in the environment and the influence of robust scheduling on decision-making. In order to capture all these elements, a scenario will be used to narrow down the possibilities which could be endless in the AOC operating environment. Concluding, the research objective is formulated as:

To evaluate decision-making processes of Airline Operations Control by simulating and analysing behaviour of controllers, sensitivities, operational uncertainties, robust scheduling and decision outcomes in a scenario using a model

1.3.2 Research framework

Several research processes have to be followed to achieve the research objective (see Figure 1).

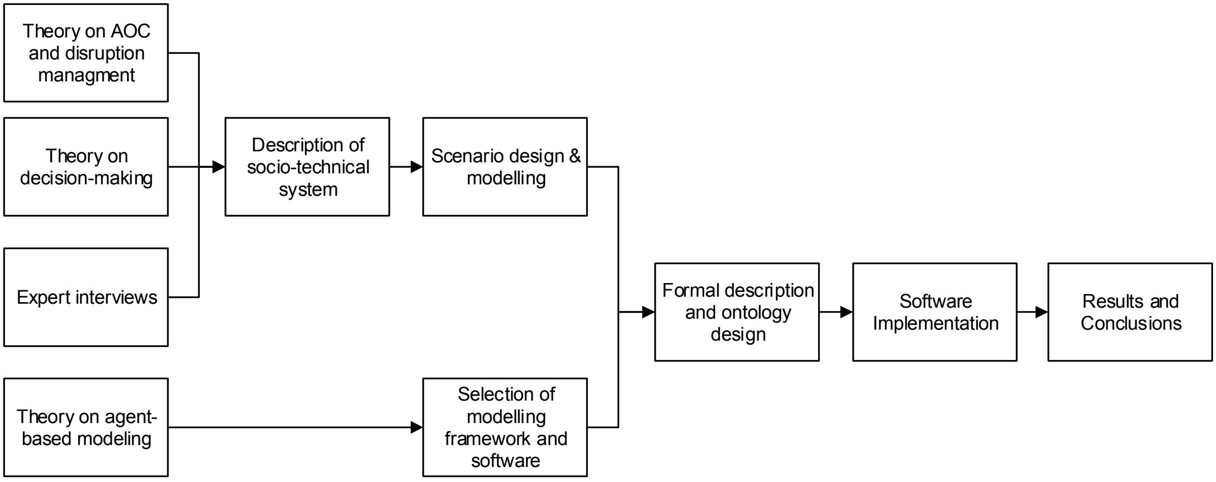


Figure 1 - concise overview of steps in this research framework

The aim of the literature review is to be familiar with all aspects of modelling and simulation towards evaluation of Airline Operations Control decision-making processes. The literature review will cover three topics: (1) theory on AOC and disruption management, (2) theory on decision making and (3) theory on agent-base modelling.

The literature review provides insights into the state of art in airline disruption management, the position of AOC in the airline operations system, the objective, lay-out structure and proceedings annex behaviour of the various decision-makers. Additionally, the type of disruptions AOC faces and the types of recovery strategies that can be deployed are identified.
Decision-making is studied in the field of cognitive psychology which is out of the scope of this study. A brief literature review is done in which explores the factors that influence decision-making performance.

Agent-based modelling is a modelling technique used to model socio-technical systems. There are various methodologies, frameworks and software available for this relatively new modelling technique. By modelling autonomous decision-makers (i.e. agents) that interact with each other or with the environment, unexpected behaviour could emerge. However, for this study, interactions among the controllers will be structured in order to effectively evaluate decision-making processes in AOC. The decision-making process follows a predefined procedure resulting in the model not being agent-based. Nonetheless, the agent-based modelling technique provides some interesting frameworks that will be used to conceptualize the socio-technical system. Therefore, a literature review is done to explore methodologies, frameworks; benefits and limitations of agent-based modelling.

Besides the literature review, two expert interviews are held with A. Blom “director Emergency flight operations” of a Dutch airline. The first interview was held to identify the need of the industry regarding to decision-making processes in AOC and to collect additional information. The second interview was held to discuss a potential collaboration for this study. Another informal meeting was conducted with T. Omondi “Chief Operating Officer” of a large African Airline to request additional information to get a better picture of AOC decision-making processes.

By using predominantly literature, the socio-technical system with all its relevant concepts, objects and interactions is analysed and described. Specifically, it will be determined what operational uncertainties the

AOC face and what kind of objectives they are striving for to achieve by using what kind of recovery strategies. In this phase also the scenarios are being presented which will be used for the modelling phase.

The next phase is the scenario design in which all the uncertainties are being conceptualized in conditions and parameters. All the tasks that the controllers will perform will be structured which will define the decision-making process. These processes will be illustrated with flowcharts so that they can be used for software implementation. In this phase also the cost equations will be defined that will eventually be used for result analysis.

The last step before software implementation is the formal description of the model. This is done by defining properties and structure of the model and by designing the input and output ontologies. For the formalization the flowcharts of the decision-making processes of the previous step will be used. The next step is the software implementation. Verification of the model is done using the flowcharts of the model design phase.

Due to the large number of possible combinations of conditions and parameters that could be evaluated, a case by case approach will be introduced. In each case a certain set of parameters and condition sequences will simulated, followed by cost calculations and a sensitivity analysis. Each case will include a discussion with regard to the research objective.

1.3.3 Research questions

Research questions are used for steering the project towards the research objective. Two types of research questions are distinguished: central and sub-questions.

The central question is of an explanatory nature and is required to be answered to achieve the research objective:

How does robust scheduling and operational uncertainties affect decision outcome and decision-making processes in Airline Operation Control?

Sub-questions are of a descriptive nature, the combined answer of all the sub-question should provide satisfactory answers to the central question:
1. What parameters and conditions are interesting to study uncertainties and robust scheduling in order to understand and evaluate decision-making processes of Airline Operations Control?
2. What aspects, criteria and setting are relevant for the scenario and model design with the aim to assess the parameters defined in sub- question 1?
3. How do controllers collect information regarding these uncertainties and how do they interact with each other and with entities located outside Airline Operations Control?
4. What recovery strategies are possible in disruption management and what are the decision criteria for executing these recovery strategies?
5. What are the sensitivities of the model parameters and conditions and what do they say about the decision-making processes?
6. How do controllers reason and what are their underlying assumptions and decision considerations?
7. How could the analysis of AOC behaviour and the conclusions drawn from this study be applied in the airline industry?

1.4 Research strategy

The literature study will be conducted by using primarily scientific articles from established journals and books. The best approach for designing a model of airline disruption management would be to use real-life airline protocols. These protocols describe lines of communication, decision-making procedures, rules and corporate culture. Due to difficulties in obtaining and freely using these protocols, the decision was made to use available literature about this topic and amend this knowledge with meetings held with industry experts. Luckily there is a large amount of qualitative data that give insight into the decision-making process of AOC. For cost calculations, equations are used that has been used in other disruption management studies.

The literature study showed that airline operational environment is a highly dynamic operating environment resulting in endless possibilities. By using a scenario it can be demonstrated on a small scale what the effect of the AOC operating environment is on the decision-making process. This scenario should be representative of a disruption that could emerge in a real-life situation. Therefore, this scenario has to be verified by industry experts. By using a frameworks of agent-based modelling, a model can be designed that promotes ontological correspondence with the real-world.

Cost calculations and sensitivity analysis of the simulations are done to discuss the effect of certain condition sequences and parameter sets on decision coutomes. This will be used to draw conclusions regarding the effect of operational uncertainties and robust scheduling on decision making processes.

1.5 Structure of the report

This report consists of eight chapters including this first chapter. Findings of the desk research regarding AOC, decision-making and agent-based modelling are described in respectively chapter two and chapter three.

The analysis and identification of the socio-technical system can be reffered to in chapter four, this includes an extensive inventory of all the relevant concepts that are required for the model design. Chapter five describes the structure of the model and the scenarios that initiates the decision making process. This chapter also includes all the tasks of the controllers and the flowcharts of the different phases of the decision-making process. Chapter six discusses the formalization of the model description which includes explanation of the language used and the design of the ontology. The analysis and the discussion of the simulation results are described in chapter seven. Conclusions that are drawn from the results analysis and recommendations for future research are discussed in chapter eight.

THIS PAGE IS INTENTIONALLY LEFT BLANK

2. Airline Operations Control

In this chapter all the theoretical knowledge concerning Airline Operations Control is laid down. Section 2.1 deals with the position of Airline Operations Control in the entire airline operational handling and the objectives it is striving to achieve. Section 2.2 discusses the core tasks of disruption management and provides insight into possible disruptions that Airline Operations Control face and the differrent types of strategies that can be deployed to recover from these disruptions. In Airline Operations Control various type of decision-makers interact with each other during the decision-making process. In section 2.3 the role of these decision-makers and the type of interaction that take place are described.

2.1 Planning and monitoring

2.1.1 Strategic Phase

Before AOC receives the schedule for execution, other departments are involved in constructing and optimizing this schedule. Development of this schedule is done in the 'strategic phase' of airline operational handling (Clarke, 1998).

The strategic phase consists of different stages with the first stage being the development of the time table (see Figure 2). In this stage, profitable routes and frequencies to fly are determined that will be used to schedule departure time, arrival time and flight destination.

The next stage of the strategic phase is resource allocation and consists of several steps. It is common practice that this starts with the aircraft resource by assigning fleet followed by individual aircraft i.e. tail assignment (Kohl et al., 2007). For the assignment the planners take into account factors like revenue per seat, noise restrictions, maintenance requirements and even gate restrictions. The result of the aircraft allocation is defined as the aircraft rotation schedule.
By using this schedule, the crew resource is allocated. This stage comprises of two actions: crew pairing and crew rostering. Crew pairing is the process of selecting crew that have to stick together during outbound and inbound flights while crew rostering involves linking designated crew with named individuals (Clausen et al., 2009).
Seat availability, pricing and revenue management is done throughout the entire planning process, from the time table development until the day of operations (Kohl et al., 2007). However, next to the major elements like aircraft, crew and passengers there are other factors considered in the strategic phase among ground staff, catering, fuelling and gates.

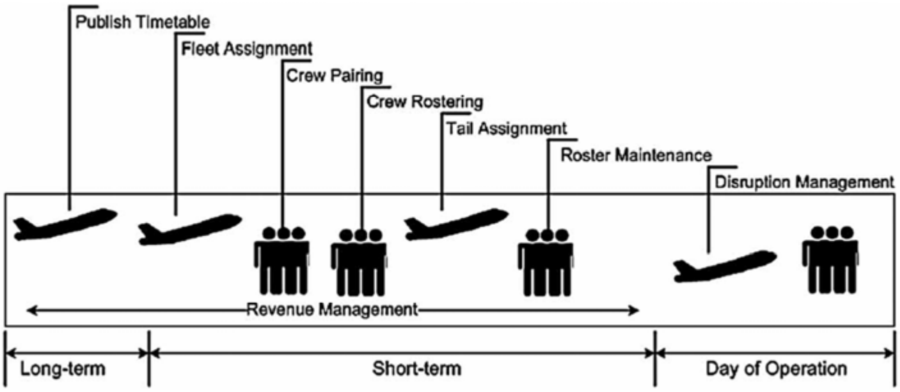


Figure 2 - Simplified illustration of scheduling of crew, passengers and aircraft in the strategic phase (Castro et al., 2014)

In addition to the time table and resource allocation, a tactical element is already introduced in the strategic phase to aid AOC, also referred to as pro-active decision-making. A certain degree of **robustness** is incorporated into the schedule to ensure continuous feasibility. A robust planning decreases the propagation of disruptions and creates opportunities in the recovery process (Lapp, 2012). Kohl et al. (2007) describes several techniques for incorporating robustness into the schedule:

1. Adding slacks in the plans – *time slack is incorporated in the plan to ensure that there is a degree of self-recovery*
2. “Crew follows each other and the aircraft” – *crew stick to each other and the aircraft during their duty, this ensures easier monitoring of airline operations*
3. Flights are “out and back” – *flights are operated with a hub and spoke structure, a cancellation of the out and ingoing flight to a certain spoke will not have an effect on the aircraft and crew schedule*
4. Having standby crew and aircraft – *there are crew pairs and aircraft standby, a reserve crew or aircraft can be valuable in the case of disruption*
5. Increased cruise speed – *flight plans are not based on maximum speed of the aircraft. In case of a delay the aircraft can fly faster to recover from the delay*

It is important to note that due to the high costs of having standby resources and adding slacks, that airlines are somewhat reserved to create robust schedules (Lapp, 2012). The most common description for slack is “non-productive” time for either crew or aircraft (Ball et al., 2006). There are several slacks that can be distinguished. This can be translated into more ground-time planned than is necessary for the aircraft, or connecting passenger or planning of crew hours that is less than the legal duty to allow for crew duty extension.

The industry is moving towards an integrated approach where the resource allocation and incorporation of robustness is done throughout the process (Belobaba et al., 2009). However, the sequential approach is still common practice in the strategic phase of airline planning.

The resulting schedule which incorporates the time table, aircraft rotations, crew allocation and other minor resources is defined as an airline system schedule (in the literature sometimes referred to as Operations Schedule, Flight Schedule or simply as Schedule).
The airline systems schedule and all its components are closely monitored until the day of operations; this process is referred to as roster maintenance. Consequently, when the day of operations is approaching, the flexibility in the airline system schedule diminishes (Kohl et al., 2007). One day before actual execution of the schedule it will transit from strategic into tactical phase (Clarke, 1998).

2.1.2 Tactical Phase

The transition of strategic into tactical phase is also a transition of responsibility. One day before the schedule is operative; the duty manager of AOC will review the schedule. There is a possibility that the duty manager rejects the schedule if it lacks robustness and/or flexibility. When the schedule is accepted by the duty manager (also referred to as the ‘handshake’- see Appendix A) the AOC will be responsible for the tactical phase of the airline operational planning.

The AOC (or in other literature referred to as OCC or AOCC) is the nerve centre of airline operations and its objective is defined as ‘*the planning and coordination of the disruption management process to achieve network punctuality and customer service while utilizing assets effectively and minimising cost.’* (Bruce, 2011)

However, airlines have different views on the objective of AOC. Some airlines’ view is that it is the responsible of AOC to have the schedule back on track at the beginning of the next day, while other airline view it as the responsibility of AOC to execute as many scheduled flights as possible (Grandeau et al., 1998). Peters (2006) defines the objective of the AOC to diminish **the difference between projected and realised quality** as much as possible (Figure 3).

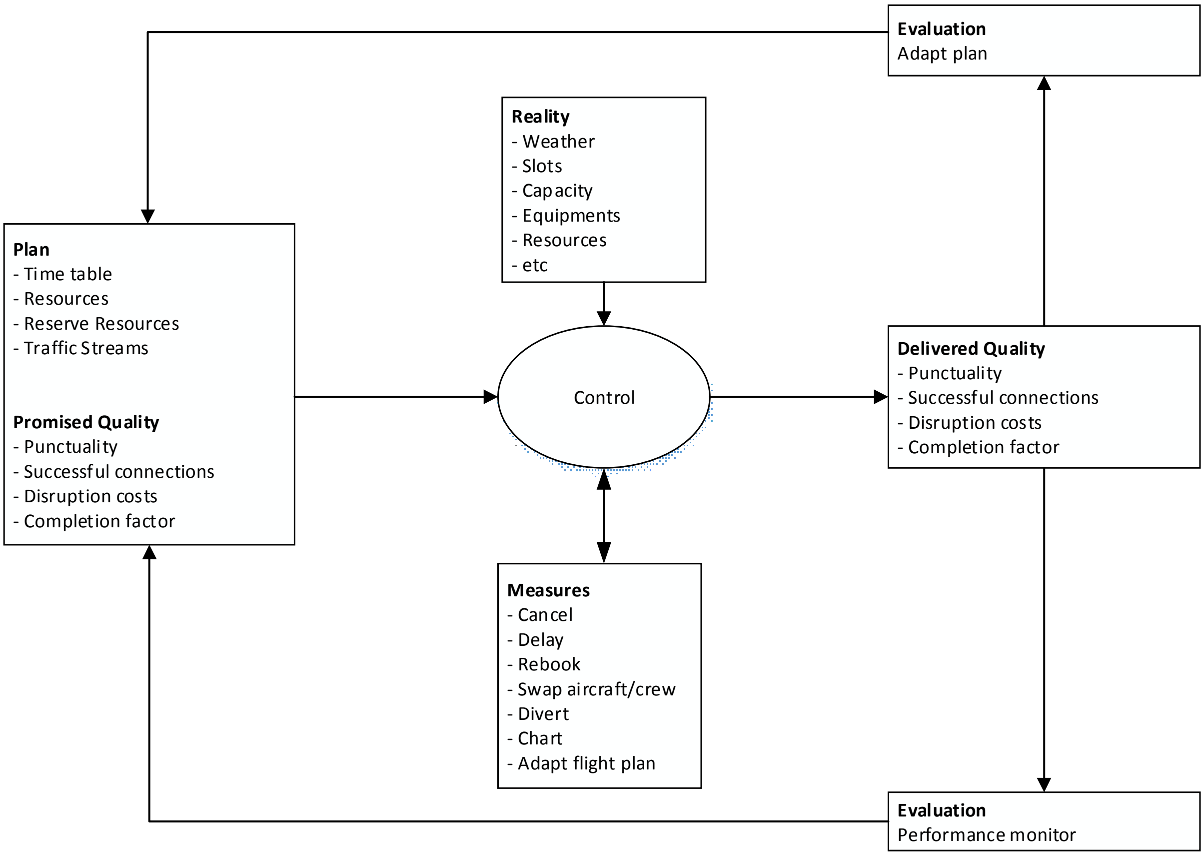


Figure 3 – AOC objectives translated from Peters (2006)

Kohl et al. (2007) identifies three objectives in disruption management: (1) get back to the plan as soon as possible, (2) minimize real costs and (3) deliver customer promise.
The reason AOC adopts the first objective is that it will make the decision-making process less complicated. Since there is a reference (i.e. original flight schedule) all decisions in the recovery process can be based on this reference. However, if the disruption is too radical, it is possible that the original flight schedule is rigorously adapted or even completely discarded which will lead to the generation of an alternative schedule (Kohl et al., 2007).

The second objective is to minimize real costs. Excess crew costs, costs of passenger compensation, rebooking expenses, hotel costs and accommodation fall into this category. Additionally, airport costs, service costs, ATC & en-route charges and fuel can be taken into account. These costs are also referred to as 'direct costs' (Castro and Oliveira, 2010).

Delivering customer promise on-time with the booked service level can also be viewed as a cost item (also referred to as passenger goodwill, soft passenger costs (Kohl et al., 2007) or quality costs (Castro and Oliveira, 2010). These costs are hard to determine since they relate to non-quantifiable items like passengers satisfaction and the likeliness that the passenger would book a ticket from this airline again in the future.

2.2 Disruptions and Recovery Strategies

2.2.1 Disruption management

As mentioned in the previous section, the core task of the AOC is disruption management. (Castro and Oliveira, 2010). Disruptions are events related to flights, aircraft, crew members and passenger that occur during schedule execution. However, a disruption is only marked as such if the event is severe to such an extent that even the incorporated robustness cannot intercept it (Clausen et al., 2009). Kohl et al. (2007) and Castro and Oliveira (2010) described disruption management as a process of continuous operation monitoring and decision-making, as illustrated in Figure 4.

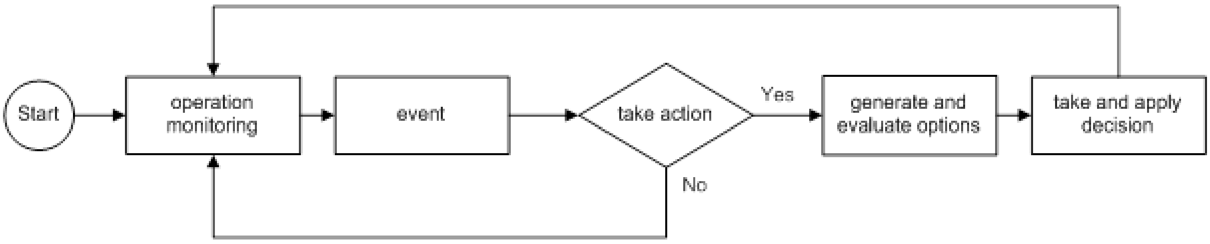


Figure 4 - The disruption management (Castro and Oliveira, 2010), (Kohl et al., 2007)

The first process in disruption management is the continuous monitoring of the operations and mirroring it with the planned airline systems schedule. Events trigger the AOC to decide whether to take action upon the event or not. If no action is required, AOC will continue with operation monitoring. If the event is recognized as a disruption, AOC will generate options and evaluate options for recovery, taking into account different perspectives e.g. passengers, aircraft, crew, ATC and airport (Castro and Oliveira, 2010). The next step in disruption management process is decision-making. Of all the generated options, the option which is acceptable with regard to the objectives of Airline Operations Control will be chosen.

2.2.2 Disruptions in Airline Operations

Barnhart and Bratu (2006) categorizes the disruptions into two groups (1) airport and airspace capacity shortage and (2) airline resource shortage i.e. there are disruptions caused by external factors acting on the operations or internal factors within the airline operations.

Regarding the external factors, Clarke (1998) describes that the main causes are due to (1) severe weather conditions, (2) runway availability and (3) airport capacity. However, it is difficult to distinguish between these factors, since severe weather conditions can lead to a decreased airport capacity.

Internal factors which cause disruptions are related to the resources aircraft, crew and ground resources.
The most common disruption related to aircraft is caused by equipment issues, also referred to as unscheduled maintenance or Aircraft On Ground (AOG). Another cause of disruptions due to the aircraft resource are the aircraft of which the scheduled maintenance took longer than expected (Barnhart and Bratu, 2006).
There are several disruptions related to the crew. Medard and Sawhney (2006) describes that flight delays and cancellations can lead to crew pairs not being able to make their connection or an infeasible crew roster. Furthermore, crewmembers calling in sick or reporting for duty could cause disruptions to the airline systems schedules. In addition, when irregular operations going on, there is a risk of crewmembers depleting their legal flight duty time during their duty (Barnhart and Bratu, 2006).
Ground resources could cause disruptions when there is a lack of them. For example, in icing conditions there is a limited number of de-icing trucks available or a lack of fuelling trucks or baggage handlers in irregular operations. (Bruce, 2011). Additional to the resources, also passengers could cause and be affected by disruptions. Passengers could be exceeding embarkation/disembarkation time or not showing up at the gate ('no show' passenger) that will lead to the time consuming activity of removing the baggage of the flight. A common passenger disruption situation is when an airline has to hold a flight so that passengers can make the connection (Wu, 2010).

Nonetheless, it has to be taken into account that the disruptions are rarely an isolated incident; a disruption in one resource could knock-on a disruptions in another resource e.g. an unavailable aircraft due to unscheduled maintenance can results in the crew not being able to connect to their next (domestic) flight, translating into a delay of that domestic flight which is affecting transit passengers that would connect to an international flight. This is also referred to as the cascading-effect (Barnhart and Bratu, 2006).

Despite the fact that robustness is incorporated in the planning, some events need to be acted upon (Kohl et al., 2007). The recovery actions are initiated to prevent further propagation, getting back to plans as soon as possible while keeping direct and quality costs down i.e. recovery policies should achieve the desired trade-off between the goal of operation recovery and the goal of commercial interests (Wu, 2010).

2.2.3 Recovery Strategies

AOC can utilize different recovery strategies for the disruptions. Wu (2010) describes that these strategies depend on the type of network the airline is operating (point-point or hub and spoke). In a point to point network, passenger disruption is not severe in contrast with crew and aircraft disruption. In a hub and spoke it would be the other way around, the crew and aircraft are flying to or from base which provides more flexibility in changing certain resources. Nonetheless, the type of recovery strategy that will be utilized depends on the circumstances.

In the field of schedule recovery studies, a distinction is made between recovering from minor and major disruptions.
In the case of a minor disruptions, common recovery tactics include speeding up turn-around processes or speeding enroute flight operations. However, for these two measures costs, have to be taken into account for allocating more ground resource staff and extra fuel costs. Regarding major disruptions, airlines are interested in both recovering the specific disrupted flight as well as reducing propagating effect on a network level. Delays and cancellations can be seen as a result from a disruption, but delaying (or re-quoting) and cancelling can also be a method for recovery (Castro and Oliveira, 2010). It can be used for instance for airport capacity shortage, airspace restrictions or when (business) transit passengers have to connect to their flight. In this case Airline Operations Control has to decide carefully which flights to proceed according schedule and which one to cancel or re-quote.

AOC has several possibilities to recover disrupted passengers and to minimize passenger convenience. Common passenger recovery strategy is to rebook these passengers on another flight of the airline its subsidiary or even on a flight of completely different airline. For passenger recovery, AOC has to take into account that there is legislation which mandates that airline must provide passengers with meals, refreshments and accommodation in certain situations (Wu, 2010).
Other major disruptions can be caused by airline resource shortage (Barnhart and Bratu, 2006). If the crew resource schedule has been disrupted, the following common strategies could be adopted: (1) deploy reserve crew, (2) positioned crew or (3) altering the crew schedule (Abdelghany et al., 2007). Altering the crew schedule can be done by swapping crew or by rebuilding crew duties (Wu, 2010). Another method of increasing crew schedule feasibility is to add slacks in the crew duty hours that could result in extending duty time for the crew.
Recovery actions for the aircraft resource are comparable to the crew actions: (1) utilization of reserve aircraft, (2) ferrying aircraft and (3) swapping aircraft/fleet and (4) diverting aircraft (Jafari and Zegordi, 2010). The aircraft swap method is a method in which the aircraft is executing another flight than initially planned (Wu, 2010). Important aspect of recovery is the reserved usage of reserve aircraft and crew. Overusing these resources could decrease flexibility for other disruptions that might take place in the network.

Next to aircraft, crew and passengers, there are various other elements in the airline system that needs to be taken into account while executing the recovery strategies. These include, airport curfew, runway availability, weather, ATC preferred routes, valuable goods and VIPs on-board.

At airlines it is current practice to recover the resources in a sequential manner. This is known as dedicated recovery. Due to the priority in costs, a typical sequence in AOC is to recover aircraft first then the crew and finally the passengers (Filar et al., 2001). Nonetheless, Clausen et al. (2009) has shown that integrated recovery is much more effective way of reducing costs and getting back to the plan as soon as possible. However, the downside of the integrated approach is that the resources are interconnected which makes the decision-making process much more difficult.

2.3 Inside Airline Operations Control

2.3.1 Structure

As mentioned earlier, AOC monitors operations, generates options and take decisions. In order to effectively monitor the various manage disruption, AOC maintains a structure with various groups monitoring specific resources and generating options to recover from any possible disruption. Additionally, decisions are made on different levels, ranging from a minor disruption of a gate-to-gate flight to major disruptions on a network level (Peters, 2006).

There are various lay out structures utilized by airlines worldwide. A typical structure of an AOC is that of an Operations Controller who is supported by several supporting groups. This structure is called "integrated-centre" and is illustrated in Figure 5 - Typical AOC structure (Clarke, 1998)(Castro et al., 2014, Clarke, 1998).

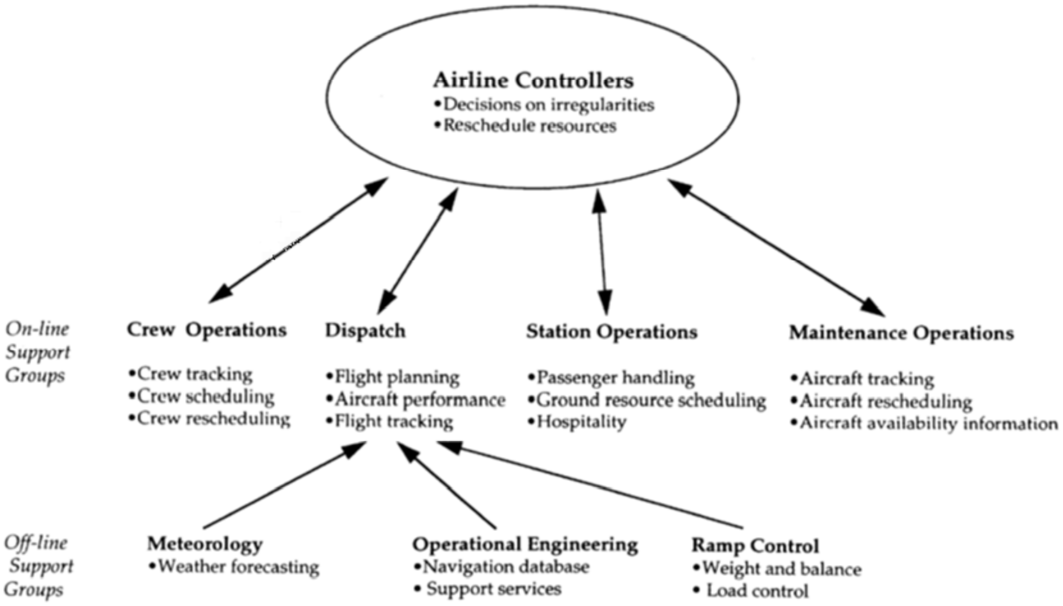


Figure 5 - Typical AOC structure (Clarke, 1998)

The Airline Controllers or referred to as Operations Controller is the main decision-maker regarding network level disruptions in this structure Furthermore, the Operations Controller coordinates the supporting groups so that they all strive towards one common objective (Kohl et al., 2007). The various supporting group make decisions on a gate-to-gate level but they also provide the Operations Controller with support regarding the resource they are responsible for.

Other typical structures which can be found in the industry are the "Decision-Centre" and the "Hub-control Centre" (Castro et al., 2014). The decision-centre is similar to the integrated centre with the difference being that aircraft control is in the same physical space as the supervisor (i.e. operations controller). The hub-control centre (HCC) is a totally different structure; in this case all the roles (i.e. supporting groups) are not in the same physical space. This requires more communication between the roles, but the main advantage is that the applicable role is physically located at the area of interest (e.g. crew related roles at crew centres or airport related roles at airports).

2.3.2 Interactions

For both structures to oversee airline operations, AOC takes actions that are defined by a combination of protocols, codes and routines (Peters, 2006). This includes "communication lines" with entities inside and outside the AOC (Feigh, 2008).

Interactions between humans is done electronically through messages and telexes or it is done verbally by phone, radio, meetings or even in person (Feigh, 2008). Some airlines believe that face to face interactions among employees would improve its ability to respond to contingencies and disruptions in its airline network (Peters, 2006) i.e. preferring the "integrated centre" type of AOC.

Within AOC some interact with technical systems to collect information. These system include schedule visualizations, weather radars and other information representation systems or data bases (Feigh, 2008).
One interesting technical system which is an area of interest in the field of disruption management is the decision-making tool. This is tool to calculate the cost effects of a certain option before a final decision is made

by the AOC. However, it requires some experience to interpret the results since there are also soft factors involved or factors that are not taken into account (see Appendix A). Peters (2006) provides an example of a controller commenting about a soft factor “The program (decision-making tool) may indicate that we need 50 minutes to turn around the plane..... But I know the people who work there, and know that they can do it in 30 minutes”. Other soft factors could be the quality costs or passenger good will that could be taken into account during the decision-making stage.

Protocols are designed to make sure that all the roles in AOC strive towards one common objective, which is to minimize the difference between projected and realized quality in performance (Peters, 2006).

2.3.3 Roles

The supervising role is the main decision-maker in AOC (Castro and Oliveira, 2010). This role is also referred to as Operations Controller, Airline Controller, Airline Operations Controller, or System Operations Controller. They have the authority and responsibility to resolve problems that develop during regular and irregular operations (Clarke, 1998). Next to having the decision-making authority, the Operations Controller coordinates the supporting groups and receives input from them.

The dispatch role (i.e. Flight Dispatch) is the supporting group that is responsible for the successful release of a flight. The tasks of Flight Dispatch include flight planning, aircraft performance calculations and flight tracking (Clarke, 1998). In some airlines, Flight Dispatch performs ATC-control related tasks e.g. requesting specific air routes and airport slots (Castro and Oliveira, 2010). At other airlines this is done by specific groups as ATC coordination or Flow Control.
Before flight, dispatch is responsible for arranging the flight plan including any weather forecasting, navigation, and load control (Clarke, 1998). The flight plan must be sent to the departure airport, a flight cannot proceed without the acceptance of the pilot. In rejection the pilot may directly interact with dispatch to request that additional flight plans be generated (Grandeau et al., 1998).
Flight dispatch is monitoring the progress of a number of flights and raises alerts with other areas when problems occur (Kohl et al., 2007). Dispatch must monitor the progress of each resource to assemble the flight. When the dispatcher realizes that deviations to the scheduled activities are sufficient to cause his flight to be delayed (i.e. a disruptions), thereby creating irregular operations, he passes this information on to the Operations Controller (Grandeau et al., 1998).
Flight dispatch is acting as an intermediary between Airline Operations AOC and Pilots, but also between AOC and ATC entities (Grandeau et al., 1998). Dispatch is able to stay in contact with the aircraft during the flight by means of sat-communication.
In the US, flight dispatch is a prominent role and shares responsibility in safety of the flight with the pilot. Therefore, dispatchers are required to have a FAA certification. At European airlines, flight following is primarily done by aircraft control (Kohl et al., 2007).

Two roles are differentiated for aircraft resource: aircraft control and maintenance services. However, some airlines combine these two roles into one.
The main tasks of aircraft control includes aircraft tracking, aircraft rescheduling and providing aircraft availability information (Clarke, 1998). Furthermore, in the case of a combination of aircraft control with maintenance services, they are also responsible for short-term maintenance scheduling and unplanned maintenance services (Kohl et al., 2007).
Aircraft control monitors the “schedule of aircraft rotations” (also referred to as aircraft schedule). They follow aircraft, and assist the Operations Controller with interchanging aircraft between flights and rotations (Grandeau et al., 1998). In a disruptive situation, aircraft control minimizes delays by changing aircraft, joining flights and rerouting or dispatching reserve aircraft (Castro and Oliveira, 2010).

Crew tracking, rescheduling and other facets of crew operation management is done by crew control. Crew control updates the schedule in terms of flying time (i.e. crew duty time) as well as qualifications of the various crew members. Additionally, crew control supports Operations Controller with crew related disruptions and recovery strategies. This involves monitoring the costs associated with the different options for how crews can be utilized (Grandeau et al., 1998).
Crew tracking involves the monitoring of individual crew member as they move through the airline’s network including the monitoring of crew check-in and check-out.

Crew rescheduling involves updating the crew roster, including the change of crew pairings in case of delays or cancellations. Furthermore, reserve crew could be called in and be re-positioned i.e. deadheading (Kohl et al., 2007, Wu, 2010).

All the tasks involving passengers, ground handling and gate scheduling is done by station operations control. However, at some airlines, passenger related tasks are separately controlled (i.e. customer services).
Decisions taken in AOC to recovery from crew or aircraft disruption typically affect passengers. The role of passenger services is to ensure minimum inconvenience for the passengers. This is done by informing and rebooking passengers. Additionally, stations operations control arranges meals, reimbursement and accommodation for passengers (Kohl et al., 2007).
Regarding ground handling, stations operations control is responsible for all ground support services involved with the turn-around process of an aircraft. (Grandeau et al., 1998).

The roles mentioned could be fulfilled by individuals or groups. The individuals who are responsible for the disruption management of AOC are referred to as controllers.

2.4 Decision-making

During disruption management controllers are in a continuous process of decision-making. Despite the fact that the study of decision-making is beyond the scope of this book, this section will provide some explanation of theories regarding decision-making processes of AOC controllers.

The terms *problem solving* and *decision-making* have been used interchangeably, but there is a difference, since decisions can be made in the absence of a problem and a problem can be solved without making a decision. Thus, decision-making will be defined as “a process involving the generation and evaluation of possible solutions to the identified problems with a view to implement the optimum solution” (Bruce, 2011).

2.4.1 Decision-making styles and Situation Awareness

Numerous studies have been done about decision-making and its relationship with uncertainty, ambiguity, risk, choice, judgement, experience and expertise (Klein, 1999). To study these factors different decision-making styles are analysed of which the most important are: rational, intuitive and naturalistic decision-making (Bruce, 2011).

Rational decision-making has been described as a logical, systematic process of analysis that occurs in a series of steps. A common rational decision-making process described by Mintzberg and Westley (2001) is (1) define, (2) diagnose, (3) design, (4) decide.
However, the problem with studies of rational decision-making is that they are focused on well-defined problems while airline operations is a complex environment with sometimes ill-defined problems and conflicting goals. This means that rational decision-making style are just partly applicable to AOC controllers (Bruce, 2011).

Decision-making can also be done by using intuition which refers to having a ‘gut-feeling’ of what might work or what might not work, or sensing what is right and wrong (Bruce, 2011). Judgment is also an intuitive type of action if it is reached by an informal and unstructered mode of reasoning, without the use of analytical methods or deliberate calculation (Kahneman and Tversky, 1982). Nonetheless, intuitive decison-making depends heavily on the use of experience to recognize key patterns that indicate the dynamics of the situation (Klein, 1999). The study of Bruce (2011) concludes that experienced AOC controllers show high intuitive decision-making.

The previous two decision-making styles are criticized since they do not account for real-world situations. Naturalistic decision-making (NDM) is an approach to analyse decision-making in a natural context. These natural settings are characterized by ill-structured problems, uncertainties, dynamic environments, shifting, ill-defined goals, competing goals, time pressure, high stakes, organization goals and norms. Many of these aspects characterise decision-making in AOC (Bruce, 2011). The model of Klein (1999) is acknowledged as the most dominant NDM-model.

To explain the model briefly, decision-maker experiences a situation in a changing context and could be collecting information to comprehend the situation. Then the decision-maker recognizes cues, goals, and the action that is required. Subsequently, the decision-maker constructs a mental simulation, this simulation project the future state of the current situation. If the decision-maker is satisfied with the simulation, a course of action is implemented.

Another major aspect in decision-making is situation awareness. Defined by Endsley (1995) as "….. the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future ".The model of Endsley (1995) is the most dominant model that explains situation awareness; in short it consists of three levels: (1) perception of the elements in the environment, (2) comprehension of the current situation and (3) projection of the future status. Situation Awareness takes place before the decision-making stage and a high degree of Situation Awareness should provide a foundation for good decision-making. Bruce (2011) described in his study that situation awareness is a key factor for AOC controllers and is developed by gathering information to cultivate decision considerations and to make sound decisions.

2.4.2 Errors in decision-making

To understand causes of pour decision-making, the question arises of how to measure decisions. The question of how decision quality is assessed is difficult to answer. Feigh (2008) described in her study that there are different benchmarking methods to determine decision quality: (1) decision outcome, (2) consistency of choice, (3) amount of time for the decision and (4) the mental effort to reach the decision. Unfortunately, each of these benchmarks has its own strengths and weaknesses and is based on rational decision-making models which are not always applicable in real-world situations.

Klein (1999) analysed more than 600 decision point in natural settings, and categorized the decisions that resulted in pour outcome into three categories (1) lack of experience, (2) lack of information and (3) due to an unsuitable mental simulation. When decision-makers are faced with time pressure it seemed that it affects their ability to collect information and to construct a decent mental simulation.
One of the causes of errors and poor decision outcome is uncertainty. Sonenshein (2007) defines uncertainty as a "...lack of information that makes constructing a plausible interpretation about a situation difficult". Uncertainty affects decision-making in various ways: it will be difficult to judge the situation as being typical (from experience) and the mental simulation will be difficult to make since there will be a lack of vision. This will lead to a doubtful course of action (Klein, 1999). There are four sources of uncertainty (1) missing information, (2) unreliable information, (3) ambiguous or conflicting information and (4) complex information.

Two tactics for managing uncertainty is to collect more information or to fill the gaps with assumptions. However, too much information could complicate the decision-making process and decrease accuracy (Tsai et al., 2008).

Since the controller at AOC collect lots of information in a complex environment, it means that they operate in an environment with a high level of uncertainty. The controllers face a constantly changing environment where certain information may not be forthcoming. Additionally, the controllers have to act and react with other controllers and system to make their decisions. These are characteristics of a socio-technical system.

THIS PAGE IS INTENTIONALLY LEFT BLANK

3. Agent-based approach

The controllers at Airline Operations Control interact with each other and with system to make decisions. These are characteristics of a socio-technical system. This type of system can be modelled using an agent-based approach. In this chapter it will be explained what an agent-based approach is (section 3.1) and what its applications and limitations are (section 3.2). The last section of this chapter provides information regarding the methodology and software used for agent-based modelling techniques (section 3.3).

Note: no agent-based model is designed in this study, since the agents will not be autonomous. However, a literature study about agent-based modelling has been conducted to identify frameworks and software used for analysing and modelling socio-technical systems

3.1 Definition of agent-based modelling

An agent-based model is defined as: "*A system modelled as a collection of autonomous decision-making entities called agents*" (Bonabeau, 2002) or as *"a new approach to modelling complex systems composed of interacting autonomous 'agents"* (Macal and North, 2010). Combining these two definitions together, it can be concluded that an agent-based model has agents that are autonomous, possess decision-making capabilities and are able to interact.

All agent-based models have the same anatomy. They consists of (1) a set of agents, (2) a set of agents' relationships and method of interaction and (3) an environment (Macal and North, 2010). These agents are the smallest element of an agent-based model. In order to classify an agent as such, they are required to possess:

1. Autonomy – there is no global controller dictating what an agent does
2. Social Ability – it is able to interact with other agents
3. Reactivity – it is able to react appropriately to stimuli coming from its environment
4. Proactivity – it has goals of goals that it pursues on its own initiative

Additionally, agents should be clearly identifiable, situated in a certain environment and designed to meet certain objectives (Jennings, 2000).

An agent has a state, which is defined as a collection of all parameters of a particular agent. The state of an agent can be changed by rules. These rules fire due to inputs from itself, other agents or the environment. The change in state can result in an action to itself, other agents or the environment (van Dam et al., 2013). There are different types of decision rules the most common used is rule-based decision rules or often called condition-actions (Gilbert, 2008). These rules based decision-rules are nested "if-then-else structures".

A structure is an important part of the modeling process of an agent-based model. It adds more realism to the model and can share characteristics with the real world (van Dam et al., 2013). An environment provides information and provides a structure or space for agent interaction. It contain what the agents may perceive and manipulate (Siegfried, 2014). The information in the environment should provide everything an agent needs to know, some environmental information is provided by the model itself, some set by the modeller, or they can be emergent. Scenarios can be tested with the model when the modeller uses a set of parameters for environmental information (van Dam et al., 2013).

3.2 Applications, benefits and limitations

3.2.1 Area of applications

Although agent-based modelling is relatively new, it already has been applied in various sciences like social, political and economic sciences (Bonabeau, 2002). The reason for the adoption is not only due to lack of other suitable modelling approach (Bankes, 2002) but because in these sciences emergent phenomena do occur and are difficult to predict.

In all these sciences, the emergent phenomena of interest can be classified into four areas:

1. Flows - evacuation, traffic and customer flow management
2. Markets - stock markets, shopbots, software agents and strategic simulation
3. Organizations - operational risk and organizational design

4. Diffusion – influence on people by their social context

Additionally, agent-based models have been used in production and manufacturing but also in Large Scale Social Technical Systems like policies and infrastructure (van Dam et al., 2013).

Organizational simulation is a promising area for agent-based modelling. Especially for modeling risk, because often risk is a property of actors in an organization. Modelling all people's activities instead of process can be easier to validate. When a reliable model of the organization is made, it is possible to change parameters and to measure how the performance of the model varies in response to these changes (Bonabeau, 2002).

3.2.2 Benefits and Limitations

In the previous it was mentioned that the most important benefit of agent-based modelling is that it captures emergent phenomena.

Another benefit of agent-based modelling is that it provides a natural description of the system, in other words, it describes what a system is doing, not what a system should be doing (Bonabeau, 2002). van Dam et al. (2013) state that this is the main difference between agent-based modelling and multi-agent systems. The latter also has autonomous agents to examine system emergence. However, the main difference is that agents in a multi-agent system are designed to achieve certain desired emergent states that are best solved from a bottom-up perspective. The third benefit of agent-based modelling identified is that it is flexible. Agents can be added or removed and features of agents can be changed with little effort.

Gilbert (2008) & Bankes (2002) mentioned that a great benefit of agent-based modelling is the ontological correspondence. There can be a direct correspondence between the agents in the model and real-world actors. For instance, an organization can include agents representing employees, customers, suppliers and any other significant actors.

With agent-based modelling having all the aforementioned benefits, it has also several limitations. Bankes (2002) identified that before capturing emergent phenomena, the human has to observe whether the occurring phenomena is indeed emergent. The benefit of ontological correspondence comes with the limitation that the humans have soft factors which are difficult to quantify and therefore difficult to model (Bonabeau, 2002).

Furthermore, due to the fact that agent-based modeling requires a certain level of detail, the model has to serve a specific purpose, a model for a general system does not work (Bonabeau, 2002). Additionally, when modelling a real-world application, it has to be considered that the model will be calculated according ticks, while the time in the real world is continuous. With the discrete time of the computer ticks, it has to be considered that in an agent-based model some processes cannot be modeled parallel, while this is the case in the real world. This problem is referred to as 'parallelism' (van Dam et al., 2013). Since agent-based modelling is a relatively new modelling concept it requires more theoretical foundation and ways to validate and accredit agent-based models (Chan et al., 2010).

3.3 Methodologies and Software

There are several methodologies and frameworks described in the literature for designing agent-based models. Wooldrigde et al. (2000) presented the "GAIA-Methodology" for Agent oriented Analysis and Design which is similar to the five step methodological framework described in Nikolic and Ghorbani (2011) and van Dam et al. (2013).

In these methodologies the first step is "system analysis". In this step the **problem and system is identified** and **conceptualized**. System identification is performed by interviewing experts, stakeholders and literature study. Conceptualization is done so that the components of the system are manageable and to see what kind of state the system can be in.

The second step is to design the model by **structuring** the agents and to identify their behaviour. Furthermore, in this step the environment is designed which consists of the components that are not part of the agents.

The third step is the **detailed design**. In this step the structuring and behaviour of the previous step need to be converted in a concept that is understandable by a computer. The ontology has to be refined and model specifications are made by creating concrete instances of the abstract classes. In the experimental phase, the desired outcome of the model is defined. This is done by determine the number of runs which are necessary, implementing a scenario and by defining parameter sweeps.

The fourth and fifth step is **software implementation** and **model evaluation**. The latter consists of three processes: verification, validation and experimentation and data analysis. The verification of an agent-based model could be measurable, but this is not always the case, especially when soft criteria have to be measured like knowledge. Three phases are considered for verification of agents (1) single agent testing - one agent is tested, (2) interaction testing – the interaction of a minimal amount of agents are tested and (3) multi-agent testing - the emergent behaviour of multiple agents is studied.

Validation of the model is usually done by performing experiment in the real world and observing the reality. However, agent-based models could be difficult to validate due to its scale and immeasurable data. The last process in model evaluation is analysing the data by experimentation. This could cause computational problem when the parameter space is too large. Validation can also be done by using scenario outputs described in the literature and compare it with outputs of the model (van Dam et al., 2013).

Although agent-based modelling is relatively new, there are numerous software environments available to model and analyse agent-based models. Common used software in the field of agent-based modelling is NetLogo and Repast.

NetLogo is a software package which is very easy to operate and is equipped with easily accessible documentation. There are also many example models from which the code can be easily extended. However, Netlogo is not capable of modelling large models and is mainly used for abstract models (Robertson, 2005).

Repast can be used for larger models, but working with this system requires knowledge of Java. The time to learn Repast without prior Java programming knowledge would be significant. Nevertheless, there is a good community and there is an extensive library of readily available models (Robertson, 2005).

Jonker et al. (2007) introduces a new formal, role-based framework to model multi-agent systems. Within this framework, roles are designed which are a subset of functionalities. These roles interacts with a conceptualized environment with an input and output interface. In this framework organizations are used as a paradigm for analysing and designing multi-agent systems. This approach captures structural and dynamic aspects of the organization. Jonker et al. (2007) identified four advantages:

1. Representation of the organization structure and dynamics by generalized models and more specific instantiated models
2. The means for simulations of different scenarios on the basis of a model and observing their results
3. Organization analysis by means of verifying static and dynamic properties against empirical data taken from real organizations, or against simulated scenarios
4. Diagnosis of inconsistencies, redundancies, conflicts and errors in organizational model by means of formal verification techniques

LEADSTO is used to model and analysis dynamic aspects of organization. LEADSTO is a software environment and a sublanguage of Trace Temporal Language (TTL). The language TTL is a variant of an order-sorted predicate logic and is introduced to address various modelling demands (Sharpanskykh and Treur, 2010). One of the features of TTL is that it is able to express both qualitative and quantitative aspects which are also present in the real world system. Furthermore, by using TTL, dynamic properties can be specified and analyzed.

LEADSTO is an executable language derived from TTL, it enables modelling direct temporal dependencies between state properties in successive states (Popova and Sharpanskykh, 2010). Additionally, properties defined in the LEADSTO format can be graphically depicted in a causal graph-like format (Bosse et al., 2007).

LEADSTO has been used in various multi-agent systems to analyse and understand system behaviour (Bosse et al., 2007, Hoogendoorn et al., 2008).

THIS PAGE IS INTENTIONALLY LEFT BLANK

4. Identification of socio-technical system

Before the model can be designed, the socio-technical system has to be identified and described (Nikolic and Ghorbani, 2011). This is done by first choosing a relevant scenario that is credible, sufficiently complex and a representative sample of the socio-technical system (section 4.1). Subsequently, an inventory is compiled of all relevant concepts of the socio-technical system (section 4.2).

4.1 Scenario Selection

AOC could be involved in various types of disruptions that result in numerous decision considerations and decision outcomes. Therefore, a scenario will be chosen to narrow down the scope of this study to have more of an in-depth analysis of its properties. This scenario will be used to describe the socio-technical system first, but will also be used to initiate the decision-making process of the model.

Three requirements for the scenario selection will be defined in order for it to be credible, complex, representative and analysable:

1. The scenario characterizes a disruption that could emerge in a real world situation and includes sufficient uncertainties
2. The behaviour of the controllers towards the scenario and uncertainties is derivable from the literature
3. The scenario includes a disruption on a gate-to-gate level that effects multiple resources so that all primary controllers play a role (see Figure 5 - Typical AOC structure (Clarke, 1998)

To comply with the above mentioned requirements, it is evident that a scenario should be selected that is used in the real-world. In the study of Bruce (2011) 52 controllers of six AOCs were exposed to a number of scenarios in a naturalistic setting. The aim of this study was to investigate decision-making processes of controllers by recording their thought processes that are verbally expressed as they provide decision consideration. With the think-aloud protocol a vast amount of qualitative data has been recorded that provide a wealth of information that describe the socio-technical system the controllers are operating in.

The controllers were being provided with an airline systems schedule of the flights and were requested to express where they would look at. In this familiarization stage, the controllers were looking into ground time of aircraft and availability of spare aircraft. After this stage, one of three scenarios was presented in the form of briefings. These scenarios represented a typical airline operational problem: (1) passenger connection problem, (2) aircraft technical problem and (3) weather problem:

A. **Passenger Connection Problem**
The time is 2100. Flight 703 operating from London (LHR) to Pacific (PCF) has been unserviceable in LHR. The aircraft has eventually departed two hours late and is picking up time in Pacific. The ETA in PCF is 2315. There is no crewing problem. All tranships are OK, except a ministerial delegation of 55 connecting with Flight 714 to Melbourne.

