An agent-based model for air transportation to capture network effects in assessing delay management mechanisms

Gérald Gurtner[a,1], Luis Delgado[a,*,1] and Damir Valput[b]

[a] School of Architecture and Cities, University of Westminster, London, United Kingdom
[b] Innaxis, Madrid, Spain

ABSTRACT

This article presents some of the results on the implementation of a decentralised delay management process that we call 4D trajectory adjustments (4DTA) obtained with Mercury, a stochastic agent-based model. The model operates within a strong agent paradigm at the level of individual flights and passengers. It includes a realistic cost model for the airlines, allowing us to have a good tactical choice model and excellent estimation of airspace user costs. Due to the inclusion of different stakeholders, including passengers, and various processes – like aircraft turnaround or passenger reaccommodation – it is able to catch European-wide network effects that are inaccessible to other models. It was used to study the 4DTA process, a blend of 'wait for passengers and tactical speed adjustments', which is shown to have a significant impact on the system. Thanks to the detailed output of the model, we are able to breakdown the effect for different classes of flights and passengers, and show that important trade-offs exist in terms of delays and costs. In particular, the introduction of such mechanism could be detrimental to non-connecting passengers, especially at secondary airports.

1. Introduction

Before the impact of COVID unfolded, the air transportation system, and in particular Air Traffic Management (ATM), was under a lot of strain. In Europe, departure delay increased from 11.3 minutes of average delay per flight in 2016 to 12.4 minutes in 2017, 14.7 minutes in 2018, and improved to 13.1 minutes in 2019 (with an average delay per delayed flight of over 28 minutes, from 27 minutes per delayed flight experienced in 2016) (EUROCONTROL, 2018, 2019a, 2020). Despite this latest improvement, 2019 presented the third worst year in terms of delay performance (average departing delay per flight) in the last 10 years (just behind 2010 and 2018), with en-route Air Traffic Flow Management (ATFM) delays during the summer season remaining a significant contributor to airlines delay (EUROCONTROL, 2020). In addition to capacity issues, flights incur delays due to a number of causes, which can then propagate through various channels to other flights: reactionary delay represented over 43% of the total departing delay experienced by airlines in 2019 (EUROCONTROL, 2020). These delays carry a certain cost for an airline, but this cost is very contextual, as it depends on the type of flights, the number of passengers carried, whether the latter are connecting at the destination, the downstream flights, etc. (Cook and Tanner, 2015). This raises the need to put in place delay management techniques that take into account these intricate effects.

In Europe, SESAR is dedicated to reduce the delay and thus put in place delay management systems and procedures (SESAR, 2020). Among these, one can cite the Extended Arrival Manager (E-AMAN) (PJ.01.01), where arrival queues at airports are built more in advance and thus can accommodate more traffic (European Commission, 2014), or User-Driven Prioritisation Process (UDPP) (PJ.07.02), among many others (SESAR, 2018). Overall the goal is to foster information sharing among stakeholders, manage more cleverly the various constraints in the airspace and find a compromise with the airspace users' needs and intents. This is embedded in the concept of 4D trajectory, which represents a contract among stakeholders.

These delay management solutions all require some assessment prior to their deployment. This is done following the SESAR innovation pipeline. However, by design, solutions are tested locally and in very specific conditions, and some of these solutions may have systemic effects that the development phase may not have identified at first due to certain degree of inter-dependency between the solution and the rest of the system.

* Corresponding author
ORCID(s): 0000-0003-2006-4653 (G. Gurtner); 0000-0003-4613-4277 (L. Delgado)
[1] These authors have contributed equally to the present work.

In this paper, we focus on a particular delay management process that we call '4 Dimensional Trajectory Adjustments (4DTA)', which enables airlines to use up to date information to actively delay departing flights to wait for connecting passengers, and to tactically modify their trajectory to recover delay. In the case study presented, we focus on the time dimension of 4DTA (adjusting cost index of flights dynamically and hence modifying the speed/time domain), leaving space adjustments to the trajectory to be accommodated in the future. In order to assess the impact of this mechanism network-wide, we use Mercury, a model able to assess the impact of delay management mechanisms in a holistic manner, taking into account the high non-linearity in the agents' behaviours and cascading effects with focus on airline costs, as well as flights and passengers metrics.

The main goal of this study is thus to assess the impact of the implementation[2] of this mechanism on the whole system, capturing delay propagation channels, through the use of an Agent-Based Model (ABM). Furthermore, we use "impact" in a broad meaning, as we are interested in seeing how other stakeholders besides airspace users are affected by the mechanisms. This in particular includes passengers, who often are not explicitly included in performance schemes, as highlighted in Section 2 .

The paper is structured as follows. First, we start with an overview of related work on current 4D trajectory management for delay management, agent-based modelling in air transportation, and performance scheme in Section 2, upon which we drew inspiration for our work. In Section 3 we explicit the scope of the model, draw some high-level requirements in order to reach our goals, and we describe how the design methodology (Gaia) that has been used to create the Mercury model leads to the creation of "roles" and "agents" in the model. Particularities of the simulator scheduling engine and messages interchange between agents are also described. Section 4 presents in more details the inner workings of the agents in the model. ABMs require an extensive calibration and validation process, and this is described in Section 5. The model is able to track flights and passengers in Europe (including explicit reactionary delay and passenger connections), which means that very low-level indicators can be obtained from the model, and the type of outcome that can be produced are also summarised here. Finally, we present the results of the simulations of the 4DTA mechanism in Section 6, which also contains an indication on the computational complexity of the model. The paper closes with the conclusions in Section 7.

2. Related work

In this section we present research work which ties into ours and has served us as basis and inspiration for developing our ABM, the mechanisms implemented in it, and the study of the emergence of complex hidden network phenomena.

2.1. Delay and 4D trajectory management

During the dispatching process of a flight, a strategically defined cost index is used to balance required fuel and time. The cost index represents the relationship between cost of time and cost of fuel used for a given flight. This enables dispatching and Flight Management System (FMS) to transform time into equivalent fuel, considering in this way the trade-off between time and fuel as single objective optimisation problem. A low cost index will instruct the aircraft to follow a trajectory that minimises fuel consumption; while higher values, conversely, will reduce time even at expenses of using more fuel (Boeing, 2017). The concept of Dynamic Cost Indexing (DCI) entails modifying the value of this parameter during the different flight phases as a function of the current situation (Cook et al., 2009). Some commercial systems, such as PACE's 'Pacelab flight profile optimizer' or Jepessen's 'FliteDeck', operate in electronic flight bags to provide updates to the crew on the trajectories to be flown (PACE, 2020; Jepessen, 2020). They aim at optimising the flight profile but do not consider an explicit representation of the cost of delay. This representation of the expected costs to tactically analyse the trade-off between cost of fuel and cost of delay is addressed in other research such as Pilot3 Consortium (2020).

Airlines' operations focus on minimising costs while maintaining flights on schedule and managing, not only flight delay but also passengers delay and connections due to their related costs – both hard (e.g., re-booking costs) and soft (i.e., potential future market loss). Current low levels of fuel cost might incentivise airlines to use higher cost indexes to recover delay, however, other costs should also be considered, such as the impact of passengers' compensation Regulation 261, maintenance or crew costs (European Commission, 2004; Cook and Tanner, 2015). In this context, other alternatives could be considered, such as actively delaying a flight for Waiting for (connecting) Passengers (WfP). The concept of coupling DCI and WfP was first explored in the project CASSIOPEIA (Molina et al., 2014) and refined

[2] Here, by "implementation", we refer to the implementation of some rules in a simulator, i.e., a piece of code runnable by a computer, and not the operational implementation, as sometimes implied in air traffic management.

in DCI-4HD2D (Cassiopeia project, 2016). Those models focused on a single-airport simulation aiming at optimising operations at a hub equipped with a collaborative extended arrival manager (E-AMAN). In this paper, we take those concepts one step further by deploying those mechanisms at a network level, instead of at a single airport, and with different uptakes while considering the wider ATM system.

Besides the tactical use of speed adjustment for airlines delay management, this technique has previously been considered in the literature for tactically managing capacity demand imbalances (Delgado and Prats, 2012; Delgado et al., 2013; Xu et al., 2017, 2020; Jones et al., 2018). However, in these cases, the approaches tend to rely on centralised solutions which do not to consider the coupling with other delay management initiatives (such as waiting for connecting passengers).

There is a tendency towards decentralised solutions for the adjustment of trajectories when dealing with ATFM issues, e.g., by assigning delay to manage capacity-demand imbalance, (Kistan et al., 2017). The research presented in this paper contributes toward these concepts.

2.2. Performance scheme

The air transportation system management is notoriously focused on some specific aspects. For instance, passenger delays are sometimes overlooked by the regulators and/or the actors of the system. Airlines monitor closely On-Time Performance (OTP), which is a standard Key Performance Indicator (KPI), and therefore ATM tends to focus on flight-centric metrics. Official Key Performance Indicators from SESAR still do not include passenger delay explicitly (SESAR, 2020). Official reports on delays from the Performance Review Body (Performance Review Body, 2018) or EUROCONTROL's CODA digest report (EUROCONTROL, 2019b) do not mention passenger delays. Note that this is an institutional issue, in other words it is not part of the mission of these bodies to estimate passenger delay: the PRB focuses on ANSP's efficiently and CODA on flights' data. However, we would like to point out the important fact that there is no similar body, which we know of, whose mission is to report on passenger delay in Europe. In other words, passenger delay is not included in any major performance scheme.

Nevertheless, there has been an effort in the past years (Cook et al., 2012, 2016) to deliver metrics which focus on the passengers themselves. This includes monitoring their full delay while taking into account, in particular, their connections. The European Regulation 261 for example is designed to compensate passengers based on their final arrival delay (European Commission, 2004). There is also a trend to consider the passenger journey as a whole, hence monitoring their door-to-door journey instead of just their gate-to-gate trip (High Level Group on Aviation Research, 2011). Even without the complete journey, the exact passenger gate-to-gate time, including connections, is seldom measured or monitored. A tool like RNEST for instance[3] is very detailed in terms of flight delay, but does not consider passengers yet[4], nor how passengers' full itineraries impact airlines' decision making processes in case of disruption.

2.3. Agent-based modelling

(Kravaris et al., 2018), The use of an ABM as a base for a network-approach is fairly novel in architecture design in the ATM domain. ABM themselves have been used in the past in ATM (Bouarfa et al., 2013; Stroeve et al., 2013; Molina et al., 2014; Delgado et al., 2017). They are typically used to test some limited operational improvements like free-routing, doing experiments a in fully-controlled environment (the computer). An example of an ABM model from the field of transport is the work of Velaga et al. (2012) who proposed a novel approach for developing an agent-based transport system platform in rural areas that would be flexible, from passengers' point of view, in choosing routes, times, modes of transport, service provider and payment systems, relying on artificial intelligence for decision making.