B. **Aircraft Mechanical Problem**
The time is 0655. Flight 705 is unserviceable in Paris (CDG). The engineers report that it has a hydraulic leak such that it may require a hydraulic pump change. If so, then they expect the pump change to take two hours. On this advice, the staff at CDG has stopped checking passengers in for Flight 705.

C. **Weather Problem**
The time is 0100. There is an alternate on Pacific due to typhoon warning level 5. The typhoon is stationary at present 140 nautical miles north east of Pacific.

However for the scenario to qualify for the description of the socio-technical system, it has to comply with the aforementioned requirements:

1. Since the scenarios of Bruce (2011) are designed and verified by a panel of industry experts, it complies with the first requirement

2. With the think-aloud protocol, controllers expressed their thought process and thus describing the actions and interactions they would take towards the scenario. These thoughts have been recorded as ‘comments’. This means that the behaviour of the controllers is derivable from the literature, meeting the second requirement
3. In both scenario A and B, the controllers were interested in solving aircraft, crew and passengers problem. Scenario C focused more on the effect of a weather disruption on a network level. This means that scenario A and B meet the third requirement

Concluding, scenario A and B meet all the requirements that have been set out. This means that scenario A and B will be selected for the description of the socio-technical system in this chapter and for the initiation of the decision-making process in the next chapter.

4.2 Inventory

In order to design a model for this scenario an inventory has to be compiled of all the actions, actors, objects, interactions and all the other relevant concept of the socio-technical system (van Dam et al., 2013, Nikolic and Ghorbani, 2011). The qualitative data of the study of Bruce (2011) is not sufficient to solely describe the socio-technical system. For this reason, the following sources are used to amend the information that is provided in the comments:

- Feigh (2008)
- Clarke (1998)
- Grandeau et al. (1998)
- Castro et al. (2014)
- Kohl et al. (2007)
- Visits at KLM OCC and interview with a director flight operations (Appendix A)
- Informal meetings with an operating officer
- Authors’ own experience as an intern in line maintenance at Schiphol airport

4.2.1 Actions of controllers

Bruce (2011) recognized for both scenarios that the controllers regard certain fundamental information to increase their situation awareness irrespective of the situation and setting. Controllers were interested in crewing and passenger loadings and transits in both scenarios. However, in scenario B controllers seem to be especially interested in information related to aircraft maintenance & repair. In order to analyze and recognize patterns of the controllers’ actions, a qualitative data analysis is used in which codes are assigned to each verbal expression (Miles et al., 2014). These verbal expressions will be further referred to as ‘comments’. In order to condense the large amount of comments into smaller number of analytic units, the codes are categorized.

The 104 comments of scenario A and B and the assigned codes are recorded in Appendix B and have resulted in twelve categories AA-AL listed in Table 1.

action cat.	Summary of action category	Comment ascription
A-A	Controllers are seeking information or requesting prognosis about the mechanical problem. Since the aircraft mechanical failure takes place at an outstation, controllers are questioning the adequateness of the **technical diagnosis** that is provided by the local technicians.	37, 38, 49, 78, 79, 87, 93, 95, 104
A-B	To repair the aircraft, controllers are looking for **spare parts** both in-house and at other airlines to repair the mechanical problem.	48, 51, 55, 80, 87, 89, 95, 104
A-C	To determine the resulting delay due to repair of the aircraft, controllers are seeking information regarding the **duration of the repair,** some controllers are rechecking whether the proposed repair time is definite or not	40, 47, 48, 50, 52, 58, 92, 93, 94
A-D	To determine whether the aircraft could be repaired on the apron, the controllers determine the **weather pattern**	58
A-E	In the case of adverse weather pattern, the controllers are looking into availability of **hangar space**	58

A-F	The controllers are looking at possibilities to dispatch a **reserve aircraft** or to join two other flights that might result in a spare aircraft that can be utilized to get the passengers back to base. Dispatching a reserve aircraft to get passengers is also referred to as **ferrying**. Additionally, controllers were looking at swapping aircraft (changing the flights in the aircraft schedule to free up an aircraft)	41, 52, 53, 60, 62, 67, 69, 73, 84, 90, 91
A-G	A considerable amount of comments relate to determine the remaining **crew duty time/flight time** and minimum rest time required to operate the aircraft back to base or to even extend crew duty hours	4, 17, 19, 20, 22, 23, 25, 26, 39, 42-45, 49, 52, 54, 55, 63, 67, 71, 83, 85, 97
A-H	To solve problems that arose from exceeding crew tour of duty or for dispatching reserve aircraft, controllers could be looking into the deployment of **reserve crew**	4, 30, 46, 67
A-I	Since there is a possibility that there is no reserve crew available at the station where the disruption takes place, the controllers are looking at possibilities to **position** crew i.e. Next to positioning crew, controllers are also interested in positioning technicians and parts to solve the mechanical problem	30, 46, 80, 88, 89, 104
A-J	The delay that resulted from recovery tactic could have effect on the **connection** of **transit passengers**. Therefore controllers are looking into where these passengers will go next (after the current flight) and whether they will make a successful connection given the expected delay	1, 2, 5, 7, 8, 9, 10, 11, 12, 13, 14, 32, 36, 49, 57, 59, 65, 73, 74, 77
A-K	The controllers are looking into the capacity at other fights, even from other airlines to determine **rebooking** possibilities for passengers	26, 28, 51, 60, 61, 66, 81, 82, 86, 96, 97, 99, 100
A-L	To make sure that the transit passengers are able to make a successful connection, the controllers look into **organizing connection measures.** The measures include aspects like increasing flight speed, holding next flight or organizing parking bays next to each other, accelerate turn-around or a combination of the aforementioned measures	13, 14, 15, 24, 32, 50, 64, 70, 71, 72, 76

Table 1 – actions of controller towards the scenario categorized

4.2.2 Objects, actors and interactions

In each action category the controllers are collecting information regarding a certain item to overcome uncertainty. These items will be referred to as “object” and do not necessarily have to be a physical component, but is merely a description about ‘what’ the controllers are collecting information.

For example, in category A of the comments, the controllers are uncertain about the adequateness of the object “technical diagnosis” and therefore information is being collected. The reason for this uncertainty is due to the fact that it is either second hand information or that the technicians on the outstation are unable to provide an adequate diagnosis. Furthermore, controllers are uncertain about the existence of certain recovery possibilities, availability of resources/parts and the favourability of weather patterns. Additional to the aforementioned uncertainties, there are also objects that are time based. These time-based objects are listed due to the fact that controllers seem to be interested in actual times regarding any recovery tactic. For instance, when crew positioning opportunities are being explored, controllers are interested in actual times of this recovery possibility i.e. the time for position flight to arrive at the station of disruption in order to determine (1) its effect on transit passengers and (2) the subsequent sectors of the aircraft. All objects are listed in Table 2 with the factors that play a role.

Action cat.	**Object**	**Uncertain about**	**Factors in the environment**
A-A	Technical Diagnosis	Adequateness	• First hand/second hand information • Expertise/Experience of local technician
A-B	Spare Part	Availability	• Inventory at outstation • Availability in maintenance pool

			• Lead-time of part
A-C	Repair time	Duration	• Complexity of the task • Manpower availability • Experience of local technicians
A-D	Weather Pattern	Favourability	• Expected weather pattern • Reliability of meteo forecast
A-E	Hangar Space	Availability	• Hangar schedule of partner airlines
A-F	Reserve Aircraft	Availability	• Robustness of schedule i.e. planned reserve aircraft • Usage of reserve aircraft by previous shift • Ability to postpone scheduled maintenance to free up a spare aircraft • Joining/cancelling flights that might end up in a spare aircraft • Ability to swap an aircraft in the aircraft schedule • Retention of reserve aircraft for other anticipated disruption on a network level
	Ferry time	Duration	• Prepare time of reserve aircraft (pre-flight checks, ground operations) • Arr/dept time of reserve aircraft • Flight time from base to station of disruption
A-G	Crew duty time	Duration	• Crew schedule (initial plan) • Cascading effect of previous flights • Possibilities to extend crew hours • Crew rest
A-H	Reserve crew	Availability	• Robustness of schedule i.e. planned reserve crew • Retention of reserve crew for other anticipated disruptions on a network level • Joining/cancelling flights that might end up in a spare crew • Qualification of reserve crew meets the requirements regarding aircraft type and route
A-I	Positioning seats	Availability	• Availability of flights from base to station of disruption • Load factor of those flights
	positioning time	Duration	• Time till positioning flight depart
A-J	Transit-buffer time	Duration	• Time required to transit/number of transit passengers • Profile of passengers
A-K	Rebooking	Possibilities	• Availability of planned flights from station of disruption to base • Load-factor of flights • Ability of ground operations to transfer baggage from the mechanically failed aircraft to the rebooking flight
	Rebooking time	Duration	• Time until rebooking flight departs
A-L	Connection measures	Possibilities	• Robustness of schedule, i.e. increasing flight speeds, accelerate turn-around time • Possibilities to delay connecting flight • Possibilities to organizing parking bays • Number of transit passengers

Table 2 – Inventory of objects and their uncertainties

In section 2.3.3 the applicable roles within AOC have been described in general. In Table 3 an inventory is provided of all the actors that could be applicable for this scenario including their responsibilities both within and outside the AOC. Categorization of the functional groups is derived from Figure 5.

Functional groups	**Actor**	**Responsibilities**	**Alternative names**
Operations Control	Operations Controller	• Meeting AOC objectives • Coordination among supporting groups • Implementation of recovery strategies	• Airline Controller • Airline Operational Manager • System Operations Controller • Airline Operations Controller
Supporting Group	Aircraft Controller	• Aircraft rotation schedule feasibility including scheduled and unscheduled maintenance • Aircraft availability information	• Maintenance Services • Aircraft Router • Maintenance Operations Control • Technical specialist
	Crew Controller	• Crew schedule feasibility • Crew availability information • Positioning crew	• Crew Tracking • Crew Scheduler • Crew Operations
	Station Operations Controller	• Passenger handling • Rebooking passengers/Seat reservations • Ground handling • Passenger accommodation	• Customer Service • Passenger Service • Station Controller • Station Operations
	Flight Dispatch	• Monitoring flight progress/aircraft load • Flight planning • Gate planning • Weather monitoring • ATC coordination	• Dispatcher • Flight Controller • Despatch • Flight dispatch and following
Off-line groups	Meteorological Bureau	• Weather monitoring	• Weather Bureau
	Ramp Control	• Load Control • Mass and balance	-
	Operational Engineering	• Navigation database	• Navigation supporting • Flight Planning support
other in airline operations	Maintenance Department	• Coordinating and planning of scheduled and unscheduled maintenance	• Maintenance Control Center
	Local Technician	• Performing technical diagnosis	
	Crew	• Operating flights within their flight duty time limitations	• Flight Crew
	Ground Operations	• Performing turn-around processes of the flights	• Ground control
	Airport	• Assigning gates and parking bays for aircraft	• Airport manager • Platform coordinator
	Partner Airlines	• Providing rebooking seats and spare parts to allied airlines	-

Table 3 – Inventory of actors in the socio-technical system

Note that only the actors that of Operations Control or the Supporting Groups are regarded as ‘controllers’.

To have a clear idea of which actors are involved in each action category and about what object they are collecting information about, the actors and action category and objects will be linked to each other and listed in Table 4. Additional to the actors and objects, the system that could be involved in each action category is described using primarily Feigh (2008), Castro et al. (2014), Bruce (2011) and Grandeau et al. (1998).

Action cat.	**Supporting group involved within AOC**	**Actors involved outside AOC**	**Object**	**technical systems involved**
A-A	• Aircraft Controller	• Local Engineer • Maintenance department	• Technical diagnosis	• Maintenance Manuals
A-B	• Aircraft Controller	• Maintenance department • Partner Airlines Part supplier	• Spare part	• Spare parts Inventory system
A-C	• Aircraft Controller	• Local Technician • Maintenance department	• Repair time	• Maintenance database • Maintenance manual
A-D	• Flight Dispatch	• Meteorological bureau	• Weather pattern	• Meteo-forecast software
A-E	• Aircraft Controller	• Maintenance department	• Hangar space	-
A-F	• Aircraft Controller • Flight Dispatch	-	• Reserve aircraft • Ferry Time	• Aircraft Schedule
A-G	• Crew Controller	• Crew	• Crew duty time	• Crew Schedule
A-H	• Crew Controller	• Reserve Crew	• Reserve Crew	• Crew Schedule
A-I	• Station Operations Controller • Crew Controller	• Partner airlines	• Seats for positioning • Positioning time	• Crew Schedule • Seat reservation system
A-J	• Station Operations Controller	• Airport manager • Duty manager	• Transit buffer time	• Time table
A-K	• Station Operations Controller	• Partner airlines • Load Control • Ground Operations	• Rebooking flight • Rebooking time	• Seat reservation system • Time table
A-L	• Flight Dispatch	• Airport manager • Ground operations • Crew • ATC Coordinator	• Connection measures	• Flight planning software

Table 4 - linking actors, objects and system

From Table 4 it can be observed that the Operations Controller is not linked with any specific comment category. The reason for this is that the Operations Controller is responsible for the coordination between the supporting group and formulation of the recovery strategy, meaning that the Operations Controller could be involved in all action categories.

Feigh (2008) recognized that Operations Controllers continuously request status updates from the supporting groups. These supporting groups are either looking into information, or providing information to the Operations Controller or other members of the supporting group. The comments of Bruce (2011) show the same interaction pattern: informing/being informed, requesting/being requested and observing/looking into certain type of information (see Table 5). These interactions are also referred to as “speech acts” and are also used in the model of Castro et al. (2014) and (Hoogendoorn et al., 2008).

Category	Interaction example	Example of comments
Inform	• Providing information to a controller • Ensure that a department is looking at certain object • Being briefed about a disruption	37, 39, 49, 52, 57, 58, 78, 95
Request	• Asking an actor to provide information regarding an object	13, 14, 16, 44, 50, 51, 52, 57, 59, 88, 91
Observe	• Checking availability of an object • Checking adequateness of a certain type of information • Looking into certain information	49, 63, 64, 72, 102

Table 5 – interaction types

4.2.3 Objectives and recovery

Each controller has responsibilities that are stated in Table 3, but there are also general objectives towards which the AOC controllers are collectively striving for. By inventorying all the objectives listed in the literature, four General Objectives (GO) categories could be identified: Executing the schedule, delivering customer service, utilizing resources effectively and minimizing costs. However, these objectives cannot be regarded as individual objectives that needed to be achieved, but are highly coupled e.g. if the first three objectives are met, then this results in reduced costs. The objectives including their literature ascriptions are listed in Table 6.

GO #	Category	General objectives (GO)	Ascribed to
GO1	Execute schedule	Completion factor (finish as much flights as possible)	Peters (2006)
		Achieve successful transit connections	Peters (2006)
		Get back to plan as soon as possible	Kohl et al. (2007)
		Maintain current operational version	Feigh (2008)
GO2	Deliver customer Service	Deliver promised service level	Kohl et al. (2007)
		Achieve customer service	Bruce (2011)
		Minimize quality costs	Castro et al. (2014)
		Achieve passenger punctuality	(Peters, 2006)
GO3	Utilize resources Effectively	Utilize assets effectively	Bruce (2011)
		Minimize utilization of reserve resources	Feigh (2008)
		get aircraft and crew back to plan as soon as possible	Kohl et al. (2007)
GO4	Minimize Costs	Minimize real costs	Kohl et al. (2007)
		Minimize direct costs	Castro and Oliveira (2014)
		Minimize quality costs	Castro and Oliveira (2014)

Table 6 – Inventory of general objectives

Feigh (2008) state that the Operations Controller weighs competing objectives when deciding which solution to implement for a given problem. Additionally, Operations Controllers are not provided specific guidance to cope with these competing objectives. This issue has also been emphasized during the interview held with a director of KLM OCC (Appendix A).

Clausen et al. (2009) state that AOC is recovering the resources in a sequential manner and that the aircraft seems to be the first resource that has to be recovered. This can also be deduced by the fact that the majority of the comments in Bruce (2011) relate to the category “aircraft schedules and patterns”. However, in scenario A, the controllers commented specifically about avoiding passengers being stranded at outstation.

To model decision-making processes, it has to be determined how the Operations Controller copes with these competing objectives and what kind of decision-making pattern is exhibited in the real-world. Table 7 includes statements that are backed by comments and literature and will be used in the subsequent chapter to model the Operations Controller’s decision-making pattern.

Importance of objectives	**Ascribed to comment/literature**
Controllers show opportunistic behaviour regarding avoiding passengers being stranded at outstation i.e. passengers have to be brought back to base regardless the effort that has to be made	• Comment#: 75, 82, 102
The aircraft is the most valuable tangible asset of an airline. Aircraft recovery is considered to be very important aspect since the controllers are very interested in its subsequent sectors of the aircraft and that they know that aircraft recovery will result in both crew and passenger recovery	• comment: #5, 6, 30 • Clausen et al. (2009) • Wu (2010)
Controllers recognized that ensuring transit passengers making a successful connection is worth utilizing reserve resources for	Comment: #60, 61, 62, 72, 73
Rebooking the passengers is not a preferred way of passenger since it is considered bad customer service	Grandeau p.159

Table 7 – Observations regarding importance of objectives

Decision outcomes are the end result of ' …response patterns exhibited by an individual when confronted with a decision situation (Bruce, 2011). For the AOC this means that recovery strategies will be deployed to recover from the disruption. Section 2.2.3 already provided an overview of the possible recovery options that can be implemented by AOC. However, to specify the decision outcomes possible for scenario A and B, these will be listed and categorized into Table 8 with its literature ascription.

Resource involved	**Decision outcome**	**ascribed to**
Aircraft	• Plan unscheduled maintenance and or position resources like parts or technicians to that aircraft on outstation to avoid long term AOG situations • Dispatch a reserve aircraft (ferrying) to pick up the passengers from outstation or to position the resources to the aircraft that has to be repaired	• Bruce (2011) • Clausen et al. (2009) • Feigh (2008)
Crew	• Extend crew hours or let crew rest • Deploy reserve crew to operate ferry flight or for positioning purposes	• Bruce (2011) • Clausen et al. (2009)
Passengers	• Rebook passengers onto other flights • Organize connection measures to make sure transit passengers will make their flight • Delay the flight to finish repair, or for the positioning crew to arrive or for the reserve aircraft to arrive • Accommodate passengers (cancel flight)	• Bruce (2011) • Wu (2010) • (Grandeau et al., 1998)

Table 8 – Inventory of recovery types

The inventory all relevant concepts of the airline operational socio-technical system provide a basis for the modeling part in the subsequent chapter.

5. Model description

By using the socio-technical analysis, the model can be designed. First, the outline of the model will be designed (section 5.1). This includes the scenario, uncertainties and the recovery. In section 5.2, these uncertainties will be conceptualized and the controllers' tasks to overcome these uncertainties will be defined. Section 5.3 discusses the entire structure of the decision-making process by using cross-functional flowcharts. In section 5.4 of this chapter the cost equations and data will be introduced that will be used to evaluate the costs of the recovery strategies. Note that these cost equations are not used during the decision-making process, but only for evaluation purposes.

5.1 Outline of the model

5.1.1 Scenario, uncertainties and objectives

In chapter 4 it was defined that scenario A and B from the study of Bruce (2011) are suitable scenarios to be used for the model. These two scenarios are combined and adapted to meet the specific needs of this study:

The time is 0900 UTC. Flight DL 1945 is about to be operated by crew 'A' from AMS to DLF with aircraft PH-TUA. During the pre-flight check, the technician reports a hydraulic leak such that it may require a hydraulic pump change. The staff at AMS (which is an outstation of DLM) has stopped checking in the passengers for the flight. There are transits passengers on board that have a connecting flight at DLF (DLM's home base). Due to company procedures, the crew contacts Flight Dispatch of Airline Operations Control department to communicate their findings.

This initiates the modeling and simulation of the decision-making process and ends when the decision-making stage is completed i.e. it ends as soon as AOC has made a final decision about the recovery strategy. Five controllers will interact with each other and with the environment to eventually formulate a recovery strategy:

1. Operations Controller (OC)
2. Aircraft Controller (AC)
3. Crew Controller (CC)
4. Station operations Controller (SC)
5. Flight Dispatch (FD)

All other actors that were specified during the socio-technical system will not be specified during the decision-making process, but could be part of the conceptualized environment.

During the decision-making process, these controllers will be confronted with uncertainties. They will collect information to formulate a recovery strategy regarding the scenario on a gate-to-gate level. These uncertainties are abstracted from the objects which were listed earlier in chapter 4 in Table 2. Even though some uncertainties could be seen as robust scheduling parameters (e.g. reserve crew availability) these will all be regarded as uncertainties since the controllers showed in the analysis to be collecting information about these objects. There are time-based and non-time based uncertainties that are listed in respectively Table 9 and Table 10.

#	**Uncertainty**	**Description**
a	Technical Diagnosis adequateness	adequateness of the technical diagnosis provided by the local technicians at AMS
b	Spare Part availability	availability of spare parts at AMS for solving the mechanical failure of aircraft TUA[1]
c	Weather Pattern favourability	favourability of the weather pattern at AMS for the repair of aircraft TUA at apron

---

[1] Aircraft TUA - the aircraft with tail number PH-TUA that has the mechanical failure

d	Hangar Space availability	availability of hangar space at AMS for the repair of aircraft TUA
e	Organizing Connection possibility	the possibility to hold the (next) connecting flight or to increase flight speed or accelerate turn around for the purpose of a successful connection of passengers
f	Positioning crew possibility	availability of seats from DLF to AMS for either positioning reserve crew , or to position resources like technicians and parts to AMS
g	Reserve crew availability	availability of reserve crew for either positioning from DLF to AMS, or for deployment to dispatch reserve aircraft from DLF to AMS
h	Rebooking possibility	possibility to rebook passengers on other flights that depart from AMS to DLF
i	Reserve aircraft availability	availability of reserve aircraft to ferry empty from DLF to AMS, to pick up the passengers A at AMS and bring them back to DLF

Table 9 – non-time based uncertainties in the simulation

	Time uncertainty	**description**
r_t	Repair time	the time that is required to repair aircraft TUA
c_t	Crew duty slack time	the crew duty time slack that is available for crew A[2] to complete the flight back to DLF
d_t	Positioning time	the time before the positioned reserve crew arrives at AMS to take over flight DL 1945
k_t	Ferry time	the time for reserve aircraft to fly from DLF to AMS i.e. the time for the passengers to wait for the reserve aircraft to arrive
b_t	Rebooking time	the time for the rebooking flight to depart from AMS to DLF
p_t	Transit-buffer time	the buffer in time the transit passengers have on flight DL 1945 to make a successful connection

Table 10 – time based uncertainties in the simulation

Controllers will strive to achieve AOC objectives and will do so by formulating the best recovery strategy possible under the uncertainties. The AOC objectives (GOs) where shown in Table 6 and were categorized in one of the following categories (GO1) Execute schedule, (GO2) Deliver high customer service level, (GO3) Utilize resources effectively and (GO4) Minimize costs. To have a list of objectives that is specific to the scenario presented in this section, the GOs will be translated into seven Scenario Objectives (SO) listed in Table 11.

R	**SO#**	**Scenario objectives**	**AOC General objective**

			GO1	GO2	GO3	GO4
Aircraft	SO1	Get aircraft TUA back to plan as soon as possible	✓		✓	✓
	SO2	Avoid usage of reserve aircraft			✓	✓
Crew	SO3	Get crew A back to plan as soon as possible	✓		✓	✓
	SO4	Avoid usage of reserve crew			✓	✓
Passengers	SO5	Avoid passengers being stranded	✓	✓		✓
	SO6	Avoid rebooking of passengers		✓		✓
	SO7	Make sure transit passengers connect successfully	✓	✓		✓

Table 11 – seven scenario objectives and their relevance towards the general AOC objectives

What can be observed from Table 11 is that all scenario objectives are striving to minimize costs.

5.1.2 Decision outcomes

To achieve the scenario objectives, the AOC will deploy a strategy to recover from the disruption that was presented in the scenario. Just like the main components of an airline systems schedule, the recovery strategy should declare something about aircraft, crew and passengers. Each of these resources will be further divided into two categories: (A) Aircraft and (B) Reserve Aircraft, (C) Crew A and (D) Reserve Crew, (E) Passengers 'A' and (F) Transit passengers.

From Table 8 of the socio-technical analysis it was already determined that there are several decision outcomes available for each resource. Table 12 presents the possible decision outcomes (DO) for each of these resources.

Res. scat.	**Resource element**	**DO #**	**The resource is:**
Aircraft	Aircraft TUA	A1	put under "unscheduled maintenance" and becomes serviceable after repair time (r_t)
		A2	AOG and awaits resources that will be positioned from DLF to AMS
		A3	AOG at AMS without any short term prospect
	Reserve aircraft	B1	Being dispatched to AMS
		B2	Not utilized/available
Crew	Crew A	C1	Crew hours extended/waiting to complete the flight back to DLF
		C2	Resting/accommodated at AMS
	Reserve crew	D1	Positioned from DLF to AMS to replace crew and operate aircraft TUA
		D2	Operating reserve aircraft
		D3	Not utilized/available
Pa sse	Pax A	E1	Experiencing a **delay** as long as the repair takes (r_t)

		E2	Experiencing a **delay** as long as the positioned crew takes to take over flight (d_t)
		E3	Experiencing a **dela**y as long as the ferrying of the reserve aircraft takes (k_t)
		E4	being rebooked and experiences a **delay** as long as the rebooking flight will depart (b_t)
		E5	Will be confronted with a **cancellation/long delay** and will be **accommodated** at AMS
	Tpax A	F1	Will make a successful connection
		F2	Will not make a successful connection

Table 12 – decision outcomes per resource

A recovery strategy for the scenario is a combination of the six types of possible decision outcomes (A-F) for each resource. This means that $3 \times 2 \times 2 \times 3 \times 5 \times 2 = 360$ combinations can be compiled with the above mentioned decision outcomes. However, some combinations produce an invalid combination e.g. it is not possible that the passengers will be accommodated at AMS <u>and</u> that they will make a successful connection on the planned connecting flight. For this reason 15 conditional statements[3] have been defined that all have to be true for a recovery strategy to be valid (see Table 13).

# conditional statement	Conditional statement that has to be true	Description
1	$(A2 \rightarrow C2) \vee$ $(A3 \rightarrow C2)$	Crew A follows aircraft TUA**:** When aircraft TUA is 'AOG', then crew A will always be accommodated
2	$(D1 \rightarrow B2) \vee$ $(D2 \leftrightarrow B1) \vee$ $(D3 \longleftrightarrow B2)$	Reserve aircraft always requires a reserve crew (1) When reserve aircraft will be dispatched, then reserve crew is also utilized (2) When reserve crew will be positioned, then it is not possible to also dispatch reserve aircraft (3) When reserve crew is not utilized, then reserve aircraft cannot be dispatched
3	$B1 \rightarrow \neg A3$	Dispatching reserve aircraft provides the opportunity to position resources and is utilized to pick up passengers When reserve aircraft is dispatched, it will not leave aircraft TUA being AOG without positioning necessary resource to that aircraft
4	$B1 \rightarrow E3$	Dispatching reserve aircraft is always done to pick up passengers from outstation
5	$D1 \rightarrow A1$	Reserve crew is only positioned when aircraft TUA is under unscheduled maintenance
6	$D1 \rightarrow C2$	When reserve crew is positioned, this means that crew A has insufficient crew duty hours (since crew A follows aircraft TUA)
7	$(\neg A1 \rightarrow \neg E1)$	When aircraft TUA is AOG, then the passengers cannot wait for repair to be finished
8	$(E5 \rightarrow C2)$	When passengers are accommodated then crew A is also accommodated
9	$C1 \rightarrow \neg E5$	When flight duty time is sufficient, then pax will **not** be accommodated
10	$(\neg D1 \rightarrow \neg E2)$	When reserve is not positioned to AMS, then passengers will not wait on positioning crew to arrive
11	$(D1 \rightarrow \neg E5)$	When reserve crew is positioned, then passengers will not be accommodated

---

[3] These conditional statements have been defined through the entire process of model design

12	$E5 \rightarrow F2$	When transit passengers are accommodated, then they will miss their connection
13	$A1 \wedge C1 \wedge F2 \rightarrow \neg(B1)$	When aircraft TUA can be repaired and crew A has sufficient flight duty time and dispatching reserve aircraft will lead to transit passengers making an unsuccessful connection then this reserve aircraft will never be dispatched
14	$A1 \wedge D1 \wedge F2 \rightarrow \neg(E2)$	When aircraft TUA can be repaired and crew A has sufficient flight duty time and transit passengers ending up making an unsuccessful connection it is assumed that passengers are not waiting for this positioned crew, but are either rebooked or waiting for repair
15	$A1 \wedge C1 \wedge F2 \rightarrow \neg(E4)$	When aircraft TUA can be repaired and crew A has sufficient flight duty time and transit passengers ending up making an unsuccessful connection, then this means that passengers are not rebooked

Table 13 – conditional statements to determine valid recovery strategies towards the scenario

These conditional statements are put in a spreadsheet with all the 360 combinations and by using truth-tables (see Table 14); all invalid recovery strategies could be identified.

P	Q	$P \rightarrow Q$	$\neg Q$	$P \rightarrow \neg Q$	$\neg P \rightarrow \neg Q$	$Q \rightarrow P$
1	1	1	0	0	1	1
1	0	0	1	1	0	0
0	1	1	0	1	1	1
0	0	1	1	1	1	1

Table 14 – Truth-tables to determine valid statements

This process resulted in twenty valid recovery strategies that are listed in Table 15.

	Aircraft decision outcome		**Crew decision outcome**		**Pax decision outcome**	
RS#	**A**	**B**	**C**	**D**	**E**	**F**
RS1	A1	B2	C1	D3	E1	F1
Aircraft TUA will be repaired, crew A and pax A will wait at the airport until repair is finished. Despite the delay due to repair, the transit pax will make a successful connection						
RS2	A1	B2	C1	D3	E1	F2
Aircraft TUA will be repaired, crew A and pax A will wait at the airport until repair is finished. Due to repair of the aircraft, the transit passengers will not make a successful connection						
RS3	A1	B2	C2	D3	E5	F2
Aircraft TUA will be repaired. However, crew A has to rest and will be accommodated. Pax A will also be accommodated and therefore the transit pax will not make a successful connection						
RS4	A1	B2	C1	D3	E4	F1
Aircraft TUA will be repaired and crew A waits at the gate until repair is finished. However, pax will be rebooked onto another flight and the transit pax will make a successful connection						
RS5	A1	B2	C2	D3	E4	F2
Aircraft TUA will be repaired and crew A is accommodated. However, pax will be rebooked onto another flight, but the transit pax will not make a successful connection						
RS6	A1	B2	C2	D3	E4	F1
Aircraft TUA will be repaired and flown back after crew A had its crew rest while the passengers will be rebooked on another flight which results in transit pax will make a successful connection						
RS7	A1	B2	C2	D1	E2	F1
Aircraft TUA will be repaired and crew A is resting. Pax are waiting for positioned reserve crew to arrive at AMS so that this crew will operate aircraft TUA back to DLF. The transit pax will make a successful connection						
RS8	A1	B1	C1	D2	E3	F1

<table>
<tr><td colspan="7">Aircraft TUA will be repaired and crew A is waiting at the airport to operate aircraft TUA. Pax A are waiting for the reserve aircraft to arrive at AMS so that they will be ferried to DLF resulting in transit pax making a successful connection</td></tr>
<tr><td>RS9</td><td>A1</td><td>B1</td><td>C2</td><td>D2</td><td>E3</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA will be repaired and crew A is accommodated to rest. Pax A are waiting for the reserve aircraft to arrive at AMS so that they will be ferried to DLF resulting in the transit pax making a unsuccessful connection</td></tr>
<tr><td>RS10</td><td>A1</td><td>B1</td><td>C2</td><td>D2</td><td>E3</td><td>F1</td></tr>
<tr><td colspan="7">Aircraft TUA will be repaired and crew A is accommodated to rest. Pax A are waiting for the reserve aircraft to arrive at AMS so that they will be ferried to DLF resulting in the transit pax making a successful connection</td></tr>
<tr><td>RS11</td><td>A1</td><td>B2</td><td>C2</td><td>D1</td><td>E4</td><td>F1</td></tr>
<tr><td colspan="7">Aircraft TUA will be repaired and crew A is resting. Reserve crew is positioned to ‘pick up’ aircraft TUA. However, pax will be rebooked onto another flight and the transit pax will make a successful connection</td></tr>
<tr><td>RS12</td><td>A1</td><td>B2</td><td>C2</td><td>D1</td><td>E4</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA will be repaired and crew A is resting. Reserve crew is positioned to ‘pick up’ aircraft TUA. However, pax will be rebooked onto another flight and the transit pax will not make a successful connection</td></tr>
<tr><td>RS13</td><td>A3</td><td>B2</td><td>C2</td><td>D3</td><td>E5</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and crew A and pax A will be accommodated without any short term prospect for recovery</td></tr>
<tr><td>RS14</td><td>A2</td><td>B2</td><td>C2</td><td>D3</td><td>E5</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and awaits resources that are positioned with another flight. Crew A and pax A will be accommodated</td></tr>
<tr><td>RS15</td><td>A2</td><td>B1</td><td>C2</td><td>D2</td><td>E3</td><td>F1</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and awaits resources that are positioned using reserve aircraft. Crew A is accommodated to wait for aircraft TUA to be serviceable again. Pax A will wait at the airport for reserve aircraft to arrive so that they will be ferried to DLF. This results in a successful connection for the transit pax</td></tr>
<tr><td>RS16</td><td>A2</td><td>B1</td><td>C2</td><td>D2</td><td>E3</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and awaits resources that are positioned using reserve aircraft. Crew A is accommodated and waits for aircraft TUA to be serviceable again. Pax A will wait at the airport for reserve aircraft to arrive so that they will be ferried to DLF. This results in an unsuccessful connection for the transit pax</td></tr>
<tr><td>RS17</td><td>A2</td><td>B2</td><td>C2</td><td>D3</td><td>E4</td><td>F1</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and awaits resources that are positioned with another flight. Crew A will be accommodated until aircraft TUA is serviceable again. Pax A will be rebooked and the transit pax will make a successful connection</td></tr>
<tr><td>RS18</td><td>A2</td><td>B2</td><td>C2</td><td>D3</td><td>E4</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and awaits resources that are positioned with another flight. Crew A will be accommodated until aircraft TUA is serviceable again. Pax A will be rebooked and the transit pax will not make a successful connection</td></tr>
<tr><td>RS19</td><td>A3</td><td>B2</td><td>C2</td><td>D3</td><td>E4</td><td>F1</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and Crew A is accommodated without any short term prospect for recovery. Pax A will be rebooked and the transit pax will make a successful connection</td></tr>
<tr><td>RS20</td><td>A3</td><td>B2</td><td>C2</td><td>D3</td><td>E4</td><td>F2</td></tr>
<tr><td colspan="7">Aircraft TUA is AOG and Crew A is accommodated without any short term prospect for recovery. Pax A will be rebooked and the transit pax will not make a successful connection</td></tr>
</table>

Table 15 – Possible recovery strategies for the scenario

5.2 Controllers and the conceptualized environment

5.2.1 Controllers’ tasks and responsibilities

By using the action inventory of Table 1 and actor description of Table 3 and Table 4 from chapter 4, the responsibilities and tasks of the controllers will be described that will be performed during the decision-making process. For the supporting groups this translates into the following responsibilities:

- **Aircraft Controller** has the responsibility to maintain feasibility of the aircraft schedule by assisting Operations Controller with aircraft availability information
- **Crew Controller** has the responsibility to maintain feasibility of the crew schedule by assisting Operations Controller with crew availability information
- **Station Operations Controller** has the responsibility to handle passengers and to manage seat reservations
- **Flight Dispatch** has responsibilities for flight planning, flight progress monitoring and weather monitoring

The tasks of the supporting group is listed in Table 16

Controller	Task ID	the task of the controller is to determine:	Ascribed to comment category
Aircraft Controller	AC1	adequateness of the technical diagnosis	A-A
	AC2	availability of spare parts for repair	A-B
	AC3	repair time of the aircraft technical problem	A-C
	AC4	hangar space availability	A-E
	AC5	reserve aircraft availability	A-F
Crew Controller	CC1	effect of repair on crew hours	A-G
	CC2	availability of reserve crew	A-H
	CC3	crew positioning time	A-I
Station Operations Controller	SC1	effect of delay on passenger connections	A-J
	SC2	rebooking possibilities	A-K
	SC3	positioning possibilities	A-I
Flight Dispatch	FD1	weather favourability at station of disruption	A-D
	FD2	possibilities to organize connection measures	A-L
	FD3	ferry time	A-K

Table 16 – tasks of the supporting group

Operations Controller has the responsibility (1) to coordinate between the controllers and (2) to choose one of the recovery strategies.

Controller	Task id	The task description
Operations Controller	OC1	Coordination includes requesting information from controllers or providing necessary information to controllers. e.g. requesting availability of reserve resources
	OC2	To choose one of the recovery strategies listed in Table 15

Table 17 – tasks of the operations controller

By using the specification of the importance of objectives inTable 7 and the scenario objectives of Table 11 a priority list for the Operations Controller can be made. Table 18 shows which objectives will be prioritized over other objectives and thus determine which recovery strategies will be preferred to be implemented.

Priority		Description	SO#	DO# to be true
1		Avoid passengers being accommodated/stranded due to cancellation or due to crew rest	SO5	$\neg E5$
2	2-1	Get aircraft TUA and crew A back to plan after repair is finished or after reserve crew is being positioned to fly the aircraft TUA and crew A back to DLF	SO1 &SO3	$(A1 \land C1)$ V $(A1 \land D1)$
	2-2	Get aircraft TUA and crew A back after crew rest		$(A1 \land C2)$
	2-3	Make sure aircraft TUA has some perspective regarding recovery by positioning parts or technicians		$A2$

3	Make sure transit passengers connect successfully	SO7	$F1$
4	Avoid usage of reserve aircraft	SO2	$B2$
5	Avoid usage of reserve crew	SO4	$D3$
6	Avoid rebooking of passengers	SO6	$E1 \vee E2 \vee E3$

Table 18 – priority of OC of implementing recovery for disruptions

For example, the first priority is to avoid that the passengers are being accommodated. This means that the recovery strategies in which E5 is true will be the least attractive to implement i.e. RS3, RS13 and RS14.

5.2.2 Conceptualized Environment

For the environment, the objects of Table 2 will be used which represents the uncertainties the controllers face during their decision-making process. These uncertainties will be conceptualized and parameterized in order to be observed by the controllers.

All the uncertainties that are not time-based will be "conditions". These conditions are Boolean valued with a closed world assumption and are presented in Table 19. The values of the subscripts can be either 0 or 1.

Condition	Conditions values	description
a_j	a_0	Inadequate technical diagnosis
	a_1	Adequate technical diagnosis
b_k	b_0	Spare parts unavailable
	b_1	Spare parts available
c_l	c_0	weather pattern unfavourable
	c_1	weather pattern favourable
d_m	d_0	hangar space unavailable
	d_1	hangar space available
e_n	e_0	organizing connection not possible
	e_1	organizing connection possible
f_o	f_0	position opportunities unavailable
	f_1	position opportunities available
g_p	g_0	reserve crew unavailable
	g_1	reserve crew available
h_q	h_0	rebooking opportunities not available
	h_1	rebooking opportunities available
i_r	i_0	reserve aircraft unavailable
	i_1	reserve aircraft available

Table 19 – Conceptualized Boolean valued environmental conditions

All the time based objects will be parameters that can be represented by integers and will have a subscript with "t". For the modelling purposes all these parameters will be in minutes (see Table 20).

Parameter	Parameter name	Description
r_t	Repair time	the time to repair aircraft TUA
d_t	Positioning time	the time before the positioned reserve crew arrives at AMS to replace crew A

k_t	Ferry time	the time for reserve aircraft to fly from DLF to AMS i.e. the time for the passengers to wait for the reserve aircraft to arrive
b_t	Rebooking time	the time for the rebooking flight to depart from AMS to DLF
c_t	Crew duty slack time	the crew duty time slack that is available for crew A[4] to complete the flight back to DLF
p_t	Transit buffer time	the slack time the transit passengers have on flight DL 1945 to make a successful connection

Table 20 – conceptualized time based parameters

From the recovery strategies it could be seen that the delay for passengers can be as long as repair time, positioning time, ferry time and rebooking time (see Expression 1).

$$d_f = \{r_t, d_t, b_t, k_t\}$$

d_f = passenger delay

Expression 1 - set of passenger delays

Therefore, these parameters will be conceptualized such that they can be regarded as delay for the passengers. For instance, when passengers are waiting for delay to be finished after repair, then the passenger delay will be as long as the repair time takes (not taking into account factors like ground handling or other turn around processes). Crew duty time (c_t) and transit buffer time (p_t) cannot be regarded as passenger delay.

5.3 Model structure

The simulation starts at the moment the scenario description of section 5.1.1 ends i.e. the moment Flight Dispatch observes the mechanical problem of aircraft TUA and ends when a recovery strategy is chosen.

The decision-making process is executed in different phases. From the socio-technical analysis it was deduced that the aircraft has to be recovered first (phase 1). The second phase is that the effect of the repair on other resources will be determined (phase 2A), or in case of an aircraft AOG, possibilities to position resources will be explored (phase 2B). The last step is the Operations Controller implementing a recovery strategy by requesting information from the supporting groups (phase 3A and 3B). A general overview of the model and its phases is illustrated in Figure 6.

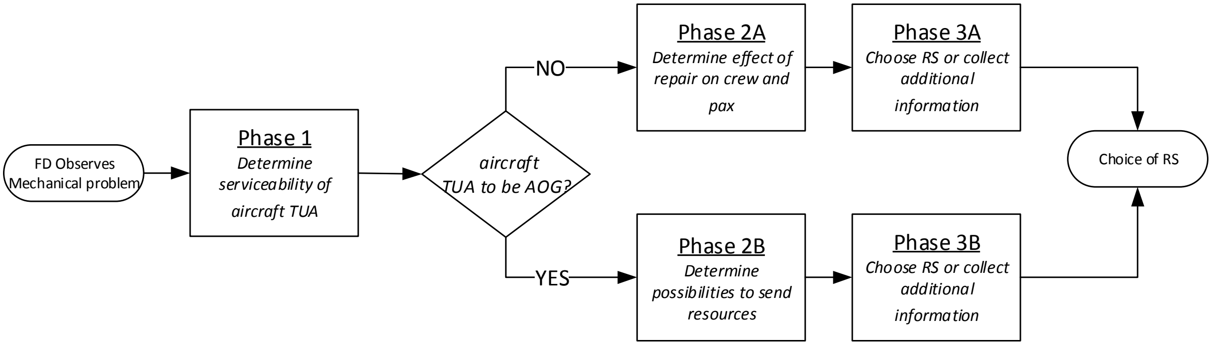


Figure 6 – The different phases in the decision-making process

For each phase it is determined which controllers are involved, which tasks are to be performed, and what environmental parameters and condition are used. The tasks will be represented by task-ids defined in section 5.2.1 and are translated into interactions. For all interactions, speech-acts will be used that were defined in Table 5 of chapter 4.

To have a clear view of the structure of each phase, cross-functional flowcharts will be used that include processes representing interactions of controllers with other controllers or with the environment (ENV). The grey colored processes are interactions with the environment; any colored processes are related to processes that are depicted on other cross-functional flowchart.

5.3.1 Phase 1

As soon as flight dispatch observes the aircraft mechanical problem, the aircraft Controller will be informed regarding this problem.

Aircraft Controller would like to turn aircraft TUA back to service and will therefore determine adequateness of technical diagnosis (AC1) and spare parts availability (AC2). From Table 1 it could be determined that the controllers were interested in weather favourability to determine whether the aircraft could be repaired on the apron or not. Therefore, the Aircraft Controller will request weather information from Flight Dispatch (FD1) to identify hangar space is necessary (AC4). The final task in this phase is to determine the duration of the repair (AC3). The tasks of the controllers in this phase are shown in Table 21.

Controller	Task-ID	Environmental conditions and parameters
AC	AC1	a_j
	AC2	b_k
	AC3	r_t
	AC4	d_m
FD	FD1	c_l

Table 21 – phase 1 controller tasks and environmental conditions and parameters observed

However, when Aircraft Controller finds out that the aircraft cannot be put under unscheduled maintenance, the Operations Controller is informed (next phase 2B). When Aircraft Controller determines that aircraft TUA can be put under unscheduled maintenance, then also Operations Controller is informed (phase 2A).

The cross-functional flowchart of phase 1 is illustrated in the next page in Figure 7.

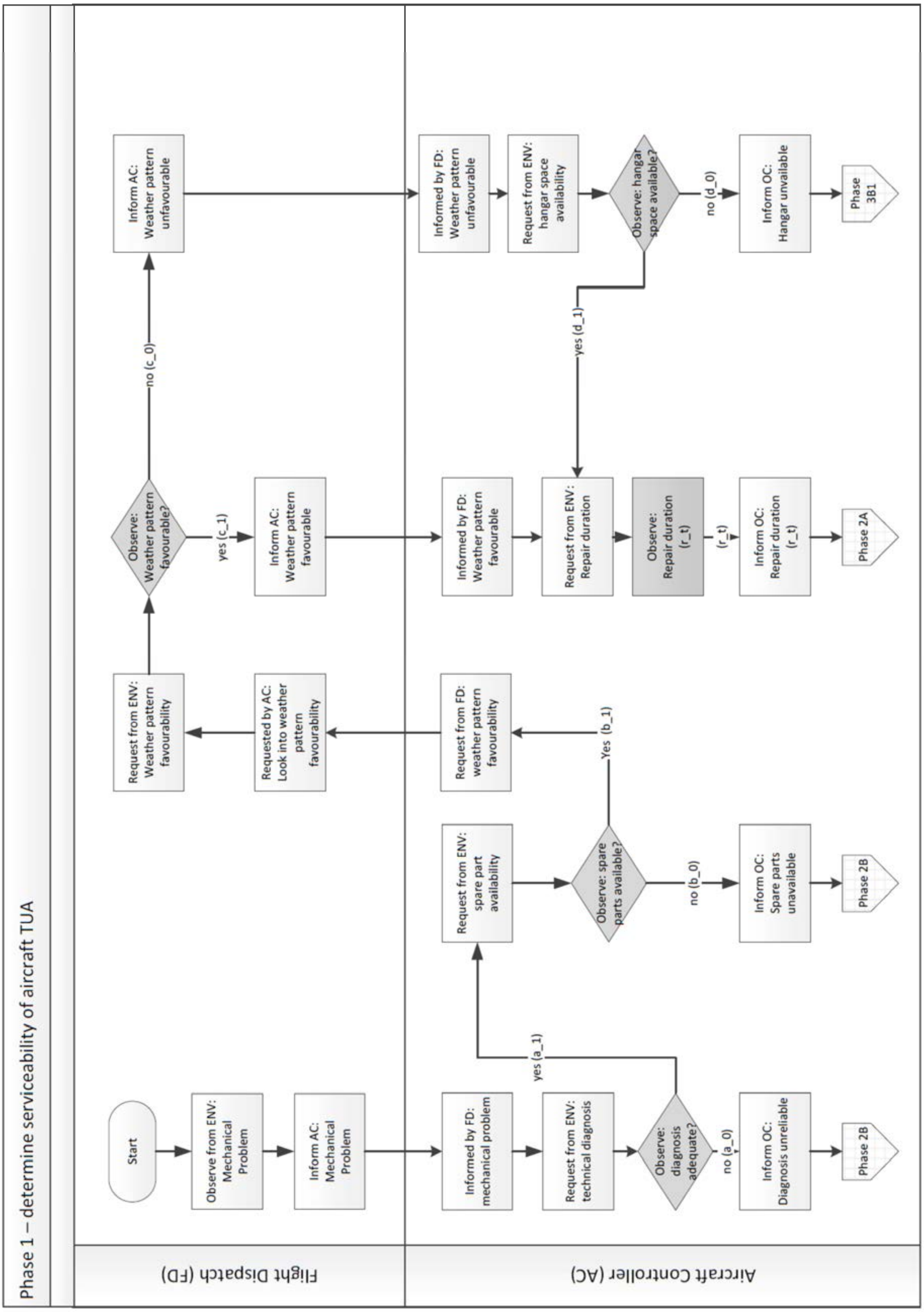


Figure 7 - Phase 1 – Cross-functional flowchart

5.3.2 Phase 2

Since phase 2 consists of two sub-phases (2A and 2B) they will be discussed separately.

Phase 2A Effect of repair on crew and passengers

From the socio-technical analysis it was evident that when the Operations Controller is informed about the repair duration, he would like to know what the effect of repair time is on passengers and crew. The Station operations Controller and Crew Controller will be requested to look into this.

The Station operations Controller will compare the transit buffer time with the repair time (SC1). When he observes that the passengers will be affected by repair time, then it will request Flight Dispatch to look into organizing connection measures (FD2). Eventually, Station operations Controller will inform OC whether the transit passengers will be affected by the repair or not.

When Crew Controller receives a request to look into the effect of repair duration on crew duty time it will also look into crew solutions. Crew Controller will compare the repair duration with crew duty slack time (CC1), look into reserve crew availability (CC2), request from Station operations Controller positioning seats availability (SC3) and looks into positioning time of the crew (CC3).
Eventually, there are three possible outcomes of Crew Controller to the Operations Controller, Crew Controller will inform that the flight duty time is sufficient (phase 3A-1) or it will inform OC about positioning crew being possible (phase 3A-2) or the lack of crew positioning possibilities (phase 3A-3).

See Table 22 for the controller tasks and applicable environmental conditions and parameters

Controller	Task-ID	Environmental conditions and parameters
CC	CC1	c_t
	CC2	g_p
	CC3	d_t
SC	SC1	p_t
	SC3	f_o
FD	FD2	e_n
OC	OC1	-

Table 22- Phase 2A controller tasks and applicable environmental conditions and parameters

The cross-functional flowchart of phase 2A is illustrated in the next page in Figure 8.

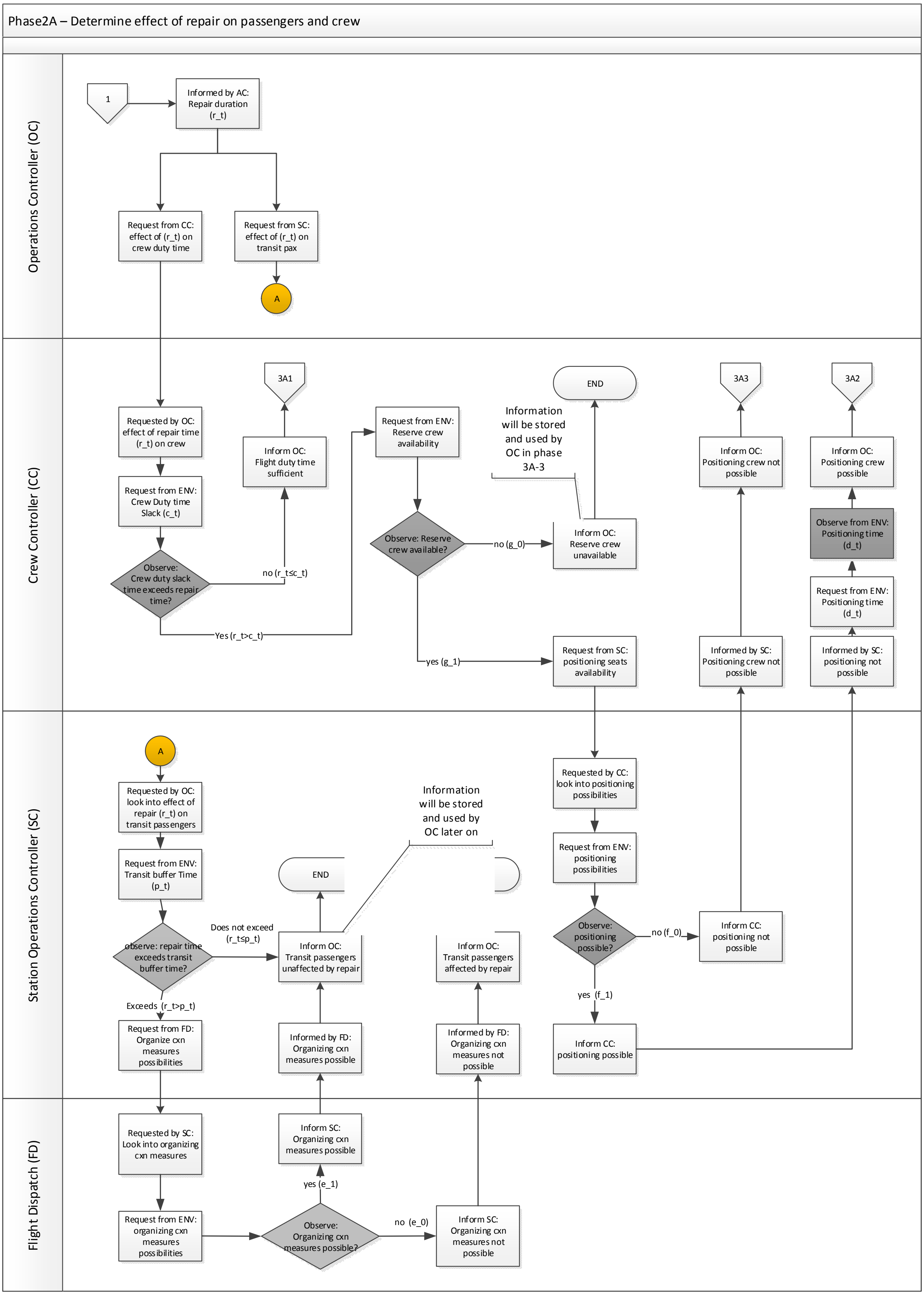


Figure 8 - Phase 2A – Cross functional flowchart

Phase 2B determine possibilities to position resources

When the Operations Controller is informed by Aircraft Controller about inadequate technical diagnosis or unavailability of parts, then he will request from Station operations Controller to look into the possibility to position resources to the outstation for the aircraft to be repaired. This phase involves only Operations Controller and Station operations Controller (see Table 23).

Controller	Task-ID	Environmental conditions and parameters
SC	SC3	f_n
OC	OC1	-

Table 23 – Phase 2B controller and task and applicable environmental parameters

After Station operations Controllers has determined whether resources can be positioned, then Operations Controller will be informed regarding these findings. When there are positioning possibilities the next phase will be 3B-1 and when there is no positioning possibility the next phase will be 3B-2 (see Figure 9).

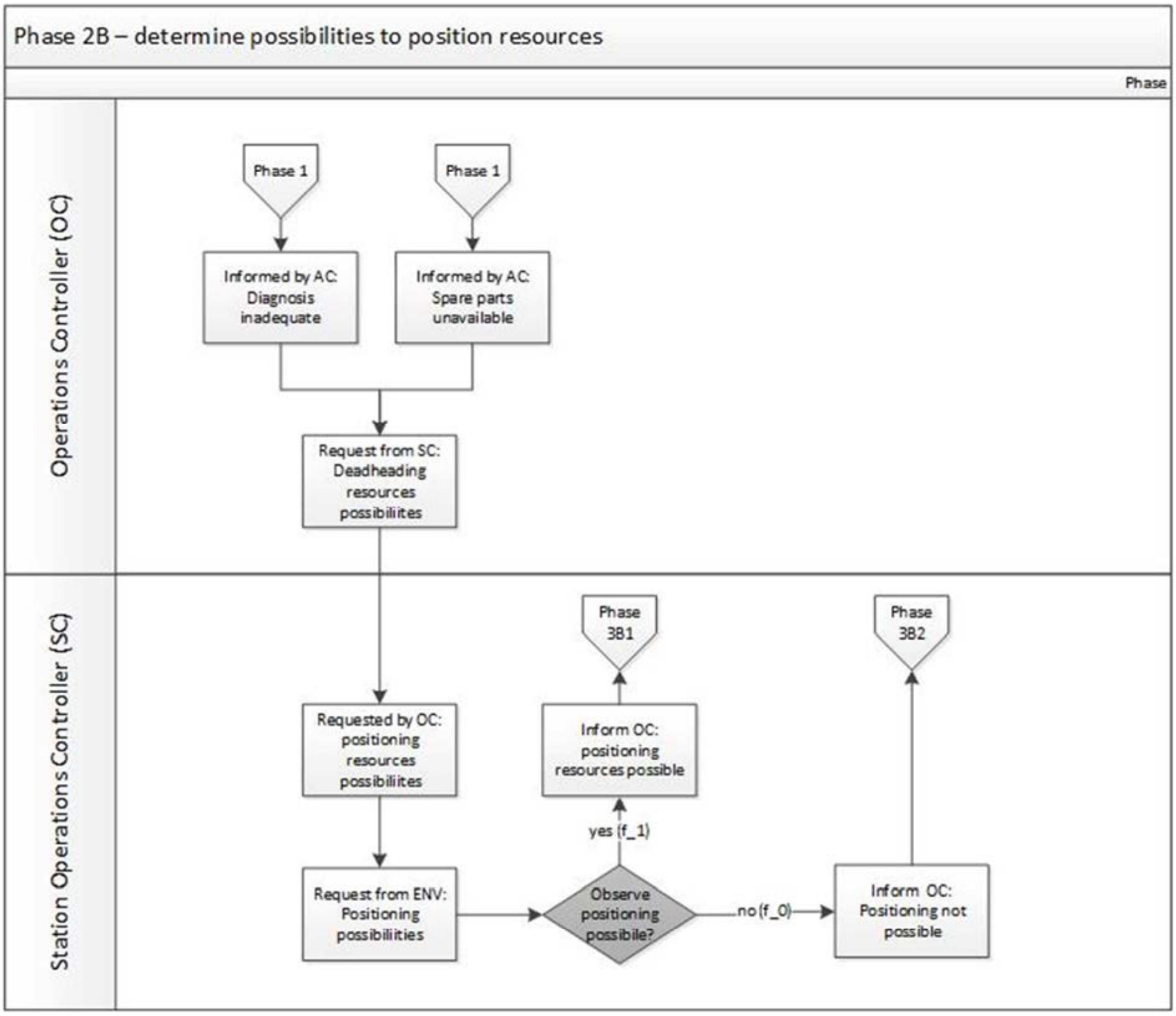


Figure 9 - Phase 2B- Cross functional flowchart

5.3.3 Phase 3

For phase 3 there are two types of flowcharts, the OC flowcharts and SG flowcharts. The SG-flowcharts includes the supporting activities of Aircraft Controller, Flight Dispatch, Station operations Controller and Crew Controller, while the OC flowcharts are about Operations controller choosing the best possible recovery strategy.
The SG-flowcharts are applicable to all flowcharts of phase 3 and are illustrated in Figure 10 and Figure 11. The colors of the start of each flowchart correspond to the request of the OC in the OC-flowchart.

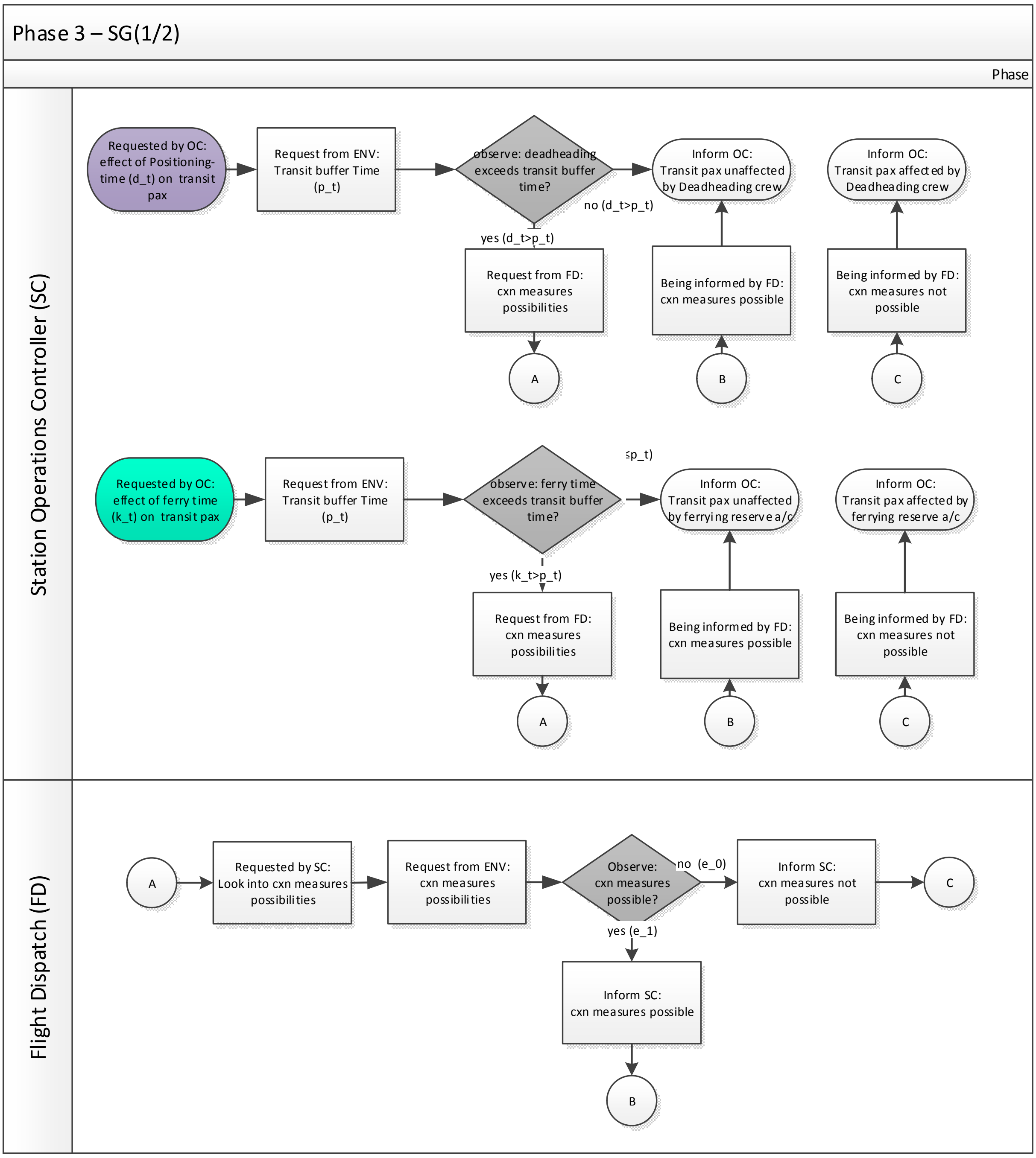


Figure 10 - Phase 3-SG1 cross functional flowchart

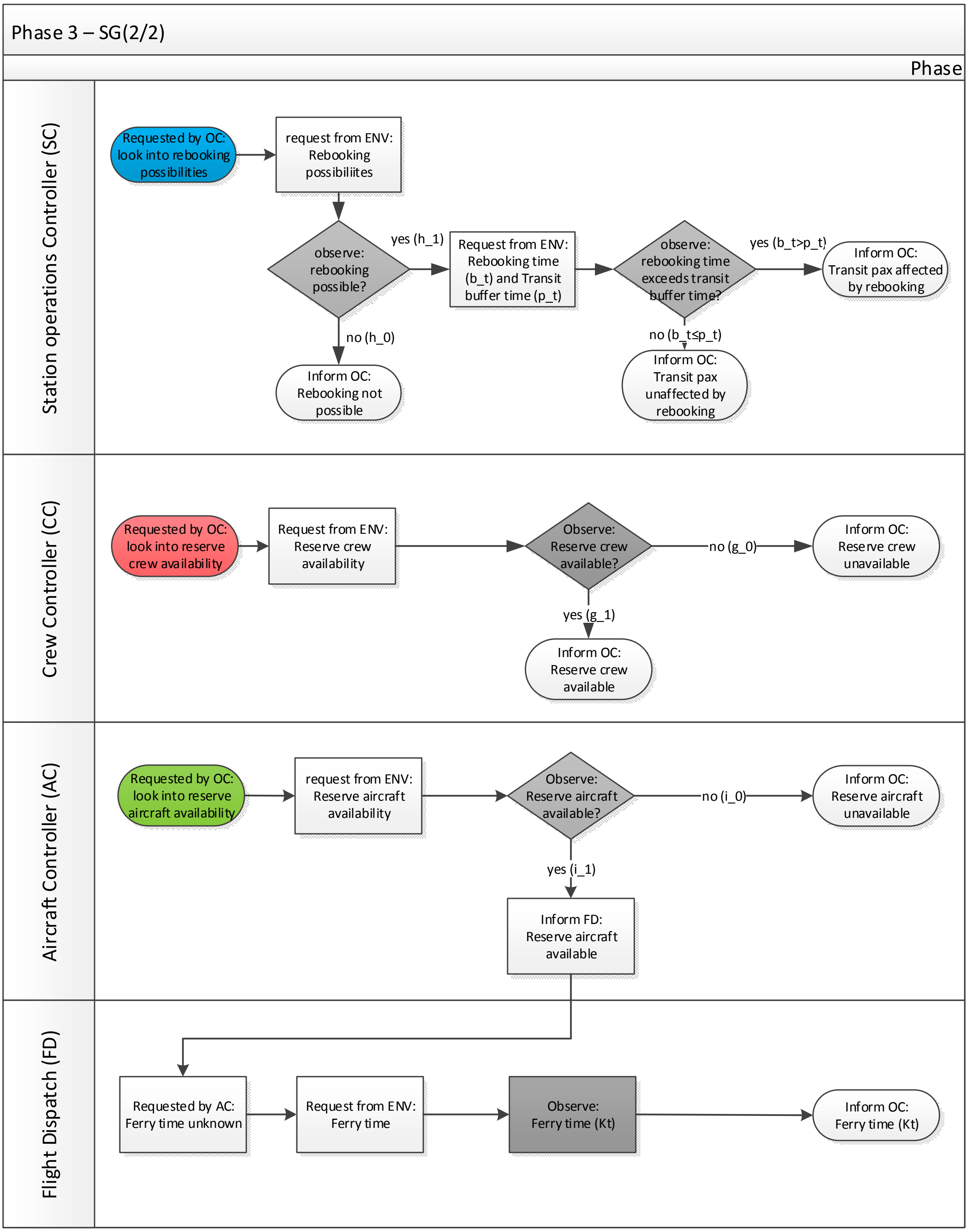


Figure 11 - Phase 3-SG2 cross functional flowchart

The OC-flowcharts provide information requests from the supporting groups (corresponding colors with the SG-flowcharts) and provides information about which recovery strategy will be chosen. For the design of the OC-flowcharts, the recovery strategies *(RS1-RS20)* in Table 15 and the priority of Scenario Objectives of Table 18 are used.

Phase 3A

When the effect of the repair on passengers and crew is being determined, then Operations Controller will be either informed that passengers are affected or unaffected by the repair. Regarding crew, Operations Controller will be either informed that crew is unaffected or affected. However, Crew Controller will also provide information regarding the positioning possibilities for the crew (see Table 24). Depending on the response of the Crew Controller in phase 2A, phase 3A can be subdivided into three OC flowcharts (phase 3A-1, 3A-2 and 3A-3).

SC response	**CC response**		
	Crew unaffected by repair	*Crew affected by repair*	
	Flight Duty time sufficient	*positioning crew possible*	*positioning crew not possible, but reserve crew available*
Transit pax unaffected by repair	3A-1	3A-2	3A-3
Transit pax affected by repair			

Table 24 – SC and CC responses from phase 2A and its subsequent phases 3A-1, 3A-2 and 3A-3

Since phase 3A is the phase in which aircraft TUA will be put under unscheduled maintenance, the recovery strategies that include decision outcome A1-($aircraft\ put\ under$ “$unscheduled\ maintenance$”) and becomes serviceable after repair time (r_t) will be chosen i.e. RS1-RS12. The approach of designing the OC-flowchart with regard to the priorities listed in Table 18 is shown for phase 3A-1. All other OC-flowcharts have been designed using the same approach.

Phase 3A-1

If Operations Controller is informed that flight duty time is sufficient, then decision outcome $C1$ ($crew\ is\ waiting$) is applicable which means that RS1, RS2 RS4 and RS8 can be implemented. From Table 18 it can be seen that the first two objectives are fulfilled with these four recovery strategies. The third priority scenario states that the AOC has to make sure that transit passengers will connect successfully and since retaining reserve aircraft and crew is prioritized above rebooking passengers, the recovery strategies will be prioritized as follows: (1) RS1, (2) RS4, (3) RS8 and (4) RS2.

The process of the Operations Controller will be as follows whether (1) delay due to repair will have a successful connection followed by (2) request rebooking opportunities and whether transit passengers connect successfully and (3) whether a reserve aircraft can be dispatched to pick up the passengers and if it will connect transit passengers successfully. The controllers and tasks involved in phase 3A-1 are listed in Table 25.

Controller	**Task-ID**	**Environmental conditions and parameters**
SC	SC1	p_t
	SC2	h_q, b_t
CC	CC1	g_p
AC	AC5	i_r
FD	FD2	e_n
	FD3	k_t
OC	OC1	-
	OC2	-

Table 25 – Controllers and their tasks in phase 3A-1

The entire OC-flowchart of phase 3A-1 can be viewed in Figure 12

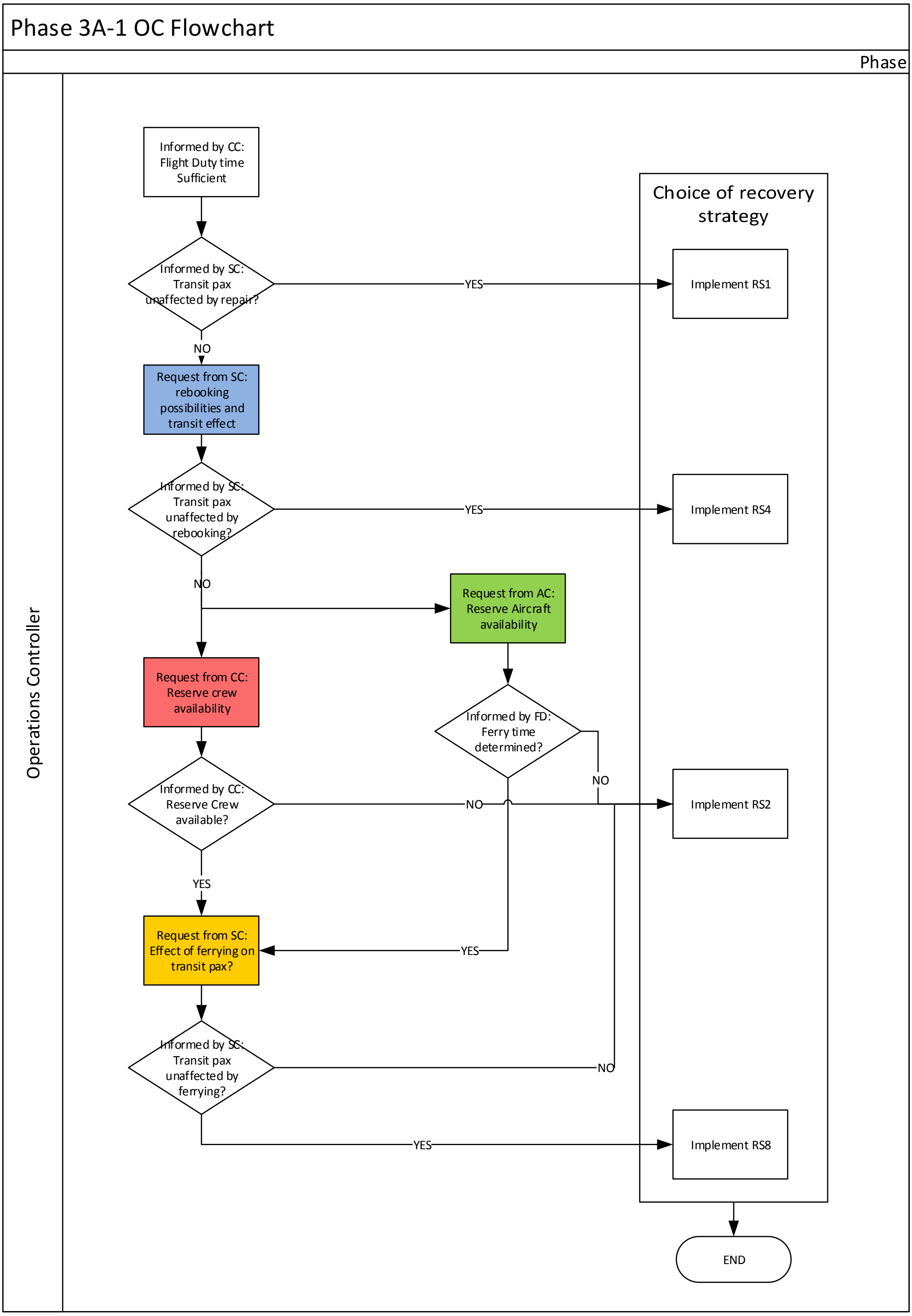


Figure 12 - Phase 3A-1 OC flowchart

Phase 3A-2
When positioning crew is possible then from Table 15 it can be determined that RS12, RS7 and RS11 is possible since they have decision outcome $D1(reserve\ crew\ is\ positioned)$. Operations Controller will request information regarding (1) the effect of positioning crew on transit passengers and (2) whether rebooking will result in successful connection of transit passengers. The tasks of the controllers in phase 3A-2 are listed in Table 26.

Controller	Task-ID	Environmental conditions and parameters
SC	SC1	p_t
	SC2	h_q, b_t
FD	FD2	e_n
OC	OC1	-
	OC2	-

Table 26 – Controllers and their task in phase 3A-2

The OC-flowchart of phase 3A-2 can be viewed in Figure 13.

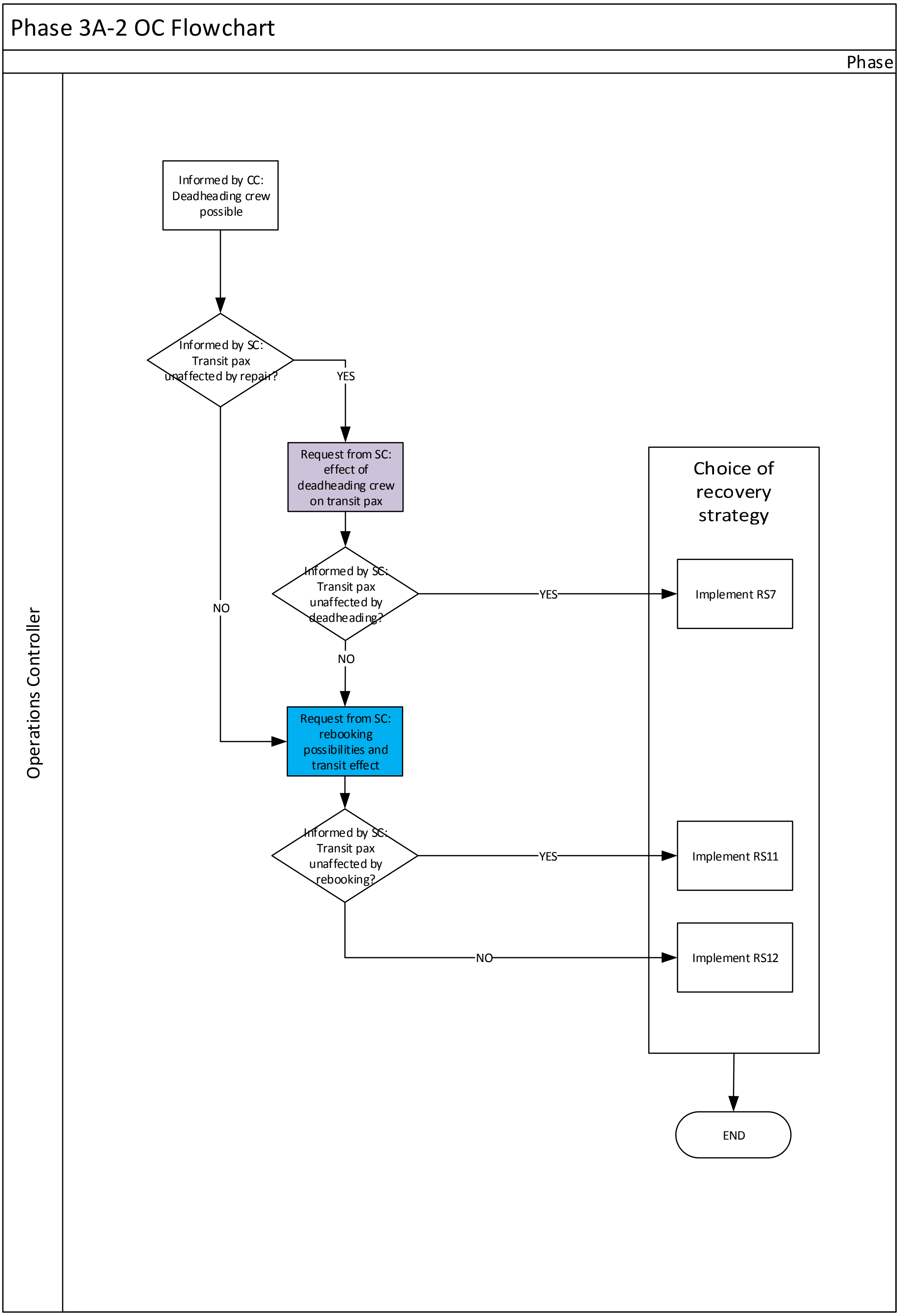


Figure 13 - Phase 3A-2 OC Flowchart

Phase 3A-3

When flight duty time is insufficient and crew positioning is not possible, the decision outcomes for crew A should be $C2$*(crew A is resting/accommodated)* and for reserve crew it is either D2 *(reserve crew is operating reserve aircraft*) or D3 *(reserve crew is not utilized).* With this information it can be determined from Table 15 that RS6, RS10, RS5, RS9 and RS3 are possible recovery strategies. Operations Controller will request information (1) whether rebooking will result in successful connection of transit passengers and (2) dispatch of reserve aircraft will result in successful connection of transit passengers. In order to avoid passengers being stranded, operations controller could decide to avoid passengers being stranded. The applicable controllers and tasks for phase 3A-3 are listed in Table 27.

Controller	Task-ID	Environmental conditions and parameters
SC	SC1	p_t
	SC2	h_q, b_t
CC	CC1	g_p
AC	AC5	i_r
FD	FD2	e_n
	FD3	k_t
OC	OC1	-
	OC2	-

Table 27 – Controllers and their tasks n phase 3A-3

The OC-flowchart of phase 3A-3 can be viewed in Figure 14.

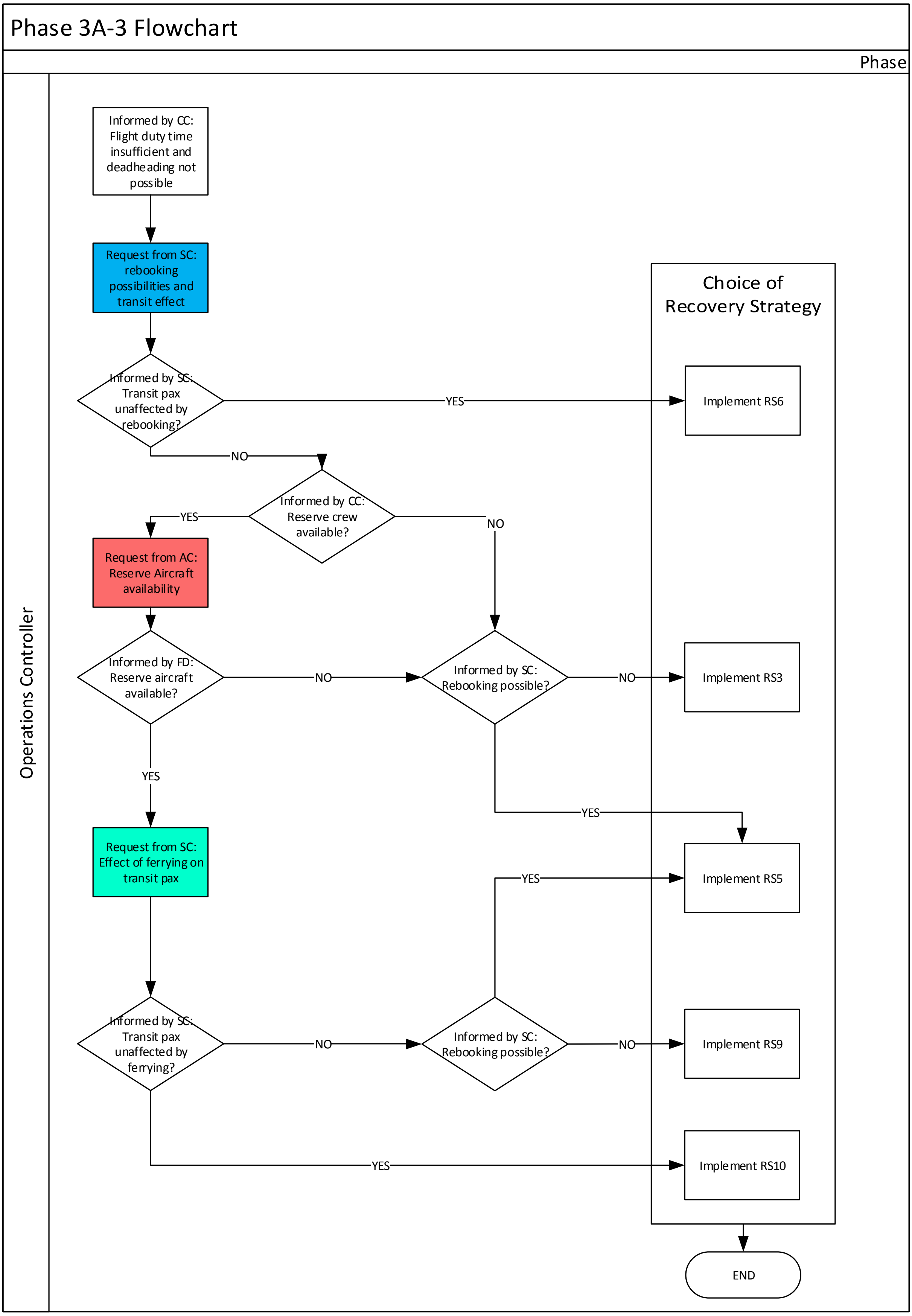


Figure 14 - Phase 3A-3 OC flowchart

Phase 3B

Depending on the response of Station operations Controller and Aircraft Controller in the previous phases the flowcharts can be subdivided into two OC-flowcharts: 3B-1 and 3B-2 (see Table 28).

AC response	**SC Response**	
Hangar unavailable	*Positioning resources possible*	*Positioning resources not possible*
Phase 3B-1		Phase 3B-2

Table 28 - AC and SC responses from resp. phase 1 and phase 2A and its subsequent phases 3B-1 and 3B-2

Phase 3B is the phase in which aircraft TUA will be AOG, this means that decision outcomes applicable for this resource are $A2$ *(aircraft TUA is AOG and awaits resources that will be transported/positioning to DLF)* or *A3 (aircraft TUA is AOG without any short term prospect).* This means that the recovery strategies RS13-RS20 can be implemented.

Phase 3B-1

When aircraft TUA awaits resources to be positioned, then this means that decision outcome A2 is applicable. From Table 15 it can be determined that the following recovery strategies include A2 *(aircraft TUA is AOG and awaits resources that will be transported/positioned to DLF)*: RS14, RS15, RS16, RS17 and RS18.

OC will request information (1) whether rebooking will result in successful connection of transit passengers and (2) dispatch of reserve aircraft will result in successful connection of transit passengers. In order to avoid passengers being stranded, the Operations Controller is also interested in rebooking passengers without successful connection. The controllers' tasks are listed in Table 29.

Controller	Task-ID	Environmental conditions and parameters
SC	SC1	p_t
	SC2	h_q, b_t
CC	CC1	g_p
AC	AC5	i_r
FD	FD2	e_n
	FD3	k_t
OC	OC1	-
	OC2	-

Table 29- Controllers and their tasks in phase 3B-1

The OC-flowchart of phase 3B-1 can be viewed in Figure 15

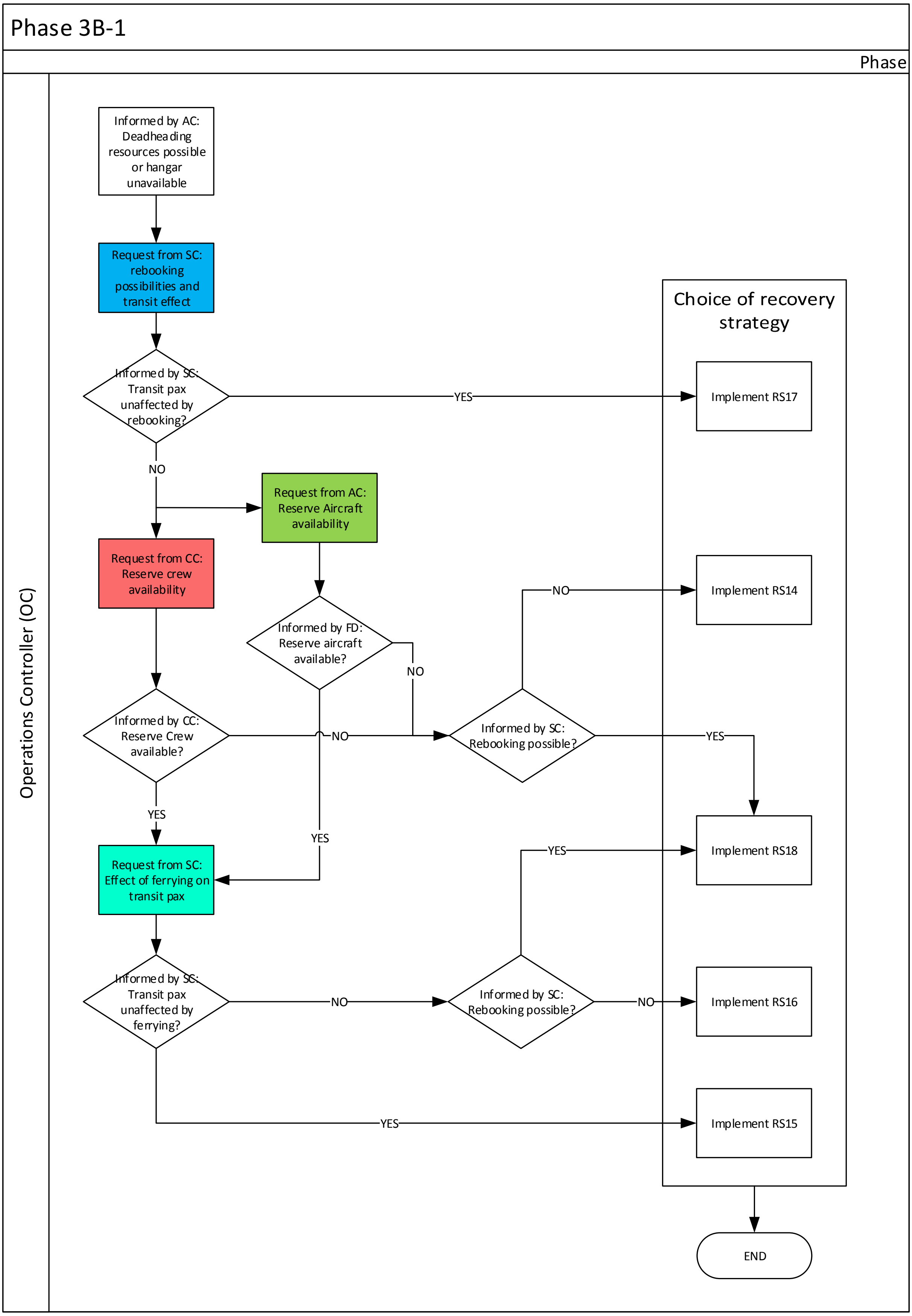


Figure 15 - Phase 3B-1 OC flowchart

Phase 3B-2

When there are no resources positioning possibilities for the aircraft, this leads to either the decision outcome *A3 (aircraft TUA is AOG without any short term prospect)* or resources are positioned by using a reserve aircraft i.e. decision outcome $A2$ *(aircraft TUA is AOG and awaits resources that will be transported/positioned to DLF)* and B1 *(reserve aircraft being dispatched).* The following recovery strategies can be implemented in this phase: RS13, RS15, RS16, RS19 and RS20.

Due to the priority of the Scenario Objectives, the Operations Controller will request information regarding (1) availability of reserve crew and aircraft and the effect of ferrying on transit passenger and (3) rebooking possibilities and its effect transit passenger. The controllers 'tasks are listed in Table 30.

Controller	Task-ID	Environmental conditions and parameters
SC	SC1	p_t
	SC2	h_q, b_t
CC	CC1	g_p
AC	AC5	i_r
FD	FD2	e_n
	FD3	k_t
OC	OC1	-
	OC2	-

Table 30 – Controllers and their tasks in phase 3B-2

The OC-flowchart of phase 3B-2 can be viewed in Figure 16.

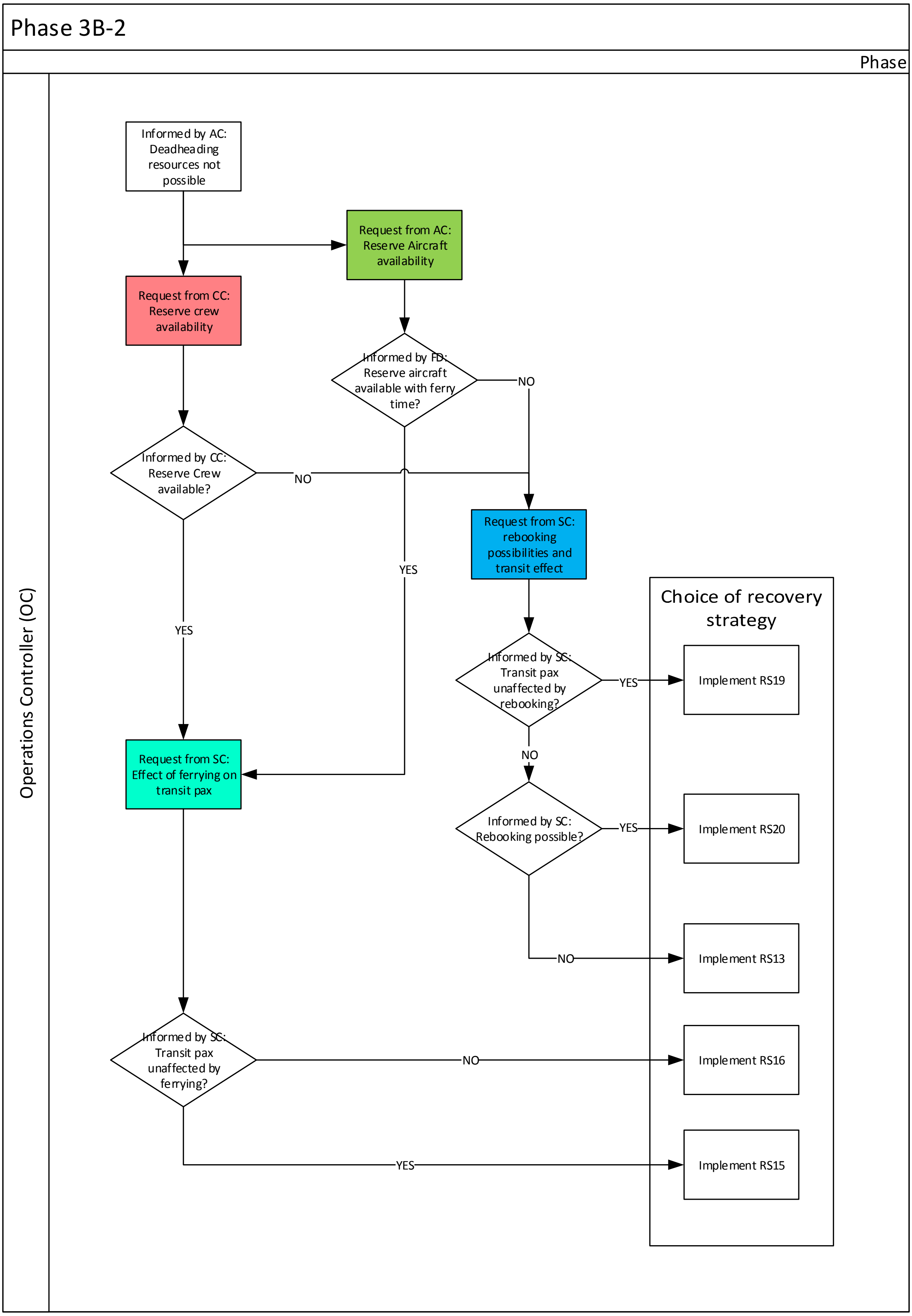


Figure 16 - Phase 3B-2 OC Flowchart

5.4 Cost modelling

One of the general AOC objectives is to minimize costs (Kohl et al., 2007), from Table 11 it could be seen that all scenario objectives contribute to this general objective (GO4). To quantify the recovery strategies, the costs of the chosen recovery strategy will be calculated after simulation. This means that costs are not explicitly taken into account during the decision-making process.
In airline operations different types of costs models can be distinguished: (1) strategic costs and (2) tactical costs. The former relates to costs that are involved during scheduling design while the latter relates to costs during disruption management of airline operational handling. For this study, only tactical costs will be considered on a gate-to-gate level.

To choose cost equations for each recovery strategy it has to be able to capture:

1. Costs of aircraft being unproductive due to mechanical problem and excess cost for crew
2. The costs of the utilization of extra resources like reserve crew and reserve aircraft
3. Real costs for passenger delay for example meals, compensation and rebooking costs
4. Quality costs are sometimes referred to as “soft” costs in the case of delayed passengers or unsuccessful connection of transit passengers

Cost equations and data are difficult to obtain due to fear of competition among airlines. Some research studies have tried to capture these costs by using equations and assumptions. Castro et al. (2014) has formulated an operating cost equation that is used in his study to quantify AOC decision outcomes. (see Equation 1).

$$C = D + \beta \cdot Q$$

$C = Operating\ Costs\ [€]$
$D = Direct\ Costs\ [€]$
$Q = Quality\ Costs\ [€]$
$\beta = weight\ coefficient\ (\beta \geq 0)$

Equation 1 – Operating Costs (Castro et al., 2014)

This equation has two main components: the direct costs and the “quality costs” which is also referred to as “soft passengers costs” or “passengers’ goodwill”. The latter component is very hard to quantify since it involves passenger satisfaction and can be altered by changing the magnitude of the weight coefficient. Direct costs ar much easier to quantify and can be subdivided into costs for aircraft, crew and passengers (Equation 2).

$$D = F + R + P$$

$F = Flight\ Costs\ [€]$
$R = Crew\ Costs\ [€]$
$P = Passenger\ Costs\ [€]$

Equation 2 – Direct costs (Castro et al., 2014)

Flight Cost are costs for fuel, maintenance, ground handling and ATC/Airport fees. For the calculations the cost of ownership will also be taken into account in the case of delays, since the aircraft is not able to generate revenue (Wu, 2010). Crew Cost include extra crew hour costs due to loss of productivity, per diem and hotel costs. Passenger costs involves cost for passengers compensation, rebooking tickets for passengers, hotel costs and meal costs if applicable. The data that will be used for the calculations in chapter 7 is provided in Table 31.

Cost component	Item	Cost (in EUR)	Remark	Literature ascripton
Flight Costs	Aircraft operating costs	66/min.	Includes all aircraft operating costs for disrupted and reserve aircraft excluding fuel	Wu (2010)

			expenses (engines off at gates), B737	
	Fuel/ATC/AIrport	4000/round trip	Fuel for ferrying reserve aircraft back and forth	Castro Castro et al. (2014)
Crew Cost	Extra crew costs	1.2/crew member/min.	Extra crew hourly rate average crew and for utilizing reserve crew	Castro Castro et al. (2014)
	Per diem	80/crew member	Crew expenses in the case of rest	Castro Castro et al. (2014)
	Hotel	103/crew member	Crew accommodation for rest	Castro Castro et al. (2014)
Passenger Cost	Compensation cost	250/passenger	>3 hours delay	EU regulation
	Rebooking costs	120/passenger	the costs of rebooking passengers for initial flight and subsequent (missed connecting) flight	KLM.com
	Hotel Cost	114/passenger	Cancellation, no distinction between economy and business	Castro Castro et al. (2014)
	Meal Costs	25/passenger	>2 hours delay	Castro Castro et al. (2014)

Table 31 – direct cost items used for calculation

For this scenario the following will be assumed for direct cost calculations

- Costs for reserve aircraft or crew are not taking into account since they do not account for tactical costs
- The normal flight time between DLF and AMS is 105 minutes, which will be used for ferrying costs
- There are a total of six crew members aboard the aircraft (i.e. crew A)
- When crew A has to rest, it will take 10 hours before they can continue with their duty (EASA, 2014)
- It is assumed that when aircraft is AOG without short term prospect, it will take 12 hours before it is brought back to service
- It is assumed that when aircraft is AOG and awaits resources to be send, that the it will take as long as positioning time plus maximum repair time i.e. $r_{t_{max}} + d_t$
- The importance of connections will be set to $(\gamma = 1)$
- There are a total of 150 passengers on board

Quality costs is very hard to define, but in the study of Castro et al. (2014) an attempt is made to capture the soft cost of passenger into an equation. This equation is shown in Equation 3.

$$Q = \left[(\gamma \cdot l_p + t_p) \cdot 1.2 \cdot d_f + (\gamma \cdot l_b + t_b) \cdot \left(0.16 \cdot {d_f}^2 + 1.19 \cdot d_f\right) \right]$$

$l_p = transit\ passengers\ in\ pleasure\ profile (economy)$
$t_p = total\ passengers\ in\ pleasure\ profile (economy)$
$l_b = transit\ passengers\ in\ business\ profile$
$t_b = total\ passengers\ in\ business\ profile$
$d_f = passenger\ delay\ in\ minutes$
$\gamma = parameter\ to\ determine\ importance\ of\ connection\ (\gamma \geq 1)$

Equation 3 – Quality cost equation (Castro et al., 2014)

As can be seen from this equation is that the quality costs for the business profile passengers rises quadratic with respect to the delay of the passengers while the passengers of the economy class rises linear with the passenger delay.

For the quality cost calculations the number of passengers per profile of Table 32 is used.

Profile	**Transit**	**non-transit**	**Total**

Economy	50	85	135
Business	5	10	15
Total	55	95	150

Table 32 - number of passenger per profile

When passengers are confronted with a cancellation or long delay resulting in accommodation, the delay time used for quality cost calculations will be five hours. This value is chosen because EU-law defines that starting from a five hour delay, passengers can choose to receive a refund or care i.e. accommodation (EU, 2004).

THIS PAGE IS INTENTIONALLY LEFT BLANK

6. Formal description

The model description has to be formalised so that it can be interpreted by a software tool. This phase of the methodology of Nikolic and Ghorbani (2011) is also referred to as detailed design. Section 6.1 provides the syntax for the language that is used for the formal description and the types of dynamic properties that are available. These dynamic properties will be described using input and output ontologies which are designed in section 6.2. In section 6.3 an example is provided of the formalization process by using a part one task of the aircraft controller. In section 6.4 the implementation into the LEADSTO-tool is discussed.

Note: even though the term agents is used to explain the theory of the modelling language, there are no agents modeled in this study, but merely controllers

6.1 Modelling language

6.1.1 Choice for modelling language and software

To describe the properties of the AOC and its environment, a language is needed. The formal description of the model will be done using Trace Temporal Language (TTL). The reason for choosing this language is that:

1. It is used in various multi-agent system studies to represent organization structures and dynamics
2. It can model qualitative and quantitative aspects
3. It is logic based and expressive
4. It can be implemented in software using a sub language “LEADSTO” as the executable language

TTL language is based on the assumption that dynamics can be described as an evolution of states over time by using order-sorted predicate logic (Bosse and Mogles, 2013). Major difference between normal order-sorted predicate logic and TTL is that the latter is used for properties that change over time i.e. dynamic properties.

6.1.2 Theory of Trace Temporal Language

To describe the behaviour of these system components in TTL, states are related to state properties. These state properties are described using ontologies that are specified by sorts, constants, variables, functions and predicates. The main literature used for the descriptions of the theory of TTL are Jonker et al. (2007), Hoogendoorn et al. (2008) and Sharpanskykh and Treur (2010).

A major aspect of the syntax of TTL is the special sorts. The first special sort that will be discussed is *STATE* which is a set of all states and can be described by a functional symbol shown in Expression 2.

$$\text{state: } TRACE\ x\ TIME\ x\ COMPONENT\_STATE\_ASPECT\ \rightarrow STATE$$

Expression 2 – state functional symbol

Expression 2 includes the following special sorts:

1. *TRACE* - set of all trace names that are denoted by γ. Traces can be thought of as a temporal description of chains of events (Sharpanskykh and Treur, 2010).
2. *TIME* - set of all linearly order time points and are denoted by *t.*
3. *COMPONENT_STATE_ASPECT* - set of all component state aspects and is described using the functional symbol in Expression 3

$$\text{comp\_aspect: } ASPECT\_COMPONENT\, x\ COMPONENT \rightarrow COMPONENT\_STATE\_ASPECT$$

Expression 3 – component function symbol

Within Expression 3 there are two special sorts that define comp_aspect:

1. *ASPECT_COMPONENT* - set of the all component aspects of a system i.e. input and output
2. *COMPONENT* - set of all component names of a system,

In order to understand the concept of “state” an example will be provided. The applicable component is agent A, the “aspect_component” is an input, the trace where it takes place is $\gamma_1$and the time point is t_1 i.e. ”the input of agent A at time point t_1 for trace γ_1” can be described using Expression 4.

$$state(\gamma_1, t_1, input(A))$$

Expression 4 - the input of agent A at time point t_1 for trace γ_1

From Expression 4 it cannot be deduced of what exactly the input of agent A is in the given time point and trace. For this reason "state properties" are required to describe the properties of particular states. The set of all state properties is denoted as *STATPROP* and its terms are denoted by "*p*". State properties are related to states by using a satisfaction relation as shown in Expression 5.

$$state(\gamma_1, t_1, input(A)) \;\; \Vdash p$$

Expression 5 - the input of agent A at time point t_1 for trace γ_1 equals " p"

For example, if we consider agent A at time point t_1 for trace γ_1 with an input which is an observation of an "event_x". Then the state property can be described by the atom "*observation(event_x)*" which is term of *STATPROP*. The expression that results in the relation between state and state property is shown in Expression 6 and in trace descriptions in Expression 7.

$$state(y_1, t_1, input(A)) \Vdash observation(event\_x)$$

Expression 6 – the input of agent A at time point t_1 for trace γ_1 equals an observation of "event_x"

$$input(A)|observation(event\_x)$$

Expression 7- trace description of

Additional to the aforementioned special sorts there is also *VALUE,* which is simply an ordered set of numbers. In Sharpanskykh and Treur (2010) an extensive explanation of the syntax and semantics of TTL is provided.