The ABM paradigm is particularly suitable to represent diverse (possibly conflicting) goals of airport stakeholders, their preferences, values and interactions; with the goal of determining optimal decisions or solutions by a group of agents as it has been done in Mercury. While they do not develop an agent-based model, in de Arruda et al. (2015), a game theory approach is introduced based on a two-sided market mechanism to determine slot allocation in A-CDM, taking into account preferences of all the essential stakeholders. This work studies the effects that emerge as a results of interactions of different actors of the system and thus is beneficial to understanding the potential of agent-based modelling approach.

[3] https://www.eurocontrol.int/simulations (accessed June 2021)

[4] EUROCONTROL has actually been developing some passenger capabilities for RNEST within some SESAR ER projects while this article was being reviewed.

Kravaris et al. (2018) explored the use of a multiagent system with reinforcement learning for solving demand-capacity imbalances by adjusting departing time of flights with the objective of reducing hotspots (congested areas) while minimising the average delay experienced by flights. This model does not consider cost functions nor the interactions of the system beyond the required to solve these capacity-demand imbalances.

Several research projects modelled passenger flows in airport terminals as multiagent systems (Enciso et al., 2016; Schultz and Fricke, 2011). Some of these studies focused on agent-based modelling the behaviour of individual passengers (Chen et al., 2018) and groups of passengers (Cheng et al., 2014) and their impact on the passenger flow. Boarding strategies were identified as one of the essential factors influencing the efficiency of the turnaround process and the efficiency of various strategies was analysed, simulated and tested in fieldwork by Schultz et al. (2013); Schultz (2018). Overall, this kind of models are particularly suited to capture human interactions and coordination, as also shown in Bouarfa et al. (2016).

ABMs have also successfully been used in the field of air transport to consider airspace safety aspects, i.e., conflict detection and resolution (Blom and Bakker, 2015; Gurtner et al., 2017). This kind of models, featuring limited information and potentially sub-optimal decision-making processes, are naturally well suited to this problem. In Mercury these flight-to-flight airborne low-level interactions for conflict management and resolution are captured by aggregated distributions and hence not explicitly considered.

The exploratory project POEM (Cook et al., 2012) designed a model able to capture passenger-centric metrics within Europe using passenger itineraries, delay distributions and simple rules for missed connections. The results showed than passenger metrics were sometimes behaving in the opposite direction with respect to flight metrics, questioning the common focus on flight delays. The model was reused and expanded in SESAR project ComplexityCosts to focus on the ATM resilience from a cost perspective. Passenger delay is required to estimate the full cost of delay for airspace users (Cook and Tanner, 2015) and hence a full mobility model was required for ComplexityCosts (Cook et al., 2016). This model was expanded again in the DATASET2050 project, to include door-to-door journeys. It was one of the first attempts to estimate the actual trip time distribution at a European scale (Kluge et al., 2018). Results showed that current values are far from targets of the European Commission (4 hours door-to-door for 90%of the passengers by 2050 (High Level Group on Aviation Research, 2011)). The SESAR project Vista (Delgado et al., 2020) then reused the same model, expanding it to include a strategic and pre-tactical layer, to estimate tactical indicators in an hypothetical future day in Europe in 2035 and 2050, given some assumptions on the traffic, and exogenous factors.

Finally, the model was reimplemented as a full agent-based model system in Domino, another SESAR exploratory research project. The goal of Domino was to highlight the dependencies of subsystems by developing new metrics tailored to capture network effects when mechanism are introduced in the ATM system (see Mazzarisi et al. (2020); Zaoli et al. (2020) for examples of these metrics and their application). Test cases including new procedures were designed, and thus the project needed a model 1) microscopic enough to capture feedback loops and produce low-level data 2) complex enough to capture the reaction of different actors to the new procedures. The new model uses a strong agent-based paradigm to allow future behavioural development. The resulting simulator, called Mercury, is presented in detail in this article.

3. Model specification and design

3.1. Model need and requirements

In summary, we notice a need for a model that is able to:

- test solutions modifying procedural rules, implying various actors like passengers or flights, in the airspace (SESAR and others) **before** large-scale exercises
- capture knock-on effects between various subsystems, in particular between flights and infrastructure facilities such as airports, at a European scale.
- compute various KPIs for stakeholders; in particular, delay distributions for passengers and flights, as well as cost,
- take into account fairly complex decision making processes from the airlines, in particular reacting to cost and not only to delay.

In this article, we are interested in testing one mechanism that we call "4 Dimensional Trajectory Adjustments". 4DTA is composed of two sub-mechanisms, Dynamic Cost Indexing (DCI) and Waiting for (connecting) Passengers (WfP), which allow a flight to:

- adjust the cost index during the flight (e.g., to either recover delay or save fuel (DCI), as suggested in Cook et al. (2009).
- delay the departure of outbound flights in order to wait for missing passengers (WfP), as implemented in Delgado et al. (2016).

4DTA also enables complex actions that are combinations of simpler actions, such as delaying a flight and then recovering part of that delay during the flight by modifying their cost index, in order to protect specific passengers with respect to the cost of delay (e.g., passengers who might miss a connection and incur on compensations due to Regulation 261 (European Commission, 2004)). More generally, it enables the airlines to adjust a flight's 4D trajectory in order to save fuel and/or reduce passenger costs relying on the updated information from other flights.

Such decisions are often made on-the-fly by relying on rules of thumb during the daily operations. However, in order to select the option which gives the lowest cost expectancy, the airlines can benefit from the assessment of the downstream effects of a delayed flight throughout the day (reactionary delay of the aircraft) plus through the rest of their network (e.g., due to connecting passengers). Mercury in particular aims at capturing such network-wide effects to provide a more informed decision making framework.