Dynamic properties are relations in time between states of agents, states of the environment or states between agents and the environment. By using ontologies and logical connectives dynamic properties can be described. To understand how a dynamic property is formalised, an example will be provided that consists of three steps (1) informal description then a (2) semi-formal description and finally a (3) formal description.

(1) When controller A observes that "event x" takes place, he will take action upon this particular event
(2) In any trace γ, at any point in time t_1 if controller A observes "event x", then at a later point in time t_2, controller A wil take action upon "event x"
(3)$\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE\ state(\gamma, t_1, input(A)) \Vdash observation(event\_x)\ \wedge \exists t_2 > t_1 \Rightarrow$
$state(\gamma, t_2, output(A)) \Vdash performing\_action(event\_x)$

For the agents to fulfill these roles, they have to be described including the interaction properties among the agents and the environment. There are five types of dynamic properties:

- *Role Property (RP)* **-** the relation between input and output state of a role that is fulfilled by the role
- *Environment Property (EP)* **–** the relation between input and output state of the conceptualized environment
- *Transfer Property (TP)* **-** the relation between output state and input states of agents
- *Environment Interaction Properties (EIP)* **–** the relation between either output to input or input to output states between the conceptualized environment and agents.
- *Interlevel Link Property (ILP)* **-** the relation between a input or output of a composite role and the input or output of one of its subrole

The properties are described using interaction ontologies consisting of only input and output ontologies. E.g. for system component A only InOnt(A) and OutOnt(A) will be used.

6.2 Ontology design

For the model in this study there are two system components: controllers and the environment for which the ontologies will be described. The ontologies consist of sorts, predicates, functions and variables.

Predicates or sometimes referred to as signatures have to be designed in such a way that they can define state properties that will be used to specify the dynamic properties. From the OC-flowcharts it can be seen that controllers inform each other, request information from each other or observe information from the environment. These interactions show similarities with the incident management study of Hoogendoorn et al. (2008) in which information exchange among the roles was formalised using TTL. For this reason the predicates used for that study will be serve as a basis and will be adjusted to meet the requirements for this study. Before the predicates can be designed, sorts have to be defined to determine the terms of the predicates.

For ontological correspondence it is important to declare (1) who is interacting with who (2) what type of message is sent (3) what the content of the message is (4) in which phase this interaction takes place and (5) what kind of recovery strategy is chosen (see Table 33).

Sort	Description
CTRL	Controllers which are involved in this scenario
MSG_TYPE	Types of message that is applicable (i.e. interaction)
MSG	Messages of one controller to the other
PHASE	Phases in which the state property takes place
RS	The recovery strategies

Table 33 – Sorts of the model

The five controllers in the scenario including their abbreviation are described in Table 34.

SORT	Terms	Description
CTRL	oc	Operations Control, the main decision-maker in the disruption management process
	ac	Aircraft Control, responsible for aircraft related disruptions and support
	fd	Flight Dispatch, responsible for pre-flight planning, ATC and weather related issues
	cc	Crew Control, responsible for crew related disruptions and support
	sc	Station operations Control, responsible for passenger related disruptions and support

Table 34 – terms of the sort CTRL

Regarding the interactions three types of interactions were identified among the controllers and environment (Table 35).

SORT	Terms	Description
MSG_TYPE	inform	informing an controller or being informed
	request	request information from the environment or other controllers
	observe	observing the environment

Table 35 - terms of the sort MESSAGE_TYPE

The terms of the sort MSG are presented in Table 36. No description is provided since the aim of the message is to be self-explanatory.

SORT	Terms	
MSG	aircraft_tua_has_mechanical_problem	positioning_not_possible
	technical_diagnosis_adequateness	positioning_crew_connects_tpax_successful
	technical_diagnosis_adequate	positioning_crew_connects_tpax_unsuccessful
	technical_diagnosis_inadequate	effect_of_kt_on_tpax
	spare_parts_availability	effect_of_dt_on_tpax
	spare_parts_available	effect_of_rt_on_tpax
	spare_parts_unavailable	tpax_unaffected_by_rt
	wx_pattern_favourability	tpax_affected_by_rt
	wx_pattern_favourable	rebooking_possibilities

	wx_pattern_unfavourable	rebooking_possible
	hangar_availability	rebooking_not_possible
	hangar_available	rebooking_connects_tpax_successful
	hangar_unavailable	rebooking_connects_tpax_unsuccessful
	reserve_aircraft_availability	repair_time
	reserve_aircraft_available	rebooking_time
	reserve_aircraft_unavailable	transit_buffer_time
	dispatch_reserve_aircraft_connects_ tpax_successful	reserve_crew_availability
	dispatch_reserve_aircraft_connects_tpax_ unsuccessful	reserve_crew_available
	organizing_cxn_measures_possibilities	reserve_crew_unavailable
	organizing_cxn_measures_possible	crew_duty_slack_time
	organizing_cxn_measures_not_possible	ferry_time
	effect_of_rt_on_crew	positioning_time
	crew_unaffected_by_rt	positioning_resources_possibilities
	positioning_possibilities	positioning_resources_possible
	positioning_possible	positioning_resources_not_possible

Table 36 – terms of the sort MSG

From the model description it was clear that there are twenty recovery strategies that could be formulated, which means that there are twenty terms for the sort 'RS' i.e. RS1-RS20. The description of these recovery strategies can be referred to in Table 15 of section 5.1.2.

To use the multi-trace application, the conditions will also be SORTS and are described in Table 37.

Sort	term	description
DIAG	a_0	Inadequate technical diagnosis
	a_1	Adequate technical diagnosis
PART	b_0	Spare parts unavailable
	b_1	Spare parts available
WX	c_0	weather pattern unfavourable
	c_1	weather pattern favourable
HANG	d_0	hangar space unavailable
	d_1	hangar space available
CONM	e_0	organizing connection not possible
	e_1	organizing connection possible
DEAD	f_0	positioning opportunities unavailable
	f_1	positioning opportunities available
RCREW	g_0	reserve crew unavailable
	g_1	reserve crew available
RBOOK	h_0	rebooking opportunities not present
	h_1	rebooking opportunities present
RAC	i_0	reserve aircraft unavailable
	i_1	reserve aircraft available

Table 37 – The sorts and terms to be used for the environment

The six time parameters will be quantitative variables. This means that these are instantiated with terms of the sort *VALUE* (i.e integers). For the completeness of the formal description these are listed in Table 38 (note: in the LEADSTO code there are no subscripts e.g. r_t is described as rt).

Variable	Description
r_t	The time that is required to repair the aircraft and prepare to fly
d_t	The time the reserve crew is positioned and ready to operate the disrupted flight
k_t	The time that the reserve aircraft is ready to take over flight A
p_t	The available buffer time for the transit passengers
b_t	The time before the rebooking flight is to depart
c_t	The available slack of crew duty time
d_f	The delay of the passengers

Table 38 – the time variables in the model

Two predicates are used during formalization: (1) communication predicate and (2) recovery predicate. The communication predicate expresses the communication between two controllers (source and destination), the type of message, the actual message, the communicated delay and the phase in which the interaction takes place. The predicate for the recovery strategy that will be formulated consist of the chosen recovery strategy and the associated delays for the passengers. These two predicates can be referred to in Table 39.

Predicate	Description
com(r: CTRL, dst: CTRL, t: MESSAGE_TYPE, v: MSG ,delay(rt:integer,dt:integer,bt:integer,kt:integer),p(‘ x: PHASE’)	the message “v” and current time parameters rt, dt, bt, kt are communicated by “r” to “dst” by using messaging type “t”, which takes place in phase ‘x’
Recovery(r: RS,df,rt,dt,bt,kt)	the chosen recovery strategy is “RS” with a passenger delay df of either “rt”, “dt”, “bt” or “kt”

Table 39 - Predicates for the model

6.3 Describing properties

The flowcharts described in section 5.3, provide information regarding the decision-making processes of the controllers. In order to model these processes, they will be described using the input and output ontologies in the previous section.

An example will be provided of the formalization of a part of the Aircraft Controller decision-making process which is illustrated in Figure 17. In phase 1 the Aircraft Controller has to determine the adequateness of the technical diagnosis (AC1). This task will be described in terms of role properties (*RPs*), environment properties (*EPs*) and environment interaction properties (EIPs) by using the input and output ontologies described earlier in this chapter. The entire model will be designed using one aggregation level, this means that no Interlevel Link Properties (ILP) is described.

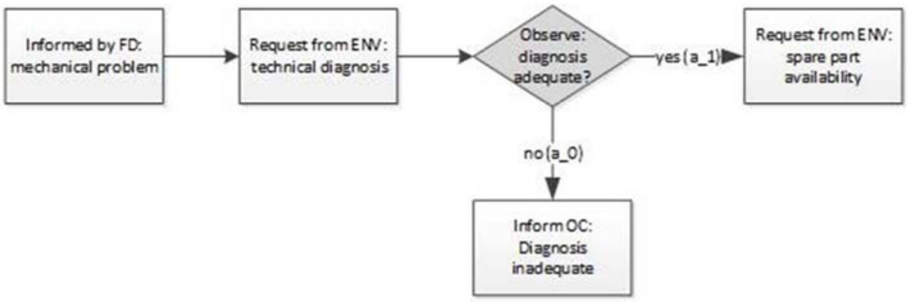


Figure 17 - part of the aircraft controller process in phase 1

In chapter 5 it was defined that there are two types of conditions possible for adequateness of technical diagnosis: technical diagnosis being adequate denoted by a_1 or inadequate denoted by a_0. The properties of both possibilities will be described below.

Semi-formal description of role property 1: In any trace γ, at any point in time $t_1$when aircraft control is informed about the mechanical problem, then at a later point in time t_2, aircraft control will request information about the technical diagnosis (*A1*).

Formal description of role property 1

$$\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, input(ac)|com(fd, ac, inform, aircraft\_tua\_has\_mechanical\_problem,\\ delay(x1, x2, x3, x4), p('1')) \ \wedge \exists t_2 > t_1\\ \Rightarrow state(\gamma, t_2, output(ac)|com(ac, env, request, technical\_diagnosis\_adequateness,\\ delay(x1, x2, x3, x4), p('1'))$$

Semi-formal description of environment interaction property 1: in any trace γ at any point in time t_1 when the aircraft controller requests the adequateness of the technical diagnosis, then at a later point in time t_2 the conceptualized environment will receive this request.

Formal description of environment interaction property 1

$$\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, output(ac)|com(ac, env, request, technical\_diagnosis\_adequateness,\\ delay(x1, x2, x3, x4), p('1')) \ \wedge \exists t_2 > t_1 \Rightarrow state(\gamma, t_2, input(env)\\ \Vdash com(ac, env, request, technical\_diagnosis\_adequateness, delay(x1, x2, x3, x4), p('1'))$$

At this point the aircraft controller requests the adequateness of the environment. Due the fact that this condition is Boolean, there are two possibilities described below.

Semi-formal description of environment property 1: in any trace γ at time point t_1 when the conceptualized environment receives a request for the technical diagnosis adequateness and the condition in this case is that it is adequate (a_1), then at a later point in time t_2 the conceptualized environment will provide an adequate technical diagnosis

Formal description of environment property 1

$$\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, input(env)\\ \Vdash com(ac, env, request, technical\_diagnosis\_adequateness, delay(x1, x2, x3, x4), p('1')) \wedge \boldsymbol{a\_1} \wedge \exists t_2\\ > t_1 \Rightarrow state(\gamma, t_2, output(env))\\ \Vdash com(env, ac, inform, technical\_diagnosis\_adequate, delay(x1, x2, x3, x4), p('1'))$$

Semi-formal description of environment interaction property 2: in any trace γ at any point in time t_1 when the environment provides an adequate technical diagnosis, then at a later point in time t_2 the aircraft controller will observe an adequate technical diagnosis.

Formal description of environment interaction property 2

$$\begin{aligned}&\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, ooutput(env))\\ &\quad \Vdash com(env, ac, inform, technical\_diagnosis\_adequate, delay(x1, x2, x3, x4), p('1'))\\ &\quad \wedge \exists t_2 > t_1 \Rightarrow state(\gamma, t_{2,}input(ac))\\ &\quad \Vdash com(env, ac, observe, technical\_diagnosis\_adequate, delay(x1, x2, x3, x4), p('1'))\end{aligned}$$

Semi-formal description of role property 2: in any trace γ at any point in time t_1 when the aircraft controller observes an adequate technical diagnosis, then at a later point in time t_2 the aircraft controller will request spare parts availability from the environment.

Formal description of role property 2

$$\begin{aligned}&\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, input(ac))\\ &\Vdash com(env, ac, observe, technical\_diagnosis\_adequate, delay(x1, x2, x3, x4), p('1')) \wedge \exists t_2 > t_1\\ &\Rightarrow state(\gamma, t_{2,}output(ac)|com(ac, env, request, spare\_parts\_availability, delay(x1, x2, x3, x4), p('1'))\end{aligned}$$

Semi-formal description of environment property 2: in any trace γ at time point t_1 when the environment receives a request for the technical diagnosis adequateness and the condition in this case is that it is <u>inadequate (a_0)</u>, then at a later point in timet_2 the conceptualized environment will provide an inadequate technical diagnosis.

Formal description of environment property 2:

$$\begin{aligned}&\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, input(env)\\ &\Vdash com(ac, env, request, technical\_diagnosis\_adequateness, delay(x1, x2, x3, x4), p('1')) \wedge \boldsymbol{a\_0} \wedge \exists t_2\\ &> t_1 \Rightarrow state(\gamma, t_{2,}output(env))\\ &\Vdash com(env, ac, inform, technical\_diagnosis\_inadequate, delay(x1, x2, x3, x4), p('1\end{aligned}$$

Semi-formal description of environment interaction property 3: in any trace γ at any point in time t_1 when the environment provides an inadequate technical diagnosis, then at a later point in time t_2 the aircraft controller will observe an inadequate technical diagnosis.

Formal description of environment interaction property 3

$$\begin{aligned}&\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, ooutput(env))\\ &\quad \Vdash com(env, ac, inform, technical\_diagnosis\_inadequate, delay(x1, x2, x3, x4), p('1'))\\ &\quad \wedge \exists t_2 > t_1 \Rightarrow state(\gamma, t_{2,}input(ac))\\ &\quad \Vdash com(env, ac, observe, technical\_diagnosis\_inadequate, delay(x1, x2, x3, x4), p('1'))\end{aligned}$$

Semi-formal: formal description of role property 3: in any trace γ at any point in time t_1 when the aircraft controller observes an inadequate technical diagnosis, then at a later point in time t_2 the aircraft controller will inform operations controller about the inadequate technical diagnosis.

Formal description of role property 3:

$$\begin{aligned}&\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1 input(ac))\\ &\quad \Vdash com(env, ac, observe, technical\_diagnosis\_inadequate, delay(x1, x2, x3, x4), p('1'))\\ &\quad \wedge \exists t_2 > t_1 \Rightarrow state(\gamma, t_{2,}output(ac))\\ &\quad \Vdash com(ac, oc, inform, technical\_diagnosis\_inadequate, delay(x1, x2, x3))\end{aligned}$$

6.4 Software Implementation

LEADSTO is a language and software environment that models and simulates dynamic processes. These dynamic processes can be modelled by specifying direct dependencies between state properties. Expression 8 is the ground expression of LEADSTO and consists of an antecedent (α), a consequent (β) and time variables (e, f, g, h).

$$\boxed{\alpha \twoheadrightarrow_{(e,f,g,h)} \beta}$$

Expression 8 – ground expression LEADSTO

This ground expression formulates that: if state property α holds for a time interval with duration g, then after a delay between e and f, state property β (consequent) will hold for a time interval length h. These expressions are also referred to as LEADSTO-rules. The dynamic properties have to be written in LEADSTO-rules so that they can be interpreted by the software. An example is provided using the role property 1 of the previous section:

Formal description of role property 1

$$\forall t_1{:}\,TIME, \forall \gamma{:}\,TRACE, state(\gamma, t_1, input(ac)|com(fd, ac, inform, aircraft\_tua\_has\_mechanical\_problem, delay(x1, x2, x3, x4), p('1')) \wedge \exists t_2 > t_1 \Rightarrow state(\gamma, t_2, output(ac)|com(ac, env, request, technical\_diagnosis\_adequateness, delay(x1, x2, x3, x4), p('1'))$$

The antecedent (α) and consequent (β) of this TTL rule can be written in trace descriptions in the following atoms (Expression 9).

$$\alpha{:}\ input(ac)|com(fd, ac, inform, aircraft\_tua\_has\_mechanical\_problem, delay(x1, x2, x3, x4), p('1'))$$
$$\beta{:}\ output(ac)|com(ac, env, request, technical\_diagnosis\_adequateness, delay(x1, x2, x3, x4), p('1'))$$

Expression 9 – formal trace description of role property 1

The LEADSTO architecture is shown in Figure 18. All the dynamic properties will be written in the LEADSTO specification file. This specification file will be loaded into the LEADSTO Simulation Tool, which will generate a trace-file.

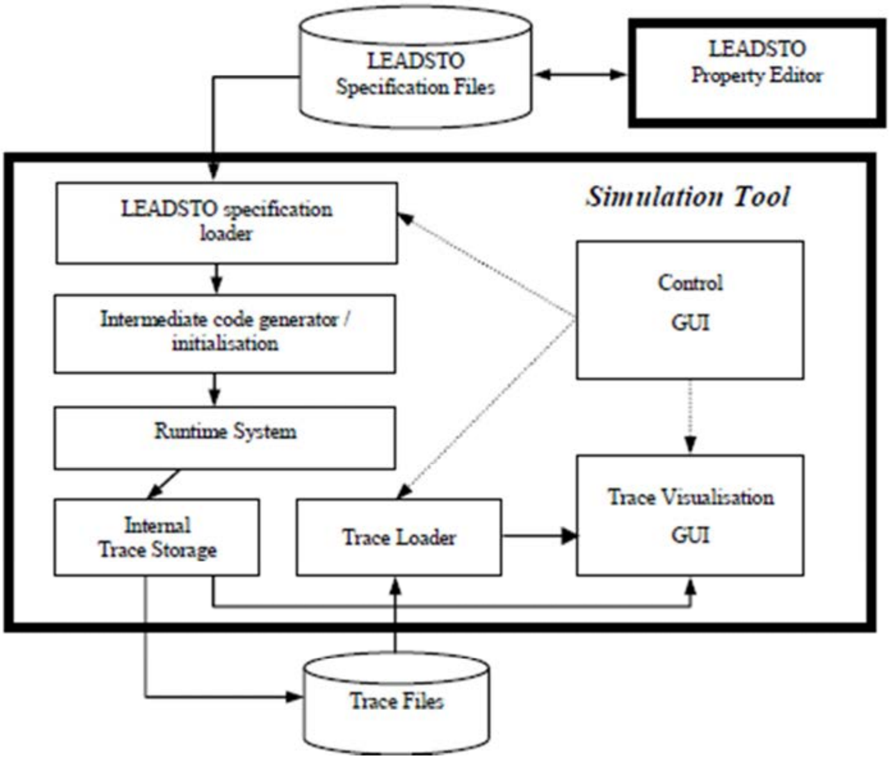


Figure 18 - LEADSTO architecture (Bosse et al., 2007)

Specification files can be generated using the LEADSTO Property Editor or can be generated by writing it into text-editor that can be saved using an "lt" extension in the filename. Example of a rule in the text editor is shown using role property 1 is provided in Expression 10.

$$leadsto([], input(ac)|com(fd, ac, inform, aircraft\_tua\_has\_mechanical\_problem, delay(x1, x2, x3, x4), p('1')),$$

$$output(ac)|com(ac, env, request, technical\_diagnosis\_adequateness,\\ delay(x1, x2, x3, x4), p('1'))\\ ,0,0,1,1)$$

Expression 10 - Leadsto rule in Specification File

In a LEADSTO specification file the following elements are present:

- Sort definitions- sort name and its terms
- Interval- the starting conditions/parameters of a certain simulation
- Multi-trace- generator of a unique combination of conditions and parameters for each simulation
- LEADSTO-rules – the role properties, environmental properties, transfer properties and environmental interaction properties

The complete specification file of the LEADSTO code for this model can be referred to in Appendix C and categorized into RPs, EPs, EIPs and TPs. This code is verified by using the flowcharts that were depicted in chapter 5. Examples of trace-files can be referred to in Appendix E and Appendix F.

THIS PAGE IS INTENTIONALLY LEFT BLANK

7. Results & Discussion

This chapter discusses the results that are obtained by the LEADSTO simulations and cost calculations. In section 7.1, the approach for the model evaluation is described. This includes which conditions sequences and parameter sets will be evaluated and what model parameter will be used as a variable. To have a structured discussion, the model evaluation will be done using four cases. In case 1 until 3 conditions are chosen that are favourable for aircraft repair, but each case will have different robust scheduling parameters and are discussed separately in section 7.2, 7.3 and 7.4. Case 4 is discussed in section 7.5 in which conditions are chosen that result in the aircraft to be On Ground. By using two different transit buffer times it will be discussed what its effect are on recovery strategy.

7.1 Simulation approach

7.1.1 Conditions and inequalities

Several simulations with the LEADSTO-simulation tool will be run to analyse the effect of conditions and parameters on choice of the recovery strategy regarding the scenario. From Table 19 in chapter 5 it could be deduced that there are nine types of environmental conditions. These will be presented as condition sets (Table 40).

Condition sets	**Condition values**	**description**
$a_j = \{a_0, a_1\}$	a_0	Inadequate technical diagnosis
	a_1	Adequate technical diagnosis
$b_k = \{b_0, b_1\}$	b_0	Spare parts unavailable
	b_1	Spare parts available
$c_l = \{c_0, c_1\}$	c_0	weather pattern unfavourable
	c_1	weather pattern favourable
$d_m = \{d_0, d_1\}$	d_0	hangar space unavailable
	d_1	hangar space available
$e_n = \{e_0, e_1\}$	e_0	organizing connection not possible
	e_1	organizing connection possible
$f_o = \{f_0, f_1\}$	f_0	positioning opportunities unavailable
	f_1	positioning opportunities available
$g_p = \{g_0, g_1\}$	g_0	reserve crew unavailable
	g_1	reserve crew available
$h_q = \{h_0, h_1\}$	h_0	rebooking opportunities unavailable
	h_1	rebooking opportunities available
$i_r = \{i_0, i_1\}$	i_0	reserve aircraft unavailable
	i_1	reserve aircraft available

Table 40 – nine environmental conditions sets to be studied

To express conditions that are applicable for a certain simulation, they are put in a sequence. These sequences are denoted by either "p" or "q" and expressed as follows: $p = (a_j, b_k, c_l, d_m, e_n, f_o, g_p, h_q, i_r)$ with the subscripts of the conditions being Boolean valued. Additional to these conditions it could be deduced from the model descriptions in section 5.3 that there are five inequalities of parameters that could influence the decision outcome of the AOC, these are presented in Table 41.

Inequalities		**parameters involved in the equality**
$r_t \leq p_t$	$r_t > p_t$	repair time (r_t) and transit buffer time (p_t)
$r_t \leq c_t$	$r_t > c_t$	repair time (r_t) and crew duty slack time (c_t)
$d_t \leq p_t$	$d_t > p_t$	positioning time (d_t) and transit buffer time (p_t)
$b_t \leq p_t$	$b_t > p_t$	rebooking time (b_t) and transit buffer time (p_t)
$k_t \leq p_t$	$k_t > p_t$	ferry time (k_t) and transit buffer time (p_t)

Table 41 - five types of inequalities

Since there are nine boolean valued conditions and five inequalities, the total number of unique condition sets and inequalities that can be tested equals: $2^9 \cdot 2^5 = 16{,}384$. Due to the large numbers of simulations that could be generated, a selection of parameters and condition sequences will be used in a case by case approach.

7.1.2 Case by case approach

In order to structure the simulation results, a case by case approach will be used to test a series of condition sequences in relation with certain parameters. As can be seen from the model description in Figure 6 that there are two major phases that affect the decision-making process: aircraft under unscheduled maintenance and aircraft AOG.

The requirements for an aircraft to 'qualify' for unscheduled maintenance is that a repair time is being formulated. From Figure 7 it can be observed that a repair time is observed when there is an adequate technical diagnosis(a_1), spare parts available (b_1) and there is either a favourable weather pattern (c_1). In addition to these conditions, recovery opportunities are presented in the form of conditions. These condition sequences are denoted by `$p$`:

$$p_1 = (a_1, b_1, c_1, d_0, e_0, f_0, g_0, h_0, i_0)$$

$$p_2 = (a_1, b_1, c_1, d_0, e_0, f_0, \boldsymbol{g_1}, h_0, \boldsymbol{i_1})$$

$$p_3 = (a_1, b_1, c_1, d_0, e_0, f_0, g_0, \boldsymbol{h_1}, i_0)$$

$$p_4 = (a_1, b_1, c_1, d_0, e_0, f_0, \boldsymbol{g_1}, \boldsymbol{h_1}, \boldsymbol{i_1})$$

$$p_5 = (a_1, b_1, c_1, d_1, e_0, \boldsymbol{f_1}, \boldsymbol{g_1}, \boldsymbol{h_1}, \boldsymbol{i_1})$$

The **bold** conditions express the difference of each condition sequence with regard to p_1.

The first condition sequence represents a situation in which there are no any other recovery opportunities than the aircraft being repaired under unscheduled maintenance (p_1), the second condition sequence is one in which reserve resources are available (p_2), the third condition sequence involves having rebooking opportunities without having reserve resources available (p_3), the fourth condition sequences involves having rebooking as well as reserve resources (p_4), the fifth is a condition sequence in which it is possible to position crew including all aforementioned recovery opportunities (p_5).

These condition sequences will be examined in three separate cases in which the repair time is exceeding:

- Case 1. Transit buffer time, but not crew duty time
- Case 2. Crew duty time, but not transit buffer time
- Case 3. Crew duty time as well as transit buffer time

These three cases will be examined by using the repair time as variable in a confined interval for comparison purposes ($r_{t_{min}} \leq r_t \leq r_{t_{max}}$). Values for crew duty slack time and transit buffer time will be assigned to depending on the case. The parameters, rebooking, ferry and positioning time will be assigned constant values to regardless of the case (see Table 42).

Parameter value	Description
$r_{t_{min}} = 120$	repair takes at least two hours (adopted from scenario)
$r_{t_{max}} = 240$	repair takes at most four hours
$k_t = 180$	three hours until ferry flight arrives and is ready to depart
$b_t = 180$	three hours until rebooking flight departs
$d_t = 180$	three hours until positioned crew arrives and is ready to operate flight

Table 42 – parameter values for case 1, 2 and 3

There are several reasons why an aircraft cannot be put under unscheduled maintenance thus resulting in the aircraft being AOG (see Figure 7 in p.39). Examples include an inadequate technical diagnosis (a_0), not having the required spare part available (b_0) or having an unfavourable weather pattern in conjunction with hangar unavailability ($c_0 \vee d_0$). For the fourth case, condition sequences are chosen with an inadequate technical diagnosis (a_0) meaning that the diagnosis that is observed by the AOC is considered to be inadequate to plan unscheduled mx for the aircraft. In order to have a consistent condition sequences with case 1-3, the same sequences are chosen with the only difference being the technical diagnosis which is inadequate. These sequences are denoted by "q":

$$q_1 = (\boldsymbol{a_0}, b_1, c_1, d_0, e_0, f_0, g_0, h_0, i_0)$$

$$q_2 = (\boldsymbol{a_0}, b_1, c_1, d_0, e_0, f_0, \boldsymbol{g_1}, h_0, \boldsymbol{i_1})$$

$$q_3 = (\boldsymbol{a_0}, b_1, c_1, d_0, e_0, f_0, g_0, \boldsymbol{h_1}, i_0)$$

$$q_4 = (\boldsymbol{a_0}, b_1, c_1, d_0, e_0, f_0, \boldsymbol{g_1}, \boldsymbol{h_1}, \boldsymbol{i_1})$$

$$q_5 = (\boldsymbol{a_0}, b_1, c_1, d_0, e_0, \boldsymbol{f_0}, \boldsymbol{g_1}, \boldsymbol{h_1}, \boldsymbol{i_1})$$

The **bold** conditions express the difference of each condition sequence with regard to p_1.

Since the aircraft is AOG, it will not be possible to formulate a repair time. This provides the opportunity to evaluate other parameters. In the previous cases the ferry time and rebooking time was held equal. For this case the relationship between these two parameters is evaluated. This is done by using the ferry time as a variable in the interval: $k_{t_{min}} \leq k_t \leq k_{t_{max}}$ while keeping rebooking time constant. For comparison purposes the ferry time interval boundaries will be equal to the repair time values of case 1-3 (see Table 43).

Parameter value	Description
$k_{t_{min}} = 120$	ferry takes at least two hours
$k_{t_{max}} = 240$	ferry takes at most four hours
$b_t = 180$	three hours until rebooking flight departs

Table 43 - parameter values for case 4

7.1.3 Calculating the costs

By simulating a condition sequence with applicable parameters and parameters, the LEADSTO simulation tool will provide a "trace.tr" i.e. trace-file. For each trace a recovery strategy is provided as a state property which is shown in Expression 11 (for description of the predicate see Table 39).

$$recovery(\text{'}RS\text{'}, d_f, r_t. d_t, b_t, k_t)$$

$d_f = delay\ of\ the\ passengers$
$r_t = repair\ time$
$d_t = positioning\ time$
$b_t = rebooking\ time$
$k_t = ferry\ time$

Expression 11- recovery strategy and associated parameters

In order to isolate the choice of recovery strategies, the associated parameters and conditions of the simulation, they are imported into a spreadsheet. Since the trace file has a standard format with delimiters, the "text-to-columns" tool of EXCEL can be used for data-filtering. By using if-then rules in the spread-sheet, costs associated with each recovery strategy can be calculated. Additional to the costs, time-steps of each simulation are provided.

A printscreen of the imported recovey strategies and cost calculations into Excel is shown in Appendix D.

7.2 Case 1 – repair time exceeding transit buffer time

7.2.1 Simulations and cost calculations of case 1

Since the first case involves evaluation of repair time that exceeds transit buffer time without exceeding crew duty time i.e. $r_{t_{max}} = c_t$. This means that the simulated interval can be written as $r_{t_{min}} \leq r_t \leq c_t$. The transit buffer time is chosen to be in the middle of this interval (see Table 44).

Parameter value	Description
$c_t = 240$	four hours of slack in crew duty time
$p_t = 180$	three hours of buffer for transit passengers

Table 44 – parameter values for case 1

Since there are five condition sequences, the same number of simulations is conducted. The results are categorized for each interval and presented in Table 45.

Interval	p_n	RS#	d_f	Time steps
$120 \leq r_t \leq 180$	$p_1 - p_5$	RS1	r_t	33
$180 < r_t \leq 240$	p_1	RS2		57
	p_2	RS8	$k_t = 180$	71
	$p_3 - p_5$	RS4	$b_t = 180$	53

Table 45 - chosen recovery strategies in case 1 with time steps

The chosen recovery strategies for this case is to either let passengers wait for repair to be finished (RS1 and RS2), or a reserve aircraft and crew is utilized to get passengers back to base (RS8) or passengers are rebooked on another flight (RS4). Transit passengers will only miss their connecting flight in RS2; in the other recovery strategies the passengers will make a successful connection. In all simulations, the aircraft and crew is brought back to plan as soon as possible. From Table 45 it can be determined that the number of time step required for reaching a decision regarding the recovery strategy differs. After the repair time exceeds the transit buffer time in $120 \leq r_t \leq 180$ only 33 steps were required, while it can be up to 71 steps in $180 < r_t \leq 240$.

It can be observed that in $120 \leq r_t \leq 180$ the recovery strategy is independent of the condition sequence. Conversely, in $180 < r_t \leq 240$ choice of recovery strategy could differ depending on the condition sequence. The simulations with p_3, p_4 and p_5 provide the same result and even though there are reserve resources available in p_4 and p_5, these are not utilized.

The operating costs that are associated with each simulation are plotted with respect to repair time and illustrated in Figure 19.

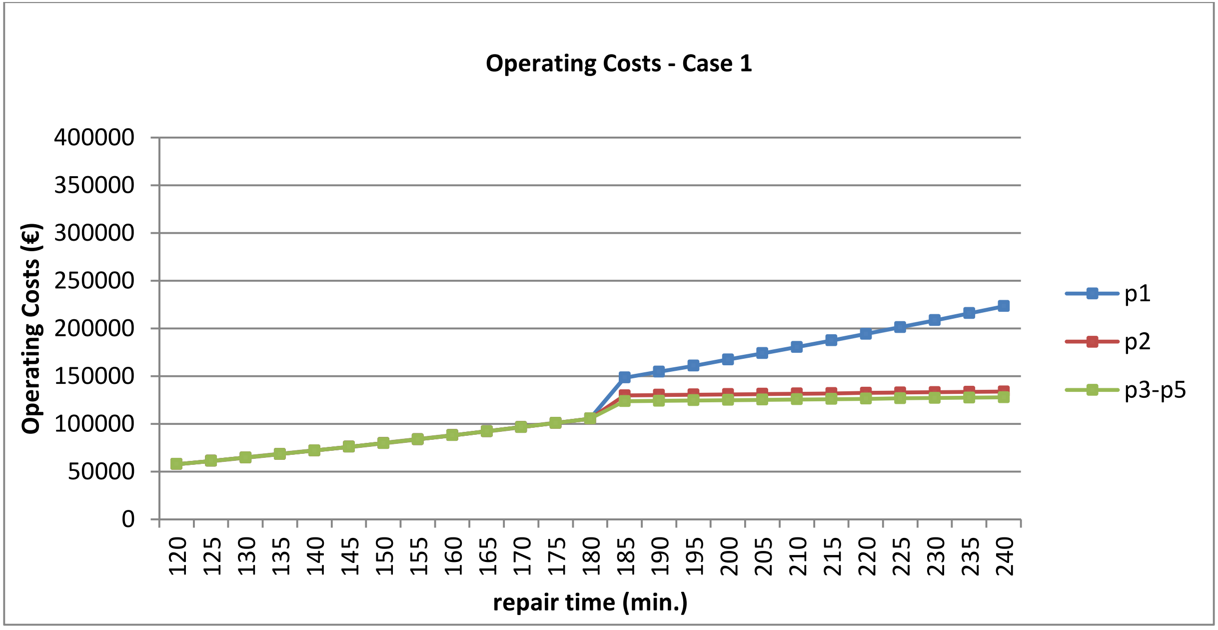


Figure 19 - plotted graph of the operating cost with respect to repair time for case 1 ($\Delta r_t = 5$)

Since the simulations show the same recovery strategies in the interval $120 \le r_t \le 180$, it is evident that the costs in this interval do not show any discrepancies. However, from Figure 19 it can be observed that in $r_t > 180$ the rise in cost differ significantly between $p_1$and the other condition sequences. Numerical values of the cost calculations are provided in Table 46.

Interval	p_n	C[5] (min. - max.)	$\frac{\partial C(p_n, r_t)}{\partial r_t}$	D[5] (min.-max.)
$120 \le r_t \le 180$	$p_1 - p_5$	€57,661- €105,353	$3.8 \cdot r_t + 202$	€12,474 -€16,836
$180 < r_t \le 240$	p_1	€143,519- €223,177	$5.1 \cdot r_t + 251$	€23,509- €27,798
	p_2	€129,559- €133,776	73	€41,042- €45,259
	$p_3 - p_5$	€123,426- €127,715		€34,909- €39,198

Table 46 – cost calculations of simulations of case 1

From Table 46 it can be observed that in $120 \le r_t \le 180$ the operating and direct cost level are the lowest. The moment the repair time surpasses 180 minutes the operating costs rises per minute of repair time either substantial or slightly depending on conditions.

It seems that for all condition sequences the operating cost depend on repair time i.e. $\partial C_{RS}(r_t)/\partial r_t \neq 0$. It can be observed that the rate of increase in operating costs is the greatest for $p_1$in $180 < r_t \le 240$. This ranges from €1147/min at r_t =181 until €1475/min at r_t=240. For the other condition sequences the operating costs rise with €73/min which equals the rise of direct costs. Since the direct costs is a cost component of the operating costs it implies that the quality costs for p_2, p_3, p_4, and p_5 do not rise in $180 < r_t \le 240$.

Next to dependency of operating cost level on repair time, cost levels can also be incremented by “one-off” costs (ΔC_{RS}). Figure 19 shows that this increment mainly takes place from r_t=180 to r_t=181. For p_1 the one-off costs increases with Δ€38,166, while the increase for p_2 and $p_3 - p_5$ are respectively Δ€23,140 and Δ€18,364. The difference between p_1 and the other condition sequences is that for the former the increase is largely caused by the quality cost component while the latter is caused by a rise in direct costs.

7.2.2 Discussion of case 1

The simulation results show that in the interval $r_t \le p_t$ and $p_t < r_t < c_t$ the chosen recovery strategies differs quite substantially. For both intervals the operating costs depend on repair time regardless of the applicable conditions. However, for the latter interval, the composition of the operating costs does depend on the condition sequence. In the absence of recovery opportunities (i.e p_1), the quality costs rises and in the case of utilizing reserve resources or rebooking passengers, the direct costs rise. This means that **when repair time exceeds transit buffer time, the composition of the operating costs changes** which implies that the transit buffer time acts as tipping point[6] for costs.

Furthermore, the results show that during interval $p_t < r_t < c_t$ utilizing reserve resources (p_2) will result in lower operating costs compared when there are no recovery opportunities (p_1). On the contrary, it also shows that the direct costs are increasing significant when the reserve resources are being utilized.

This implies that the Operations Controller has to make **a decision between delivering customer service and utilizing resources effectively**. This is an example of competing objectives. In the model design, the transit passenger connecting successfully was prioritized above retention of reserve resources (see Table 18) which explains the utilization of reserve resources in p_2. However, it could have been very well possible that in another situation in the real-world Operations Controller would prefer retaining reserve resources and thus decreasing the customer service level.

---

[5] passenger compensation of €33,750 not taken into account for comparison purposes

[6] the ferry time and rebooking time are held constant and favourable in this case ($k_t \le p_t$). and ($b_t \le p_t$), in case 3 and case 4, these two inequalites are evaluated

The simulations show that there is a difference in chosen recovery strategy when repair time is either observed in $r_t \leq p_t \; or \; p_t < r_t < c_t$. The repair was an uncertainty which was of interest in the study of Bruce (2011). Typical factors that influence the repair duration is availability of manpower, complexity of the task (Table 2). This uncertainty could be overcome by collecting information from local technicians and maintenance departments, but also from technical systems like maintenance manuals. This means that this information is highly distributed and prone to uncertainty which can be overcome by making estimations by the aircraft controller.

From Figure 6 it could be determined that the repair duration is determined before it is compared with transit buffer time (phase 1 and phase 2A). This means that the aircraft controller is not aware whether his repair time estimations will be in $r_t \leq p_t$ or $p_t < r_t < c_t$ which has shown to have great implication on the choice of the recovery strategy. Additionally, from Table 45 it can be seen that the number of time-steps required to come up with a decision also depends on repair time. When the time-steps differ this implies that the decision-making process also differs. To show what the extra time steps are occurred by, two simulations are used with condition sequence p_2. One with $r_t = 170$ which is within $r_t \leq p_t$ and one with $r_t = 200$ which is within $p_t < r_t < c_t$. The former repair estimation results in RS1, while the latter repair time estimation results in RS8, which means that reserve resources will be utilized to retrieve passengers.

The traces can be referred to in Appendix E and Appendix F and the difference in interactions between these two simulations is highlighted in Appendix F and shown in Table 47.

	Input					
Output	OC	AC	CC	SC	FD	ENV
OC		Δ1	Δ1	Δ2	-	-
AC	-		-	-	Δ1	Δ1
CC	Δ1	-		-	-	Δ1
SC	Δ2	-	-		Δ1	Δ2
FD	Δ1	-	-	Δ1		Δ2
ENV	-	Δ1	Δ1	Δ2	Δ2	

Table 47 – discrepancies between r_t=170 and r_t=200 in terms of interactions

From Table 47 it can be seen that all controllers will have additional interactions. By using task descriptions of Table 16 and Table 17 and the traces that resulted from these simulations, the following discrepancies between these two simulations have been found in terms of tasks that are done:

1. OC has additional coordination tasks (OC1)
2. FD has to determine ferry time (FD3)
3. FD has to look whether organizing connections can be established (FD2)
4. AC has to look into availability of reserve aircraft (AC5)
5. CC has to look into availability of reserve crew (CC2)
6. SC has to look for rebooking opportunities (SC5)
7. SC has to look whether the ferry time of the reserve aircraft will provide successful transit passenger connections (SC1)

This means that a repair time difference of 30 minutes could lead to a great difference in the decision-making process. Simulation results show that when, for instance, the repair time is assumed in a higher region with respect to other scheduling parameters, that it results in a **significant increase in number of interactions and thus task load** for the controllers. Feigh (2008) mentioned that increased coordination tasks could lead to less time for gathering information this could eventually lead to **unnecessary deployment of costly recovery strategies** including utilization of reserve resources.

7.3 Case 2 – repair time exceeding crew duty time

7.3.1 Simulations and cost calculations of case 2

In contrast with previous case, it will be evaluated what the effect of repair time that exceeds crew duty slack time is instead of transit buffer time. For this reason, crew duty slack time (c_t) is interchanged with transit buffer time (p_t). This lead to the following simulated interval: $r_{t_{min}} \leq r_t \leq p_t$ i.e. $r_{t_{max}} = p_t$. Crew duty slack time will be chosen to be in the middle of the interval (see Table 48).

Parameter value	**Description**
$c_t = 180$	three hours of slack in crew duty time
$p_t = 240$	four hours of buffer for transit passengers

Table 48 - parameter values for case 2

The simulations resulted in the AOC choosing five distinct recovery strategies which are presented in Table 49.

Interval	$\boldsymbol{p_n}$	**RS#**	$\boldsymbol{d_f}$	**Time-steps**
$120 \leq r_t \leq 180$	$p_1 - p_5$	RS1	r_t	33
$180 < r_t \leq 240$	p_1	RS3	$crew\ rest$	45
	p_2	RS10	$k_t = 180$	75
	$p_3 - p_4$	RS6	$b_t = 180$	49
	p_5	RS11		57

Table 49 - deployed recovery strategies in case 2

Just like in case 1, during $120 \leq r_t \leq 180$ the passengers wait for the repair to be finished. However, in $180 < r_t \leq 240$, the passengers and crew are accommodated (RS3), or a reserve aircraft is dispatched (RS10), rebooking passengers (RS6) or combining rebooking with positioning of crew (RS11). The transit passengers will make a successful connection in all recovery strategies except in RS3. In all the cases, the disrupted aircraft is recovered after crew rest except of RS11 which is recovered after aircraft repair.

The time-steps show a small decrease in the condition sequences p_1, p_3 and p_4 compared to case 1. In the other condition sequence a small increase is observed.
Additionally, it can be seen that p_3 and p_4 result in the same recovery strategy even though in p_4 reserve resources are present, but not utilized. However, in p_5 reserve crew is used and passengers are rebooked, while reserve aircraft is available and retained.

The costs that are associated with each simulation are plotted with respect to repair time and illustrated in Figure 20.

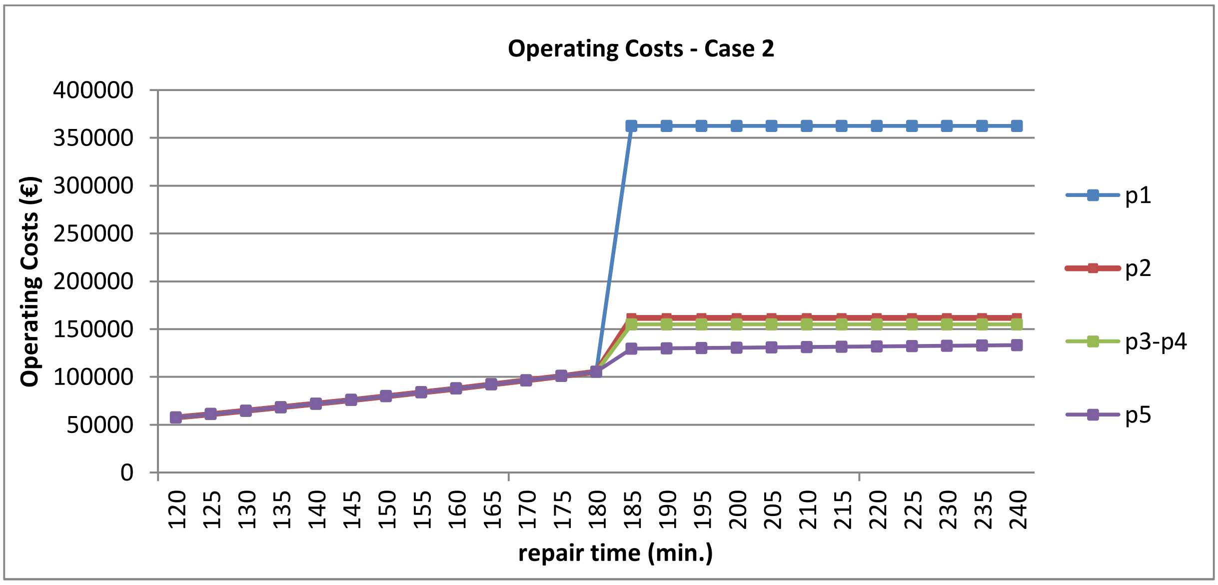


Figure 20 - plotted graph of the operating cost with respect to repair time for case 2 ($\Delta r_t = 5$)

From Figure 20 it can be deduced that the operating cost levels between p_1 and the other condition sequences is quite substantial. It can be determined that just like case 1, the operating costs decreases when added recovery opportunities. Furthermore, it can be deduced from Figure 20 that the operating costs increases right after $r_t = 180$. Another aspect to be highlighted is that there is little to no increase in operating costs with respect to repair time in $r_t > 180$ (see Table 50).

Interval	p_n	C [7] **(min. - max.)**	$\frac{\partial C(p_n, r_t)}{\partial r_t}$	D **(min-max)**
$120 \leq r_t \leq 180$	$p_1 - p_5$	€57,661- €105,353	$3.8 \cdot r_t + 202$	€12,474 -€16,836
$180 < r_t \leq 240$	p_1	€362,464	0	€72,160
	p_2	€161,696		€73,180
	$p_3 - p_4$	€154,977		€66,460
	p_5	€129,232- €133,126	73	€40,716- €44,610

Table 50 - costs for the simulations in case 2

Table 50 shows that only for p_5 the operating costs depend on repair time. It can be determined that the operating costs increases with €73/min and can be entirely allocated to direct costs. This implies that for neither condition sequence the quality costs increases per increment of repair time in the interval $180 < r_t \leq 240$.

The one-off costs in this case are considerably higher than the ones in case 1. Especially in p_1 this seems the case. When repair time increments with one minute from $r_t = 180$ to $r_t = 181$ the "one-off" operating costs ranges from around Δ€260,000 in p_1 to around Δ€25,000 in p_5. In the condition sequences p_2 – p_5 these one-off costs are entirely caused by an increase in direct costs while in p_1 these are mostly due to an increase in quality costs. This increase in quality costs is caused by the stranded passengers resulting in costs for accommodation and missed connection for transit passengers.

This implies that the recovery opportunities presented in $p_2 - p_5$ will have much more impact in terms of decreasing quality costs.

---

[7] passenger compensation of €33,750 not taken into account for comparison purposes

Even though p_5 combines positioning crew and rebooking it still results in a lower cost than rebooking passengers only. This has to do with the fact that in p_5, the disrupted aircraft is recovered much quicker due to replacement of crew A with reserve crew.

7.3.2 Discussion of case 2

Results of case 2 have shown that $r_t \leq c_t$ has the same characeteristics as $r_t \leq p_t$ in case 1. However, repair time during the $c_t < r_t < p_t$ shows different characteristics than $p_t < r_t < c_t$. From the simulations it can be concluded that in the latter interval the operating cost level is independent on repair time when the nature of the disruption shifts from being an aircraft mechanical failure into a crew problem that is not solvable by positioning (reserve) crew. Additionally, the result show that the quality cost is independent of repair time under all conditions when repair time exceeds crew duty time. Concluding, repair time passing the crew duty slack will result in a significant “one-off” increase in direct cost compared to case 1. Furthermore, in the case of no recovery opportunities being present the rise in “one-off” quality cost will also be significant.

In the comments of Bruce (2011), controllers were very much interested into accurate time estimations i.e. actual times. However, simulations have shown that accurate repair time estimation are not required since operating costs is independent of repair time (i.e. $\partial C_{RS}(r_t)/\partial r_t = 0$). This is true when there no positioning opportunities $(p_1 - p_4)$ and the repair time being in the interval $c_t < r_t < p_t$, This means **that the aircraft controller collecting additional information for accurate repair time estimations seems to be unnecessary in the abovementioned situation** and could result in an unnecessary long decision-making process.

Another aspect that is interesting is the effect of having rebooking opportunities. The results have shown that having rebooking opportunities $(p_3 - p_4)$ in contrast with no recovery opportunities at all will (p_1) beneficially impact the operating cost with €207,487 and could retain reserve resources. In the comments of Bruce (2011) numerous comments related to rebooking passengers. Controllers were seeking into possibilities to rebook the passengers on flights of its own carrier, but also on other carriers. The main controller that is responsible for overcoming this uncertainty is the *station operations controller.* These rebooking opportunities, as mentioned in Table 2, depend on multiple factors including availability of those flights and load factor (percentage of available seats). Other factors include if ground handling is able to send baggage over to the rebooking flight within the rebooking time. This means that the ability to find rebooking opportunities for the passengers and thus **overcoming this uncertainty can reduce costs significantly** when controllers are confronted with a crew problem. The study of Bruce (2011) already provided insights into that predominantly controllers with experience identify advanced decision considerations by requesting information. The current study has shown that, for instance, identifying rebooking opportunities could result in some cases in significant lower operating costs and retention of reserve resources.

7.4 Case 3 – repair time exceeding crew duty and transit buffer time

7.4.1 Simulations and cost calculations of case 3

In the previous two cases, the effect of repair time exceeding crew duty time and transit buffer time were evaluated independently. In this case it will be examined when repair time exceeds both these parameters. Furthermore, it will also be evaluated when transit buffer time will be less than rebooking time and ferry time i.e. evaluating the inequalities $b_t > p_t$ and $k_t > p_t$ Three values of transit buffer time and crew duty time will be chosen (1) at the middle of the simulated interval (=180 minutes), (2) at an increment of five minutes (=185 minutes) and (3) a decrement of five minutes (=175 minutes). This result in a total of nine combinations of transit buffer and crew duty slack time (see Table 51)

		c_t		
		175	180	185
p_t	185	**1**	**2**	**3**
	180	**4**	**5**	**6**
	175	**7**	**8**	**9**

Table 51 - combinations of transit buffer time and crew duty time for case 3

The simulated interval ($r_{t_{min}} \leq r_t \leq r_{t_{max}}$) and all other relevant parameters are the same as in case 1 and case 2.