Since we want to capture network effects, the model needs to take into account the most important channels of propagation of delay. First, the fact that aircraft are used throughout the day by different flights, and thus can propagate delays (reactionary delay). Second, passengers, or at least groups of them, need to be modelled in such a way that flight may be delayed because of them and to capture the costs of their delay and missed connections. Third, the model needs to take into account exogenous sources of delays, for instance ATFM, turn-around, or taxi delays. Fourth, the reactions of the system to delays and/or congestion need to be included, for instance through air traffic management regulations and airlines' actions. Lastly, the model needs to allow dynamical decisions from agents, i.e., an information management service giving them the right level of information in 'real' time and allowing them to modify their resources (e.g., aircraft allocation) based on this information, whenever it is allowed. This dynamical aspect is crucial, since operations face lots of uncertainty on a daily basis, which trigger decisions that have knock-on effects on the system.

Note finally that we are not interested in safety issues *per se*. Indeed, we consider them only indirectly, through the definition of capacities for different subsystems and the modelling of the consequences of enforcing them (i.e., creating delays).

3.2. The Gaia methodology

In order to build the model, we use the Gaia methodology (Wooldridge et al., 2000). Gaia is a methodology for agent-oriented analysis and design applicable to multi-agent systems founded on the view that a multi-agent system is a computational organisation consisting of various interacting roles. The methodology is particularly well suited to the development of large-scale applications with specific characteristics, like high heterogeneity among agents, static organisational structure, and static roles for agents.

Gaia deals with both the macro (societal) and the micro (agent) levels of design. Gaia divides the different activities between two phases: analysis and design. The objective of the analysis is to develop an understanding of the system and its structure, capturing the different roles and their interactions. The design phase transforms the abstract models derived from the analysis into models at the level of detail which allows their further implementation.

As shown in Figure 1 from Wooldridge et al. (2000), from the requirements of the system a set of roles and their interactions are derived. The roles are grouped to create agents which provide a set of services and the communication protocols identified during the interaction models will define the acquaintance model (see below) between the agents. The objective of each of these activities is summarised below.

- Roles model: Identify the different roles that exist in the system. A role describes what an entity is expected to do. The roles are characterised by their permissions and responsibilities.
- Interaction model: This model captures the relationship between the roles, defining the protocols which describe the interactions between them, i.e., the communication purpose, initiator, responder, inputs and outputs and processing information.

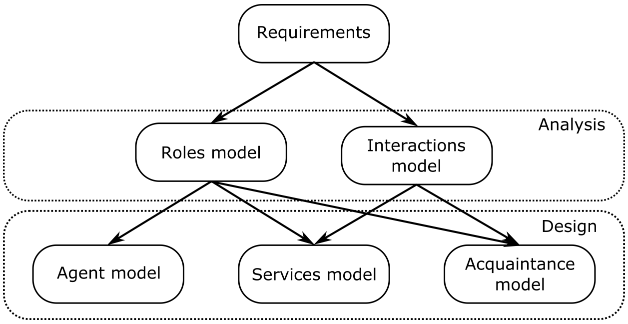

Figure 1: Flow of design in Gaia's model (from Wooldridge et al. (2000)).

- Agent model: The agent model is created by aggregating roles into different agent entities.
- Services model: The services that each agent provides are defined based on the functions the agents provide, which are in turn derived from their associated roles and protocols.
- Acquaintance model: This model simply describes the communications links between the agents.

The ABM system (the agents, their services and relationships between them) was described following Gaia. However, this methodology does not prescribe a specific implementation approach. Therefore, there is no imposition on the simulator engine nor on the approach to be followed for the implementation of the communication channels between the agents. Different alternatives were therefore considered. For example, step-wise modelling or event-driven approach for the engine implementation; and the use of a shared ontology and standardised communication messages, such as a FIPA-compatible communication protocol (Foundation for Intelligent Physical Agents, 2012), the use of a communication middle server, or direct communication between agents in a closer object-oriented modelling approach, for the communication channels. Section 3.7 presents the details on the approach selected for Mercury for these implementation aspects. A full description of the different aspects of the specification and design of Mercury can be found in Domino Consortium (2018).

3.3. Roles in Mercury

Following the Gaia methodology (Wooldridge et al., 2000), a role can be seen as an abstract description of an entity's expected function which is defined by four attributes:

- Responsibilities: these determine functionality. They are divided between:
  - Liveness properties, which describe the states that the agent must bring about, given certain environmental conditions. For example, the different tasks that an agent needs to perform when a given environmental condition are met or a given message is received;
  - Safety properties, which are invariants that must be assured by the role.
- Permissions: rights associated with a role, which describe the resources that are available to the role to realise its responsibilities (e.g., reading, modifying or creating data).
- Activities: computations associated with the role that can be carried out without interacting with other agents, which can be seen as private actions in the context of software engineering.
- Protocols: these define the way that a role can interact with other roles.

Following the high-level goals and requirements explained in sections 1 and 3.1, thirty-nine roles were identified in Mercury as listed in Table 3. We focused on the main roles that may have an impact on the creation and propagation of delay in the system. Keeping the description fairly high level, we left for instance the detailed en-route tactical deconfliction out of the modelling, only including explicit queues and capacities at airports and modelling instead the impact of those actions on the uncertainty on route duration. The reader is referred to Domino Consortium (2018) for a detailed description of these roles.

The level of description has been chosen to strike a good balance between computational complexity and microscopic description of the model, while keeping the goals above in mind. In particular, roles aggregate in general more microscopic tasks that are carried out in reality by atomic entities, humans or machine. This is important to keep in mind since inner, non-modelled interactions may actually have an impact on the roles described here, in particular in terms of efficiency of how the task is carried out. Moreover, modifications of procedural rules – e.g., to test a new solution – can obviously happen explicitly only at a larger scale than the one described by the model. In other words, only modifications of the inner decisions of the agents (activities/responsibilities) or their interactions (protocols) can be done with model. However, modifications at a lower level than the level of aggregation described here may be modelled in some instances through the modification of the activities themselves. As an example, the introduction of a better flight management system for the pilot lies below the level of aggregation presented here. However the relationship of the new system with the pilot may can be captured by the modification of an activity within a bigger agent, the Flight agent.