The simulation results of **combinations 1-5** show the same pattern of choice in recovery strategy **as in case 2**. The only difference that could be noticed is that the point at which a certain recovery strategy is chosen moves with the crew duty slack time. For instance, with p_1, the difference between simulations of combination 1 and 2 is that at the former, RS3 is chosen from $r_t > 175$ and in the latter from $r_t > 180$. This means that repair time exceeding transit buffer time does not have effect on the choice of recovery strategy i.e. transit buffer time is insensitive with respect to repair time with these combinations.

Simulations with **combination 6** show a pattern that is a mixture of the simulations in case 1 and case 2. In this case the connecting passengers and reserve resource utilization is presented in the table with simulation results (seeTable 52).

Interval	$\boldsymbol{p_n}$	**RS#**	$\boldsymbol{d_f}$	**Reserve resources utilized**	**Transit pax connecting successfully?**
$120 \leq r_t \leq 180$	$p_1 - p_5$	RS1	r_t	-	yes
$180 < r_t \leq 185$	p_1	RS2			no
	p_2	RS8	$k_t = 180$	aircraft and crew	yes
	$p_3 - p_5$	RS4	$b_t = 180$	-	
$185 < r_t \leq 240$	p_1	RS3	$crew\ rest$		no
	p_2	RS10	$k_t = 180$	aircraft and crew	yes
	$p_3 - p_4$	RS6	$b_t = 180$	-	
	p_5	RS11		crew	

Table 52 – Simulations in case 3 with p_t=180 and c_t=185

Three intervals can be distinguished in this simulation which means that there are two tipping points at which the recovery strategy is changing. At $r_t = 180$ and $r_t = 185$, respectively the values of transit buffer and crew duty slack time.

Simulations with **combination 7** resulted in recovery strategies that have not been chosen in the previous cases and are shown in Table 53.

Interval	$\boldsymbol{p_n}$	**RS#**	$\boldsymbol{d_f}$	**Reserve resources utilized**	**Transit pax connecting successfully?**
$120 \leq r_t \leq 175$	$p_1 - p_5$	RS1	r_t	-	yes
$175 < r_t \leq 240$	p_1	RS3	$crew\ rest$	-	no
	p_2	RS9	$k_t = 180$	aircraft and crew	
	$p_3 - p_4$	RS5	$b_t = 180$	-	
	p_5	RS12		crew	

Table 53 - simulations in case 3 of combination 7 (p_t=175 and c_t=175)

It can be seen from Table 53 that there is just one tipping point and that the same recovery types as case 2 are being applied i.e. cancelling, utilizing reserve aircraft and crew, rebooking passengers and positioning crew. However, the difference between these simulations with the simulations of case 2 is that in this case **no transit passenger is connecting successfully** in $r_t > 175$. Due to the fact that the same recovery tactics are applied as case 2 implies that also in this case repair time with respect to transit buffer time is insensitive.

Simulations of **combination 8 and 9** have similarities with combination 7 in terms of recovery tactics i.e. cancelling or resulting in unsuccessful connection of transit passengers. However, in this case there are also two tipping points at which recovery strategies are changed. The simulation results of combination 8 are shown in Table 54.

Interval	p_n	RS#	d_f	Reserve resources utilized	Transit pax connecting successfully?
$120 \le r_t \le 175$	$p_1 - p_5$	RS1	r_t	-	yes
$175 < r_t \le 180$	$p_1 - p_5$	RS2			no
$180 < r_t \le 240$	p_1	RS3	$crew\ rest$		
	p_2	RS9	$k_t = 180$	aircraft and crew	
	$p_3 - p_4$	RS5	$b_t = 180$	-	
	p_5	RS12		crew	

Table 54 - simulations in case 3 of combination 8 i.e. p_t=175 and c_t=180

The simulations in Table 54 show a difference with the simulations shown in Table 53 in terms of deployed recovery strategy. In $175 < r_t \le 180$, RS2 is being chosen which means that the passengers will have to wait until repair is finished, resulting in transit passengers making an unsuccessful connection.

The costs of combination 8 will presented and plotted with respect to repair time (see Figure 21).

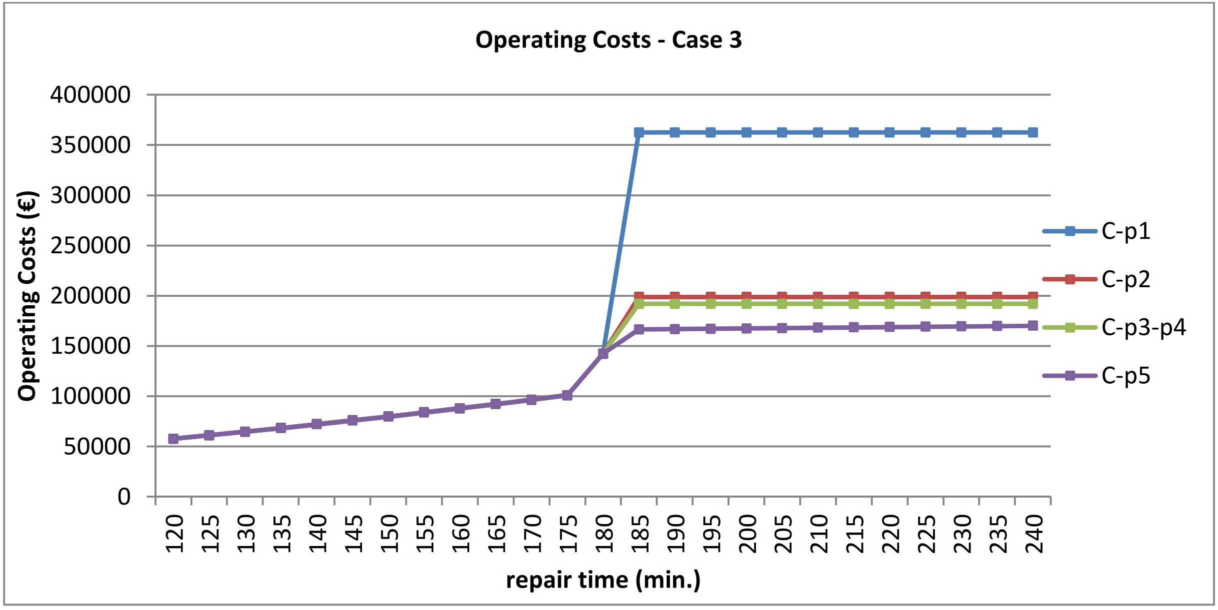


Figure 21 - plotted graph of operating cost with respect to repair time for case 3 of combination 8 i.e. p_t=175 and c_t=180 ($\Delta r_t = 5$)

From Figure 21 it can be deduced that the cost levels is decreasing with added recovery opportunities (as the previous cases). Additionally, it can be seen that the costs in all condition sequences are elevated compared to case 2 except for p_1, which results in the same cost level as case 2. Another aspect that could be observed is the two tipping points at $r_t = 175$ and $r_t = 180$.

Numerical values of the cost calculations of combination 8 are provided in Table 55.

Interval	$\boldsymbol{p_n}$	$\boldsymbol{C}$[8] **(min. - max.)**	$\frac{\partial C(p_n, r_t)}{\partial r_t}$	$\boldsymbol{D}$ **(min-max)**
$120 \leq r_t \leq 175$	$p_1 - p_5$	€57,661- €100,851	$3.8 \cdot r_t + 202$	€12,474 -€16,473
$175 < r_t \leq 180$	$p_1 - p_5$	€137,587-€142,322	$5.1 \cdot r_t + 251$	€23,145-€23,436
$180 < r_t \leq 240$	p_1	€362,464	0	€72,160
	p_2	€198,666		€79,780
	$p_3 - p_4$	€191,946		€73,060
	p_5	€166,466- €170,096	73	€47,316-€51,210

Table 55 - costs for the simulations in case 3 of combination 8 (p_t=175 and c_t=180)

From Table 55 it can be deduced that the operating costs always depend on repair time in $r_t \leq 180$. From $r_t > 180$ it seems that only in p_5 the operating costs depend on repair time which seems to be accountable to direct costs just like case 2.

Regarding one-off costs it shows that these are higher than in the cases 1-2 for all condition sequences except for p_1. The one-off costs are Δ€56,344, Δ€49,624 and Δ€24,144 for respectively p_2, p_3/p_4 and p_5 when $r_t = 180$ increases to $r_t = 181$. This is caused by the fact that in all these occasions the transit passengers will make an unsuccessful connection which means that there is an increase in quality costs due to passenger dissatisfaction and an increase in rebooking fee for the transit passenger due to missed connection.

7.4.2 Discussion of case 3

From the simulations it is observed that when $c_t > p_t$ is true that there are three intervals in which recovery strategies are being deployed (1) $r_t \leq p_t$, (2) $p_t < r_t \leq c_t$ and (3) $r_t > p_t$ implying that there are two tipping points in the repair time interval. Furthermore, simulations show that when $c_t \leq p_t$ there are just two intervals in which recovery strategies are deployed (1) $r_t \leq c_t$ and (2) $r_t > c_t$ and that there is just one tipping point at which the recovery strategy changes in this interval. This means that the repair time with respect to transit buffer time is completely insensitive. Additionally this means that **moving crew duty slack time equals moving the tipping point dictating operating costs for recovery.**

Simulations show that when transit buffer time is smaller with respect to rebooking time and ferry time $b_t > p_t$ and $k_t > p_t$ that it will result in elevated quality costs and direct costs if repair time exceeds crew duty time. This is only true for situations in which there are recovery opportunities present. Conversely, in the case of having no recovery opportunities, it seems that transit buffer time is insensitive when crew duty time exceeds repair time.

Since the transit buffer time can be insensitive, it also means that looking into organizing connections could be useless. From the description of the flowcharts (Figure 8) it can be deduced that the tasks of checking whether crew and passengers are affected by repair, are parallel. This suggests that when $r_t > c_t$ applies then the following tasks are performed which do not contribute to the choice of the recovery strategy: determine effect of delay on passenger connections (SC1) and determine possibilities to organize connections (FD2).

---

[8] passenger compensation of €33,750 not taken into account for comparison purposes

By “organizing connections” controllers are seeking ways to ensure transit passengers make a successful connection. This has been conceptualized using the conditions e_n. Additional simulations with organizing connection possible e_1, have verified that it is the same as increasing the transit buffer to such an extent that it can absorb the delay that resulted from the disruption. Various comments in the study of Bruce (2011) relate to controllers seeking ways to make sure that transit passengers will make a successful connection (i.e. organizing favourable connection). During the strategic phase, flights are usually not planned at their maximum speeds due to restrictions of ATC or due to fuel saving motivations, this leaves room for the flight dispatch to increase cruise speeds. Next to this measure certain comments related to requesting gates that will promote less connecting time or acceleration of turn- around times. However, not only the robust scheduling is important, also the ability to put these measures into account is important. Flight dispatch is responsible for this task (see Table 16), it is demonstrated in the simulation that when transit buffer time is increased around rebooking, ferry or repair time, that the saved costs can be up to €36,970 (See p_2 between Table 50 and Table 55).

Transit buffer time and crew duty slack time are implemented during the strategic phase. However, these slacks can also be influenced by swapping crew or cascading effects of previous flights during the tactical phase. Simulations results have shown that these slacks play a major role in the choice of recovery strategies, direct costs, utilization of reserve resources and delivering customer service. However, these simulations have also shown that these two parameters do not share the same characteristics.
Concluding, buffers and slack are important parameters that act as tipping point in certain cases. It has been demonstrated that these buffers and slacks can be used to postpone in some cases very costly recovery strategies and usage of resources. Furthermore, the simulation show that transit buffer and crew duty slack do not share the same characteristics.
However, the simulations also have shown that in some cases a large transit buffer time will not have any effect on recovery strategy, while this imposes costs due to robust scheduling.

7.4 Case 4 –Aircraft on Ground

7.4.1 Simulations and cost calculations of case 4

In this case simulations are done in which the Aircraft is on Ground due to lack of an adequate technical diagnosis. For this case the variable ferry time (k_t) will be selected to be evaluated in the interval $k_{t_{min}} \leq k_t \leq k_{t_{max}}$. As mentioned in 7.1.2, the condition sequences denoted by ‘*q*’ will be used for simulations.

To address the effect of transit buffer time in relation with rebooking time and ferry time, two values of transit buffer time will be chosen (1) $p_t = 180$ and (2) $p_t = 175$

The LEADSTO simulation with $\boldsymbol{p_t = 180}$, provided the following recovery strategies (see Table 56)

Interval	$\boldsymbol{q_n}$	**RS#**	$\boldsymbol{d_f}$	**Reserve resources utilized**	**Transit pax connect successfully?**
-	q_1	RS13	*cancel*	-	no
$120 < k_t \leq 180$	q_2, q_4	RS15	k_t	aircraft and crew	yes
$180 < k_t \leq 240$		RS16			no
-	q_3	RS19	$b_t = 180$	-	yes
	q_5	RS17	$b_t = 180$		

Table 56 – Simulation result of case 4 with $p_t = 180$

From Table 56 it can be observed that there are five different recovery strategies chosen: cancelling flight and accommodating passengers and crew (RS13), dispatching reserve aircraft to get passengers back and to position resources (RS15 and RS16), rebooking passengers without positioning resources (RS19) and rebooking passengers with positioning resources on another flight (RS17). It seems that reserve resources are retained when positioning opportunities are present. Furthermore, it seems that ferry time is only sensitive in q_2 and q_4 which are the condition sequence that will lead to the utilization of the reserve resources. Another aspect which is striking is that in $180 < k_t \leq 240$ for q_4 resources are utilized and this results in passengers making an unsuccessful connection, while it could have been possible to rebook the passengers with a successful connection[9].

The costs of the chosen recovery strategies are shown in Table 57.

Interval	q_n	C[10] **(min. - max.)**	$\frac{\partial C(q_n, k_t)}{\partial k_t}$	D **(min.-max.)**
-	q_1	€389,188	0	€98,884
$120 < k_t \leq 180$	q_2& q_4	€100,919-€148,610	$3.8 \cdot k_t + 202$	€55,732-€60,094
$180 < k_t \leq 240$		€186,776-€266,435	$5.1 \cdot k_t + 251$	€66,766-€71,056
-	q_3	€163,701	0	€75,184
	q_5	€141,891		€53,374

Table 57 – Cost of case 4 simulations with $p_t = 180$

Table 57 shows that operating cost dependency on ferry time has the same values as repair time dependency on operating costs under certain circumstances. Furthermore, the cost calculation shows that direct costs are decreasing with added recovery opportunities. However, this is not the case for operating costs which seem to be the highest in q_1 and, depending on the ferry time it could either be lower or higher with added recovery opportunities.

The cost level of $q_1$is the highest of all simulations so far in terms of direct and quality costs. The high direct costs are caused by accommodating passengers and crew, but also due to AOG of the disrupted aircraft. Another observation that can be made is that in $q_1$and q_3 the direct costs are higher than in q_2, q_4 and q_5. This can be explained by the fact that there is no recovery regarding the disrupted aircraft that will remain AOG.

The simulations with $\boldsymbol{p_t = 175}$, provided the following recovery strategies (see Table 58.)

Interval	q_n	**RS**	d_f	**Reserve resources utilized**	**Transit pax connect successfully?**
-	q_1	RS13	*cancel*	-	no
$120 < k_t \leq 175$	q_2, q_4	RS15	k_t	aircraft and crew	yes
$175 < k_t \leq 240$		RS16			no
-	q_3	RS20	$b_t = 180$	-	
$120 < k_t \leq 175$	q_5	RS15	k_t	aircraft and crew	yes

---

[9] since $b_t \leq p_t$

[10] passenger compensation of €33,750 not taken into account for comparison purposes

$175 < k_t \leq 240$		RS18	$b_t = 180$	-	no

Table 58 - Simulation result of case 4 with $p_t = 175$

From Table 58 it can be deduced that in contrast with the simulation with $p_t = 180$, there are differences in recovery strategies in q_3 and q_5. Regarding q_3, the only difference is that the passengers will not be able to make a successful connection. However, for q_5 it seems that there is a sudden shift in utilization of reserve resources are being utilized in $k_t \leq 175$ and passengers are rebooked and resources are send in $k_t > 175$. Another aspect that differs with the previous simulations is that in q_2 and q_4 there due to a decrease of 5 minutes in transit-buffer this will result in transit passengers will make a successful connection.

The costs of the simulations with $\boldsymbol{p_t = 175}$ are shown in Table 59.

Interval	$\boldsymbol{q_n}$	$\boldsymbol{C}$[11] **(min. - max.)**	$\boldsymbol{\frac{\partial C(q_n, k_t)}{\partial k_t}}$	$\boldsymbol{D}$ **(min. – max.)**
-	q_1	€389,188	0	€98,884
$120 < k_t \leq 175$	q_2, q_4	€100,919-€144,108	$3.8 \cdot k_t + 202$	€55,732-€59,730
$175 < k_t \leq 240$		€180,845-€266,435	$5.1 \cdot k_t + 251$	€66,403-€71,056
-	q_3	€200,670	0	€81,784
$120 < k_t \leq 175$	q_5	€100,919-€144,108	$3.8 \cdot k_t + 202$	€55,732-€59,730
$175 < k_t \leq 240$		€178,860	0	€59,974

Table 59 - Cost of case 4 simulations with $p_t = 175$

The cost calculations show major differences in q_3 and q_5 compared with $p_t = 180$. For q_3 there is an increase in operating costs and direct costs due to transit passengers connecting unsuccessfully. For q_5 the costs seems to depend on the ferry time and could be lower in this case when the ferry time is relatively low.

From the simulations with $p_t = 175$ it can be concluded that in most cases the operating and direct costs will rise. Additionally, the simulations have shown that a decrease in transit buffer time could result in utilization of reserve resources. For case 4 it has shown that a decreased buffer without having reserve aircraft or positioning possibilities imposes the greatest implications on operating costs.

7.4.2 Discussion of case 4

When comparing the costs of the simulations of condition sequences denoted by "q" with the condition sequences of "p" it can be observed that for case 1 the costs were much lower. For instance, the direct costs of simulations with p_3 in case 1 resulted in direct costs of up to €39,198 while for q_3 this seems to be €75,184. The difference of q_3 with p_3, is the fact that the technical diagnosis is considered to be inadequate by the aircraft controller. In the comments of (Bruce, 2011) focused on getting an adequate technical diagnosis through local engineers or the maintenance department and others put great efforts in spare parts availability. These results have shown that these tasks could promote very low cost recovery strategies.

When transit buffer time is decreased with five minutes, then the pattern of the recovery strategy will change in p_5. The reason for this change in pattern can be explained by the fact that the relation of rebooking time and ferry time changes with respect to the transit buffer time. This results in reserve resources to be utilized when the transit buffer time is decreased.

---

[11] passenger compensation of €33,750 not taken into account for comparison purposes

In the case of $p_t = 180$ it can be deduced that in q_3, passengers are rebooked resulting in transit passengers connecting successfully, while in q_4, reserve aircraft is dispatched resulting in transit passengers connecting unsuccessfully[12].

When taking into account that in q_4 the passengers could also have been rebooked resulting in successful connection means that the *Operations Controller* has intentionally let transit passengers connect unsuccessfully for the purpose of sending resources to outstation. This is an example of **competing objectives**. Dispatching a reserve aircraft with technicians on board to the disrupted aircraft on outstation could result in acceleration in getting the disrupted aircraft back to plan. This is also reflected in lower direct costs in q_4 compared to q_3. This is quite a remarkable observation, that in **this case utilizing reserve resources instead of rebooking passengers could decrease direct costs.**

The simulations in this case show that rebooking opportunities are insensitive in q_4. On the contrary with case 1-3, in which the reserve resources have shown to be insensitive in p_4. This could be explained by the sequence of the decision-making processes. In Figure 16, the order of was to first request information regarding reserve aircraft availability from aircraft controller (task AC5) and reserve crew availability from crew controller (task CC2), while in Table 27, first rebooking possibilities are requested from station operations controller (task SC2). This can be verified by two traces with exactly the same parameters simulating condition sequences q_4 and p_4 shown in respectively Appendix G and Appendix H. From this observation it can be concluded that the uncertainty of an inadequate technical diagnosis influences the decision-making process significantly in terms of the sequence of the tasks for aircraft controller, crew controller and station operations controller. Another item which has to be considered regarding this observation is that in q_4 the priority of the objective is to position resources to the aircraft which is AOG, while in p_4, the priority is to make sure that transit passengers connect successfully.

---

[12] in the case of $k_t > p_t$

8. Conclusion & Future Research

In this chapter the conclusions of this study are presented in section 8.1. Based on these conclusions and limitations several recommendations are made for future research in section 8.2.

8.1 Conclusion

AOC controllers rarely have time to explain their reason for decision- making (Bruce, 2011). By *modelling and simulation of decision-making processes in Airline Operations Control* it can improve understanding and evaluation of the effect of both internal as external factors on the choice of recovery strategies. By quantifying this decision outcome, a sensitivity analysis is performed to study its relationship towards certain parameters and conditions. Since LEADSTO models organization dynamic, it can be used to understand behaviour on a local level. This section will answer the research question which was defined in chapter 1:

How does robust scheduling and operational uncertainties affect decision outcome and decision-making processes of Airline Operation Control?

The conclusions that can be drawn from this study are categorized into three themes: (8.1.1) robust scheduling (8.1.2) operational uncertainties and (8.1.3) decision outcome.

8.1.1 Robust Scheduling

Airlines desire to increase profitability and revenue by optimizing airline systems schedules (Belobaba et al., 2009). The airline system schedule is composed during the strategic phase of airline operational handling and during this stage flexibilities are incorporated into this schedule which is referred to as robust scheduling. Robust scheduling involves planning slacks, buffers and standby reserve resources (Kohl et al., 2007). However, airlines limit the incorporation of robustness due to high (strategic) costs that is involved with these measures. The next stage of airline operational handling is the tactical phase in which the AOC is responsible for maintaining the current airline system schedule and manage disruptions if necessary. However, buffers and slacks can change in magnitude due to cascading effect in the schedule and it is also very common that standby reserve resources have been utilized in a former disruption or by a former shift which further decreases the robustness of the schedule.

From the analysis of the socio-technical system it is observed that the first thing controllers do when starting the shift is to look at the schedule to identify ground time, spare aircraft and passenger loadings. These are all components that define robustness in a schedule and have shown in this analysis to be important piece of information in the decision-making process to build up initial situation awareness. By conceptualizing buffers, slacks and availability of planned reserve resource some interesting conclusions are drawn based on the simulations.

It is demonstrated that increased buffers and slacks could lead to a degree of self-recovery. This means that AOC does not have to take action with regard to a disruption if the delay can be “absorbed” by the slack and buffers which results in fewer tasks to be performed by controllers. These tasks include looking for recovery opportunities which could also promote less usage of reserve resources or using other costly recovery opportunities. Conversely, a smaller buffer or slack will result in more difficulties in achieving AOC objectives and will result in being more dependent on availability of recovery opportunities.

Simulations show that transit buffer time and crew duty slack time act as tipping point in terms of operating costs. Depending on the available recovery opportunities, the operating costs could either rise or flatten out after a delay exceeds this buffer or slack. The composition of the operating costs (i.e. direct or quality costs) and behaviour of these costs have shown to depend on the slack or buffer that is exceeded and on the available recovery opportunities. Delay exceeding crew duty time could result in more “one-off” costs, while delay exceeding transit buffer time results in more operating cost dependencies on the delay.

By simulating crew duty slack and transit buffer time first separately and then combined it is demonstrated that these two parameters have different characteristics. When combining these two parameters in a simulation it is shown that when repair time exceeds crew duty slack time in certain cases the transit buffer time becomes insensitive. This could be explained that repair time only exceeding transit buffer time leads to a passenger-only problem, but when repair time exceeds crew duty slack time, this leads to aircraft and crew problem cascading into a passenger problem. Since the transit buffer time is insensitive to crew duty time in certain cases, simulations have shown that unnecessary tasks could be performed by controllers that work in parallel. For instance, when *Crew Controller* already identified repair time exceeding crew duty time, then it is not necessary for the *Station Operations Controller* to compare repair time with transit buffer, neither does *Flight Dispatch* has to look to organize favourable connections.

Having planned reserve resources at the controllers' disposal have shown in the simulation to provide satisfactory results in terms of decreasing quality costs i.e. delivering customer service level. In most cases however, utilization of these reserve resources could lead to increased direct costs. In an AOG situation though, in which there are no positioning opportunities, dispatching reserve aircraft will lead to decreased direct costs. Concluding, when rebooking opportunities are not present or an AOG situation occurs, reserve resources shown to be valuable resources. Conversely, in one case it has been demonstrated that decreasing transit buffer would lead to utilization of reserve resources.

This study shows that robustness of the schedule has major impact on the decision-making process of AOC and on the operating costs required for recovery. In order to effectively evaluate decision-making performance it is important to include evaluation of the robustness of the airline system schedule.

8.1.2 Operational Uncertainties

The complex and dynamic nature of the airline operating environment creates a high level of uncertainty that can lead to errors and poor decision outcome (Bruce, 2011, Klein, 1999). Two tactics for overcoming uncertainties is to collect information or fill the 'gaps' with assumptions (Klein, 1999). The study of Bruce (2011) provided a wealth of qualitative data regarding the requests of information by the controllers. For this study, these comments have been analysed thoroughly to understand what kind of uncertainties the controllers face during the scenario. By conceptualizing these uncertainties into parameters and conditions, the simulations provide insights into its effects on the decision-making process and decision outcome.

The simulation results show that operational uncertainties could have major impact on the choice of recovery strategy affecting the chance of recovery for aircraft, crew and passengers. It is shown that if operational uncertainties are not overcome that it will lead to a decrease in choice of recovery strategies and possibly opting for a more expensive recovery strategies. Equally, when certain operational uncertainties are overcome and recovery opportunities are being discovered, it could lead to less recovery costs and retention of reserve resource. Surprisingly, there are operational uncertainties that do not have to be overcome, since they have shown to be insensitive during the simulations.

From the analysis of the socio-technical system it was evident that some information could be highly distributed and unreliable. Simulations show that when controllers assume a certain value to be high, that it results in a significant increase in number of interactions and task load for the controllers. The simulations also show that uncertainty could lead to a different decision-making processes and unnecessary deployment of costly recovery strategies including utilization of reserve resources.

Castro et al. (2014) identified that there are difficulties in automating decision-making process when there is a degree of uncertainty in the system and that the human still plays a major role in these circumstances. However, the effect of these uncertainties on the choice of recovery strategy has not been studied. In the current study it has been demonstrated that uncertainties and the way the controllers cope with these uncertainties play a significant role in the choice of recovery strategy, but also affect the decision-making process in terms of tasks to be done and interactions that take place. By modelling these uncertainties it could be demonstrated what its sensitivities and implications on AOC objectives are.

8.1.3 Decision outcome

AOC weighs objectives when a decision is made regarding a given disruption. However, these objectives can be conflicting or ambiguous (Feigh, 2008). Additionally, Operations Controllers are not given explicit guidance of the relative importance of the different objectives (Peters, 2006). In the current study an inventory is made of the AOC objectives described in the literature. These objectives are categorized into (1) Execute schedule, (2) Deliver high customer service level, (3) Utilize resources effectively and (4) Minimize costs. With the simulation results it can be determined what the implications of the decision outcome are in towards the objectives and decision-making processes.

By simulating the Operations Controller with a predefined priority of objectives it can be concluded that AOC objectives can be highly coupled. This study shows that in some cases decisions will lead to increased customer service level, but will lead to less effective use of (reserve) resources. Conversely, in another case this study has demonstrated that customer service level is traded in to get the aircraft back to plan so that it can execute its schedule. The choice for a recovery strategy is not in all cases a compromise between objectives. It is also demonstrated that certain recovery strategies can have positive impact in terms of customer service level and executing the schedule with little direct cost. This depends on the available recovery opportunities and the favourability of certain time parameters.

This study has identified two main tasks of the Operations Controller: coordination and formulation of recovery strategies. The coordination task includes requesting information from the supporting group. During design it is observed that these information requests from the Operations Controller depend on his priority of the objectives. This implies that the relative importance of the objective of the Operations Controller has influence on coordination tasks towards other controllers. This can also be verified by the fact that the controllers responded differently towards the scenario in Bruce (2011), some commented mainly about looking for opportunities to repair the aircraft while other commented mainly about looking for rebooking of passengers.

The Operations Controller is the main responsible decision-maker to meet AOC objectives. However, this study shows that conflict of interest could arise during the decision-making process. For instance, the aircraft controller (responsible for aircraft schedule feasibility) could be requested to free up a reserve aircraft to bring passengers back while the aircraft controller could be more interested in ensuring his aircraft schedule feasibility than to bring passengers back.

This study shows that by simulating a decision-making process, it can be evaluated whether an Operations Controller is confronted with competing objectives and it can be identified how the relative importance of these objectives affect the choice of recovery strategy Additionally, by simulating this decision-making process it can be evaluated what the implications are when opting for an alternative recovery strategy.

8.1.4 Closing remarks

Klein (1999) describes that when decision-maker construct mental simulations before making a decision that it could improve their decision-making performance. With models like the one in this study, controllers could use this for training purposes to enhance their mental simulation skills by observing the sensitivities of their judgements and assumption on certain AOC objectives. Additionally, due to the prospective nature of the model, newly designed AOC protocols can be simulated and evaluated before being implemented to look for redundancies or potential hazards i.e. using a pre-mortem managerial strategy.

The conclusions presented in this study should provide more insights into evaluation of decision-making processes in AOC. By modelling and simulations of the decision-making process it could enhance the awareness of airline in terms of why certain operating costs are made or why reserve resources have been utilized. The most important conclusion that can be drawn from this study is that effective evaluation of decision-making processes should be done using a holistic approach i.e. considering aspect like robustness of the schedule, relative importance of the (sometimes) competing objectives, uncertainties and the available means to overcome these uncertainties.

Even though the usefulness of this model has been demonstrated in this study, there are some limitations and numerous refinements and recommendations for further research.

8.2 Limitations & Recommendations for further research

The most important limitation regarding this study is the lack of validation of the decision-making process. Having real-life protocols would have increased the fidelity of the model. Other aspects that limits this study is the fact that one specific scenario is selected with the controllers choosing a recovery strategy on a gate-to-gate level. In real-life there are numerous scenarios possible under different uncertainties and decision considerations including decision considerations on a network level.

Modelling humans is a challenging task since they have soft factors that add complexity to the system. However, at the right abstraction level and the available decision-making models it could be very well possible to enhance the understanding of decision-makers individually as well as collectively. The literature stresses that there are numerous aspects of decision-making that affect decision-outcome and decision quality. Interesting future studies about decision-making in AOC would include researching:

- different scenarios in which different disruptions are being introduced to see correlations with the decision-making process of this study and to validate these scenarios with real-life protocols that describe lines of communication, decision-making procedures, rules and corporate culture
- Adding scenario objectives that would strive for increasing passenger punctuality
- the effect of experience and expertise on the decision-outcome and its effect on decision-making performance of the entire AOC by designing controllers that would produce more decision considerations on a different level
- information supply to overcome uncertainties and the time pressure that limits collecting this information
- the effect of a lack of situation awareness and coordination in case of agents acting in parallel and its influence on recovery strategies
- the effect of competing objectives more thoroughly by designing different Operations Controller ‘profiles’ with each having its own relative importance in terms of objectives and to see its effect on coordination and tasks performed by the supporting group
- decision-making processes on a network level to the study the propagation of decisions into the entire airline schedule

References

ABDELGHANY, K., ABDELGHANY, A. & EKOLLU, G. 2007. An integrated decision support tool for airlines schedule recovery during irregular operations. *European journal of operational research,* 185**,** 825-848.

AIR FRANCE-KLM 2015. Annual Report 2015. Paris.

BALL, M., BARNHART, C., NEMHAUSER, G. & ODONI, A. 2006. Air Transportation: Irregular Operations and Control. *Handbook in OR & MS,* 14.

BANKES, S. 2002. Agent-based modeling: A revolution? *PNAS,* 99**,** 2.

BARNHART, C. & BRATU, S. 2006. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling,* 9**,** 297-298.

BELOBABA, P., ODONI, A. & BARNHART, C. 2009. The Global Airline Industry. West Sussex: Wiley.

BONABEAU, E. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *PNAS,* 99**,** 8.

BORSHCHEV, A. & FILIPPOV, A. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. The 22nd International Conference of the System Dynamics Society, 2004 Oxford, England.

BOSSE, T., JONKER, C., VAN DER MEIJ, L. & TREUR, J. 2007. A language and environment for analysis of dynamics by simulation. *International Journal of Artificial Intelligence Tools,* 16**,** 435-464.

BOSSE, T. & MOGLES, N., M., 2013. Studying Aviation Incidents by Agent-Based Simulation and Analysis: A Case Study on an Runway Incursion Incident. *In:* FILIPE, J. (ed.) *International Conference on Agents and Artificial Intelligence.* Barcelona.

BOUARFA, S., BLOM, H. A. P. & CURRAN, R. 2016. Agent-Based Modeling and Simulation of Coordination by Airline Operations Control. *IEEE Transactions on Emerging Topics in Computing,* 4**,** 12.

BRUCE, P. 2011. *Understanding Decision-making Processes in Airline Operations Control,* Surrey, Ashgate publishing company.

CASTRO, A. J. M. & OLIVEIRA, E. 2010. A new concept for disruption managent in airline operations control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* 225.

CASTRO, A. J. M., OLIVEIRA, E. & ROCHA, P., R., 2014. A New Approach for Disruption Management in Airline Operations Control. *Studies in Computational Intelligence,* 562.

CASTRO, A. J. M., ROCHA P. & E., O. 2012. Towards an Autonomous and Intelligent Airline Operations Control. *15th International IEEE Conference on Intelligent Transportation Systems.* Anchorage, Alaska.

CHAN, W., MACAL, M. & SON, Y. Agent-based simulation tutorial - Simulation of emergent behavior and differences between agent-based simualtion and discrete event simulation. *In:* JOHANSSON, B., JAIN, S., MONTOYA-TORRES, J., HUGAN, J., YÜSECAN, E.,, ed. Winter Simulation Conference, 2010 New Jersey. IEEE.

CLARKE, M. D. D. 1998. Irregular Airline Operations: A review of the state-of-the-practice in airline operations control centers. *Journal of Air Transport Management,* 4**,** 67-76.

CLAUSEN, J., LARSEN, A., LARSEN, J. & REZANOVA, N. 2009. Disruption management in the airline industry - Concepts, models and methods. *Computers & Operations Research,* 37**,** 809-821.

EASA 2014. ANNEX II - 'SUBPART FTL
FLIGHT AND DUTY TIME LIMITATIONS AND REST REQUIREMENTS. *In:* AGENCY, E. A. S. (ed.).

ENDSLEY, M., R., 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *The Journal of the Human Factors and Ergonomics Society,* 37**,** 32-64.

EU. 2004. Air passenger rights. Available: http://europa.eu/youreurope/citizens/travel/passenger-rights/air/index_en.htm.

FEIGH, K., M.,. 2008. *Design of cognitive work support systems for airline operations.* Georgia Institute of Technology.

FILAR, A., MANYEM, P. & WHITE, K. 2001. How Airlines and Airports recover from Schedule Perturbations: A Survey. *Annals of Operations Research,* 108**,** 315-333.

GILBERT, N. 2008. *Agent-based models,* Thousand Oaks, Sage Publications, Inc.

GRANDEAU, S., CLARKE, M., D., & MATHAISEL, D. F. X. 1998. The Processes of Airline System Operations Control. *In:* YA, G. (ed.) *Operations Research in the Airline Industry.* New York: Springer.

HOOGENDOORN, M., JONKER, C., M., , VAN MAANEN, P. P. & SHARPANSKYKH, A. 2008. Formal Analysis of empirical traces in incident management. *Reliability Engineering & System Safety,* 93 (2008)**,** 1422-1433.

IGBO, K. E. 2013. Managing Disruptions in Complex Adaptive Systems A Case for resilience in Airline Operations Control. *In:* CHAUDET, H., PELLEGRIN, L., BONNARDEL, N., (ed.) *International Conference on Naturalistic Decision Making.* Marseille, France: Arpege Science Publishing.

JAFARI, N. & ZEGORDI, S. H. 2010. Simultaneous recovery model for aircraft and passengers. *Journal of the Franklin Institute,* 348**,** 1638-1655.

JENNINGS, N. 2000. On agent-based software engineering. *Artificial Intelligence,* 117**,** 277-296.

JONKER, C., M., , SHARPANSKYKH, A., TREUR, J. & YOLUM, P. 2007. A framework for formal modeling and analysis of organizations. *Applied Intelligence,* 27**,** 49-66.

KAHNEMAN, D. & TVERSKY, A. 1982. Variants of uncertainty. *Cognition,* 11**,** 143-157.

KLEIN, G. 1999. *Sources of Power: How People Make Decisions,* Cambridge, Massachusetts, The MIT Press.

KOHL, N., LARSEN, A., LARSEN, J., ROSS, A. & TIOURINE, S. 2007. Airline disruption management - Perspectives, experiences and outlook. *Journal of Air Transport Management,* 13**,** 149-162.

LAPP, M. 2012. *Methods for Improving Robustness and Recovery in Aviation Planning.* Doctor of Philosophy, University of Michigan.

MACAL, C. J. & NORTH, M. J. 2010. Tutorial on agent-based modeling and simulation. *Journal of Simulation,* 4**,** 151-162.

MATHAISEL, D. 1996. Decision support for airline system operations control and irregular operations. *Computers & Operations Research,* 23**,** 1083-1098.

MEDARD, C. P. & SAWHNEY, N. 2006. Airline crew scheduling from planning to operations. *European journal of operational research,* 183**,** 1013-1027.

MILES, M., B.,, HUBERMAN, A., M., & SALDAÑA, J. 2014. Qualitative Data Analysis: A methods Sourcebook. *In:* SALMON, H. (ed.) 3 ed. California: SAGE Publications Ltd.

MINTZBERG, H. & WESTLEY, F. 2001. Decision Making: It's not what you think. *In:* NUTT, P. C., WILSON, D.C., (ed.) *Handbook of Decision Making.* West Sussex: John Wiley & Sons.

NIKOLIC, I. & GHORBANI, A. 2011. A Method for Developing Agent-based Models of Socio-technical Systems. *International Conference on Networking, Sensing and Control.* Delft, The Netherlands: IEEE.

PETERS, P., F., 2006. *Time, Innovation and Mobilities: Travel in Technological Cultures,* New York, Routledge.

PETERSON, E., NEELS, K., BARCZI, N. & GRAHAM, T. 2013. The Economic Cost of Airline Flight Delay. *Journal of Transport Economics and Policy,* 47**,** 107-232.

POPOVA, V. & SHARPANSKYKH, A. 2010. Modeling organizational performance indicators. *Information Systems,* 35**,** 505-527.

ROBERTSON, D. 2005. Agent-Based Modeling Toolkits NetLogo, RePast, and Swarm. *Academy of Management Learning & Education,* 4**,** 525-527.

SHARPANSKYKH, A. & TREUR, J. 2010. A Temporal Trace Language for Formal Modelling and Analysis of Agent Systems. *In:* DASTANI, M., ET AL (ed.) *Specification and Verification of Multi-agent Systems.*

SHAVELL, Z., A., 2001. Effects of Schedule Disruptions on the Economics of Airline Operations. *In:* DONOHUE, G., L., (ed.) *Air Transportation Systems Engineering.* Virgina: American Institute of Aeronautics and Astronautics.

SIEGFRIED, R. 2014. Modeling and Simulation of Complex Systems. *A Framework for Efficient Agent-Based Modeling and Simulation.* Wiesbaden: Springer DE.

SONENSHEIN, S. 2007. The Role of Construction, Intuition, and Justification in Responding to Ethical Issues at work: the sensemaking-intuition model. *Academy of Management Review,* 32**,** 1022-1040.

TSAI, C., I., , KLAYMAN, J. & HASTIE, R. 2008. Effect of amount of information on judgement accuracy and confidence. *Organizational Behavior and Human Decision Processes,* 107**,** 97-105.

VAN DAM, K., NIKOLIC, I. & LUKSZO, Z. 2013. Agent-Based Modelling of Socio-Technical Systems. Dordrecht: Springer Science+Business Media Dordrecht 2013.

VAN DER ZAKEN, G. 2015. *ATO Research* [Online]. TU Delft. Available: http://www.lr.tudelft.nl/nl/organisatie/afdelingen/control-and-operations/section-air-transport-and-operations/research/?binary=%2Fimg%2FThrustcoef002.gif.

WOOLDRIGDE, M., JENNINGS, N. & KINNY D. 2000. The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*.

WU, C. L. 2010. *Airline Operations and Delay Management: Insights from Airline Economics, Networks and Strategic Schedule Planning,* Farnham, England, Ashgate.

Appendix A

Notes about Interview with an KLM AOC director

Date of meeting 06/05/2013
Time of meeting 10:00 – 12:00
Location of meeting KLM OCC (Schiphol Oost)
Attendance

Name	Title	Organization
Arjen Blom	Director of emergency management*	Air-France KLM
Soufiane Bouarfa	PhD Student Air Transport & Operations	TU Delft
Kamal Belhadji	MSc Student Air Transport & Operations	TU Delft

*The emergency management department within the KLM OCC is involved whenever there is a disaster involving a KLM aircraft anywhere in the world.

OCC environment

The OCC environment consists of a central front office located at the main floor. Other units are located on both sides of the front office on the first floor (emergency unit on the right side, management units on the left side). Operators at the front office work in an open space using different displays that assist them during the decision-making process. The lay out of the main floor is configured in such a way that the divisions who need to interact more often are close to each other. E.g. the operations controllers and duty manager operations are close to each other in the front rows, whereas the flight planners and technical specialists are in the back rows. In the front of the main floor there is a large board which displays relevant information such as wind direction and speed at the hub airport. The screen also shows flights with major delays that need OCC response.

OCC responsibility

The OCC is responsible for controlling the flight operations of the KLM fleet according to the schedule. The schedule is received by the OCC at 17:00 o'clock before the day of operation. The schedule originates from both a long-term and medium-term planning. The long-term planning starts seasons ahead and is done twice a year. The medium-term planning starts from 2 to 5 weeks before the flight depending on the type of resource being scheduled. E.g. crew scheduling happens often 5 weeks before the operation. Once printed, it is unlikely that KLM will change it since it would be expensive (roster commitment).

OCC Operators

The front office consists out of several divisions:

- Senior Operations Controllers
- Duty Manager Operations
- Technical Specialists
- Flight Dispatcher/load controllers
- Crew Controller
- Station Controller
- Passenger rebooking/accommodation
- Other divisions (e.g. management units)

Controllers shift

The front office operates 24/7 in three shifts of 9 hours. At the beginning of each shift, there is a handover session which includes a half hour overlap. In this half hour, a briefing takes place between the duty manager and controllers from the different divisions. After the so-called handshake, the senior operations controller becomes the main responsible in terms of decision-making. The duty manager takes actions in case of escalations (e.g. when a certain flight has been cancelled). Arjen agrees that the beginning of a shift is important for the controller to build up his situation awareness.

Decision Support Systems

The decision support systems were designed based on feedback from experienced controllers. Therefore, experts are involved in trainings dedicated to new controllers who need to use these tools. Arjen recognizes the limitations of these tools, since they only solve part of the operational problem. Operators use these systems merely as guidance, and weighs different aspects before making a decision. Therefore, different operations controllers have different decision-making strategies. According to Arjen, some might use spare aircraft/crew in less urgent circumstances, while others would use them only in restricting situations.

Decision-making styles
Even though there are protocols, controllers have freedom in their decision-making. Arjen mentions again the difference in usage of spare aircraft/crew among the shifts

Decision considerations by the controllers
One main focus of the controllers is departure punctuality. Another consideration by the controllers is also fuel efficiency. E.g. if controllers have two aircraft available and have to decide which one would go to New York (7.5 hours flight) and which one to LA (11 hours). The Aircraft that is more fuel efficient would be assigned to the long flight.

Importance of Checklist at KLM OCC
Arjen stresses the importance of checklists in KLM. In the emergency department for instance, when a disaster occurs, the appropriate file is selected depending on the nature, location, and the context of the incident. Each file is made up of checklists that show the actions that need to be undertaken (e.g. which divisions to be contacted). The checklists are divided across multiple operators at the OCC who can carry out part of the tasks in the crisis coordination process. Arjen is very satisfied about this way of working.

Communication
In the OCC, operators communicate face to face (e.g. briefing sessions), or through conference calls (e.g. with the pilot). The OCC also communicates with other external parties such as ATC, ground services, hub controllers, etc. ATC is contacted by the flight dispatcher. Arjen gave an example of interactions between the pilot, OCC, and ATC:
At KLM, there is always a trade-off between payload and fuel. For a long flight to Hong-Kong for instance, an aircraft would have reserve fuel for half an hour. However, if the pilot experiences severe head-wind, he would then contact the OCC. The OCC would then contact ATC to request priority for the flight to be rerouted.

Difficulties to evaluate decision-making
Arjen stresses that since there are multiple operators involved in the process of decision-making and that it is difficult to know exactly who or what lead to a certain decision.

Role of operators in the OCC
Arjen strongly agree that the OCC cannot be fully automated and the human still plays a vital role in airline operations control. He also recognizes the need of having a good situation awareness in order to reduce the chance of operational errors and improve decision-making performance.

Proactive decision-making
Rather than being reactive to different disruptions, KLM also incorporates a proactive approach through predicting events before they happen. An example given by Arjen was the prediction of low-pressure areas 5 days before the day of operations and adjustment of the flight schedule if necessary.

Projects related to OCC
Arjen referred to a “proficiency program” that KLM launched one year ago.

Areas of improvement
According to Arjen, the main challenge is when weather prediction is not accurate and controllers need to make decisions in a limited time. Another concern is the extent of usefulness of robust scheduling and the difficulties to evaluate decisions made in hindsight.

Appendix B

List of comments of the controllers for Scenario A and Scenario B in both Domestic and International Simulations

#	page	Bruce#	Comment description	Code
			Scenario 1 international simulation	
1	75	203	I'd check that the information is correct. Obviously from that point of view I'd ask what their revised estimated time of arrival is, if it is already in flight. And I would monitor that as it goes	Actual times, transit time, curfew
2	75	400	It's just not sufficient to work on block (scheduled flight) time because that wouldn't be absolutely accurate in today's operations. I think we need to look for actual times	transit times, curfew times
3	76	401	Sometimes having additional information just complicates everything	n/a
4	75	306	How are we off for crew hours? Can we extend the crew duty? Have we got any crew laying over? Have you got another crew? What about that inbound crew that operated flight 706? So the crew that has reported needs 12 hours rest?	crew duty time, crew rest, reserve crew
5	76	307	Ok, first I'm going to be looking at the impact that flight 741 being delayed will have on its arrival at Melbourne and its subsequent sector	transit passengers, network disruption
6	76	311	Most of the delay is not the issue for me. The issue for me is the late arrival coming into Melbourne, where the aircraft is going next and where it goes from there	aircraft schedule, network disruption
7	76	405	Of course, with an hour delay, I would not think that the crew would be problem. Of course we'd look into other passenger connections...there may be passengers from Pacific to Melbourne	transit passengers
8	76	502	Would you be able to tell me whether this delay of about an hour is going to give any connection problems for the passengers going to Melbourne or Sydney	transit passengers
9	77	211	I have an input from the aiport manager who is there on the spot for me to glean what is achievable and I would question his timings	transit passengers, curfew times
10	77	211	A 30 minute connection is what twigged as probably unachievable, how accurate are we with an ETD? The consequences of it, I'd be looking at next, but the question is, how accurate is a connection time of 30 minutes for 120 passengers?	transit passengers, transit time
11	77	217	even though the duty manager has told us we're not going to get away until 2345 he's only given us half an hour transit to get these 55 politicians to transit and knowing what a delegation's like, I think they'd need a longer time than 30 minutes	transit passengers
12	77	403	Well now, that means you've got a curfew time. I mean you can arrive in Sydney obviously, but not depart because of the curfew	transit passengers
13	77	502	So I would ask the airport manager again as to what's the best time they can get the passengers from flight 703, to flight 741, providing we can organize parking bays	transit passengers, organize parking bays,
14	77	315	There isn't much of an opportunity to pick up any late running, so it's just a matter of …asking all stations to do their best to get the aircraft turn-around as quickly as possible	transit passengers, turn around processes
15	77	309	Generally, we'd be looking at getting advanced speed crossings, if that was at all achieve on any of the sectors, to minimise the impact	flight speeds
16	78	400	We might want to ask our representatives in Sydney ahead of time, try and find out from the relevant authorities consider giving up curfew dispensation to operate during curfew hours	curfew times
17	78	201	Instead of coming out for an 1130 UTC departure, it maybe a 1300 departure so we get the crewing people to their crew time for the departures that we are going set and what that does is save us any crewing problems	crew duty time
18	78	211	So I start to look at things like, can we go from Pacific, straight to Sydney, and then via Melbourne; in other words come through Sydney first. I appreciate the loads are full, so our ability to carry them through to Sydney and pick up the joining load may be quite hard, but that's the intention	rerouting, curfew
19	78	302	Well the holding of the crew at home base means that until they've reported, their duty doesn't start. So it actually means allows me more ability to utilize this crew and extend them further if, for any reason, we have subsequently got another..delay of some sort. It just gives me that additional time factor	crew duty time
20	79	305	I would check that... and make sure that my tech[nical, pilot] crew hours are on [the trip plan and] that they're not going into disrupt[ion time] or anything, OK, so I go to my crewing person ... and what are [the crew's] hours [limitations]? OK, so that would be fine, crewing wise.	crew duty time
21	79	212	Not a lot [going through my mind] at the moment. Where are we heading with this again? We're running	n/a

22	80	500	[In relation to] this aircraft... [is the] maximum capacity ... 450? I’d just check whether there are any other alternatives that you can use. So the other option is... we could upset the other 350 passengers. What I am concerned [with] is the different sectors ... We have a crew change in Melbourne too? OK, what is the FTI (Flight time interval)? How about the crew hours for this evening? They operate the two sectors. What is the flight time limitation on this?	crew duty time
23	80	205	What you would need there [is to] probably hold the [flight] 742 crew by up to an hour if that was what the delay was ... with Sydney, you [will] possibly be cribbing curfew to get out, so with an hour delay, 2230, it is still on that, but if it went any later you have to start worrying about Sydney curfew. If it got really late and you weren't going to make Sydney curfew, well you might have to look at coming straight [from Pacific] to Sydney and [reversing the routing of the flight] ... going from Pacific to Sydney and then to Melbourne.	crew duty time, rerouting, curfew
24	80	310	He needs an hour on the ground, so that's now a two hour delay for that flight... now that's getting a bit ugly ... making it a two hour delay here. So that pushes [the flight] out... I 'm just extending [the delay] on the chart. Now that really has put the cat amongst the pigeons there. Now [that] just does change how I thought about the previous [information]. There has to be a cut-off somewhere. I 'm not willing to delay this thing indefinitely.	holding next flight
25	81	302	... my point of view [with] that time factor is whether or not I can actually hold the crew at home base ... well, the holding of the crew at home base means that until they've [signed on for duty], their duty doesn't start, so it actually allows me more time; more ability to actually utilize this crew and extend them further... It just gives me that additional time factor; another hour up my sleeve	extend crew duty time
26	82	501	[I would] look for other airlines going down to Melbourne and put the passengers onto that... , if that's the case, then we'd check [that] the crews are OK and we'll need to know what sort of delay to get them across.	rebooking, crew duty time
27	82	213	Based on those times there, eyeballing what we've got, without a calculator or anything like that... it's going to blow out some time for the next day for the flight 741 , so that's the same time as [another aircraft].	Aircraft Schedule
28	83	508	Do we have another flight in Paris? [I'd] get the traffic staff to check other airlines to protect the passengers. [I 'd ask] how many first and business [class passengers] ... and ask them to book [the passengers] [to hotels]	rebooking, accommodation
29	83	305	... Say that's going to be delayed an hour or so straight away I would check the crew hours [and] see I f they can cope with an hour's delay or not. I 'd advise the [passenger handling people] who take care of the on-carriage (onward connections) [then] I 'd go to my aircraft maintenance [controller] and check [the latest information about] this delay	crew duty time, repair time
30	83	504	... The first thing we'd check is that everything is OK to handle these two [flights] both airports, Manila and Taipei. Then we'd have a chat with the crew [to see If] the crew hours [are] OK or not. I f not OK, then we'd have to send crews from home base to pick up the aircraft to [go to] those two airports. Then we'd have passenger services put [the crews] on something special arranged. We would have to alert ground handling	crew duty time, positioning
31	84	203	There'd be slot time implications out of Paris. Because of the European situation, you have to let flight dispatch [and] ATC know [and] you have to obtain the nearest slot [take-off time] out of Paris for that. [For] anywhere in Europe, that is the case	ATC slots
32	84	403	[Regarding the] Melbourne passengers, so [we have] half an hour to spare. [This is] a bit tight. Ah, normally we find out the schedule ... normally we give one hour to get the passengers across. With this running late, this time [we have] about half an hour for this transit - OK you'd get a delay to the flight	transit passengers, holding flight
33	84	214	We'll leave the [flight] 741 as it is. We won't touch that yet. [We’ll] wait and see. There's no point jumping the gun. Let's just see what the final story is from the engineers I think	technical diagnosis
34	84	307	It doesn't appear that there would be any maintenance problems with that running late. So I would... just issue the re-schedule signal	Repair time, requote/delay
35	85	211	So, OK, It's fantastic he can do it, but I am still reticent to say that we are only copping a one hour delay. I would believe it would be more, I wouldn't want to start publishing one hour [which then might become a] rolling delay, I believe [that] the time...published [should be] something achievable. What is an achievable time?	Repair time, requote/delay
36	85	217	...the duty manager has ...only given us half an hour transit [time] to get these 55 politicians [to transfer flights] and knowing what a delegation's like, I think they'd need longer than 30 minutes basically to get the [passengers across]. The ramp might be ok to get the bags „, off pretty quickly but I think to move a group of that size „, and get them on board [the connecting flight] ,„ might take a bit longer than that	transit time
			Scenario 2 international simulation	
37	88	502	The problem is the Paris engineers’ information update. Sometimes the information from our own part is the second-hand information. I would double check with the Paris staff first to see what the answer is.	technical diagnosis

38	88	205	…it looks like we wouldn't press the button on anything else until we had further word from an engineer. It's a regular happening for someone to be told that the aircraft is not going anywhere, and then ten minutes later a magical fix has been found, so we wouldn't take any action at this stage of the game.	technical diagnosis, repair time
39	88	211	So the question that has been raised would also have been to the crewing section – my expertise not being crewing … to ensure that they also are looking and giving to me the best information they can on availability of crews.	crew duty time
40	89	201	Ok … being a port where we don't have any ability to change aircraft, we are basically locked into that delay … basically the information the engineer has given us is … what we would stick to … a two hour delay.	repair time
41	89	311	I'm looking to see if there is any other aircraft anywhere around Paris … There's obviously the London one, but I wouldn't want to amend that at all. They're a full ship anyway. I can't do anything about that.	reserve aircraft, rebooking
42	89	211	I need to know where my crews are … would a one hour delay Paris to Pacific necessarily put them at a point where they exceed the tour of duty?	crew duty time
43	89	405	For now our main consideration is the crew's flight time limitation. Let's say there is another delay. The crew will not be able to operate to their destination.	crew duty time
44	89	305	I'd go to the technical crew desk and ask them if they have any pilots we can call out … we would ask the tech crew desk to have a look at the crew that's in Paris. Have they had the minimum rest requirements? What time can …they report to go?	crew duty time, crew rest
45	90	500	Ok, we have a …problem with the crew hours …what's the maximum discretion they can use?	crew duty time
46	90	400	…can we check whether it's possible to get a crew from a neighbouring station, position them to Paris and thereafter operate from Paris to Pacific in one tour of crew duty time?	positioning crew, reserve crew
47	90	307	OK. I would like to check again with Maintenance that the time they have given us is absolute.	repair time
48	90	315	What I would do is contact our Maintenance Watch people here and ask them to get in touch with the Maintenance people at Paris, just to confirm that is the problem and that the part is available …I'd be concerned that the original indication of a potential two hour delay seems to be now more or less an indefinite or an unknown delay and I want to get the potential … impact of that.	spare parts, repair time
49	91	311	OK, check with Maintenance. Make sure they are aware of the mechanical problem with the aircraft as well. Regarding crewing … what hours, what flight hour limitations have we got? Next we could look at where the passengers are going to once they get (to Pacific)	technical diagnosis, crew duty time, transit passengers
50	91	213	Based on the engineer's decision, you would publish the delay of 0855 on advice that if they can get it ready earlier; we would be looking at going. Then I'd contact …Despatch asking if a faster time can be organised to make up time a) for the passengers and b) for the crew tour of duty. Just trying to think. I'd stay in contact with Maintenance Watch to make sure that they keep us in the loop if they fix the part earlier.	repair time, increase flight speeds, crew duty time
51	91	216	Right … so I'd offer my resources to actually try and help him find where a maintenance part is, because I'm a hunter … I'd go to another airline Maintenance department and put the two of them in touch with each other straight away …I'd ask at this stage for the local people to start looking at other carriers to see what capacity is around. I'd make sure I speak with the captain as soon as the captain checks in.	spare parts, rebooking
52	91	401	If we have this information and …there is no aircraft you can drop in to Paris, we get a telephone line with the duty engineer. We tell him the constraints and the time we have for the delay. We want him to pass his time of serviceability. **If that time is way beyond the crew time,** we would prefer to stand the crew down … but if the serviceability is between five and six hours, the crew can't operate the flight.	reserve aircraft, repair time, crew duty time
53	93	213	Ok. Well, I'd be looking at either … no I wouldn't because that would affect … I was looking or going to look at swapping aircraft to the flight 700/705 aircraft, but that would be pointless because it would start causing problems into London. What I'd be looking for is an aircraft that could pick up the flight out of Paris, but really I thought they were in Pacific, so delete reference to that altogether.	reserve aircraft
54	93	201	I'd certainly be keeping the engineer going with the pump change even though we have lost the original crew to crew tour of duty problems. There is still a change of using the inbound flight 706 crew for a 1200 departure, once they have had their rest. If 1200 UTC is their first time they can operate certainly there is a chance of getting the aircraft out of Paris …even though that is going to be a four hour delay, we have quite a long extended ground time in Pacific. When the aircraft gets back it is on the ground for about 14 hours it looks like.	crew duty time
55	94	212	I guess now it's just a waiting game for more information from maintenance as to whether there's a replacement maintenance part or not. I'd give them a bit of time to sort it out obviously …probably give them say about 15 minutes to make their phone calls and then I would chase up.	spare part

56	94	501	It's basically the same information that we have already. Then there is no point hanging around. The soon we step down (sign off the crews, the sooner we can depart after crew rest.	crew duty time
57	94	310	OK … hydraulic leak. Now that's …in my own mind I'm thinking that could be actually quite a problem. Immediately I'd get the rest of … my group looking at factors. I'd just do a quick brief, tell them what the problem is and seeing if this thing fell over permanently what in terms of long-term effect …	transit passengers, crew duty time
58	95	503	…just the time required to change that part and the other factor …if the weather is not looking too good, then they could probably sit it out for half an hour, the rain's going to pass on and between times to and from the hangar and back again, there's not a lot of opportunity to make up time. So it looks like 1100 is totally unrealistic, so there's really no point in keeping the passengers at the airport for two to three hours then to be told after that the flight's not operating	Repair time, weather, hangar
			Scenario 1 Domestic Simulation	
59	122	206	Immediately, with a 1715 arrival I think, well, how long ... and that would be the question I would be asking of Sydney ... how long are we looking at to tranship them if indeed they do get in at 1715?	Transit passengers
60	122	207	Then the first thing I would be doing is looking at the reservations system and [finding out] what loads have I got [and] what aeroplanes have I got going [from] Sydney [to] Adelaide at that time?	reserve aircraft, rebooking
61	122	208	That's the last flight of the day? Has [another carrier] got any seats? ... Do we have another flight [from] Sydney [to] Adelaide an hour before it? ... I may also ring flight planning [to ask] what kind of high speed plans we can get... and are the crew going to be in Sydney for an on-time departure?	rebooking, increase flight speeds
62	123	209	... I can see a swap that could be done between aircraft ... neither aircraft has got overnight maintenance required. We'd ring Maintenance Scheduling and check [their response to the situation] that if we did do the swap, we'd have NBV in Sydney and NBR in Perth	reserve aircraft, aircraft swap
63	123	313	What I'd be looking at ... is the crew going to Adelaide and back to Sydney? OK, so I'd be checking their hours ... if we have a delay of half an hour, would that be OK?	crew duty time
64	124	314	I would actually check the Sydney to Adelaide flying time and Adelaide to Sydney flying time [and] get a flight plan to see if there was a short time, [then] check the turn-around time in Adelaide, and see what they can do …	flight speeds, turnaround time, transit passengers
65	124	215	Moving disabled people onto an aircraft takes ages. We would have to load them ... we usually do load them before the other passengers ... 25 is a large number and [the transfer] is going to take ages. There is a very serious risk [that] if we did wait for them, we wouldn't get [Flight] 845 back into Sydney [due to the curfew].	transit passengers, transit time
66	124	202	Do they have to go on [Flight] 826? Is there no other Sydney-Adelaide flight with seats available that we can put them on? Either that or can I send them via Melbourne and then across to Adelaide on another service? The next option is, can I send them directly Brisbane-Adelaide ... or can [I] send them again Brisbane-Melbourne-Adelaide?	rebooking
67	125	206	I'd be looking for another aeroplane in Adelaide that wouldn't be curfew affected. Now, if we go down the board, you have got NBW into Adelaide [at] 1945 [which then] goes Adelaide-Melbourne at 2020. I would be asking Crewing whether they could use that crew to come Adelaide-Sydney	Reserve aircraft, Crew swap
68	125	101	Well, it would affect our delay on [Flight] 826 which means we possibly couldn't do It. We could perhaps get a dispensation under extenuating circumstances from Sydney [authorities] to get back into Sydney	Curfew
69	125	206	So a scheduled number two engine change on that ... so then I have got to go cap my hand to Maintenance and say, well... if we are in a curfew situation with NBR, how are you going to feel about me throwing out the ... engine change on NBW?	Reserve aircraft
70	125	105	Obviously the first thing is [that] you need to confirm times out of Brisbane and times into Sydney and obviously probably more so with Sydney with regards to the holding. Now obviously first they are going to hold [the 'flight or connection] and that's confirmed.	Holding flight
71	126	104	This one's out at 1845 and it's coming in when? Trying to have a look to see what's going where. That's the only Sydney [to] Adelaide flight .. what's our crew doing? And what are they doing the next day? So that's the old curfew back ... How many passengers are on [flight] 845? [are there] any other operators with a flight? I honestly don't know where I 'm going with this.	Crew duty time, holding flight, rebooking
72	126	207	... do we hold for the 70 passengers and the 25 wheelchairs? [With a] 1715 arrival [at the] international [terminal for an] 1805 [departure at the domestic terminal] there is no way they can do that in 50 minutes ... you've got a Sydney curfew problem at 2230 with the aeroplane coming back [from Adelaide] ... the first thing I would be doing is looking at the reservations system [to tell] me what loads have I got going Sydney to Adelaide at that time	Holding flight, curfew, rebooking

73	127	208	At this stage for a 50 minute tranship [time] I 'm not going to waste a whole heap of time ... worrying about it because at this stage it's OK. If [the international flight leaves Brisbane] later than 1530, that's when I 'm concerned. Just my first thought would be ... well can we cancel something else and put on a supplementary flight? That would be one thing I would be looking at in the future.	Transit passengers, reserve aircraft, Reserve crew
74	127	206	So there is not really the option of [getting the passengers] off in Brisbane and sending them some other way to Adelaide, so obviously what the next question would be if we run [flight] 826 that they are connecting with 25 minutes late and we take a standard turnaround in Adelaide, then we're only going to run 20 minutes late back into Sydney. We should be able to make curfew [at] 2250, so I would be asking Crewing what the crew are doing	rebooking, turnaround
75	128	312	Well, I would always leave [the decision] to as late as possible. I mean we want to get these passengers to Adelaide tonight. We don't want to put 70 people in the hotel. You can't take the [able] ones and leave the disabled ones behind. I mean that shouldn't really be an option.	accommodation
76	129	102	Going right back to my initial thought - my gut feeling - [with] 70 passengers, it's going to take a while to offload them, but I think 70 out of 150 is a significant amount of people. I f it was ten and we could do that other thing, I would have gone with that. But with 70, it's like half the flight so I think you've got to hold.	holding next flight
77	129	313	You've got to figure out [that] if you're not going to get back into Sydney [due to the curfew], then you'll have bigger problems in the morning. I'd have a cut-off point, but I think with half an hour extra, they could probably do that [connection time].	transit passengers, curfew
			Scenario 2 Domestic Simulation	
78	132	206	Well I suppose I'd be immediately on the phone to Maintenance in Melbourne to find out exactly what was looked at in Melbourne ...	technical diagnosis
79	132	103	What was the synopsis on the initial problem? It's going to be grounded. Are there [maintenance] parts available? I 'd ask this of the engineers	technical diagnosis, spare parts
80	132	202	Are we talking ten minutes? Are we talking hours? Is there a part involved that we need to now replace because it didn't work the first time...? Is that part available in Canberra? If not, how soon can we get a part up into Canberra? Is there another flight going to Canberra out of Sydney? Maybe we can get a part out of Sydney We don't have to get it out of Melbourne	spare parts, positioning resources
81	132	206	... My initial question would be what capacity have we got Sydney [to] Canberra, to get the passengers back to Canberra?	Rebooking, rerouting
82	132	301	What other resource do we have to try and get these people away? What other flights do we have available from Melbourne [to] Canberra for the remainder of the day with seats available?	Rebooking
83	132	304	Do any of these crews actually go in [to Canberra] and then turn around and come out the next morning?	Crew schedule
84	133	210	I guess my first thought was Ok, get this thing [flight 876] back. We are going to lose it for the day What then is my next available aeroplane? I've gone straight to NBV which is available. However, it is doing a 1910 service to Adelaide for which I don't have another aeroplane	Reserve aircraft
85	133	210	I'm just thinking... [Flight] 879 Melbourne to Hobart is going to be late whatever we do. Is the tour of duty for both crew[s] sufficient, so [that] we can delay the 879 Melbourne to Hobart [service]?	Crew duty time
86	133	208	I 'm thinking that if we sent the aircraft to Canberra ... then our problem is in Canberra which would be [Flight] 879 ... [which has] 130 people on it. A quick glance down the [display] to see if we have got another [flight] going an hour later ... [Flight 889] is half full. That means we can get 70 [passengers on Flight 889, so] ... we've got 60 people left. Do we have any other Canberra to Melbourne [flights]?	Subsequent sectors, rebooking
87	134	301	We would have confidence in Maintenance watch and [would] get the aircraft [crew] to speak to maintenance watch. Now I'd be asking them ... [as] they know the defect: 'what's your prognosis?' Obviously if it's going to Canberra, and the [captain] reckons he's going to be [operationally grounded] there, have we got parts there ... with an engineer ... to get [the aircraft] serviceable?	Technical diagnosis, spare parts
88	134	202	... [I'd also be] communicating with Melbourne [and asking] ... would it be an easy fix? Are we talking ten minutes, are we talking hours, is there a part involved that we need to now replace because it didn't work the first time. Is there a part available in Canberra? I if not, how soon can we get a part up [there]?	Spare parts, positioning
89	134	206	So I would be looking at... how many seats have we got Canberra to Sydney at about the same time? [At] 1920 Canberra to Sydney, you have got 50 seats. Does that connect with anything Sydney to Adelaide?	Positioning parts

90	134	209	I am also looking at the loadings to see if there's anything that has a light load that we could perhaps cancel, and [then we could] combine [flights].	Reserve aircraft
91	135	204	Could we create the circumstances [where] we can even swap with a wide body [aircraft] to free up an aircraft that will allow us to go Melbourne [then] Launceston [to] Hobart and probably back the same way?	Reserve aircraft
92	135	105	The engineers told me four hours, but it could be half an hour The guys could get in there and say, look ... it could be half an hour [to fix].	Repair time
93	135	207	What I am saying is ... if it's only an instrument indication problem it could be a loose wire when they changed the instrument around. The guy gets in and [finds] the plug's come loose, just twigs it, and the aeroplane goes. [I f that were the case] we [would have seen] it turns back [to Melbourne] with 150 people on it, or no reason whatsoever	Repair time, technical diagnosis
94	135	101	From what I've experienced, [the time of serviceability] can also go the other way, where a one hour fix can blow out to four hours.	Repair time
95	135	105	We've got a bit of time to play with, not much - sort of an hour or so. We can go through and have an assessment as to what we can do, have a look at what we've got and get whoever is on with me perhaps to do a brainstorm [to] see what we can come up with. [We could also] get suggestions from the guys actually on the front there and see what we can do. But at this stage that's my quick fix	Technical diagnosis, spare parts
96	136	104	I'm just looking for another flight out of Melbourne [to operate to] Sydney and hopefully connect this to Canberra from Sydney ... [but] I don't know if there is one there. So, I've got rid of 70 [passengers] haven't I? So, I've got to get rid of another 60. If I could get them to Sydney; I could possibly put them on [Flight] 849 if I could get them away. He's probably already gone hasn't he? That's right he has too. Possibly I'd be over-nighting the 60 others maybe. I don't know, not having much experience.	Rebooking, accommodating
97	136	210	Where's my Sydney [to] Adelaide ... doesn't help. I would have explored the possibility of sending the Adelaide [passengers] via Sydney actually on [Flight] 888 [as] there [are] probably enough seats on that. That doesn't appear to work. I think I would be looking to put... [Flight 840] on the next available aeroplane which appears to be NBN, coming off Flight 839 from Adelaide at 2010. That would be an hour and a half delay on [Flight] 840. [The] crew has sufficient hours for that.	Rebooking, Crew duty time
98	136	210	Now that aircraft is supposed to be in Sydney for an APU rectification. Given the gravity of the problem, I'd be negotiating with Maintenance for another slot on that APU.	Aircraft schedule
99	137	104	1850, so we've got heaps of passengers to move. It is huge. Right we can offload a few to [Flight 889] here.	Rebooking
100	137	304	.. OK, maybe ... if we decided that we want to cancel that flight, I'd be looking at getting some capacity between Melbourne and Sydney because we've got a Flight 849 that is going Sydney [to] Canberra.	Rebooking
101	138	101	Alright then, we have a major disruption ... we have got the engineers organized and Melbourne airport advised regarding slot availability ... There is no emergency. [This has been] established from the crew. The approach and landing will be normal... we then have to look after the 150 people that are on the ground with [Flight] 876 and 130 coming out of Canberra [for] Melbourne.	n/a
102	139	314	... I 'm now looking to see how to move the people out of Canberra and out of Melbourne. I would return that aircraft and I would ask them to rectify the aircraft as soon as possible and I would continue with the flight…	n/a
103	139	208	Canberra aren't going to like this very much because I 'm splitting the load. They would probably want me to run [Flight] 879 off the back of 889 and take the delay on 889 till later which is 2030. I think 1 will go with my original way at this stage [as] I have got a couple of options open to me depending on what the port want to do ...	Rebooking
104	139	304	Is there any way we can get him to Sydney? OK, we've only got one flight there but that's carrying many [passengers]. Can it be fixed in Sydney? [Are there] parts [in Sydney]? [I 'm just trying] to see whether it's going to be worth the risk or not. Sometimes it's gut feeling. Sometimes it [depends] on who the captain is.	Spare parts, technical diagnosis, positioning technicians

Appendix C

Leadsto-code

```
content(type(save_lt_editor('c:/Users/KamalB/Desktop/Thesis 8.0/Model0.6/model0.8'))).
content(generator(app(leadsto_software, 127, [lteditor:1, psprinting:1]))).
content(run([date('Mon Mar 21 11:09:47 2016')])).

/*multi_trace_section*/
model(model_name('F':'POS','G':'RCREW','H':'RBOOK','I':'RAC','R':'RT','C':'CT','P':'PT')).
/*'A':'DIAG','B':'PART',,,,,'H':'RBOOK',','F':'DEAD',,,'E':'CONM',,dt:between(2,3),rt:between(2,3),,dt:between(2,3)'C':'WX',,'T':'DTIME','D':'HANG','E':'CONM',,*/
sortdef('DIAG',[a_0,a_1]).  /*A*/
sortdef('PART',[b_0,b_1]).     /*B*/
sortdef('WX',[c_0,c_1]).                          /*C*/
sortdef('HANG',[d_0,d_1]).  /*D*/
sortdef('CONM',[e_0,e_1]).   /*E*/
sortdef('POS',[f_0,f_1]).        /*F*/
sortdef('RCREW',[g_0,g_1]).  /*G*/
sortdef('RBOOK',[h_0,h_1]).  /*H*/
sortdef('RAC',[i_0,i_1]). /*I*/
sortdef('RT',[120,125,130,135,140,145,150,155,160,165,170,175,180,185,190,195,200,205,210,215,220,225,230,235,240]).
sortdef('CT',[175,180,185]).
sortdef('PT',[175,180,185]).
/*multi-trace-inputs */
interval([],      range(0,        k                  ),                 a_1              ).
interval([],      range(0,        k                  ),                 b_1              ).
interval([],      range(0,        k                  ),                 c_1              ).
interval([],      range(0,        k                  ),                 d_0              ).
interval([],      range(0,        k                  ),                 e_0              ).
interval([],      range(0,        k                  ),                 'F'              ).
interval([],      range(0,        k                  ),                 'G').
interval([],      range(0,        k                  ),                 'H'              ).
interval([],      range(0,        k                  ),                 'I'              ).
interval([],      range(0,        k                  ),                 rt_('R')         ).
interval([],      range(0,        k                  ),                 dt_(180)).
interval([],      range(0,        k                  ),                 bt_(180)         ).
interval([],      range(0,        k                  ),                 kt_(180)         ).
interval([],      range(0,        k                  ),                 ct_('C')         ).
interval([],      range(0,        k                  ),                 pt_('P')         ).
constant(k,90).
end_time(95).

/*sorts*/
sortdef('MSG_TYPE',[inform,request,observe]).
sortdef('PHASE',['3A1','3A2','3A3','3B1','3B2','x']).
sortdef('CTRL',[ac,oc,fd,cc,sc]).
sortdef('MSG',[aircraft_tua_has_mechanical_problem,technical_diagnosis_adequateness,technical_diagnosis_adequate,technical_diagnosis_inadequate,spare_parts_availability,spare_p
arts_available,spare_parts_unavailable,wx_pattern_favourability,wx_pattern_favourable,wx_pattern_unfavourable,hangar_availability,hangar_available,hangar_unavailable,reserve_air
craft_availability,reserve_aircraft_available,reserve_aircraft_unavailable,dispatch_reserve_aircraft_connects_tpax_successful,dispatch_reserve_aircraft_connects_tpax_unsuccessful,org
anizing_cxn_measures_possibilities,organizing_cxn_measures_possible,organizing_cxn_measures_not_possible,effect_of_rt_on_crew,crew_unaffected_by_rt,reserve_crew_availability,r
eserve_crew_available,reserve_crew_unavailable,deadheading_possibilities,deadheading_possible,deadheading_not_possible,deadheading_crew_connects_tpax_successful,deadheadin
g_crew_connects_tpax_unsuccessful,
effect_of_rt_on_tpax,effect_of_dt_on_tpax,effect_of_kt_on_tpax,tpax_unaffected_by_rt,tpax_affected_by_rt,rebooking_possibilities,rebooking_possible,rebooking_not_possible,reboo
king_connects_tpax_successful,rebooking_connects_tpax_unsuccessful,repair_time,rebooking_time,transit_slack_time,transit_slack_time_crt,transit_slack_time_cdt,transit_slack_time
_cbt,transit_slack_time_ckt,crew_duty_slack_time,ferry_time,deadheading_time,deadheading_resources_possibilities,deadheading_resources_possible,deadheading_resources_not_po
ssible]).
sortdef('RS',['RS1','RS2','RS3','RS4','RS5','RS6','RS7','RS8','RS9','RS10','RS11','RS12','RS13','RS14','RS15','RS16','RS17','RS18','RS19','RS20']).


/*start*/
interval([],range(0,1),input(fd)|com(env,fd,observe,aircraft_tua_has_mechanical_problem,delay(x1,x2,x3,x4),p(x))).

/*ROLE PROPERTIES*/
leadsto([],input(fd)|com(env,fd,observe,aircraft_tua_has_mechanical_problem,delay(x1,x2,x3,x4),p(x)),output(fd)|com(fd,ac,inform,aircraft_tua_has_mechanical_problem,delay(x1,x2,x
3,x4),p(x)),standard).

/*AC-phase1*/
leadsto([],input(ac)|com(fd,ac,inform,aircraft_tua_has_mechanical_problem,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,env,request,technical_diagnosis_adequateness,delay(x1,x2,x3,x4
),p(x)),standard).
leadsto([],input(ac)|com(env,ac,observe,technical_diagnosis_adequate,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,env,request,spare_parts_availability,delay(x1,x2,x3,x4),p(x)),standard)
.
leadsto([],input(ac)|com(env,ac,observe,technical_diagnosis_inadequate,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,oc,inform,technical_diagnosis_inadequate,delay(x1,x2,x3,x4),p(x)),st
andard).
leadsto([],input(ac)|com(env,ac,observe,spare_parts_available,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,fd,request,wx_pattern_favourability,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(ac)|com(env,ac,observe,spare_parts_unavailable,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,oc,inform,spare_parts_unavailable,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(ac)|com(fd,ac,inform,wx_pattern_favourable,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,env,request,repair_time,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(ac)|com(fd,ac,inform,wx_pattern_unfavourable,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,env,request,hangar_availability,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(ac)|com(env,ac,observe,hangar_available,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,env,request,repair_time,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(ac)|com(env,ac,observe,hangar_unavailable,delay(x1,x2,x3,x4),p(x)),output(ac)|com(ac,oc,inform,hangar_unavailable,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([rt:integer],input(ac)|com(env,ac,observe,repair_time(rt),delay(rt,x2,x3,x4),p(x)),output(ac)|com(ac,oc,inform,repair_time(rt),delay(rt,x2,x3,x4),p(x)),standard).
/*AC-phase'3A1'*/
leadsto([rt:integer],input(ac)|com(oc,ac,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A1')),output(ac)|com(ac,env,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('
3A1')),standard).
leadsto([rt:integer],input(ac)|com(env,ac,observe,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p('3A1')),output(ac)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p(
'3A1')),standard).
leadsto([rt:integer],input(ac)|com(env,ac,observe,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A1')),output(ac)|com(ac,fd,inform,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A1'
)),standard).
/*AC-phase3A3*/
leadsto([rt:integer],input(ac)|com(oc,ac,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A3')),output(ac)|com(ac,env,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('
3A3')),standard).
leadsto([rt:integer],input(ac)|com(env,ac,observe,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p('3A3')),output(ac)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p(
'3A3')),standard).
leadsto([rt:integer],input(ac)|com(env,ac,observe,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A3')),output(ac)|com(ac,fd,inform,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A3'
```

```
)),standard).
/*AC-phase3B1*/
leadsto([],input(ac)|com(oc,ac,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B1')),output(ac)|com(ac,env,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(ac)|com(env,ac,observe,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B1')),output(ac)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(ac)|com(env,ac,observe,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B1')),output(ac)|com(ac,fd,inform,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B1')),standard).
/*AC-phase3B2*/
leadsto([],input(ac)|com(oc,ac,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B2')),output(ac)|com(ac,env,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],input(ac)|com(env,ac,observe,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B2')),output(ac)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],input(ac)|com(env,ac,observe,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B2')),output(ac)|com(ac,fd,inform,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B2')),standard).
/*FD-phase1*/
leadsto([],input(fd)|com(ac,fd,request,wx_pattern_favourability,delay(x1,x2,x3,x4),p(x)),output(fd)|com(fd,env,request,wx_pattern_favourability,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(fd)|com(env,fd,observe,wx_pattern_favourable,delay(x1,x2,x3,x4),p(x)),output(fd)|com(fd,ac,inform,wx_pattern_favourable,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(fd)|com(env,fd,observe,wx_pattern_unfavourable,delay(x1,x2,x3,x4),p(x)),output(fd)|com(fd,ac,inform,wx_pattern_unfavourable,delay(x1,x2,x3,x4),p(x)),standard).
/*FD-phase2A*/
leadsto([rt:integer],input(fd)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,x4),p(x)),output(fd)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_possible,delay(rt,x2,x3,x4),p(x)),output(fd)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_not_possible,delay(rt,x2,x3,x4),p(x)),output(fd)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,x4),p(x)),standard).
/*FD-phase'3A1'*/
leadsto([rt:integer],input(fd)|com(ac,fd,inform,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A1')),output(fd)|com(fd,env,request,ferry_time,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(env,fd,observe,ferry_time(kt),delay(rt,x2,x3,kt),p('3A1')),output(fd)|com(fd,oc,inform,ferry_time(kt),delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A1')),output(fd)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A1')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A1')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A1')),standard).
/*FD-phase3A2*/
leadsto([rt:integer,dt:integer],input(fd)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,dt,x3,x4),p('3A2')),output(fd)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_possible,delay(rt,dt,x3,x4),p('3A2')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_not_possible,delay(rt,dt,x3,x4),p('3A2')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
/*FD-phase3A3*/
leadsto([rt:integer],input(fd)|com(ac,fd,inform,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A3')),output(fd)|com(fd,env,request,ferry_time,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(env,fd,observe,ferry_time(kt),delay(rt,x2,x3,kt),p('3A3')),output(fd)|com(fd,oc,inform,ferry_time(kt),delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A3')),output(fd)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A3')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A3')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A3')),standard).
/*FD-phase3B1*/
leadsto([],input(fd)|com(ac,fd,inform,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B1')),output(fd)|com(fd,env,request,ferry_time,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([kt:integer],input(fd)|com(env,fd,observe,ferry_time(kt),delay(x1,x2,x3,kt),p('3B1')),output(fd)|com(fd,oc,inform,ferry_time(kt),delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],input(fd)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B1')),output(fd)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B1')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B1')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B1')),standard).
/*FD-phase3B2*/
leadsto([],input(fd)|com(ac,fd,inform,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B2')),output(fd)|com(fd,env,request,ferry_time,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([kt:integer],input(fd)|com(env,fd,observe,ferry_time(kt),delay(x1,x2,x3,kt),p('3B2')),output(fd)|com(fd,oc,inform,ferry_time(kt),delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],input(fd)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B2')),output(fd)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B2')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],input(fd)|com(env,fd,observe,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B2')),output(fd)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B2')),standard).
/*OC2A*/
leadsto([rt:integer],input(oc)|com(ac,oc,inform,repair_time(rt),delay(rt,x2,x3,x4),p(x)),and(output(oc)|com(oc,sc,request,effect_of_rt_on_tpax,delay(rt,x2,x3,x4),p(x)),output(oc)|com(oc,cc,request,effect_of_rt_on_crew,delay(rt,x2,x3,x4),p(x))),standard).
/*OC-Phase3A-1*/
leadsto([rt:integer],and(input(oc)|com(cc,oc,inform,crew_unaffected_by_rt,delay(rt,x2,x3,x4),p('3A1')),input(oc)|com(sc,oc,inform,tpax_unaffected_by_rt,delay(rt,x2,x3,x4),p(x))),recovery('RS1',rt,rt,x2,x3,x4),standard).
leadsto([rt:integer],and(input(oc)|com(cc,oc,inform,crew_unaffected_by_rt,delay(rt,x2,x3,x4),p('3A1')),input(oc)|com(sc,oc,inform,tpax_affected_by_rt,delay(rt,x2,x3,x4),p(x))),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer,bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(rt,x2,bt,x4),p('3A1')),recovery('RS4',bt,rt,x2,bt,x4),standard).
leadsto([rt:integer],input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A1')),and(output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A1')),output(oc)|com(oc,cc,request,reserve_crew_availability,delay(rt,x2,x3,x4),p('3A1'))),standard).
leadsto([rt:integer,bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A1')),and(output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A1')),output(oc)|com(oc,cc,request,reserve_crew_availability,delay(rt,x2,x3,x4),p('3A1'))),standard).
leadsto([rt:integer],input(oc)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p('3A1')),recovery('RS2',rt,rt,x2,x3,x4),standard).
leadsto([rt:integer],input(oc)|com(cc,oc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A1')),recovery('RS2',rt,rt,x2,x3,x4),standard).
leadsto([rt:integer,kt:integer],and(input(oc)|com(fd,oc,inform,ferry_time(kt),delay(rt,x2,x3,kt),p('3A1')),input(oc)|com(cc,oc,inform,reserve_crew_available,delay(rt,x2,x3,x4),p('3A1'))),output(oc)|com(oc,sc,request,effect_of_kt_on_tpax,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(rt,x2,x3,kt),p('3A1')),recovery('RS8',kt,rt,x2,x3,kt),standard).
leadsto([rt:integer,kt:integer],input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(rt,x2,x3,kt),p('3A1')),recovery('RS2',rt,rt,x2,x3,kt),standard).
/*OC-Phase3A-2*/
leadsto([rt:integer,dt:integer],and(input(oc)|com(cc,oc,inform,deadheading_time(dt),delay(rt,dt,x3,x4),p('3A2')),input(oc)|com(sc,oc,inform,tpax_unaffected_by_rt,delay(rt,x2,x3,x4),p(x))),output(oc)|com(oc,sc,request,effect_of_dt_on_tpax,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],and(input(oc)|com(cc,oc,inform,deadheading_time(dt),delay(rt,dt,x3,x4),p('3A2')),input(oc)|com(sc,oc,inform,tpax_affected_by_rt,delay(rt,x2,x3,x4),p(x))),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(oc)|com(sc,oc,inform,deadheading_crew_connects_tpax_successful,delay(rt,dt,x3,x4),p('3A2')),recovery('RS7',dt,rt,dt,x3,x4),standard).
```

```
leadsto([rt:integer,dt:integer],input(oc)|com(sc,oc,inform,deadheading_crew_connects_tpax_unsuccessful,delay(rt,dt,x3,x4),p('3A2')),output(oc)|com(oc,sc,request,rebooking_possibilit
ies,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer,bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(rt,dt,bt,x4),p('3A2')),recovery('RS11',bt,rt,dt,bt,x4),standard).
leadsto([rt:integer,dt:integer],input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,dt,x3,x4),p('3A2')),recovery('RS7',dt,rt,dt,bt,kt),standard).
leadsto([rt:integer,dt:integer,bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,dt,bt,x4),p('3A2')),recovery('RS12',bt,rt,dt,bt,x4),standard).
/*OC-Phase3A-3*/
leadsto([rt:integer],input(oc)|com(cc,oc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A3')),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A3')),st
andard).
leadsto([rt:integer],input(oc)|com(cc,oc,inform,deadheading_not_possible,delay(rt,x2,x3,x4),p('3A3')),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A3')),st
andard).
leadsto([rt:integer,bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(rt,x2,bt,x4),p('3A3')),recovery('RS6',bt,rt,x2,bt,x4),standard).
leadsto([rt:integer],and(input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A3')),input(oc)|com(cc,oc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A3'))
),recovery('RS3',x,rt,x2,x3,x4),standard).
leadsto([rt:integer],and(input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A3')),input(oc)|com(cc,oc,inform,deadheading_not_possible,delay(rt,x2,x3,x4),p('3A3')
)),output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer,bt:integer],and(input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A3')),input(oc)|com(cc,oc,inform,reserve_crew_unavailable,
delay(rt,x2,x3,x4),p('3A3'))),recovery('RS5',bt,rt,x2,bt,x4),standard).
leadsto([rt:integer,bt:integer],and(input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A3')),input(oc)|com(cc,oc,inform,deadheading_not_possible
,delay(rt,x2,x3,x4),p('3A3'))),output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer],and(input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A3')),input(oc)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p('3A
3'))),recovery('RS3',x,rt,x2,x3,x4),standard).
leadsto([rt:integer,bt:integer],and(input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A3')),input(oc)|com(ac,oc,inform,reserve_aircraft_unavailab
le,delay(rt,x2,x3,x4),p('3A3'))),recovery('RS5',bt,rt,x2,bt,x4),standard).
leadsto([rt:integer,kt:integer],and(input(oc)|com(fd,oc,inform,ferry_time(kt),delay(rt,x2,x3,kt),p('3A3')),input(oc)|com(cc,oc,inform,deadheading_not_possible,delay(rt,x2,x3,x4),p('3A3')
)),output(oc)|com(oc,sc,request,effect_of_kt_on_tpax,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,bt:integer,kt:integer],and(input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(rt,x2,x3,kt),p('3A3')),input(oc)|com(sc,oc,inform,r
ebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A3'))),recovery('RS5',bt,rt,x2,bt,x4),standard).
leadsto([rt:integer,kt:integer],and(input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(rt,x2,x3,kt),p('3A3')),input(oc)|com(sc,oc,inform,rebooking_
not_possible,delay(rt,x2,x3,x4),p('3A3'))),recovery('RS9',kt,rt,x2,x3,kt),standard).
leadsto([rt:integer,kt:integer],input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(rt,x2,x3,kt),p('3A3')),recovery('RS10',kt,rt,x2,x3,kt),standard).
/*OC-Phase2B*/
leadsto([],input(oc)|com(ac,oc,inform,technical_diagnosis_inadequate,delay(x1,x2,x3,x4),p(x)),output(oc)|com(oc,sc,request,deadheading_resources_possibilities,delay(x1,x2,x3,x4),p(x)
),standard).
leadsto([],input(oc)|com(ac,oc,inform,spare_parts_unavailable,delay(x1,x2,x3,x4),p(x)),output(oc)|com(oc,sc,request,deadheading_resources_possibilities,delay(x1,x2,x3,x4),p(x)),stand
ard).
/*OC-Phase3B-1*/
leadsto([],input(oc)|com(ac,oc,inform,hangar_unavailable,delay(x1,x2,x3,x4),p('3B1')),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(oc)|com(sc,oc,inform,deadheading_resources_possible,delay(x1,x2,x3,x4),p('3B1')),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B1')),sta
ndard).
leadsto([bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(x1,x2,bt,x4),p('3B1')),recovery('RS17'),standard).
leadsto([],input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B1')),and(output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B1')),out
put(oc)|com(oc,cc,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B1'))),standard).
leadsto([bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(x1,x2,bt,x4),p('3B1')),and(output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay
(x1,x2,x3,x4),p('3B1')),output(oc)|com(oc,cc,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B1'))),standard).
leadsto([],and(input(oc)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B1')),input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B1'))),reco
very('RS14'),standard).
leadsto([],and(input(oc)|com(cc,oc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B1')),input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B1'))),recove
ry('RS14'),standard).
leadsto([bt:integer],and(input(oc)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B1')),input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(x
1,x2,bt,x4),p('3B1'))),recovery('RS18'),standard).
leadsto([bt:integer],and(input(oc)|com(cc,oc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B1')),input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(x1,
x2,bt,x4),p('3B1'))),recovery('RS18'),standard).
leadsto([kt:integer],and(input(oc)|com(fd,oc,inform,ferry_time(kt),delay(x1,x2,x3,kt),p('3B1')),input(oc)|com(cc,oc,inform,reserve_crew_available,delay(x1,x2,x3,x4),p('3B1'))),output(oc
)|com(oc,sc,request,effect_of_kt_on_tpax,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(x1,x2,x3,kt),p('3B1')),recovery('RS15'),standard).
leadsto([kt:integer],and(input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(x1,x2,x3,kt),p('3B1')),input(oc)|com(sc,oc,inform,rebooking_not_possi
ble,delay(x1,x2,x3,x4),p('3B1'))),recovery('RS16'),standard).
leadsto([bt:integer,kt:integer],and(input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(x1,x2,x3,kt),p('3B1')),input(oc)|com(sc,oc,inform,rebooking_
connects_tpax_unsuccessful,delay(x1,x2,bt,x4),p('3B1'))),recovery('RS18'),standard).
/*OC-Phase3B-2*/
leadsto([],input(oc)|com(sc,oc,inform,deadheading_resources_not_possible,delay(x1,x2,x3,x4),p('3B2')),and(output(oc)|com(oc,ac,request,reserve_aircraft_availability,delay(x1,x2,x3,x4
),p('3B2')),output(oc)|com(oc,cc,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B2'))),standard).
leadsto([],input(oc)|com(ac,oc,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B2')),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B2')),standar
d).
leadsto([],input(oc)|com(cc,oc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B2')),output(oc)|com(oc,sc,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(x1,x2,bt,x4),p('3B2')),recovery('RS17'),standard).
leadsto([],input(oc)|com(sc,oc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B2')),recovery('RS13'),standard).
leadsto([bt:integer],input(oc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(x1,x2,bt,x4),p('3B2')),recovery('RS20'),standard).
leadsto([kt:integer],and(input(oc)|com(fd,oc,inform,ferry_time(kt),delay(x1,x2,x3,kt),p('3B2')),input(oc)|com(cc,oc,inform,reserve_crew_available,delay(x1,x2,x3,x4),p('3B2'))),output(oc
)|com(oc,sc,request,effect_of_kt_on_tpax,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(x1,x2,x3,kt),p('3B2')),recovery('RS15'),standard).
leadsto([kt:integer],input(oc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(x1,x2,x3,kt),p('3B2')),recovery('RS16'),standard).
/*CC-Phase2A*/
leadsto([rt:integer],input(cc)|com(oc,cc,request,effect_of_rt_on_crew,delay(rt,x2,x3,x4),p(x)),output(cc)|com(cc,env,request,crew_duty_slack_time,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer,ct:integer],and(input(cc)|com(env,cc,observe,crew_duty_slack_time(ct),delay(rt,x2,x3,x4),p(x)),<=(rt,ct)),output(cc)|com(cc,oc,inform,crew_unaffected_by_rt,delay(rt
,x2,x3,x4),p('3A1')),efgh(0,0,1,25)).
leadsto([rt:integer,ct:integer],and(input(cc)|com(env,cc,observe,crew_duty_slack_time(ct),delay(rt,x2,x3,x4),p(x)),>(rt,ct)),output(cc)|com(cc,env,request,reserve_crew_availability,dela
y(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],input(cc)|com(env,cc,observe,reserve_crew_unavailable,delay(rt,x2,x3,x4),p(x)),output(cc)|com(cc,oc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A3')),ef
gh(0,0,1,45)).
leadsto([rt:integer],input(cc)|com(env,cc,observe,reserve_crew_available,delay(rt,x2,x3,x4),p(x)),output(cc)|com(cc,sc,request,deadheading_possibilities,delay(rt,x2,x3,x4),p(x)),standar
d).
leadsto([rt:integer],input(cc)|com(sc,cc,inform,deadheading_not_possible,delay(rt,x2,x3,x4),p(x)),output(cc)|com(cc,oc,inform,deadheading_not_possible,delay(rt,x2,x3,x4),p('3A3')),efg
h(0,0,1,45)).
leadsto([rt:integer],input(cc)|com(sc,cc,inform,deadheading_possible,delay(rt,x2,x3,x4),p(x)),output(cc)|com(cc,env,request,deadheading_time,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer,dt:integer],input(cc)|com(env,cc,observe,deadheading_time(dt),delay(rt,dt,x3,x4),p(x)),output(cc)|com(cc,oc,inform,deadheading_time(dt),delay(rt,dt,x3,x4),p('3A2'))
,standard).
/*CC-Phase'3A1'*/
leadsto([rt:integer],input(cc)|com(oc,cc,request,reserve_crew_availability,delay(rt,x2,x3,x4),p('3A1')),output(cc)|com(cc,env,request,reserve_crew_availability,delay(rt,x2,x3,x4),p('3A1')
),standard).
leadsto([rt:integer],input(cc)|com(env,cc,observe,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A1')),output(cc)|com(cc,oc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A1'
)),standard).
leadsto([rt:integer],input(cc)|com(env,cc,observe,reserve_crew_available,delay(rt,x2,x3,x4),p('3A1')),output(cc)|com(cc,oc,inform,reserve_crew_available,delay(rt,x2,x3,x4),p('3A1')),efg
h(0,0,1,8)).
```

```
/*CC-Phase3B1*/
leadsto([],input(cc)|com(oc,cc,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B1')),output(cc)|com(cc,env,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(cc)|com(env,cc,observe,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B1')),output(cc)|com(cc,oc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(cc)|com(env,cc,observe,reserve_crew_available,delay(x1,x2,x3,x4),p('3B1')),output(cc)|com(cc,oc,inform,reserve_crew_available,delay(x1,x2,x3,x4),p('3B1')),standard).
/*CC-Phase3B2*/
leadsto([],input(cc)|com(oc,cc,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B2')),output(cc)|com(cc,env,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],input(cc)|com(env,cc,observe,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B2')),output(cc)|com(cc,oc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],input(cc)|com(env,cc,observe,reserve_crew_available,delay(x1,x2,x3,x4),p('3B2')),output(cc)|com(cc,oc,inform,reserve_crew_available,delay(x1,x2,x3,x4),p('3B2')),efgh(0,0,1,12)).
/*SC-phase2A*/
leadsto([rt:integer],input(sc)|com(oc,sc,request,effect_of_rt_on_tpax,delay(rt,x2,x3,x4),p(x)),output(sc)|com(sc,env,request,transit_slack_time_crt,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer,pt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_crt(pt),delay(rt,x2,x3,x4),p(x)),<=(rt,pt)),output(sc)|com(sc,oc,inform,tpax_unaffected_by_rt,delay(rt,x2,x3,x4),p(x)),efgh(0,0,1,35)).
leadsto([rt:integer,pt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_crt(pt),delay(rt,x2,x3,x4),p(x)),>(rt,pt)),output(sc)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,x4),p(x)),output(sc)|com(sc,oc,inform,tpax_unaffected_by_rt,delay(rt,x2,x3,x4),p(x)),efgh(0,0,1,35)).
leadsto([rt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,x4),p(x)),output(sc)|com(sc,oc,inform,tpax_affected_by_rt,delay(rt,x2,x3,x4),p(x)),efgh(0,0,1,35)).
/*SC-CC-phase2A*/
leadsto([rt:integer],input(sc)|com(cc,sc,request,deadheading_possibilities,delay(rt,x2,x3,x4),p(x)),output(sc)|com(sc,env,request,deadheading_possibilities,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],input(sc)|com(env,sc,observe,deadheading_possible,delay(rt,x2,x3,x4),p(x)),output(sc)|com(sc,cc,inform,deadheading_possible,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],input(sc)|com(env,sc,observe,deadheading_not_possible,delay(rt,x2,x3,x4),p(x)),output(sc)|com(sc,cc,inform,deadheading_not_possible,delay(rt,x2,x3,x4),p(x)),standard).
/*SC-phase2B*/
leadsto([],input(sc)|com(oc,sc,request,deadheading_resources_possibilities,delay(x1,x2,x3,x4),p(x)),output(sc)|com(sc,env,request,deadheading_resources_possibilities,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],input(sc)|com(env,sc,observe,deadheading_resources_possible,delay(x1,x2,x3,x4),p(x)),output(sc)|com(sc,oc,inform,deadheading_resources_possible,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(sc)|com(env,sc,observe,deadheading_resources_not_possible,delay(x1,x2,x3,x4),p(x)),output(sc)|com(sc,oc,inform,deadheading_resources_not_possible,delay(x1,x2,x3,x4),p('3B2')),standard).
/*SCphase3A1*/
leadsto([rt:integer],input(sc)|com(oc,sc,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A1')),output(sc)|com(sc,env,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer],input(sc)|com(env,sc,observe,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A1')),output(sc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer],input(sc)|com(env,sc,observe,rebooking_possible,delay(rt,x2,x3,x4),p('3A1')),and(output(sc)|com(sc,env,request,rebooking_time,delay(rt,x2,x3,x4),p('3A1')),output(sc)|com(sc,env,request,transit_slack_time_cbt,delay(rt,x2,x3,x4),p('3A1'))),standard).
leadsto([rt:integer,pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('3A1')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(rt,x2,x3,x4),p('3A1')),<=(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(rt,x2,bt,x4),p('3A1')),standard).
leadsto([rt:integer,pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('3A1')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(rt,x2,x3,x4),p('3A1')),>(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(sc)|com(oc,sc,request,effect_of_kt_on_tpax,delay(rt,x2,x3,kt),p('3A1')),output(sc)|com(sc,env,request,transit_slack_time_ckt,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('3A1')),<=(kt,pt)),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('3A1')),>(kt,pt)),output(sc)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A1')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A1')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_unsuccessful,delay(rt,x2,x3,kt),p('3A1')),standard).
/*SC-phase3A2*/
leadsto([rt:integer,dt:integer],input(sc)|com(oc,sc,request,effect_of_dt_on_tpax,delay(rt,dt,x3,x4),p('3A2')),output(sc)|com(sc,env,request,transit_slack_time_cdt,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,pt:integer,dt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cdt(pt),delay(rt,dt,x3,x4),p('3A2')),<=(dt,pt)),output(sc)|com(sc,oc,inform,deadheading_crew_connects_tpax_successful,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,pt:integer,dt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cdt(pt),delay(rt,dt,x3,x4),p('3A2')),>(dt,pt)),output(sc)|com(sc,fd,request,organizing_cxn_measures_possibilities,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,dt,x3,x4),p('3A2')),output(sc)|com(sc,oc,inform,deadheading_crew_connects_tpax_successful,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,dt,x3,x4),p('3A2')),output(sc)|com(sc,oc,inform,deadheading_crew_connects_tpax_unsuccessful,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(sc)|com(oc,sc,request,rebooking_possibilities,delay(rt,dt,x3,x4),p('3A2')),output(sc)|com(sc,env,request,rebooking_possibilities,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(sc)|com(env,sc,observe,rebooking_not_possible,delay(rt,dt,x3,x4),p('3A2')),output(sc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],input(sc)|com(env,sc,observe,rebooking_possible,delay(rt,dt,x3,x4),p('3A2')),and(output(sc)|com(sc,env,request,rebooking_time,delay(rt,dt,x3,x4),p('3A2')),output(sc)|com(sc,env,request,transit_slack_time_cbt,delay(rt,dt,x3,x4),p('3A2'))),standard).
leadsto([rt:integer,pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(rt,dt,x3,x4),p('3A2')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(rt,dt,x3,x4),p('3A2')),<=(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(rt,dt,bt,x4),p('3A2')),standard).
leadsto([rt:integer,pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(rt,dt,x3,x4),p('3A2')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(rt,dt,x3,x4),p('3A2')),>(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,dt,bt,x4),p('3A2')),standard).
/*SC-phase3A3*/
leadsto([rt:integer],input(sc)|com(oc,sc,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A3')),output(sc)|com(sc,env,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer],input(sc)|com(env,sc,observe,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A3')),output(sc)|com(sc,oc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A3')),efgh(0,0,1,25)).
leadsto([rt:integer],input(sc)|com(env,sc,observe,rebooking_possible,delay(rt,x2,x3,x4),p('3A3')),and(output(sc)|com(sc,env,request,rebooking_time,delay(rt,x2,x3,x4),p('3A3')),output(sc)|com(sc,env,request,transit_slack_time_cbt,delay(rt,x2,x3,x4),p('3A3'))),standard).
leadsto([rt:integer,pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('3A3')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(rt,x2,x3,x4),p('3A3')),<=(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(rt,x2,bt,x4),p('3A3')),standard).
leadsto([rt:integer,pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('3A3')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(rt,x2,x3,x4),p('3A3')),>(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(rt,x2,bt,x4),p('3A3')),efgh(0,0,1,25)).
leadsto([rt:integer,kt:integer],input(sc)|com(oc,sc,request,effect_of_kt_on_tpax,delay(rt,x2,x3,kt),p('3A3')),output(sc)|com(sc,env,request,transit_slack_time_ckt,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('3A3')),<=(kt,pt)),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_successful,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('3A3')),>(kt,pt)),output(sc)|com(sc,fd,request,organizing_cxn_
```

```
measures_possibilities,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A3')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects
_tpax_successful,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A3')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_con
nects_tpax_unsuccessful,delay(rt,x2,x3,kt),p('3A3')),standard).
/*SC-phase3B1*/
leadsto([],input(sc)|com(oc,sc,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B1')),output(sc)|com(sc,env,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],input(sc)|com(env,sc,observe,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B1')),output(sc)|com(sc,oc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B1')),efgh(0,0,1,
35)).
leadsto([],input(sc)|com(env,sc,observe,rebooking_possible,delay(x1,x2,x3,x4),p('3B1')),and(output(sc)|com(sc,env,request,rebooking_time,delay(x1,x2,x3,x4),p('3B1')),output(sc)|com(
sc,env,request,transit_slack_time_cbt,delay(x1,x2,x3,x4),p('3B1'))),standard).
leadsto([pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('3B1')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(x1,x2,x3,
x4),p('3B1')),<=(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(x1,x2,bt,x4),p('3B1')),standard).
leadsto([pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('3B1')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(x1,x2,x3,
x4),p('3B1')),>(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(x1,x2,bt,x4),p('3B1')),efgh(0,0,1,35)).
leadsto([kt:integer],input(sc)|com(oc,sc,request,effect_of_kt_on_tpax,delay(x1,x2,x3,kt),p('3B1')),output(sc)|com(sc,env,request,transit_slack_time_ckt,delay(x1,x2,x3,kt),p('3B1')),stand
ard).
leadsto([pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('3B1')),<=(kt,pt)),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_
connects_tpax_successful,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('3B1')),>(kt,pt)),output(sc)|com(sc,fd,request,organizing_cxn_measures_
possibilities,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B1')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_suc
cessful,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B1')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpa
x_unsuccessful,delay(x1,x2,x3,kt),p('3B1')),standard).
/*SC-phase3B2*/
leadsto([],input(sc)|com(oc,sc,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B2')),output(sc)|com(sc,env,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],input(sc)|com(env,sc,observe,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B2')),output(sc)|com(sc,oc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],input(sc)|com(env,sc,observe,rebooking_possible,delay(x1,x2,x3,x4),p('3B2')),and(output(sc)|com(sc,env,request,rebooking_time,delay(x1,x2,x3,x4),p('3B2')),output(sc)|com(
sc,env,request,transit_slack_time_cbt,delay(x1,x2,x3,x4),p('3B2'))),standard).
leadsto([pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('3B2')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(x1,x2,x3,
x4),p('3B2')),<=(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_successful,delay(x1,x2,bt,x4),p('3B2')),standard).
leadsto([pt:integer,bt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('3B2')),input(sc)|com(env,sc,observe,rebooking_time(bt),delay(x1,x2,x3,
x4),p('3B2')),>(bt,pt)),output(sc)|com(sc,oc,inform,rebooking_connects_tpax_unsuccessful,delay(x1,x2,bt,x4),p('3B2')),standard).
leadsto([kt:integer],input(sc)|com(oc,sc,request,effect_of_kt_on_tpax,delay(x1,x2,x3,kt),p('3B2')),output(sc)|com(sc,env,request,transit_slack_time_ckt,delay(x1,x2,x3,kt),p('3B2')),stand
ard).
leadsto([pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('3B2')),<=(kt,pt)),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_
connects_tpax_successful,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([pt:integer,kt:integer],and(input(sc)|com(env,sc,observe,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('3B2')),>(kt,pt)),output(sc)|com(sc,fd,request,organizing_cxn_measures_
possibilities,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B2')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpax_suc
cessful,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],input(sc)|com(fd,sc,inform,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B2')),output(sc)|com(sc,oc,inform,dispatch_reserve_aircraft_connects_tpa
x_unsuccessful,delay(x1,x2,x3,kt),p('3B2')),standard).
/*EP*/
/*EP-tillphase3*/
leadsto([],and(input(env)|com(ac,env,request,technical_diagnosis_adequateness,delay(x1,x2,x3,x4),p(x)),a_1),output(env)|com(env,ac,inform,technical_diagnosis_adequate,delay(x1,x2
,x3,x4),p(x)),standard).
leadsto([],and(input(env)|com(ac,env,request,technical_diagnosis_adequateness,delay(x1,x2,x3,x4),p(x)),a_0),output(env)|com(env,ac,inform,technical_diagnosis_inadequate,delay(x1,
x2,x3,x4),p(x)),standard).
leadsto([],and(input(env)|com(ac,env,request,spare_parts_availability,delay(x1,x2,x3,x4),p(x)),b_1),output(env)|com(env,ac,inform,spare_parts_available,delay(x1,x2,x3,x4),p(x)),standa
rd).
leadsto([],and(input(env)|com(ac,env,request,spare_parts_availability,delay(x1,x2,x3,x4),p(x)),b_0),output(env)|com(env,ac,inform,spare_parts_unavailable,delay(x1,x2,x3,x4),p(x)),stan
dard).
leadsto([],and(input(env)|com(fd,env,request,wx_pattern_favourability,delay(x1,x2,x3,x4),p(x)),c_1),output(env)|com(env,fd,inform,wx_pattern_favourable,delay(x1,x2,x3,x4),p(x)),stan
dard).
leadsto([],and(input(env)|com(fd,env,request,wx_pattern_favourability,delay(x1,x2,x3,x4),p(x)),c_0),output(env)|com(env,fd,inform,wx_pattern_unfavourable,delay(x1,x2,x3,x4),p(x)),st
andard).
leadsto([],and(input(env)|com(ac,env,request,hangar_availability,delay(x1,x2,x3,x4),p(x)),d_1),output(env)|com(env,ac,inform,hangar_available,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([],and(input(env)|com(ac,env,request,hangar_availability,delay(x1,x2,x3,x4),p(x)),d_0),output(env)|com(env,ac,inform,hangar_unavailable,delay(x1,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(rt,x2,x3,x4),p(x)),g_1),output(env)|com(env,cc,inform,reserve_crew_available,delay(rt,x2,x3,x4),
p(x)),standard).
leadsto([rt:integer],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(rt,x2,x3,x4),p(x)),g_0),output(env)|com(env,cc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x
4),p(x)),standard).
leadsto([rt:integer],and(input(env)|com(sc,env,request,deadheading_possibilities,delay(rt,x2,x3,x4),p(x)),f_1),output(env)|com(env,sc,inform,deadheading_possible,delay(rt,x2,x3,x4),p(
x)),standard).
leadsto([rt:integer],and(input(env)|com(sc,env,request,deadheading_possibilities,delay(rt,x2,x3,x4),p(x)),f_0),output(env)|com(env,sc,inform,deadheading_not_possible,delay(rt,x2,x3,x
4),p(x)),standard).
leadsto([rt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,x4),p(x)),e_1),output(env)|com(env,fd,inform,organizing_cxn_measures_po
ssible,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,x4),p(x)),e_0),output(env)|com(env,fd,inform,organizing_cxn_measures_no
t_possible,delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer],and(input(env)|com(ac,env,request,repair_time,delay(x1,x2,x3,x4),p(x)),rt_(rt)),output(env)|com(env,ac,inform,repair_time(rt),delay(rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_crt,delay(rt,x2,x3,x4),p(x)),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_crt(pt),delay(
rt,x2,x3,x4),p(x)),standard).
leadsto([rt:integer,dt:integer],and(input(env)|com(cc,env,request,deadheading_time,delay(rt,x2,x3,x4),p(x)),dt_(dt)),output(env)|com(env,cc,inform,deadheading_time(dt),delay(rt,dt,x
3,x4),p(x)),standard).
leadsto([rt:integer,ct:integer],and(input(env)|com(cc,env,request,crew_duty_slack_time,delay(rt,x2,x3,x4),p(x)),ct_(ct)),output(env)|com(env,cc,inform,crew_duty_slack_time(ct),delay(
rt,x2,x3,x4),p(x)),standard).
leadsto([],and(input(env)|com(sc,env,request,deadheading_resources_possibilities,delay(x1,x2,x3,x4),p(x)),f_1),output(env)|com(env,sc,inform,deadheading_resources_possible,delay(x
1,x2,x3,x4),p(x)),standard).
leadsto([],and(input(env)|com(sc,env,request,deadheading_resources_possibilities,delay(x1,x2,x3,x4),p(x)),f_0),output(env)|com(env,sc,inform,deadheading_resources_not_possible,de
lay(x1,x2,x3,x4),p(x)),standard).
/*EP-phase'3A1'*/
leadsto([rt:integer],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A1')),h_1),output(env)|com(env,sc,inform,rebooking_possible,delay(rt,x2,x3,x4),p('3
A1')),standard).
leadsto([rt:integer],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A1')),h_0),output(env)|com(env,sc,inform,rebooking_not_possible,delay(rt,x2,x3,x4
),p('3A1')),standard).
leadsto([rt:integer,bt:integer],and(input(env)|com(sc,env,request,rebooking_time,delay(rt,x2,x3,x4),p('3A1')),bt_(bt)),output(env)|com(env,sc,inform,rebooking_time(bt),delay(rt,x2,x3,
x4),p('3A1')),standard).
leadsto([rt:integer,kt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_ckt,delay(rt,x2,x3,kt),p('3A1')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time
_ckt(pt),delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A1')),i_1),output(env)|com(env,ac,inform,reserve_aircraft_available,delay(rt,x
2,x3,x4),p('3A1')),standard).
```

```
leadsto([rt:integer],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A1')),i_0),output(env)|com(env,ac,inform,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(rt,x2,x3,x4),p('3A1')),g_1),output(env)|com(env,cc,inform,reserve_crew_available,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(rt,x2,x3,x4),p('3A1')),g_0),output(env)|com(env,cc,inform,reserve_crew_unavailable,delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer,kt:integer],and(input(env)|com(fd,env,request,ferry_time,delay(rt,x2,x3,x4),p('3A1')),kt_(kt)),output(env)|com(env,fd,inform,ferry_time(kt),delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_cbt,delay(rt,x2,x3,x4),p('3A1')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('3A1')),standard).
leadsto([rt:integer,kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A1')),e_1),output(env)|com(env,fd,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A1')),standard).
leadsto([rt:integer,kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A1')),e_0),output(env)|com(env,fd,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A1')),standard).
/*EP-phase'3A2'*/
leadsto([rt:integer,dt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_cdt,delay(rt,dt,x3,x4),p('3A2')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_cdt(pt),delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,dt,x3,x4),p('3A2')),e_1),output(env)|com(env,fd,inform,organizing_cxn_measures_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,dt,x3,x4),p('3A2')),e_0),output(env)|com(env,fd,inform,organizing_cxn_measures_not_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(rt,dt,x3,x4),p('3A2')),h_1),output(env)|com(env,sc,inform,rebooking_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(rt,dt,x3,x4),p('3A2')),h_0),output(env)|com(env,sc,inform,rebooking_not_possible,delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer,bt:integer],and(input(env)|com(sc,env,request,rebooking_time,delay(rt,dt,x3,x4),p('3A2')),bt_(bt)),output(env)|com(env,sc,inform,rebooking_time(bt),delay(rt,dt,x3,x4),p('3A2')),standard).
leadsto([rt:integer,dt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_cbt,delay(rt,dt,x3,x4),p('3A2')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_cbt(pt),delay(rt,dt,x3,x4),p('3A2')),standard).
/*EP-phase'3A3'*/
leadsto([rt:integer],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A3')),h_1),output(env)|com(env,sc,inform,rebooking_possible,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(rt,x2,x3,x4),p('3A3')),h_0),output(env)|com(env,sc,inform,rebooking_not_possible,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer,bt:integer],and(input(env)|com(sc,env,request,rebooking_time,delay(rt,x2,x3,x4),p('3A3')),bt_(bt)),output(env)|com(env,sc,inform,rebooking_time(bt),delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer,kt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_ckt,delay(rt,x2,x3,kt),p('3A3')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A3')),i_1),output(env)|com(env,ac,inform,reserve_aircraft_available,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(rt,x2,x3,x4),p('3A3')),i_0),output(env)|com(env,ac,inform,reserve_aircraft_unavailable,delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer,kt:integer],and(input(env)|com(fd,env,request,ferry_time,delay(rt,x2,x3,x4),p('3A3')),kt_(kt)),output(env)|com(env,fd,inform,ferry_time(kt),delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_cbt,delay(rt,x2,x3,x4),p('3A3')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('3A3')),standard).
leadsto([rt:integer,kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A3')),e_1),output(env)|com(env,fd,inform,organizing_cxn_measures_possible,delay(rt,x2,x3,kt),p('3A3')),standard).
leadsto([rt:integer,kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(rt,x2,x3,kt),p('3A3')),e_0),output(env)|com(env,fd,inform,organizing_cxn_measures_not_possible,delay(rt,x2,x3,kt),p('3A3')),standard).
/*EP-phase'3B1'*/
leadsto([],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B1')),h_1),output(env)|com(env,sc,inform,rebooking_possible,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B1')),h_0),output(env)|com(env,sc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([bt:integer],and(input(env)|com(sc,env,request,rebooking_time,delay(x1,x2,x3,x4),p('3B1')),bt_(bt)),output(env)|com(env,sc,inform,rebooking_time(bt),delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([kt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_ckt,delay(x1,x2,x3,kt),p('3B1')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B1')),g_1),output(env)|com(env,cc,inform,reserve_crew_available,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B1')),g_0),output(env)|com(env,cc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B1')),i_1),output(env)|com(env,ac,inform,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B1')),i_0),output(env)|com(env,ac,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([kt:integer],and(input(env)|com(fd,env,request,ferry_time,delay(x1,x2,x3,x4),p('3B1')),kt_(kt)),output(env)|com(env,fd,inform,ferry_time(kt),delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_cbt,delay(x1,x2,x3,x4),p('3B1')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('3B1')),standard).
leadsto([kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B1')),e_1),output(env)|com(env,fd,inform,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B1')),standard).
leadsto([kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B1')),e_0),output(env)|com(env,fd,inform,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B1')),standard).
/*EP-phase'3B2'*/
leadsto([],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B2')),h_1),output(env)|com(env,sc,inform,rebooking_possible,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],and(input(env)|com(sc,env,request,rebooking_possibilities,delay(x1,x2,x3,x4),p('3B2')),h_0),output(env)|com(env,sc,inform,rebooking_not_possible,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([bt:integer],and(input(env)|com(sc,env,request,rebooking_time,delay(x1,x2,x3,x4),p('3B2')),bt_(bt)),output(env)|com(env,sc,inform,rebooking_time(bt),delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([kt:integer,pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_ckt,delay(x1,x2,x3,kt),p('3B2')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B2')),g_1),output(env)|com(env,cc,inform,reserve_crew_available,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],and(input(env)|com(cc,env,request,reserve_crew_availability,delay(x1,x2,x3,x4),p('3B2')),g_0),output(env)|com(env,cc,inform,reserve_crew_unavailable,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B2')),i_1),output(env)|com(env,ac,inform,reserve_aircraft_available,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([],and(input(env)|com(ac,env,request,reserve_aircraft_availability,delay(x1,x2,x3,x4),p('3B2')),i_0),output(env)|com(env,ac,inform,reserve_aircraft_unavailable,delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([kt:integer],and(input(env)|com(fd,env,request,ferry_time,delay(x1,x2,x3,x4),p('3B2')),kt_(kt)),output(env)|com(env,fd,inform,ferry_time(kt),delay(x1,x2,x3,kt),p('3B2')),standard).
```