One of the obvious limitations of the level of description below is the fact that we do not include explicitly air traffic control. We assume that the regulations are enough to avoid major incidents, and that air traffic control has little effect on the gate-to-gate delay. As highlighted below in section 4.1.4, we also assume that the ATC will be able to grant the adjustment of cruising speed when tactically modified in the advanced version of the mechanism, an assumption that can be studied further in future work, in particular from the safety point of view.

Finally, note that the activities described in the following are often carried out with quite a high level of efficiency and a perfect rationality. In reality, where humans take decision, they usually tend to be limited by various factors, and be prone to behavioural effects. This is not taken into account in this article, but is under study in a subsequent work within the SESAR ER-4 BEACON project.

Table 1 presents an example of a role. This particular role is called "Departure Slot Requester (DSR)" and is tasked with requesting a slot and estimating the push-back time for a flight. It is composed of several 'protocols' and 'activities'. Their combination forms its liveness responsibility: check back continuously when the 'push-back ready' event is triggered. If so, request a departing slot and wait for it. In parallel, ask for a taxi-out estimation, then wait for it. When all the requested information arrives, compute an estimated push-back time. The role should have a certain number of access permissions to various resources to fulfil its tasks, for instance 'read' the flight status (e.g., listening to the push-back ready event), or 'update' the flight status after push-back time computation. Finally, it has some 'safety responsibilities', making sure the inner consistency of the model is preserved during its process.

3.4. Interaction model

Following the agent paradigm, roles have a limited access to information, and cannot access private information pertaining to other roles. Hence, roles need to interact, i.e., delegate computations by asking other roles to perform them using their own memory. The interaction model allows us to list all the possible interactions between two roles.

In the Gaia methodology, an interaction is described by their:

- purpose: brief textual description of the nature of the interaction;
- initiator: the role(s) responsible for starting the interaction;
- responder: the role(s) with which the initiator interacts;
- inputs: information used by the role initiator while enacting the protocol;
- outputs: information supplied by/to the protocol responder during the course of the interaction;
- processing: brief textual description of any processing that the protocol initiator performs during the course of the interaction.

Sixty-three interactions were defined in the model. Table 2 shows an example of such an interaction. In this interaction, the role "Departure Slot Requester (DSR)" triggers the protocol "*RequestDepartingSlot*". Doing so, it communicates with the role "Departure Slot Provider (DSP)", by providing the flight number of the flight requesting the slot. The responder does not return anything within this interaction. Indeed, as a design choice, we decided to use interactions for one-way messages, not two-way. Hence, accompanying this example, there is another interaction in which the DSP is the initiator and the DSR the responder, with DSR communicating the results of the slot allocation to DSP.

Role schema	Departure Slot Requester (DSR)
Description	When the flight is ready to push-back, the role requests a departing slot, requests a taxi-out estimation, and computes and sets its push-back time: • Request departure slot, • Request estimate taxi-out time, • Computes push-back time.
Protocols and activities	CheckPushBackReady *RequestDepartureSlot* *WaitForDepartureSlot* *RequestTaxiOutEstimation* *WaitForTaxiOutEstimation* ComputePushBackTime
Permissions	*reads flight status // (ready)* *flight departing slot // departing slot of flight* *flight plan // information on flight plan - speed, distances, etc.* *updates flight status to push-back ready* *flight AOBT* *generates departing slot for the flight* *TakeoffTime flight*
Liveness responsibilities	DSR = (CheckPushBackReady. (*RequestDepartureSlot*.*WaitForDepartureSlot* || *RequestTaxiOutEstimation*.*WaitForTaxiOutEstimation*) .ComputePushBackTime)$^{\omega}$
Safety responsibilities	• AOBT = Final EOBT = Push-back time • AOBT = max(EOBT, Departing slot - taxi out estimation)

Table 1
Example of a role description in the model ("Departure Slot Requester (DSR)").

3.5. Agent definition

Once all the roles and their interactions are defined, they are grouped into higher-level entities, which will represent the different agent types. This process is in general largely arbitrary and adaptable as the specification of the system evolves into a specific designed architecture. In Mercury, this process was guided by existing entities in the ATM domain, naming and representing themselves as coherent (such as the Airline Operating Centre (AOC)). This facilitates the identification of agent types with well-known entities in the ATM community. Another important consideration when defining the agent types is the private nature of the information available within an agent. An agent is the 'atomic level', above which information is not freely accessible. Within an agent, i.e., within and between its roles, all information can be shared. This has also repercussions in terms of performance (e.g., time required to access and process information) and behaviour (e.g., information available when performing decisions).

Following these guidelines, 8 agent types were described as indicated in Table 3. The inner working of the model and details on the decision making process of the agents are presented in Section 4.

The 'Airline Operating Centre' (AOC) manages the passengers and the airline's fleet using its knowledge on their state (e.g., delay, expected costs). In a given instantiation of the model there is one AOC agent instance per airline. The agent considers the tactical reassignment of passengers to flights (if they need to be reallocated due to missed-connections) and the dispatching processes of flights (e.g., selection of flight plan, or 4DTA strategies). More information on the 4DTA mechanism is presented in Section 4.1.3.