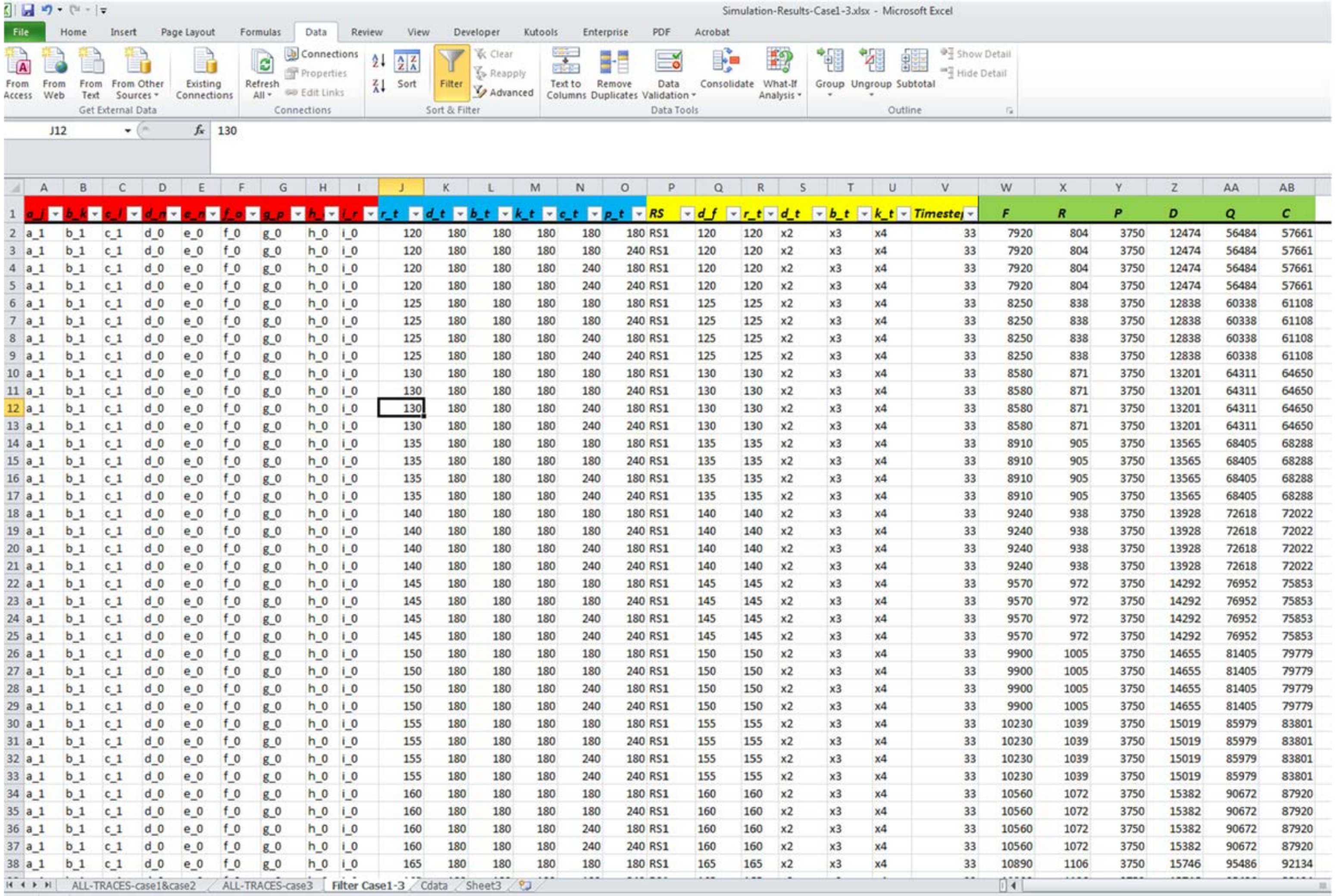
```
leadsto([pt:integer],and(input(env)|com(sc,env,request,transit_slack_time_cbt,delay(x1,x2,x3,x4),p('3B2')),pt_(pt)),output(env)|com(env,sc,inform,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('3B2')),standard).
leadsto([kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B2')),e_1),output(env)|com(env,fd,inform,organizing_cxn_measures_possible,delay(x1,x2,x3,kt),p('3B2')),standard).
leadsto([kt:integer],and(input(env)|com(fd,env,request,organizing_cxn_measures_possibilities,delay(x1,x2,x3,kt),p('3B2')),e_0),output(env)|com(env,fd,inform,organizing_cxn_measures_not_possible,delay(x1,x2,x3,kt),p('3B2')),standard).
/*TRANSFER PROPERTIES,agent-agent*/
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE'],output('R')|com('R','DST','T','M',delay(x1,x2,x3,x4),p('P')),input('DST')|com('R','DST','T','M',delay(x1,x2,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer],output('R')|com('R','DST','T','M',delay(rt,x2,x3,x4),p('P')),input('DST')|com('R','DST','T','M',delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',bt:integer],output('R')|com('R','DST','T','M',delay(x1,x2,bt,x4),p('P')),input('DST')|com('R','DST','T','M',delay(x1,x2,bt,x4),p('P')),standard).
leadsto(['R':'CTRL', 'DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',kt:integer],output('R')|com('R','DST','T','M',delay(x1,x2,x3,kt),p('P')),input('DST')|com('R','DST','T','M',delay(x1,x2,x3,kt),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,dt:integer],output('R')|com('R','DST','T','M',delay(rt,dt,x3,x4),p('P')),input('DST')|com('R','DST','T','M',delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,kt:integer],output('R')|com('R','DST','T','M',delay(rt,x2,x3,kt),p('P')),input('DST')|com('R','DST','T','M',delay(rt,x2,x3,kt),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,dt:integer,bt:integer],output('R')|com('R','DST','T','M',delay(rt,dt,bt,x4),p('P')),input('DST')|com('R','DST','T','M',delay(rt,dt,bt,x4),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,bt:integer],output('R')|com('R','DST','T','M',delay(rt,x2,bt,x4),p('P')),input('DST')|com('R','DST','T','M',delay(rt,x2,bt,x4),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,kt:integer],output('R')|com('R','DST','T',ferry_time(kt),delay(rt,x2,x3,kt),p('P')),input('DST')|com('R','DST','T',ferry_time(kt),delay(rt,x2,x3,kt),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','P':'PHASE',kt:integer],output('R')|com('R','DST','T',ferry_time(kt),delay(x1,x2,x3,kt),p('P')),input('DST')|com('R','DST','T',ferry_time(kt),delay(x1,x2,x3,kt),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer],output('R')|com('R','DST','T',repair_time(rt),delay(rt,x2,x3,x4),p('P')),input('DST')|com('R','DST','T',repair_time(rt),delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,dt:integer],output('R')|com('R','DST','T',deadheading_time(dt),delay(rt,dt,x3,x4),p('P')),input('DST')|com('R','DST','T',deadheading_time(dt),delay(rt,dt,x3,x4),p('P')),standard).
/*ENVIRONMENT INTERACTION PROPERTIES,agent-env*/
leadsto(['R':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE'],output('R')|com('R',env,'T','M',delay(x1,x2,x3,x4),p('P')),input(env)|com('R',env,'T','M',delay(x1,x2,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer],output('R')|com('R',env,'T','M',delay(rt,x2,x3,x4),p('P')),input(env)|com('R',env,'T','M',delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,dt:integer],output('R')|com('R',env,'T','M',delay(rt,dt,x3,x4),p('P')),input(env)|com('R',env,'T','M',delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['R':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',kt:integer],output('R')|com('R',env,'T','M',delay(x1,x2,x3,kt),p('P')),input(env)|com('R',env,'T','M',delay(x1,x2,x3,kt),p('P')),standard).
leadsto(['R':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,kt:integer],output('R')|com('R',env,'T','M',delay(rt,x2,x3,kt),p('P')),input(env)|com('R',env,'T','M',delay(rt,x2,x3,kt),p('P')),standard).
/*ENVIRONMETN INTERACTION PROPERTIES,env-agent*/
leadsto(['DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE'],output(env)|com(env,'DST','T','M',delay(x1,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,'M',delay(x1,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,dt:integer],output(env)|com(env,'DST','T','M',delay(rt,dt,x3,x4),p('P')),input('DST')|com(env,'DST',observe,'M',delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,dt:integer],output(env)|com(env,'DST','T','M',delay(rt,dt,x3,x4),p('P')),input('DST')|com(env,'DST',observe,'M',delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer,kt:integer],output(env)|com(env,'DST','T','M',delay(rt,x2,x3,kt),p('P')),input('DST')|com(env,'DST',observe,'M',delay(rt,x2,x3,kt),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',kt:integer],output(env)|com(env,'DST','T','M',delay(x1,x2,x3,kt),p('P')),input('DST')|com(env,'DST',observe,'M',delay(x1,x2,x3,kt),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','M':'MSG','P':'PHASE',rt:integer],output(env)|com(env,'DST','T','M',delay(rt,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,'M',delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer],output(env)|com(env,'DST','T',repair_time(rt),delay(rt,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,repair_time(rt),delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,pt:integer],output(env)|com(env,'DST','T',transit_slack_time_crt(pt),delay(rt,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_crt(pt),delay(rt,x2,x3,x4),p('P')),standard).

leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,dt:integer],output(env)|com(env,'DST','T',deadheading_time(dt),delay(rt,dt,x3,x4),p('P')),input('DST')|com(env,'DST',observe,deadheading_time(dt),delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,ct:integer],output(env)|com(env,'DST','T',crew_duty_slack_time(ct),delay(rt,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,crew_duty_slack_time(ct),delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,kt:integer,pt:integer],output(env)|com(env,'DST','T',transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_ckt(pt),delay(rt,x2,x3,kt),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,kt:integer],output(env)|com(env,'DST','T',ferry_time(kt),delay(rt,x2,x3,kt),p('P')),input('DST')|com(env,'DST',observe,ferry_time(kt),delay(rt,x2,x3,kt),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,bt:integer],output(env)|com(env,'DST','T',rebooking_time(bt),delay(rt,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,rebooking_time(bt),delay(rt,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,pt:integer],output(env)|com(env,'DST','T',transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_cbt(pt),delay(rt,x2,x3,x4),p('P')),standard).

leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,pt:integer,dt:integer],output(env)|com(env,'DST','T',transit_slack_time_cdt(pt),delay(rt,dt,x3,x4),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_cdt(pt),delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,bt:integer,dt:integer],output(env)|com(env,'DST','T',rebooking_time(bt),delay(rt,dt,x3,x4),p('P')),input('DST')|com(env,'DST',observe,rebooking_time(bt),delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',rt:integer,pt:integer,dt:integer],output(env)|com(env,'DST','T',transit_slack_time_cbt(pt),delay(rt,dt,x3,x4),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_cbt(pt),delay(rt,dt,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',bt:integer],output(env)|com(env,'DST','T',rebooking_time(bt),delay(x1,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,rebooking_time(bt),delay(x1,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',pt:integer],output(env)|com(env,'DST','T',transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_cbt(pt),delay(x1,x2,x3,x4),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',kt:integer],output(env)|com(env,'DST','T',ferry_time(kt),delay(x1,x2,x3,kt),p('P')),input('DST')|com(env,'DST',observe,ferry_time(kt),delay(x1,x2,x3,kt),p('P')),standard).
leadsto(['DST':'CTRL','T':'MSG_TYPE','P':'PHASE',kt:integer,pt:integer],output(env)|com(env,'DST','T',transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('P')),input('DST')|com(env,'DST',observe,transit_slack_time_ckt(pt),delay(x1,x2,x3,kt),p('P')),standard).
```

Appendix D

Excel sheet with conditions (red), parameters (blue), decision outcome (yellow) and cost implication (green)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	a_j	b_k	c_l	d_m	e_n	f_o	g_p	h_	i_r	r_t	d_t	b_t	k_t	c_t	p_t	RS	d_f	r_t	d_t	b_t	k_t	Timeste	F	R	P	D	Q	C
2	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	120	180	180	180	180	180	RS1	120	120	x2	x3	x4	33	7920	804	3750	12474	56484	57661
3	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	120	180	180	180	180	240	RS1	120	120	x2	x3	x4	33	7920	804	3750	12474	56484	57661
4	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	120	180	180	180	240	180	RS1	120	120	x2	x3	x4	33	7920	804	3750	12474	56484	57661
5	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	120	180	180	180	240	240	RS1	120	120	x2	x3	x4	33	7920	804	3750	12474	56484	57661
6	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	125	180	180	180	180	180	RS1	125	125	x2	x3	x4	33	8250	838	3750	12838	60338	61108
7	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	125	180	180	180	180	240	RS1	125	125	x2	x3	x4	33	8250	838	3750	12838	60338	61108
8	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	125	180	180	180	240	180	RS1	125	125	x2	x3	x4	33	8250	838	3750	12838	60338	61108
9	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	125	180	180	180	240	240	RS1	125	125	x2	x3	x4	33	8250	838	3750	12838	60338	61108
10	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	130	180	180	180	180	180	RS1	130	130	x2	x3	x4	33	8580	871	3750	13201	64311	64650
11	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	130	180	180	180	180	240	RS1	130	130	x2	x3	x4	33	8580	871	3750	13201	64311	64650
12	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	130	180	180	180	240	180	RS1	130	130	x2	x3	x4	33	8580	871	3750	13201	64311	64650
13	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	130	180	180	180	240	240	RS1	130	130	x2	x3	x4	33	8580	871	3750	13201	64311	64650
14	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	135	180	180	180	180	180	RS1	135	135	x2	x3	x4	33	8910	905	3750	13565	68405	68288
15	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	135	180	180	180	180	240	RS1	135	135	x2	x3	x4	33	8910	905	3750	13565	68405	68288
16	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	135	180	180	180	240	180	RS1	135	135	x2	x3	x4	33	8910	905	3750	13565	68405	68288
17	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	135	180	180	180	240	240	RS1	135	135	x2	x3	x4	33	8910	905	3750	13565	68405	68288
18	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	140	180	180	180	180	180	RS1	140	140	x2	x3	x4	33	9240	938	3750	13928	72618	72022
19	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	140	180	180	180	180	240	RS1	140	140	x2	x3	x4	33	9240	938	3750	13928	72618	72022
20	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	140	180	180	180	240	180	RS1	140	140	x2	x3	x4	33	9240	938	3750	13928	72618	72022
21	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	140	180	180	180	240	240	RS1	140	140	x2	x3	x4	33	9240	938	3750	13928	72618	72022
22	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	145	180	180	180	180	180	RS1	145	145	x2	x3	x4	33	9570	972	3750	14292	76952	75853
23	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	145	180	180	180	180	240	RS1	145	145	x2	x3	x4	33	9570	972	3750	14292	76952	75853
24	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	145	180	180	180	240	180	RS1	145	145	x2	x3	x4	33	9570	972	3750	14292	76952	75853
25	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	145	180	180	180	240	240	RS1	145	145	x2	x3	x4	33	9570	972	3750	14292	76952	75853
26	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	150	180	180	180	180	180	RS1	150	150	x2	x3	x4	33	9900	1005	3750	14655	81405	79779
27	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	150	180	180	180	180	240	RS1	150	150	x2	x3	x4	33	9900	1005	3750	14655	81405	79779
28	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	150	180	180	180	240	180	RS1	150	150	x2	x3	x4	33	9900	1005	3750	14655	81405	79779
29	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	150	180	180	180	240	240	RS1	150	150	x2	x3	x4	33	9900	1005	3750	14655	81405	79779
30	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	155	180	180	180	180	180	RS1	155	155	x2	x3	x4	33	10230	1039	3750	15019	85979	83801
31	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	155	180	180	180	180	240	RS1	155	155	x2	x3	x4	33	10230	1039	3750	15019	85979	83801
32	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	155	180	180	180	240	180	RS1	155	155	x2	x3	x4	33	10230	1039	3750	15019	85979	83801
33	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	155	180	180	180	240	240	RS1	155	155	x2	x3	x4	33	10230	1039	3750	15019	85979	83801
34	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	160	180	180	180	180	180	RS1	160	160	x2	x3	x4	33	10560	1072	3750	15382	90672	87920
35	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	160	180	180	180	180	240	RS1	160	160	x2	x3	x4	33	10560	1072	3750	15382	90672	87920
36	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	160	180	180	180	240	180	RS1	160	160	x2	x3	x4	33	10560	1072	3750	15382	90672	87920
37	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	160	180	180	180	240	240	RS1	160	160	x2	x3	x4	33	10560	1072	3750	15382	90672	87920
38	a_1	b_1	c_1	d_0	e_0	f_0	g_0	h_0	i_0	165	180	180	180	180	180	RS1	165	165	x2	x3	x4	33	10890	1106	3750	15746	95486	92134

Appendix E

Simulation of case 1 with condition sequence p_2 and r_t=170

```
content(type(savedtrace('c:/Users/KamalB/Desktop/Thesis 11.0-final/sim11.lt'))).
content(generator(app(leadsto_software, 127, [psprinting:1, showtrace:8, simalgo:11]))).
content(source(file('c:/Users/KamalB/Desktop/Thesis 11.0-final/sim11.lt', [size(69367), path('c:/Users/KamalB/Desktop/Thesis 11.0-
final/sim11.lt'), mdate('Tue May 31 17:33:33 2016')]))).
content(run([date('Tue May 31 17:33:41 2016')])).
atom_trace('input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd,
observe, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))), [range(1, 95, unknown), range(0, 1, true)]).
atom_trace('output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac,
inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))), [range(2, 95, unknown), range(1, 2, true), range(0, 1,
unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac,
inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))), [range(3, 95, unknown), range(2, 3, true), range(0, 2,
unknown)]).
atom_trace('output(ac)|com(ac, env, request, technical_diagnosis_reliability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env,
request, technical_diagnosis_reliability, delay(x1, x2, x3, x4), p(x))), [range(4, 95, unknown), range(3, 4, true), range(0, 3, unknown)]).
atom_trace('input(env)|com(ac, env, request, technical_diagnosis_reliability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env,
request, technical_diagnosis_reliability, delay(x1, x2, x3, x4), p(x))), [range(5, 95, unknown), range(4, 5, true), range(0, 4, unknown)]).
atom_trace('output(env)|com(env, ac, inform, technical_diagnosis_reliable, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform,
technical_diagnosis_reliable, delay(x1, x2, x3, x4), p(x))), [range(6, 95, unknown), range(5, 6, true), range(0, 5, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, technical_diagnosis_reliable, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe,
technical_diagnosis_reliable, delay(x1, x2, x3, x4), p(x))), [range(7, 95, unknown), range(6, 7, true), range(0, 6, unknown)]).
atom_trace('output(ac)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request,
spare_parts_availability, delay(x1, x2, x3, x4), p(x))), [range(8, 95, unknown), range(7, 8, true), range(0, 7, unknown)]).
atom_trace('input(env)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request,
spare_parts_availability, delay(x1, x2, x3, x4), p(x))), [range(9, 95, unknown), range(8, 9, true), range(0, 8, unknown)]).
atom_trace('output(env)|com(env, ac, inform, spare_parts_available, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform,
spare_parts_available, delay(x1, x2, x3, x4), p(x))), [range(10, 95, unknown), range(9, 10, true), range(0, 9, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, spare_parts_available, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe,
spare_parts_available, delay(x1, x2, x3, x4), p(x))), [range(11, 95, unknown), range(10, 11, true), range(0, 10, unknown)]).
atom_trace('output(ac)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, fd, request,
wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))), [range(12, 95, unknown), range(11, 12, true), range(0, 11, unknown)]).
atom_trace('input(fd)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(ac, fd, request,
wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))), [range(13, 95, unknown), range(12, 13, true), range(0, 12, unknown)]).
atom_trace('output(fd)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, env, request,
wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))), [range(14, 95, unknown), range(13, 14, true), range(0, 13, unknown)]).
atom_trace('input(env)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(fd, env, request,
wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))), [range(15, 95, unknown), range(14, 15, true), range(0, 14, unknown)]).
atom_trace('output(env)|com(env, fd, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, fd, inform,
wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))), [range(16, 95, unknown), range(15, 16, true), range(0, 15, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe,
wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))), [range(17, 95, unknown), range(16, 17, true), range(0, 16, unknown)]).
atom_trace('output(fd)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac, inform,
wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))), [range(18, 95, unknown), range(17, 18, true), range(0, 17, unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac, inform,
wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))), [range(19, 95, unknown), range(18, 19, true), range(0, 18, unknown)]).
atom_trace('output(ac)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, repair_time,
delay(x1, x2, x3, x4), p(x))), [range(20, 95, unknown), range(19, 20, true), range(0, 19, unknown)]).
atom_trace('input(env)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, repair_time,
delay(x1, x2, x3, x4), p(x))), [range(21, 95, unknown), range(20, 21, true), range(0, 20, unknown)]).
atom_trace('output(env)|com(env, ac, inform, repair_time(170), delay(170, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform,
repair_time(170), delay(170, x2, x3, x4), p(x))), [range(22, 95, unknown), range(21, 22, true), range(0, 21, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, repair_time(170), delay(170, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe,
repair_time(170), delay(170, x2, x3, x4), p(x))), [range(23, 95, unknown), range(22, 23, true), range(0, 22, unknown)]).
atom_trace('output(ac)|com(ac, oc, inform, repair_time(170), delay(170, x2, x3, x4), p(x))', (output(ac)|com(ac, oc, inform,
repair_time(170), delay(170, x2, x3, x4), p(x))), [range(24, 95, unknown), range(23, 24, true), range(0, 23, unknown)]).
atom_trace('input(oc)|com(ac, oc, inform, repair_time(170), delay(170, x2, x3, x4), p(x))', (input(oc)|com(ac, oc, inform, repair_time(170),
delay(170, x2, x3, x4), p(x))), [range(25, 95, unknown), range(24, 25, true), range(0, 24, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(170, x2, x3, x4), p(x))', (output(oc)|com(oc, sc, request,
effect_of_rt_on_tpax, delay(170, x2, x3, x4), p(x))), [range(26, 95, unknown), range(25, 26, true), range(0, 25, unknown)]).
atom_trace('output(oc)|com(oc, cc, request, effect_of_rt_on_crew, delay(170, x2, x3, x4), p(x))', (output(oc)|com(oc, cc, request,
effect_of_rt_on_crew, delay(170, x2, x3, x4), p(x))), [range(26, 95, unknown), range(25, 26, true), range(0, 25, unknown)]).

atom_trace('input(sc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(170, x2, x3, x4), p(x))', (input(sc)|com(oc, sc, request,
effect_of_rt_on_tpax, delay(170, x2, x3, x4), p(x))), [range(27, 95, unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('input(cc)|com(oc, cc, request, effect_of_rt_on_crew, delay(170, x2, x3, x4), p(x))', (input(cc)|com(oc, cc, request,
effect_of_rt_on_crew, delay(170, x2, x3, x4), p(x))), [range(27, 95, unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('output(sc)|com(sc, env, request, transit_slack_time_crt, delay(170, x2, x3, x4), p(x))', (output(sc)|com(sc, env, request,
```

```
transit_slack_time_crt, delay(170, x2, x3, x4), p(x))), [range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('output(cc)|com(cc, env, request, crew_duty_slack_time, delay(170, x2, x3, x4), p(x))', (output(cc)|com(cc, env, request, crew_duty_slack_time, delay(170, x2, x3, x4), p(x))), [range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('input(env)|com(sc, env, request, transit_slack_time_crt, delay(170, x2, x3, x4), p(x))', (input(env)|com(sc, env, request, transit_slack_time_crt, delay(170, x2, x3, x4), p(x))), [range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).
atom_trace('input(env)|com(cc, env, request, crew_duty_slack_time, delay(170, x2, x3, x4), p(x))', (input(env)|com(cc, env, request, crew_duty_slack_time, delay(170, x2, x3, x4), p(x))), [range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).
atom_trace('output(env)|com(env, sc, inform, transit_slack_time_crt(180), delay(170, x2, x3, x4), p(x))', (output(env)|com(env, sc, inform, transit_slack_time_crt(180), delay(170, x2, x3, x4), p(x))), [range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('output(env)|com(env, cc, inform, crew_duty_slack_time(240), delay(170, x2, x3, x4), p(x))', (output(env)|com(env, cc, inform, crew_duty_slack_time(240), delay(170, x2, x3, x4), p(x))), [range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('input(cc)|com(env, cc, observe, crew_duty_slack_time(240), delay(170, x2, x3, x4), p(x))', (input(cc)|com(env, cc, observe, crew_duty_slack_time(240), delay(170, x2, x3, x4), p(x))), [range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, transit_slack_time_crt(180), delay(170, x2, x3, x4), p(x))', (input(sc)|com(env, sc, observe, transit_slack_time_crt(180), delay(170, x2, x3, x4), p(x))), [range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('output(cc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(170, x2, x3, x4), p(\'3A1\'))', (output(cc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(170, x2, x3, x4), p('3A1'))), [range(56, 95, unknown), range(31, 56, true), range(0, 31, unknown)]).
atom_trace('input(oc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(170, x2, x3, x4), p(\'3A1\'))', (input(oc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(170, x2, x3, x4), p('3A1'))), [range(57, 95, unknown), range(32, 57, true), range(0, 32, unknown)]).
atom_trace('recovery(\'RS1\', 170, 170, dt, bt, kt)', recovery('RS1', 170, 170, dt, bt, kt), [range(58, 95, unknown), range(33, 58, true), range(0, 33, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, tpax_unaffected_by_rt, delay(170, x2, x3, x4), p(x))', (output(sc)|com(sc, oc, inform, tpax_unaffected_by_rt, delay(170, x2, x3, x4), p(x))), [range(66, 95, unknown), range(31, 66, true), range(0, 31, unknown)]).

atom_trace('input(oc)|com(sc, oc, inform, tpax_unaffected_by_rt, delay(170, x2, x3, x4), p(x))', (input(oc)|com(sc, oc, inform, tpax_unaffected_by_rt, delay(170, x2, x3, x4), p(x))), [range(67, 95, unknown), range(32, 67, true), range(0, 32, unknown)]).
atom_trace(a_1, a_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(b_1, b_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(c_1, c_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(d_1, d_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(e_0, e_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(f_0, f_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(g_1, g_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(h_0, h_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(i_1, i_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('rt_(170)', rt_(170), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('pt_(180)', pt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('bt_(180)', bt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('kt_(180)', kt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('ct_(240)', ct_(240), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('dt_(dt)', dt_(dt), [range(90, 95, unknown), range(0, 90, true)]).
times(0, 95, 95).
```