The next most relevant agent type is the 'Flight'. These agents are tasked with the operation of the aircraft in the network and with the trajectory integration. They start by waiting for the departure of the flight, then they manage the trajectory to the arrival at the destination. In particular, they are tasked with operating the trajectory, which involves the computation of actual times of passage through waypoints, while considering probabilistic environment variables, such as wind or variations in route length due to ATC action. The agents will compute performance indicators such as

RequestDepartingSlot

Initiator	Responder
Departure Slot Requester (DSR)	Departure Slot Provider (DSP)
Inputs	
flight id	
Processing	
Request a departure slot when flight is ready.	
Outputs	
None	

Table 2
Example of interaction. The protocol *RequestDepartingSlot* is used by the DSR role from Table 1. The other end of the interaction is the DSP, which after this interaction will compute the slot available and return it to the DSR as part of another interactions.

Agent type	Roles
Airline Operating Centre	Airline Flight Planner, Dynamic Cost Index Computer, Passenger Reallocation, Turnaround Operations, Airline Passenger Handler
Flight	Aircraft Departing Handler, Departure Slot Requester, Flight Plan Constraint Updater, Flight Plan Updater, Flight Arrival Information Provider, Ground Arrival Handler, Operate Trajectory, Potential Delay Recovery Provider
Ground Airport	Ground Handler, Provide Connecting Times, Taxi-out Estimator, Taxi-Out Provider, Taxi-In Provider
E-AMAN	Strategic Arrival Queue Builder, Arrival Queue Planned Updater, Arrival Cancellation Handler, Flight In AMAN Handler, Arrival Planner Provider, Arrival Tactical Provider, Slot Assigner, Arrival Planner Provider Queue, Arrival Tactical Provider Queue
DMAN	Strategic Departure Queue Builder, Departure Slot Provider, Departure Queue Updater, Departure Cancellation Handler
Radar	Disseminate Flight Plan, Disseminate Cancellation FP, Disseminate Flight Position Updat, Radar Augment Flight Plan
Network Manager	Network Manager Flight Plan Processing, Network Manager Accept and Disseminate FP, Network Manager Cancel FP, Flight Swap Processor
Flight Swapper	Swap Engine

Table 3
All agent types in the model, with their roles.

fuel usage and interact with other elements such as the departure or the arrival manager. There is one Flight agent per flight in the simulation.

The 'Ground Airport' agents process arriving passengers (computing the actual transfer time between flights in the terminals) and the arrival of flights (providing the turnaround times). Both processes rely on probabilistic modelling of their distributions considering characteristics of operations such as airport, airline, aircraft and passenger types. One Ground Airport agent is instantiated per airport in the model.

Each airport has also an associated 'E-AMAN' agent which manages the arrival queue of slots needed to respect the arrival capacity. This agent can be instantiated either with a planning and an execution horizon, or just with an execution horizon (for airports which do not have extended arrival planning capabilities). The agent will build and manage this queue of arrival slots and it will communicate with the flights to create the arrival sequence. More details on the behaviour and functionalities of the 'E-AMAN' agent are provided in Section 4.4. The 'DMAN' (departure manager) agent type is similar to the E-AMAN, albeit simpler. The 'DMAN' agents manage the departing queue at the airport and authorises the push-back of the flights. All airports have an E-AMAN (which could have or not and extended planning horizon) and a DMAN to manage the capacity and tactical queues at their runways. Note that the capacities at the runways can be adjusted and vary through the day.

The 'Radar' is the agent which is in charge of broadcasting the flight position to all the interested agents. A flexible subscription notification architecture has been implemented to allow this. When a flight crosses significant waypoints, the 'Radar' notifies subscriber agents. In practice, it mainly communicates with the AOC and the E-AMAN (e.g., the AOC is notified when a flight reaches its Top Of Climb (TOC) so that delay management strategies (4DTA) can be implemented; the E-AMAN is notified when a flight enters its planning horizon, so that it can be considered in the arrival sequencing). There is only one Radar agent instance per simulation, which is visible to all flights in the model.

The final two agent types are the 'Network Manager' (NM) and the 'Flight Swapper'. The NM is in charge of modelling the processes of the ATFM Network. It receives the flight plan request and, if required, assigns ATFM delay. This delay can be either probabilistic or with explicit modelling of regulations, i.e., modelling the slot queues. The 'Flight Swapper' is responsible for the management of ATFM slots swaps following the principles of User-Driven Prioritisation Process (UDPP) (Pilon, 2016; SESAR, 2018). In the model there is one instance of each.

3.6. Services and acquaintances model

Once the agent types have been defined as collections of roles, they inherit the protocols and activities of the latter. While role's activities become agent's activities, protocols are turned into services. Indeed, protocols can trigger interactions between roles which are part of the same agent, or between roles belonging to different agents. In the first case, the interaction can be reduced to a a direct request between roles, given that they share the same memory space within the agent. In the second case, a message is sent between agents to require some computation (or data) using non-shared resources. This can be viewed as a 'service' provided by one agent to the other.

For instance, the interaction described in Table 2 involves the Departing Slot Requester and the Departure Slot Provider. The former being part of Flight and the latter of DMAN (see Table 3), this interaction will trigger an exchange between these two agents. In other words, the DMAN has a service, called 'provide departing slots', which can be provided to various agents. Hence, the service list of each agent can be understood as that agent's Application Programming Interface (API), or its public interface. The standardisation of these interactions increases the reusability and flexibility of the model. See Section 3.7 for more information on these communication protocols.

Finally, the reduction from roles to agents creates 'acquaintances', i.e., required visibility between agents. This creates a high-level view of the model, which is useful to estimate the coupling of the system. This information could also be considered if the agents are deployed in a distributed architecture as communication links will be require. As part of the agentification of the roles, this coupling can be considered if computational or communication issues arise, leading to a potential different definition of agent types. This acquaintance model is shown in Figure 2, where it is easy to observe the centrality of the AOC agent.

3.7. Model scheduling and messaging

External communication (between agents) is handled via a unified message system. For this, we considered using a standard such as the XMPP, a set of open protocols usually used for instant messaging, but usable for decentralised data processing (Saint-Andre, 2011). However, the available implementations of XMPP in Python (e.g., SPADE) are not fast enough for a large model such as Mercury: the number of interactions between agents for the modelling of a day of operations in Europe in Mercury is around 1.4M messages. Hence, a simplified proto-messaging system was developed. Any process within the model can only access the 'box letter' of other agents (plus a few public parameters), which ensures the privacy of each agent and the standardisation of the interactions.

The agent-based model architecture is static unless some events evolve the environment or trigger the interaction between the agents. There are several common framework for this modelling engines:

- Step-wise simulations: it is a simple framework, but results (output) of the simulation might depend on the order of computation within a step.
- Event-driven simulations: agents react to events being triggered by other agents or by the environment. Once an event is resolved, the simulation jumps to the next event in the queue. Events can be rescheduled or cancelled, if needed, and new events created as required. This approach might weaken the agent paradigm if events are shared among agents. Finally, note that the reaction of one agent to an event might require the interaction of this agent with others in a message-driven approach.
- Message-driven simulations: agents react to messages being sent by other agents, by the environment, or by a user. This setup could be similar to an event driven simulation, but the event-driven approach provides a

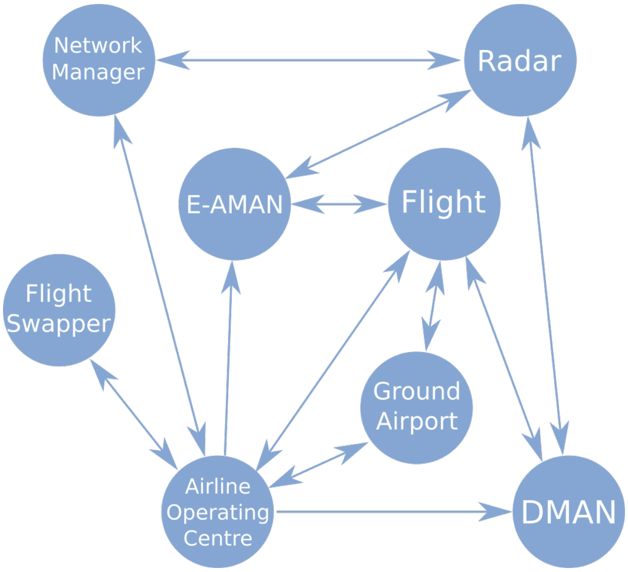

Figure 2: Acquaintances between agent types, derived from interactions between their underlying roles.

simplified setup when the triggers of actions are changes in the environment (e.g., a given milestone, such as push-back, is reached).

Mercury is an event-driven simulator. Initial events are pre-defined at the initialisation of the simulation. They are linked to the milestones of each flight: push-back, landing, etc. Table 4 shows a list of all the events in the model. Once an event is triggered, one or more agents react to it. This usually implies some internal computations (activities) and some messaging between agents to delegate some of the computations (services) or request information. Once all the activities (including the interaction between agents) that are triggered by the event are completed, the simulation jumps to the next event in the queue. The action of the agents might modify the scheduled events, and in some cases internal events for synchronisation might also be created.

Some of the events in the simulation might be triggered at the same time (e.g., two flights with the same SOBT). Therefore, they need to be treated in parallel, leading to a typical case of concurrent programming. A system of queues and resources (using Python module Simpy[5]) ensures that these races are properly solved. A single queue of scheduled events is built for the whole system, with all the events stacked on a single timeline. As indicated, events are created, modified, and cancelled dynamically based on actions triggered by the previous ones.

The model has been calibrated to simulate one day of operation (see Section 5) of all commercial flights departing or leaving Europe. In a full simulation there are approximately 27k Flight instances, 300 AOC instances, 800 Ground airport, E-AMAN, and DMAN instances each, one Network Manager and one Radar instance. On top of that, around 400k passenger groups (passengers are bundled by itineraries, see Section 5) are created, representing 3.4M passengers. Around 7000 aircraft resources are created (as aircraft are traced across Europe, explicitly modelling reactionary delay), along with approximately 43k flight plans (as the AOC can select which flight plan to use for each flight from a pool of pre-computed flight plans between each origin and destination).

4. Description of Agents

In this section, the inner working of the model is described in more detail, in particular how agents make decisions.

4.1. Airline Operating Centre (AOC) Agent

One of the most important agent in the model is the AOC. This agent manages the operations of the airline, including flights (fleet management) and passengers. These decisions are driven by the estimation of expected losses due to cost functions.

[5] https://simpy.readthedocs.io/ (accessed June 2021)

Event	Short description
FP submission	First submission of flight plan for a flight. This is normally triggered 3 hours before the flight SOBT.
Delay estimation	AOC checks the status of the flight and a random non-ATFM delay is drawn. This is normally triggered 1 hour before the flight EOBT.
Passenger check	AOC checks which passengers are not ready to board their flights, 5 minutes before EOBT.
Pushback ready	Aircraft is ready to push-back. The flight requests a departure slot.
Pushback	The flight is off-block and begins taxi-out. Connecting passengers which are not boarded are rebooked.
Takeoff	The flight begins an "operate trajectory" activity which integrates the trajectory between pre-defined waypoints in the flight plan (with stochastic noise).
Flight Crossing Point	A waypoint is crossed by the flight during its trajectory execution. This type of event triggered by the flight and captured by the Radar for the broadcast the position of the flight to interested parties in the model.
Landing	The flight reaches its final trajectory point. It begins taxi-in.
Flight Arrival	The flight arrives at the gate. Turnaround and connecting passenger processes begin.