Appendix F

Simulation of case 1 with condition sequence p_2 and r_t=200 (grey highlighted stateproperties are additional state properties compared with r_t=170)

```
content(type(savedtrace('c:/Users/KamalB/Desktop/Thesis 11.0-final/sim11.lt'))).
content(generator(app(leadsto_software, 127, [psprinting:1, showtrace:8, simalgo:11]))).
content(source(file('c:/Users/KamalB/Desktop/Thesis 11.0-final/sim11.lt', [size(69367), path('c:/Users/KamalB/Desktop/Thesis 11.0-final/sim11.lt'), mdate('Tue May 31 17:34:34
2016')]))).
content(run([date('Tue May 31 17:34:49 2016')])).
atom_trace('input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(1, 95, unknown), range(0, 1, true)]).
atom_trace('output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(2, 95, unknown), range(1, 2, true), range(0, 1, unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(3, 95, unknown), range(2, 3, true), range(0, 2, unknown)]).
atom_trace('output(ac)|com(ac, env, request, technical_diagnosis_reliability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, technical_diagnosis_reliability, delay(x1, x2,
x3, x4), p(x))), [range(4, 95, unknown), range(3, 4, true), range(0, 3, unknown)]).
atom_trace('input(env)|com(ac, env, request, technical_diagnosis_reliability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, technical_diagnosis_reliability, delay(x1, x2,
x3, x4), p(x))), [range(5, 95, unknown), range(4, 5, true), range(0, 4, unknown)]).
atom_trace('output(env)|com(env, ac, inform, technical_diagnosis_reliable, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, technical_diagnosis_reliable, delay(x1, x2, x3,
x4), p(x))), [range(6, 95, unknown), range(5, 6, true), range(0, 5, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, technical_diagnosis_reliable, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, technical_diagnosis_reliable, delay(x1, x2, x3,
x4), p(x))), [range(7, 95, unknown), range(6, 7, true), range(0, 6, unknown)]).
atom_trace('output(ac)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4),
p(x))), [range(8, 95, unknown), range(7, 8, true), range(0, 7, unknown)]).
atom_trace('input(env)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4),
p(x))), [range(9, 95, unknown), range(8, 9, true), range(0, 8, unknown)]).
atom_trace('output(env)|com(env, ac, inform, spare_parts_available, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, spare_parts_available, delay(x1, x2, x3, x4), p(x))),
[range(10, 95, unknown), range(9, 10, true), range(0, 9, unknown)]).

atom_trace('input(ac)|com(env, ac, observe, spare_parts_available, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, spare_parts_available, delay(x1, x2, x3, x4), p(x))),
[range(11, 95, unknown), range(10, 11, true), range(0, 10, unknown)]).

atom_trace('output(ac)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4),
p(x))), [range(12, 95, unknown), range(11, 12, true), range(0, 11, unknown)]).
atom_trace('input(fd)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))),
[range(13, 95, unknown), range(12, 13, true), range(0, 12, unknown)]).
atom_trace('output(fd)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4),
p(x))), [range(14, 95, unknown), range(13, 14, true), range(0, 13, unknown)]).
atom_trace('input(env)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4),
p(x))), [range(15, 95, unknown), range(14, 15, true), range(0, 14, unknown)]).
atom_trace('output(env)|com(env, fd, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, fd, inform, wx_pattern_favourable, delay(x1, x2, x3, x4),
p(x))), [range(16, 95, unknown), range(15, 16, true), range(0, 15, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))),
[range(17, 95, unknown), range(16, 17, true), range(0, 16, unknown)]).

atom_trace('output(fd)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))),
[range(18, 95, unknown), range(17, 18, true), range(0, 17, unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))),
[range(19, 95, unknown), range(18, 19, true), range(0, 18, unknown)]).
atom_trace('output(ac)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))), [range(20, 95,
unknown), range(19, 20, true), range(0, 19, unknown)]).
atom_trace('input(env)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))), [range(21, 95,
unknown), range(20, 21, true), range(0, 20, unknown)]).
atom_trace('output(env)|com(env, ac, inform, repair_time(200), delay(200, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, repair_time(200), delay(200, x2, x3, x4), p(x))),
[range(22, 95, unknown), range(21, 22, true), range(0, 21, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, repair_time(200), delay(200, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, repair_time(200), delay(200, x2, x3, x4), p(x))), [range(23,
95, unknown), range(22, 23, true), range(0, 22, unknown)]).
atom_trace('output(ac)|com(ac, oc, inform, repair_time(200), delay(200, x2, x3, x4), p(x))', (output(ac)|com(ac, oc, inform, repair_time(200), delay(200, x2, x3, x4), p(x))), [range(24, 95,
unknown), range(23, 24, true), range(0, 23, unknown)]).
atom_trace('input(oc)|com(ac, oc, inform, repair_time(200), delay(200, x2, x3, x4), p(x))', (input(oc)|com(ac, oc, inform, repair_time(200), delay(200, x2, x3, x4), p(x))), [range(25, 95,
unknown), range(24, 25, true), range(0, 24, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(200, x2, x3, x4), p(x))', (output(oc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(200, x2, x3, x4), p(x))),
[range(26, 95, unknown), range(25, 26, true), range(0, 25, unknown)]).

atom_trace('output(oc)|com(oc, cc, request, effect_of_rt_on_crew, delay(200, x2, x3, x4), p(x))', (output(oc)|com(oc, cc, request, effect_of_rt_on_crew, delay(200, x2, x3, x4), p(x))),
[range(26, 95, unknown), range(25, 26, true), range(0, 25, unknown)]).

atom_trace('input(sc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(200, x2, x3, x4), p(x))', (input(sc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(200, x2, x3, x4), p(x))),
[range(27, 95, unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('input(cc)|com(oc, cc, request, effect_of_rt_on_crew, delay(200, x2, x3, x4), p(x))', (input(cc)|com(oc, cc, request, effect_of_rt_on_crew, delay(200, x2, x3, x4), p(x))),
[range(27, 95, unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('output(cc)|com(cc, env, request, crew_duty_slack_time, delay(200, x2, x3, x4), p(x))', (output(cc)|com(cc, env, request, crew_duty_slack_time, delay(200, x2, x3, x4),
p(x))), [range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('output(sc)|com(sc, env, request, transit_slack_time_crt, delay(200, x2, x3, x4), p(x))', (output(sc)|com(sc, env, request, transit_slack_time_crt, delay(200, x2, x3, x4), p(x))),
[range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('input(env)|com(cc, env, request, crew_duty_slack_time, delay(200, x2, x3, x4), p(x))', (input(env)|com(cc, env, request, crew_duty_slack_time, delay(200, x2, x3, x4),
p(x))), [range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).
atom_trace('input(env)|com(sc, env, request, transit_slack_time_crt, delay(200, x2, x3, x4), p(x))', (input(env)|com(sc, env, request, transit_slack_time_crt, delay(200, x2, x3, x4), p(x))),
[range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).
atom_trace('output(env)|com(env, cc, inform, crew_duty_slack_time(240), delay(200, x2, x3, x4), p(x))', (output(env)|com(env, cc, inform, crew_duty_slack_time(240), delay(200, x2,
x3, x4), p(x))), [range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('output(env)|com(env, sc, inform, transit_slack_time_crt(180), delay(200, x2, x3, x4), p(x))', (output(env)|com(env, sc, inform, transit_slack_time_crt(180), delay(200, x2,
x3, x4), p(x))), [range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('input(cc)|com(env, cc, observe, crew_duty_slack_time(240), delay(200, x2, x3, x4), p(x))', (input(cc)|com(env, cc, observe, crew_duty_slack_time(240), delay(200, x2, x3,
x4), p(x))), [range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, transit_slack_time_crt(180), delay(200, x2, x3, x4), p(x))', (input(sc)|com(env, sc, observe, transit_slack_time_crt(180), delay(200, x2, x3,
x4), p(x))), [range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('output(sc)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))', (output(sc)|com(sc, fd, request,
organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))), [range(32, 95, unknown), range(31, 32, true), range(0, 31, unknown)]).
atom_trace('input(fd)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))', (input(fd)|com(sc, fd, request, organizing_cxn_measures_possibilities,
delay(200, x2, x3, x4), p(x))), [range(33, 95, unknown), range(32, 33, true), range(0, 32, unknown)]).
```

```
atom_trace('output(fd)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))', (output(fd)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))), [range(34, 95, unknown), range(33, 34, true), range(0, 33, unknown)]).
atom_trace('input(env)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))', (input(env)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(200, x2, x3, x4), p(x))), [range(35, 95, unknown), range(34, 35, true), range(0, 34, unknown)]).
atom_trace('output(env)|com(env, fd, inform, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))', (output(env)|com(env, fd, inform, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))), [range(36, 95, unknown), range(35, 36, true), range(0, 35, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))), [range(37, 95, unknown), range(36, 37, true), range(0, 36, unknown)]).
atom_trace('output(fd)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))', (output(fd)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))), [range(38, 95, unknown), range(37, 38, true), range(0, 37, unknown)]).
atom_trace('input(sc)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))', (input(sc)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(200, x2, x3, x4), p(x))), [range(39, 95, unknown), range(38, 39, true), range(0, 38, unknown)]).
atom_trace('output(cc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(200, x2, x3, x4), p(\'3A1\'))', (output(cc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(200, x2, x3, x4), p('3A1'))), [range(56, 95, unknown), range(31, 56, true), range(0, 31, unknown)]).
atom_trace('input(oc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(200, x2, x3, x4), p(\'3A1\'))', (input(oc)|com(cc, oc, inform, crew_unaffected_by_rt, delay(200, x2, x3, x4), p('3A1'))), [range(57, 95, unknown), range(32, 57, true), range(0, 32, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, rebooking_possibilities, delay(200, x2, x3, x4), p(\'3A1\'))', (output(oc)|com(oc, sc, request, rebooking_possibilities, delay(200, x2, x3, x4), p('3A1'))), [range(58, 95, unknown), range(41, 58, true), range(0, 41, unknown)]).
atom_trace('input(sc)|com(oc, sc, request, rebooking_possibilities, delay(200, x2, x3, x4), p(\'3A1\'))', (input(sc)|com(oc, sc, request, rebooking_possibilities, delay(200, x2, x3, x4), p('3A1'))), [range(59, 95, unknown), range(42, 59, true), range(0, 42, unknown)]).
atom_trace('output(sc)|com(sc, env, request, rebooking_possibilities, delay(200, x2, x3, x4), p(\'3A1\'))', (output(sc)|com(sc, env, request, rebooking_possibilities, delay(200, x2, x3, x4), p('3A1'))), [range(60, 95, unknown), range(43, 60, true), range(0, 43, unknown)]).
atom_trace('input(env)|com(sc, env, request, rebooking_possibilities, delay(200, x2, x3, x4), p(\'3A1\'))', (input(env)|com(sc, env, request, rebooking_possibilities, delay(200, x2, x3, x4), p('3A1'))), [range(61, 95, unknown), range(44, 61, true), range(0, 44, unknown)]).
atom_trace('output(env)|com(env, sc, inform, rebooking_not_possible, delay(200, x2, x3, x4), p(\'3A1\'))', (output(env)|com(env, sc, inform, rebooking_not_possible, delay(200, x2, x3, x4), p('3A1'))), [range(62, 95, unknown), range(45, 62, true), range(0, 45, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, rebooking_not_possible, delay(200, x2, x3, x4), p(\'3A1\'))', (input(sc)|com(env, sc, observe, rebooking_not_possible, delay(200, x2, x3, x4), p('3A1'))), [range(63, 95, unknown), range(46, 63, true), range(0, 46, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, rebooking_not_possible, delay(200, x2, x3, x4), p(\'3A1\'))', (output(sc)|com(sc, oc, inform, rebooking_not_possible, delay(200, x2, x3, x4), p('3A1'))), [range(64, 95, unknown), range(47, 64, true), range(0, 47, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, rebooking_not_possible, delay(200, x2, x3, x4), p(\'3A1\'))', (input(oc)|com(sc, oc, inform, rebooking_not_possible, delay(200, x2, x3, x4), p('3A1'))), [range(65, 95, unknown), range(48, 65, true), range(0, 48, unknown)]).
atom_trace('output(oc)|com(oc, ac, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (output(oc)|com(oc, ac, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p('3A1'))), [range(66, 95, unknown), range(49, 66, true), range(0, 49, unknown)]).
atom_trace('output(oc)|com(oc, cc, request, reserve_crew_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (output(oc)|com(oc, cc, request, reserve_crew_availability, delay(200, x2, x3, x4), p('3A1'))), [range(66, 95, unknown), range(49, 66, true), range(0, 49, unknown)]).
atom_trace('input(ac)|com(oc, ac, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (input(ac)|com(oc, ac, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p('3A1'))), [range(67, 95, unknown), range(50, 67, true), range(0, 50, unknown)]).
atom_trace('input(cc)|com(oc, cc, request, reserve_crew_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (input(cc)|com(oc, cc, request, reserve_crew_availability, delay(200, x2, x3, x4), p('3A1'))), [range(67, 95, unknown), range(50, 67, true), range(0, 50, unknown)]).
atom_trace('output(cc)|com(cc, env, request, reserve_crew_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (output(cc)|com(cc, env, request, reserve_crew_availability, delay(200, x2, x3, x4), p('3A1'))), [range(68, 95, unknown), range(51, 68, true), range(0, 51, unknown)]).
atom_trace('output(ac)|com(ac, env, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (output(ac)|com(ac, env, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p('3A1'))), [range(68, 95, unknown), range(51, 68, true), range(0, 51, unknown)]).
atom_trace('input(env)|com(cc, env, request, reserve_crew_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (input(env)|com(cc, env, request, reserve_crew_availability, delay(200, x2, x3, x4), p('3A1'))), [range(69, 95, unknown), range(52, 69, true), range(0, 52, unknown)]).
atom_trace('input(env)|com(ac, env, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p(\'3A1\'))', (input(env)|com(ac, env, request, reserve_aircraft_availability, delay(200, x2, x3, x4), p('3A1'))), [range(69, 95, unknown), range(52, 69, true), range(0, 52, unknown)]).
atom_trace('output(env)|com(env, ac, inform, reserve_aircraft_available, delay(200, x2, x3, x4), p(\'3A1\'))', (output(env)|com(env, ac, inform, reserve_aircraft_available, delay(200, x2, x3, x4), p('3A1'))), [range(70, 95, unknown), range(53, 70, true), range(0, 53, unknown)]).
atom_trace('output(env)|com(env, cc, inform, reserve_crew_available, delay(200, x2, x3, x4), p(\'3A1\'))', (output(env)|com(env, cc, inform, reserve_crew_available, delay(200, x2, x3, x4), p('3A1'))), [range(70, 95, unknown), range(53, 70, true), range(0, 53, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, reserve_aircraft_available, delay(200, x2, x3, x4), p(\'3A1\'))', (input(ac)|com(env, ac, observe, reserve_aircraft_available, delay(200, x2, x3, x4), p('3A1'))), [range(71, 95, unknown), range(54, 71, true), range(0, 54, unknown)]).
atom_trace('input(cc)|com(env, cc, observe, reserve_crew_available, delay(200, x2, x3, x4), p(\'3A1\'))', (input(cc)|com(env, cc, observe, reserve_crew_available, delay(200, x2, x3, x4), p('3A1'))), [range(71, 95, unknown), range(54, 71, true), range(0, 54, unknown)]).
atom_trace('output(ac)|com(ac, fd, inform, reserve_aircraft_available, delay(200, x2, x3, x4), p(\'3A1\'))', (output(ac)|com(ac, fd, inform, reserve_aircraft_available, delay(200, x2, x3, x4), p('3A1'))), [range(72, 95, unknown), range(55, 72, true), range(0, 55, unknown)]).
atom_trace('input(fd)|com(ac, fd, inform, reserve_aircraft_available, delay(200, x2, x3, x4), p(\'3A1\'))', (input(fd)|com(ac, fd, inform, reserve_aircraft_available, delay(200, x2, x3, x4), p('3A1'))), [range(73, 95, unknown), range(56, 73, true), range(0, 56, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, tpax_affected_by_rt, delay(200, x2, x3, x4), p(x))', (output(sc)|com(sc, oc, inform, tpax_affected_by_rt, delay(200, x2, x3, x4), p(x))), [range(74, 95, unknown), range(39, 74, true), range(0, 39, unknown)]).
atom_trace('output(fd)|com(fd, env, request, ferry_time, delay(200, x2, x3, x4), p(\'3A1\'))', (output(fd)|com(fd, env, request, ferry_time, delay(200, x2, x3, x4), p('3A1'))), [range(74, 95, unknown), range(57, 74, true), range(0, 57, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, tpax_affected_by_rt, delay(200, x2, x3, x4), p(x))', (input(oc)|com(sc, oc, inform, tpax_affected_by_rt, delay(200, x2, x3, x4), p(x))), [range(75, 95, unknown), range(40, 75, true), range(0, 40, unknown)]).
atom_trace('input(env)|com(fd, env, request, ferry_time, delay(200, x2, x3, x4), p(\'3A1\'))', (input(env)|com(fd, env, request, ferry_time, delay(200, x2, x3, x4), p('3A1'))), [range(75, 95, unknown), range(58, 75, true), range(0, 58, unknown)]).
atom_trace('output(env)|com(env, fd, inform, ferry_time(180), delay(200, x2, x3, 180), p(\'3A1\'))', (output(env)|com(env, fd, inform, ferry_time(180), delay(200, x2, x3, 180), p('3A1'))), [range(76, 95, unknown), range(59, 76, true), range(0, 59, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, ferry_time(180), delay(200, x2, x3, 180), p(\'3A1\'))', (input(fd)|com(env, fd, observe, ferry_time(180), delay(200, x2, x3, 180), p('3A1'))), [range(77, 95, unknown), range(60, 77, true), range(0, 60, unknown)]).
atom_trace('output(fd)|com(fd, oc, inform, ferry_time(180), delay(200, x2, x3, 180), p(\'3A1\'))', (output(fd)|com(fd, oc, inform, ferry_time(180), delay(200, x2, x3, 180), p('3A1'))), [range(78, 95, unknown), range(61, 78, true), range(0, 61, unknown)]).

atom_trace('output(cc)|com(cc, oc, inform, reserve_crew_available, delay(200, x2, x3, x4), p(\'3A1\'))', (output(cc)|com(cc, oc, inform, reserve_crew_available, delay(200, x2, x3, x4), p('3A1'))), [range(79, 95, unknown), range(55, 79, true), range(0, 55, unknown)]).
atom_trace('input(oc)|com(fd, oc, inform, ferry_time(180), delay(200, x2, x3, 180), p(\'3A1\'))', (input(oc)|com(fd, oc, inform, ferry_time(180), delay(200, x2, x3, 180), p('3A1'))), [range(79, 95, unknown), range(62, 79, true), range(0, 62, unknown)]).
atom_trace('input(oc)|com(cc, oc, inform, reserve_crew_available, delay(200, x2, x3, x4), p(\'3A1\'))', (input(oc)|com(cc, oc, inform, reserve_crew_available, delay(200, x2, x3, x4), p('3A1'))), [range(80, 95, unknown), range(56, 80, true), range(0, 56, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(200, x2, x3, 180), p(\'3A1\'))', (output(oc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(200, x2, x3, 180), p('3A1'))), [range(80, 95, unknown), range(63, 80, true), range(0, 63, unknown)]).
atom_trace('input(sc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(200, x2, x3, 180), p(\'3A1\'))', (input(sc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(200, x2, x3, 180), p('3A1'))), [range(81, 95, unknown), range(64, 81, true), range(0, 64, unknown)]).
atom_trace('output(sc)|com(sc, env, request, transit_slack_time_ckt, delay(200, x2, x3, 180), p(\'3A1\'))', (output(sc)|com(sc, env, request, transit_slack_time_ckt, delay(200, x2, x3, 180), p('3A1'))), [range(82, 95, unknown), range(65, 82, true), range(0, 65, unknown)]).
atom_trace('input(env)|com(sc, env, request, transit_slack_time_ckt, delay(200, x2, x3, 180), p(\'3A1\'))', (input(env)|com(sc, env, request, transit_slack_time_ckt, delay(200, x2, x3, 180), p('3A1'))), [range(83, 95, unknown), range(66, 83, true), range(0, 66, unknown)]).
atom_trace('output(env)|com(env, sc, inform, transit_slack_time_ckt(180), delay(200, x2, x3, 180), p(\'3A1\'))', (output(env)|com(env, sc, inform, transit_slack_time_ckt(180), delay(200, x2, x3, 180), p('3A1'))), [range(84, 95, unknown), range(67, 84, true), range(0, 67, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, transit_slack_time_ckt(180), delay(200, x2, x3, 180), p(\'3A1\'))', (input(sc)|com(env, sc, observe, transit_slack_time_ckt(180), delay(200, x2, x3, 180), p('3A1'))), [range(85, 95, unknown), range(68, 85, true), range(0, 68, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, dispatch_reserve_aircraft_connects_tpax_successful, delay(200, x2, x3, 180), p(\'3A1\'))', (output(sc)|com(sc, oc, inform,
```

```
dispatch_reserve_aircraft_connects_tpax_successful, delay(200, x2, x3, 180), p('3A1'))), [range(86, 95, unknown), range(69, 86, true), range(0, 69, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, dispatch_reserve_aircraft_connects_tpax_successful, delay(200, x2, x3, 180), p(\'3A1\'))', (input(oc)|com(sc, oc, inform,
dispatch_reserve_aircraft_connects_tpax_successful, delay(200, x2, x3, 180), p('3A1'))), [range(87, 95, unknown), range(70, 87, true), range(0, 70, unknown)]).
atom_trace('recovery(\'RS8\', 180, 200, dt, bt, 180)', recovery('RS8', 180, 200, dt, bt, 180), [range(88, 95, unknown), range(71, 88, true), range(0, 71, unknown)]).
atom_trace(a_1, a_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(b_1, b_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(c_1, c_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(d_1, d_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(e_0, e_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(f_0, f_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(g_1, g_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(h_0, h_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(i_1, i_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('rt_(200)', rt_(200), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('pt_(180)', pt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('bt_(180)', bt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('kt_(180)', kt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('ct_(240)', ct_(240), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('dt_(dt)', dt_(dt), [range(90, 95, unknown), range(0, 90, true)]).
times(0, 95, 95).
```

Appendix G

Simulation of condition sequence q_4 (highlighted area shows the request of operations controller regarding reserve resources)

```
content(type(savedtrace('c:/Users/KamalB/Desktop/Thesis 11.0-final/Model/sim11.lt'))).
content(generator(app(leadsto_software, 127, [psprinting:1, showtrace:8, simalgo:11]))).
content(source(file('c:/Users/KamalB/Desktop/Thesis 11.0-final/Model/sim11.lt', [size(69542), path('c:/Users/KamalB/Desktop/Thesis 11.0-final/Model/sim11.lt'), mdate('Thu Jun 23
11:17:04 2016')]))).
content(run([date('Thu Jun 21 11:17:10 2016')])).
atom_trace('input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(1, 95, unknown), range(0, 1, true)]).
atom_trace('output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(2, 95, unknown), range(1, 2, true), range(0, 1, unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(3, 95, unknown), range(2, 3, true), range(0, 2, unknown)]).
atom_trace('output(ac)|com(ac, env, request, technical_diagnosis_adequateness, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, technical_diagnosis_adequateness,
delay(x1, x2, x3, x4), p(x))), [range(4, 95, unknown), range(3, 4, true), range(0, 3, unknown)]).
atom_trace('input(env)|com(ac, env, request, technical_diagnosis_adequateness, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, technical_diagnosis_adequateness,
delay(x1, x2, x3, x4), p(x))), [range(5, 95, unknown), range(4, 5, true), range(0, 4, unknown)]).
atom_trace('output(env)|com(env, ac, inform, technical_diagnosis_inadequate, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, technical_diagnosis_inadequate,
delay(x1, x2, x3, x4), p(x))), [range(6, 95, unknown), range(5, 6, true), range(0, 5, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, technical_diagnosis_inadequate, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, technical_diagnosis_inadequate, delay(x1,
x2, x3, x4), p(x))), [range(7, 95, unknown), range(6, 7, true), range(0, 6, unknown)]).
atom_trace('output(ac)|com(ac, oc, inform, technical_diagnosis_inadequate, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, oc, inform, technical_diagnosis_inadequate, delay(x1, x2,
x3, x4), p(x))), [range(8, 95, unknown), range(7, 8, true), range(0, 7, unknown)]).
atom_trace('input(oc)|com(ac, oc, inform, technical_diagnosis_inadequate, delay(x1, x2, x3, x4), p(x))', (input(oc)|com(ac, oc, inform, technical_diagnosis_inadequate, delay(x1, x2, x3,
x4), p(x))), [range(9, 95, unknown), range(8, 9, true), range(0, 8, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, deadheading_resources_possibilities, delay(x1, x2, x3, x4), p(x))', (output(oc)|com(oc, sc, request, deadheading_resources_possibilities,
delay(x1, x2, x3, x4), p(x))), [range(10, 95, unknown), range(9, 10, true), range(0, 9, unknown)]).
atom_trace('input(sc)|com(oc, sc, request, deadheading_resources_possibilities, delay(x1, x2, x3, x4), p(x))', (input(sc)|com(oc, sc, request, deadheading_resources_possibilities,
delay(x1, x2, x3, x4), p(x))), [range(11, 95, unknown), range(10, 11, true), range(0, 10, unknown)]).
atom_trace('output(sc)|com(sc, env, request, deadheading_resources_possibilities, delay(x1, x2, x3, x4), p(x))', (output(sc)|com(sc, env, request, deadheading_resources_possibilities,
delay(x1, x2, x3, x4), p(x))), [range(12, 95, unknown), range(11, 12, true), range(0, 11, unknown)]).
atom_trace('input(env)|com(sc, env, request, deadheading_resources_possibilities, delay(x1, x2, x3, x4), p(x))', (input(env)|com(sc, env, request, deadheading_resources_possibilities,
delay(x1, x2, x3, x4), p(x))), [range(13, 95, unknown), range(12, 13, true), range(0, 12, unknown)]).
atom_trace('output(env)|com(env, sc, inform, deadheading_resources_not_possible, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, sc, inform,
deadheading_resources_not_possible, delay(x1, x2, x3, x4), p(x))), [range(14, 95, unknown), range(13, 14, true), range(0, 13, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, deadheading_resources_not_possible, delay(x1, x2, x3, x4), p(x))', (input(sc)|com(env, sc, observe, deadheading_resources_not_possible,
delay(x1, x2, x3, x4), p(x))), [range(15, 95, unknown), range(14, 15, true), range(0, 14, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, deadheading_resources_not_possible, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(sc)|com(sc, oc, inform,
deadheading_resources_not_possible, delay(x1, x2, x3, x4), p('3B2'))), [range(16, 95, unknown), range(15, 16, true), range(0, 15, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, deadheading_resources_not_possible, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(oc)|com(sc, oc, inform, deadheading_resources_not_possible,
delay(x1, x2, x3, x4), p('3B2'))), [range(17, 95, unknown), range(16, 17, true), range(0, 16, unknown)]).
atom_trace('output(oc)|com(oc, ac, request, reserve_aircraft_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(oc)|com(oc, ac, request, reserve_aircraft_availability, delay(x1, x2,
x3, x4), p('3B2'))), [range(18, 95, unknown), range(17, 18, true), range(0, 17, unknown)]).
atom_trace('output(oc)|com(oc, cc, request, reserve_crew_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(oc)|com(oc, cc, request, reserve_crew_availability, delay(x1, x2, x3,
x4), p('3B2'))), [range(18, 95, unknown), range(17, 18, true), range(0, 17, unknown)]).
atom_trace('input(ac)|com(oc, ac, request, reserve_aircraft_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(ac)|com(oc, ac, request, reserve_aircraft_availability, delay(x1, x2, x3,
x4), p('3B2'))), [range(19, 95, unknown), range(18, 19, true), range(0, 18, unknown)]).
atom_trace('input(cc)|com(oc, cc, request, reserve_crew_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(cc)|com(oc, cc, request, reserve_crew_availability, delay(x1, x2, x3, x4),
p('3B2'))), [range(19, 95, unknown), range(18, 19, true), range(0, 18, unknown)]).
atom_trace('output(cc)|com(cc, env, request, reserve_crew_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(cc)|com(cc, env, request, reserve_crew_availability, delay(x1, x2, x3,
x4), p('3B2'))), [range(20, 95, unknown), range(19, 20, true), range(0, 19, unknown)]).
atom_trace('output(ac)|com(ac, env, request, reserve_aircraft_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(ac)|com(ac, env, request, reserve_aircraft_availability, delay(x1,
x2, x3, x4), p('3B2'))), [range(20, 95, unknown), range(19, 20, true), range(0, 19, unknown)]).
atom_trace('input(env)|com(cc, env, request, reserve_crew_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(env)|com(cc, env, request, reserve_crew_availability, delay(x1, x2, x3,
x4), p('3B2'))), [range(21, 95, unknown), range(20, 21, true), range(0, 20, unknown)]).
atom_trace('input(env)|com(ac, env, request, reserve_aircraft_availability, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(env)|com(ac, env, request, reserve_aircraft_availability, delay(x1, x2,
x3, x4), p('3B2'))), [range(21, 95, unknown), range(20, 21, true), range(0, 20, unknown)]).
atom_trace('output(env)|com(env, cc, inform, reserve_crew_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(env)|com(env, cc, inform, reserve_crew_available, delay(x1, x2, x3,
x4), p('3B2'))), [range(22, 95, unknown), range(21, 22, true), range(0, 21, unknown)]).
atom_trace('output(env)|com(env, ac, inform, reserve_aircraft_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(env)|com(env, ac, inform, reserve_aircraft_available, delay(x1, x2,
x3, x4), p('3B2'))), [range(22, 95, unknown), range(21, 22, true), range(0, 21, unknown)]).
atom_trace('input(cc)|com(env, cc, observe, reserve_crew_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(cc)|com(env, cc, observe, reserve_crew_available, delay(x1, x2, x3, x4),
p('3B2'))), [range(23, 95, unknown), range(22, 23, true), range(0, 22, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, reserve_aircraft_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(ac)|com(env, ac, observe, reserve_aircraft_available, delay(x1, x2, x3,
x4), p('3B2'))), [range(23, 95, unknown), range(22, 23, true), range(0, 22, unknown)]).
atom_trace('output(ac)|com(ac, fd, inform, reserve_aircraft_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(ac)|com(ac, fd, inform, reserve_aircraft_available, delay(x1, x2, x3, x4),
p('3B2'))), [range(24, 95, unknown), range(23, 24, true), range(0, 23, unknown)]).
atom_trace('input(fd)|com(ac, fd, inform, reserve_aircraft_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(fd)|com(ac, fd, inform, reserve_aircraft_available, delay(x1, x2, x3, x4),
p('3B2'))), [range(25, 95, unknown), range(24, 25, true), range(0, 24, unknown)]).
atom_trace('output(fd)|com(fd, env, request, ferry_time, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(fd)|com(fd, env, request, ferry_time, delay(x1, x2, x3, x4), p('3B2'))), [range(26, 95,
unknown), range(25, 26, true), range(0, 25, unknown)]).
atom_trace('input(env)|com(fd, env, request, ferry_time, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(env)|com(fd, env, request, ferry_time, delay(x1, x2, x3, x4), p('3B2'))), [range(27, 95,
unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('output(env)|com(env, fd, inform, ferry_time(240), delay(x1, x2, x3, 240), p(\'3B2\'))', (output(env)|com(env, fd, inform, ferry_time(240), delay(x1, x2, x3, 240), p('3B2'))),
[range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, ferry_time(240), delay(x1, x2, x3, 240), p(\'3B2\'))', (input(fd)|com(env, fd, observe, ferry_time(240), delay(x1, x2, x3, 240), p('3B2'))),
[range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).

atom_trace('output(fd)|com(fd, oc, inform, ferry_time(240), delay(x1, x2, x3, 240), p(\'3B2\'))', (output(fd)|com(fd, oc, inform, ferry_time(240), delay(x1, x2, x3, 240), p('3B2'))),
[range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('input(oc)|com(fd, oc, inform, ferry_time(240), delay(x1, x2, x3, 240), p(\'3B2\'))', (input(oc)|com(fd, oc, inform, ferry_time(240), delay(x1, x2, x3, 240), p('3B2'))),
[range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(oc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(x1, x2, x3, 240),
p('3B2'))), [range(32, 95, unknown), range(31, 32, true), range(0, 31, unknown)]).
atom_trace('input(sc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(sc)|com(oc, sc, request, effect_of_kt_on_tpax, delay(x1, x2, x3, 240),
p('3B2'))), [range(33, 95, unknown), range(32, 33, true), range(0, 32, unknown)]).
atom_trace('output(sc)|com(sc, env, request, transit_slack_time_ckt, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(sc)|com(sc, env, request, transit_slack_time_ckt, delay(x1, x2, x3, 240),
p('3B2'))), [range(34, 95, unknown), range(33, 34, true), range(0, 33, unknown)]).
atom_trace('output(cc)|com(cc, oc, inform, reserve_crew_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (output(cc)|com(cc, oc, inform, reserve_crew_available, delay(x1, x2, x3, x4),
p('3B2'))), [range(35, 95, unknown), range(23, 35, true), range(0, 23, unknown)]).
```

```
atom_trace('input(env)|com(sc, env, request, transit_slack_time_ckt, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(env)|com(sc, env, request, transit_slack_time_ckt, delay(x1, x2, x3, 240), p('3B2'))), [range(35, 95, unknown), range(34, 35, true), range(0, 34, unknown)]).
atom_trace('input(oc)|com(cc, oc, inform, reserve_crew_available, delay(x1, x2, x3, x4), p(\'3B2\'))', (input(oc)|com(cc, oc, inform, reserve_crew_available, delay(x1, x2, x3, x4), p('3B2'))), [range(36, 95, unknown), range(24, 36, true), range(0, 24, unknown)]).
atom_trace('output(env)|com(env, sc, inform, transit_slack_time_ckt(180), delay(x1, x2, x3, 240), p(\'3B2\'))', (output(env)|com(env, sc, inform, transit_slack_time_ckt(180), delay(x1, x2, x3, 240), p('3B2'))), [range(36, 95, unknown), range(35, 36, true), range(0, 35, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, transit_slack_time_ckt(180), delay(x1, x2, x3, 240), p(\'3B2\'))', (input(sc)|com(env, sc, observe, transit_slack_time_ckt(180), delay(x1, x2, x3, 240), p('3B2'))), [range(37, 95, unknown), range(36, 37, true), range(0, 36, unknown)]).
atom_trace('output(sc)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(sc)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p('3B2'))), [range(38, 95, unknown), range(37, 38, true), range(0, 37, unknown)]).
atom_trace('input(fd)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(fd)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p('3B2'))), [range(39, 95, unknown), range(38, 39, true), range(0, 38, unknown)]).
atom_trace('output(fd)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(fd)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p('3B2'))), [range(40, 95, unknown), range(39, 40, true), range(0, 39, unknown)]).
atom_trace('input(env)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(env)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(x1, x2, x3, 240), p('3B2'))), [range(41, 95, unknown), range(40, 41, true), range(0, 40, unknown)]).
atom_trace('output(env)|com(env, fd, inform, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(env)|com(env, fd, inform, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p('3B2'))), [range(42, 95, unknown), range(41, 42, true), range(0, 41, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(fd)|com(env, fd, observe, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p('3B2'))), [range(43, 95, unknown), range(42, 43, true), range(0, 42, unknown)]).
atom_trace('output(fd)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(fd)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p('3B2'))), [range(44, 95, unknown), range(43, 44, true), range(0, 43, unknown)]).
atom_trace('input(sc)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(sc)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(x1, x2, x3, 240), p('3B2'))), [range(45, 95, unknown), range(44, 45, true), range(0, 44, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, dispatch_reserve_aircraft_connects_tpax_unsuccessful, delay(x1, x2, x3, 240), p(\'3B2\'))', (output(sc)|com(sc, oc, inform, dispatch_reserve_aircraft_connects_tpax_unsuccessful, delay(x1, x2, x3, 240), p('3B2'))), [range(46, 95, unknown), range(45, 46, true), range(0, 45, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, dispatch_reserve_aircraft_connects_tpax_unsuccessful, delay(x1, x2, x3, 240), p(\'3B2\'))', (input(oc)|com(sc, oc, inform, dispatch_reserve_aircraft_connects_tpax_unsuccessful, delay(x1, x2, x3, 240), p('3B2'))), [range(47, 95, unknown), range(46, 47, true), range(0, 46, unknown)]).
atom_trace('recovery(\'RS16\')', recovery('RS16'), [range(48, 95, unknown), range(47, 48, true), range(0, 47, unknown)]).
atom_trace(a_0, a_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(b_1, b_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(c_1, c_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(d_0, d_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(e_0, e_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(f_0, f_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(g_1, g_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(h_1, h_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(i_1, i_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('rt_(240)', rt_(240), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('dt_(180)', dt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('bt_(180)', bt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('kt_(240)', kt_(240), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('ct_(180)', ct_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('pt_(180)', pt_(180), [range(90, 95, unknown), range(0, 90, true)]).
times(0, 95, 95).
```

Appendix H

Simulation with condition sequence p_4 (highlighted area shows the request of operations controller regarding rebooking opportunities)

```
content(type(savedtrace('c:/Users/KamalB/Desktop/Thesis 11.0-final/Model/sim11.lt'))).
content(generator(app(leadsto_software, 127, [psprinting:1, showtrace:8, simalgo:11]))).
content(source(file('c:/Users/KamalB/Desktop/Thesis 11.0-final/Model/sim11.lt', [size(69542), path('c:/Users/KamalB/Desktop/Thesis 11.0-final/Model/sim11.lt'), mdate('Thu Jun 23
11:37:23 2016')]))).
content(run([date('Thu Jun 21 11:37:29 2016')])).
atom_trace('recovery(\'RS6\', 180, 240, x2, 180, x4)', recovery('RS6', 180, 240, x2, 180, x4), [range(57, 95, true), range(0, 57, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(1, 95, unknown), range(0, 1, true)]).
atom_trace('output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(2, 95, unknown), range(1, 2, true), range(0, 1, unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac, inform, aircraft_tua_has_mechanical_problem,
delay(x1, x2, x3, x4), p(x))), [range(3, 95, unknown), range(2, 3, true), range(0, 2, unknown)]).
atom_trace('output(ac)|com(ac, env, request, technical_diagnosis_adequateness, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, technical_diagnosis_adequateness,
delay(x1, x2, x3, x4), p(x))), [range(4, 95, unknown), range(3, 4, true), range(0, 3, unknown)]).
atom_trace('input(env)|com(ac, env, request, technical_diagnosis_adequateness, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, technical_diagnosis_adequateness,
delay(x1, x2, x3, x4), p(x))), [range(5, 95, unknown), range(4, 5, true), range(0, 4, unknown)]).
atom_trace('output(env)|com(env, ac, inform, technical_diagnosis_adequate, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, technical_diagnosis_adequate, delay(x1,
x2, x3, x4), p(x))), [range(6, 95, unknown), range(5, 6, true), range(0, 5, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, technical_diagnosis_adequate, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, technical_diagnosis_adequate, delay(x1, x2,
x3, x4), p(x))), [range(7, 95, unknown), range(6, 7, true), range(0, 6, unknown)]).
atom_trace('output(ac)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4),
p(x))), [range(8, 95, unknown), range(7, 8, true), range(0, 7, unknown)]).
atom_trace('input(env)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, spare_parts_availability, delay(x1, x2, x3, x4),
p(x))), [range(9, 95, unknown), range(8, 9, true), range(0, 8, unknown)]).
atom_trace('output(env)|com(env, ac, inform, spare_parts_available, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, spare_parts_available, delay(x1, x2, x3, x4), p(x))),
[range(10, 95, unknown), range(9, 10, true), range(0, 9, unknown)]).

atom_trace('input(ac)|com(env, ac, observe, spare_parts_available, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, spare_parts_available, delay(x1, x2, x3, x4), p(x))),
[range(11, 95, unknown), range(10, 11, true), range(0, 10, unknown)]).

atom_trace('output(ac)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4),
p(x))), [range(12, 95, unknown), range(11, 12, true), range(0, 11, unknown)]).
atom_trace('input(fd)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(ac, fd, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))),
[range(13, 95, unknown), range(12, 13, true), range(0, 12, unknown)]).
atom_trace('output(fd)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4),
p(x))), [range(14, 95, unknown), range(13, 14, true), range(0, 13, unknown)]).
atom_trace('input(env)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4), p(x))', (input(env)|com(fd, env, request, wx_pattern_favourability, delay(x1, x2, x3, x4),
p(x))), [range(15, 95, unknown), range(14, 15, true), range(0, 14, unknown)]).
atom_trace('output(env)|com(env, fd, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (output(env)|com(env, fd, inform, wx_pattern_favourable, delay(x1, x2, x3, x4),
p(x))), [range(16, 95, unknown), range(15, 16, true), range(0, 15, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))),
[range(17, 95, unknown), range(16, 17, true), range(0, 16, unknown)]).

atom_trace('output(fd)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (output(fd)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))),
[range(18, 95, unknown), range(17, 18, true), range(0, 17, unknown)]).
atom_trace('input(ac)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))', (input(ac)|com(fd, ac, inform, wx_pattern_favourable, delay(x1, x2, x3, x4), p(x))),
[range(19, 95, unknown), range(18, 19, true), range(0, 18, unknown)]).
atom_trace('output(ac)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))', (output(ac)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))), [range(20, 95,
unknown), range(19, 20, true), range(0, 19, unknown)]).
atom_trace('input(env)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))', (input(env)|com(ac, env, request, repair_time, delay(x1, x2, x3, x4), p(x))), [range(21, 95,
unknown), range(20, 21, true), range(0, 20, unknown)]).
atom_trace('output(env)|com(env, ac, inform, repair_time(240), delay(240, x2, x3, x4), p(x))', (output(env)|com(env, ac, inform, repair_time(240), delay(240, x2, x3, x4), p(x))),
[range(22, 95, unknown), range(21, 22, true), range(0, 21, unknown)]).
atom_trace('input(ac)|com(env, ac, observe, repair_time(240), delay(240, x2, x3, x4), p(x))', (input(ac)|com(env, ac, observe, repair_time(240), delay(240, x2, x3, x4), p(x))), [range(23,
95, unknown), range(22, 23, true), range(0, 22, unknown)]).
atom_trace('output(ac)|com(ac, oc, inform, repair_time(240), delay(240, x2, x3, x4), p(x))', (output(ac)|com(ac, oc, inform, repair_time(240), delay(240, x2, x3, x4), p(x))), [range(24, 95,
unknown), range(23, 24, true), range(0, 23, unknown)]).
atom_trace('input(oc)|com(ac, oc, inform, repair_time(240), delay(240, x2, x3, x4), p(x))', (input(oc)|com(ac, oc, inform, repair_time(240), delay(240, x2, x3, x4), p(x))), [range(25, 95,
unknown), range(24, 25, true), range(0, 24, unknown)]).
atom_trace('output(oc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(240, x2, x3, x4), p(x))', (output(oc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(240, x2, x3, x4), p(x))),
[range(26, 95, unknown), range(25, 26, true), range(0, 25, unknown)]).

atom_trace('output(oc)|com(oc, cc, request, effect_of_rt_on_crew, delay(240, x2, x3, x4), p(x))', (output(oc)|com(oc, cc, request, effect_of_rt_on_crew, delay(240, x2, x3, x4), p(x))),
[range(26, 95, unknown), range(25, 26, true), range(0, 25, unknown)]).

atom_trace('input(sc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(240, x2, x3, x4), p(x))', (input(sc)|com(oc, sc, request, effect_of_rt_on_tpax, delay(240, x2, x3, x4), p(x))),
[range(27, 95, unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('input(cc)|com(oc, cc, request, effect_of_rt_on_crew, delay(240, x2, x3, x4), p(x))', (input(cc)|com(oc, cc, request, effect_of_rt_on_crew, delay(240, x2, x3, x4), p(x))),
[range(27, 95, unknown), range(26, 27, true), range(0, 26, unknown)]).
atom_trace('output(cc)|com(cc, env, request, crew_duty_slack_time, delay(240, x2, x3, x4), p(x))', (output(cc)|com(cc, env, request, crew_duty_slack_time, delay(240, x2, x3, x4),
p(x))), [range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('output(sc)|com(sc, env, request, transit_slack_time_crt, delay(240, x2, x3, x4), p(x))', (output(sc)|com(sc, env, request, transit_slack_time_crt, delay(240, x2, x3, x4), p(x))),
[range(28, 95, unknown), range(27, 28, true), range(0, 27, unknown)]).
atom_trace('input(env)|com(cc, env, request, crew_duty_slack_time, delay(240, x2, x3, x4), p(x))', (input(env)|com(cc, env, request, crew_duty_slack_time, delay(240, x2, x3, x4),
p(x))), [range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).
atom_trace('input(env)|com(sc, env, request, transit_slack_time_crt, delay(240, x2, x3, x4), p(x))', (input(env)|com(sc, env, request, transit_slack_time_crt, delay(240, x2, x3, x4), p(x))),
[range(29, 95, unknown), range(28, 29, true), range(0, 28, unknown)]).
atom_trace('output(env)|com(env, cc, inform, crew_duty_slack_time(180), delay(240, x2, x3, x4), p(x))', (output(env)|com(env, cc, inform, crew_duty_slack_time(180), delay(240, x2,
x3, x4), p(x))), [range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('output(env)|com(env, sc, inform, transit_slack_time_crt(180), delay(240, x2, x3, x4), p(x))', (output(env)|com(env, sc, inform, transit_slack_time_crt(180), delay(240, x2,
x3, x4), p(x))), [range(30, 95, unknown), range(29, 30, true), range(0, 29, unknown)]).
atom_trace('input(cc)|com(env, cc, observe, crew_duty_slack_time(180), delay(240, x2, x3, x4), p(x))', (input(cc)|com(env, cc, observe, crew_duty_slack_time(180), delay(240, x2, x3,
x4), p(x))), [range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, transit_slack_time_crt(180), delay(240, x2, x3, x4), p(x))', (input(sc)|com(env, sc, observe, transit_slack_time_crt(180), delay(240, x2, x3,
x4), p(x))), [range(31, 95, unknown), range(30, 31, true), range(0, 30, unknown)]).
atom_trace('output(cc)|com(cc, env, request, reserve_crew_availability, delay(240, x2, x3, x4), p(x))', (output(cc)|com(cc, env, request, reserve_crew_availability, delay(240, x2, x3, x4),
p(x))), [range(32, 95, unknown), range(31, 32, true), range(0, 31, unknown)]).
atom_trace('output(sc)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))', (output(sc)|com(sc, fd, request,
```

```
organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))), [range(32, 95, unknown), range(31, 32, true), range(0, 31, unknown)]).
atom_trace('input(env)|com(cc, env, request, reserve_crew_availability, delay(240, x2, x3, x4), p(x))', (input(env)|com(cc, env, request, reserve_crew_availability, delay(240, x2, x3, x4),
p(x))), [range(33, 95, unknown), range(32, 33, true), range(0, 32, unknown)]).
atom_trace('input(fd)|com(sc, fd, request, organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))', (input(fd)|com(sc, fd, request, organizing_cxn_measures_possibilities,
delay(240, x2, x3, x4), p(x))), [range(33, 95, unknown), range(32, 33, true), range(0, 32, unknown)]).
atom_trace('output(env)|com(env, cc, inform, reserve_crew_available, delay(240, x2, x3, x4), p(x))', (output(env)|com(env, cc, inform, reserve_crew_available, delay(240, x2, x3, x4),
p(x))), [range(34, 95, unknown), range(33, 34, true), range(0, 33, unknown)]).
atom_trace('output(fd)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))', (output(fd)|com(fd, env, request,
organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))), [range(34, 95, unknown), range(33, 34, true), range(0, 33, unknown)]).
atom_trace('input(cc)|com(env, cc, observe, reserve_crew_available, delay(240, x2, x3, x4), p(x))', (input(cc)|com(env, cc, observe, reserve_crew_available, delay(240, x2, x3, x4), p(x))),
[range(35, 95, unknown), range(34, 35, true), range(0, 34, unknown)]).
atom_trace('input(env)|com(fd, env, request, organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))', (input(env)|com(fd, env, request,
organizing_cxn_measures_possibilities, delay(240, x2, x3, x4), p(x))), [range(35, 95, unknown), range(34, 35, true), range(0, 34, unknown)]).
atom_trace('output(cc)|com(cc, sc, request, deadheading_possibilities, delay(240, x2, x3, x4), p(x))', (output(cc)|com(cc, sc, request, deadheading_possibilities, delay(240, x2, x3, x4),
p(x))), [range(36, 95, unknown), range(35, 36, true), range(0, 35, unknown)]).
atom_trace('output(env)|com(env, fd, inform, organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))', (output(env)|com(env, fd, inform,
organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))), [range(36, 95, unknown), range(35, 36, true), range(0, 35, unknown)]).
atom_trace('input(fd)|com(env, fd, observe, organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))', (input(fd)|com(env, fd, observe,
organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))), [range(37, 95, unknown), range(36, 37, true), range(0, 36, unknown)]).
atom_trace('input(sc)|com(cc, sc, request, deadheading_possibilities, delay(240, x2, x3, x4), p(x))', (input(sc)|com(cc, sc, request, deadheading_possibilities, delay(240, x2, x3, x4),
p(x))), [range(37, 95, unknown), range(36, 37, true), range(0, 36, unknown)]).
atom_trace('output(fd)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))', (output(fd)|com(fd, sc, inform,
organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))), [range(38, 95, unknown), range(37, 38, true), range(0, 37, unknown)]).
atom_trace('output(sc)|com(sc, env, request, deadheading_possibilities, delay(240, x2, x3, x4), p(x))', (output(sc)|com(sc, env, request, deadheading_possibilities, delay(240, x2, x3,
x4), p(x))), [range(38, 95, unknown), range(37, 38, true), range(0, 37, unknown)]).
atom_trace('input(env)|com(sc, env, request, deadheading_possibilities, delay(240, x2, x3, x4), p(x))', (input(env)|com(sc, env, request, deadheading_possibilities, delay(240, x2, x3,
x4), p(x))), [range(39, 95, unknown), range(38, 39, true), range(0, 38, unknown)]).
atom_trace('input(sc)|com(fd, sc, inform, organizing_cxn_measures_not_possible, delay(240, x2, x3, x4), p(x))', (input(sc)|com(fd, sc, inform, organizing_cxn_measures_not_possible,
delay(240, x2, x3, x4), p(x))), [range(39, 95, unknown), range(38, 39, true), range(0, 38, unknown)]).
atom_trace('output(env)|com(env, sc, inform, deadheading_not_possible, delay(240, x2, x3, x4), p(x))', (output(env)|com(env, sc, inform, deadheading_not_possible, delay(240, x2, x3,
x4), p(x))), [range(40, 95, unknown), range(39, 40, true), range(0, 39, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, deadheading_not_possible, delay(240, x2, x3, x4), p(x))', (input(sc)|com(env, sc, observe, deadheading_not_possible, delay(240, x2, x3, x4),
p(x))), [range(41, 95, unknown), range(40, 41, true), range(0, 40, unknown)]).
atom_trace('output(sc)|com(sc, cc, inform, deadheading_not_possible, delay(240, x2, x3, x4), p(x))', (output(sc)|com(sc, cc, inform, deadheading_not_possible, delay(240, x2, x3, x4),
p(x))), [range(42, 95, unknown), range(41, 42, true), range(0, 41, unknown)]).
atom_trace('input(cc)|com(sc, cc, inform, deadheading_not_possible, delay(240, x2, x3, x4), p(x))', (input(cc)|com(sc, cc, inform, deadheading_not_possible, delay(240, x2, x3, x4),
p(x))), [range(43, 95, unknown), range(42, 43, true), range(0, 42, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, tpax_affected_by_rt, delay(240, x2, x3, x4), p(x))', (output(sc)|com(sc, oc, inform, tpax_affected_by_rt, delay(240, x2, x3, x4), p(x))),
[range(74, 95, unknown), range(39, 74, true), range(0, 39, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, tpax_affected_by_rt, delay(240, x2, x3, x4), p(x))', (input(oc)|com(sc, oc, inform, tpax_affected_by_rt, delay(240, x2, x3, x4), p(x))), [range(75,
95, unknown), range(40, 75, true), range(0, 40, unknown)]).
atom_trace('output(cc)|com(cc, oc, inform, deadheading_not_possible, delay(240, x2, x3, x4), p(\'3A3\'))', (output(cc)|com(cc, oc, inform, deadheading_not_possible, delay(240, x2, x3,
x4), p('3A3'))), [range(88, 95, unknown), range(43, 88, true), range(0, 43, unknown)]).
atom_trace('input(oc)|com(cc, oc, inform, deadheading_not_possible, delay(240, x2, x3, x4), p(\'3A3\'))', (input(oc)|com(cc, oc, inform, deadheading_not_possible, delay(240, x2, x3,
x4), p('3A3'))), [range(89, 95, unknown), range(44, 89, true), range(0, 44, unknown)]).
atom_trace(a_1, a_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(b_1, b_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(c_1, c_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(d_0, d_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(e_0, e_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(f_0, f_0, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(g_1, g_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(h_1, h_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace(i_1, i_1, [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('rt_(240)', rt_(240), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('dt_(180)', dt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('bt_(180)', bt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('kt_(240)', kt_(240), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('ct_(180)', ct_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('pt_(180)', pt_(180), [range(90, 95, unknown), range(0, 90, true)]).
atom_trace('output(oc)|com(oc, sc, request, rebooking_possibilities, delay(240, x2, x3, x4), p(\'3A3\'))', (output(oc)|com(oc, sc, request, rebooking_possibilities, delay(240, x2, x3, x4),
p('3A3'))), [range(90, 95, unknown), range(45, 90, true), range(0, 45, unknown)]).
atom_trace('input(sc)|com(oc, sc, request, rebooking_possibilities, delay(240, x2, x3, x4), p(\'3A3\'))', (input(sc)|com(oc, sc, request, rebooking_possibilities, delay(240, x2, x3, x4),
p('3A3'))), [range(91, 95, unknown), range(46, 91, true), range(0, 46, unknown)]).
atom_trace('output(env)|com(env, sc, inform, rebooking_possible, delay(240, x2, x3, x4), p(\'3A3\'))', (output(env)|com(env, sc, inform, rebooking_possible, delay(240, x2, x3, x4),
p('3A3'))), [range(91, 95, unknown), range(49, 91, true), range(0, 49, unknown)]).
atom_trace('output(env)|com(env, sc, inform, rebooking_time(180), delay(240, x2, x3, x4), p(\'3A3\'))', (output(env)|com(env, sc, inform, rebooking_time(180), delay(240, x2, x3, x4),
p('3A3'))), [range(91, 95, unknown), range(53, 91, true), range(0, 53, unknown)]).
atom_trace('output(env)|com(env, sc, inform, transit_slack_time_cbt(180), delay(240, x2, x3, x4), p(\'3A3\'))', (output(env)|com(env, sc, inform, transit_slack_time_cbt(180),
delay(240, x2, x3, x4), p('3A3'))), [range(91, 95, unknown), range(53, 91, true), range(0, 53, unknown)]).
atom_trace('output(sc)|com(sc, env, request, rebooking_possibilities, delay(240, x2, x3, x4), p(\'3A3\'))', (output(sc)|com(sc, env, request, rebooking_possibilities, delay(240, x2, x3,
x4), p('3A3'))), [range(92, 95, unknown), range(47, 92, true), range(0, 47, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, rebooking_possible, delay(240, x2, x3, x4), p(\'3A3\'))', (input(sc)|com(env, sc, observe, rebooking_possible, delay(240, x2, x3, x4),
p('3A3'))), [range(92, 95, unknown), range(50, 92, true), range(0, 50, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, transit_slack_time_cbt(180), delay(240, x2, x3, x4), p(\'3A3\'))', (input(sc)|com(env, sc, observe, transit_slack_time_cbt(180), delay(240,
x2, x3, x4), p('3A3'))), [range(92, 95, unknown), range(54, 92, true), range(0, 54, unknown)]).
atom_trace('input(sc)|com(env, sc, observe, rebooking_time(180), delay(240, x2, x3, x4), p(\'3A3\'))', (input(sc)|com(env, sc, observe, rebooking_time(180), delay(240, x2, x3, x4),
p('3A3'))), [range(92, 95, unknown), range(54, 92, true), range(0, 54, unknown)]).
atom_trace('input(env)|com(sc, env, request, rebooking_possibilities, delay(240, x2, x3, x4), p(\'3A3\'))', (input(env)|com(sc, env, request, rebooking_possibilities, delay(240, x2, x3,
x4), p('3A3'))), [range(93, 95, unknown), range(48, 93, true), range(0, 48, unknown)]).
atom_trace('output(sc)|com(sc, env, request, rebooking_time, delay(240, x2, x3, x4), p(\'3A3\'))', (output(sc)|com(sc, env, request, rebooking_time, delay(240, x2, x3, x4), p('3A3'))),
[range(93, 95, unknown), range(51, 93, true), range(0, 51, unknown)]).
atom_trace('output(sc)|com(sc, env, request, transit_slack_time_cbt, delay(240, x2, x3, x4), p(\'3A3\'))', (output(sc)|com(sc, env, request, transit_slack_time_cbt, delay(240, x2, x3, x4),
p('3A3'))), [range(93, 95, unknown), range(51, 93, true), range(0, 51, unknown)]).
atom_trace('output(sc)|com(sc, oc, inform, rebooking_connects_tpax_successful, delay(240, x2, 180, x4), p(\'3A3\'))', (output(sc)|com(sc, oc, inform,
rebooking_connects_tpax_successful, delay(240, x2, 180, x4), p('3A3'))), [range(93, 95, unknown), range(55, 93, true), range(0, 55, unknown)]).
atom_trace('input(env)|com(sc, env, request, rebooking_time, delay(240, x2, x3, x4), p(\'3A3\'))', (input(env)|com(sc, env, request, rebooking_time, delay(240, x2, x3, x4), p('3A3'))),
[range(94, 95, unknown), range(52, 94, true), range(0, 52, unknown)]).
atom_trace('input(env)|com(sc, env, request, transit_slack_time_cbt, delay(240, x2, x3, x4), p(\'3A3\'))', (input(env)|com(sc, env, request, transit_slack_time_cbt, delay(240, x2, x3, x4),
p('3A3'))), [range(94, 95, unknown), range(52, 94, true), range(0, 52, unknown)]).
atom_trace('input(oc)|com(sc, oc, inform, rebooking_connects_tpax_successful, delay(240, x2, 180, x4), p(\'3A3\'))', (input(oc)|com(sc, oc, inform,
rebooking_connects_tpax_successful, delay(240, x2, 180, x4), p('3A3'))), [range(94, 95, unknown), range(56, 94, true), range(0, 56, unknown)]).
times(0, 95, 95).
```