Table 4
Events used in the simulation.

4.1.1. Cost function

In general, airlines' costs are notoriously complex to estimate. A decision on one flight can have cascading effects on several others and numerous passengers.

During the pre-tactical and tactical operations, airlines focus on minimising the expected total cost. The costs that are modelled in Mercury include direct operating costs, which are defined based on the selected flight plan, namely: fuel and en-route charges; and costs due to unforeseen circumstances during the operations, in particular 'cost of delay'. The model does not consider costs that are fixed (or already considered) on the day of operations (such as nominal crew cost, or aircraft depreciation) as these are not affected by the tactical operational decisions that are performed in the model. Note that for example the extra crew cost due to delay will be considered as part of the 'cost of delay'.

We consider the 'cost of delay' as the extra cost that airlines experience due to the delay of their flights. These costs can be broken down and examined carefully for different types of aircraft and companies, as done in (Cook and Tanner, 2015).

A breakdown of the costs considered in the model is listed in Table 5. For the cost experience by the airline due to passenger delay, two type of costs are considered: 'hard' costs, which have a direct monetary translation, such as compensation costs for passengers due to Regulation 261 (European Commission, 2004); and 'soft' costs, i.e., costs that represent a future loss for the company in terms of, for instance, market share.

The cost of delay is typically a non-linear function of delay. The reasons for that are multiple. First, negative delays (flights that arrive before scheduled) are usually of no monetary value to the airline, or even in some cases could be detrimental. Second, high delays are typically proportionally more costly than small delays (cost has been shown to increase in average as roughly quadratic with delay (Cook and Tanner, 2015)). Finally, the cost functions tend to exhibit significant noncontinuous increments, usually for some specific values of delay. This is linked to events such as passengers missing connections or a flight breaching an airport curfew. Note that, due to the estimation of reactionary delay, the flight breaching the curfew could be the one at the end of the day, even if the delay is experienced in the morning. Mercury estimate these costs so that they can be considered even if the flight is not the one which would experience the curfew as explained below. Typical cost functions are displayed in Figure 3.

Most of the costs of delay are based on Cook and Tanner (2015). Depending on the phase of the flight, the AOC uses different cost estimation functions, as some costs relate to departing delay while others to arrival delay (e.g., the departure delay impacts the duty of care cost, but the estimated arrival delay should be considered for the passenger compensation costs computations). Moreover, in some cases, cost will be already accrued and therefore it is not possible to recover it with mitigation strategies (e.g., the departure delay and its associated cost cannot be recovered once the flight is airborne, but the arrival delay can be adjusted with modification of the cost index, which might impact the cost of delay due to passenger missed connections if some connections can be protected). Note that in all cases, delay is computed with respect to the schedules and the times at the gate, and hence buffers play an important role.

Curfews are explicitly modelled in Mercury. In real operations, flights infringing curfews, i.e., planning to land

Type of cost	Implicit /Explicit	Approximated /Exact	Description
Cost of maintenance due to delay	Implicit	Approximated	Additional cost incurred by airline due to delay as some maintenance processes are linked with the usage of the aircraft.
Cost of crew due to delay	Implicit	Approximated	Additional cost because of overtime and crew rotation.
Passenger compensation	Explicit	Exact	EU Regulation 261 (European Commission, 2004) entitles passengers to compensation in case of delays. The amount that passengers are entitled to compensation depends on the great circle distance between origin and destination, the magnitude of the delay, and the entitlement is linked with the reason of the delay (e.g., weather). These parameters have been carefully implemented in detail in the model. Based on interaction with airlines (Cook et al., 2016), we also considered that only a percentage (11%) of the passengers claims the compensation.
Duty of care	Explicit	Approximated	EU regulations also entitles passengers to have meals, drinks, and hotel accommodation in case of departure delays.
Soft cost	Explicit	Approximated	Delays are also detrimental to the airline's image, potentially damaging future market share.
Curfew cost	Explicit	Approximated	Some airports have curfews. Flights breaching these curfews may incur a wide range of costs, from a small fine to an interdiction to land, which is very disruptive for the airlines operation.
Fuel costs	Explicit	Exact	Computed using BADA 4 performances (EUROCONTROL, 2015). This includes the fuel that is used nominally (as expected by the flight plan), plus the variations (extra-usage or savings) due to uncertainty (e.g., wind), and delay and delay management strategies (e.g., holding or adjusting the cost index).
Airspace charges	Explicit	Exact	This consider airspace charges airlines incur based on their flight plan. These fees usually depend on the route and the aircraft MTOW. The model considers the 39 regions managed by EUROCONTROL CRCO plus the airspaces of Egypt, Belarus, Morocco, Uzbekistan and Ukraine. Other surrounding countries which follow different charging schemes are also modelled: Algeria, Iceland, Russia, Tunisia. This allows us to compare the cost of different routes even when they use adjacent airspaces to the core European ones.

Table 5
Costs considered in the model. Cost of delay are defined as in Cook and Tanner (2015), except for fuel, airspace charges and curfews, see text.

after a certain time of the day, may face either:

- a fine to pay (soft curfew), or
- a rejection of their flight plan, i.e., flights cannot be planned to land after a certain hour (hard curfew).

As reported in Boeing (2019), curfew application can be very complex. For example, the curfew may be active only for a certain type of aircraft, for flights coming from a certain direction, for departures, for arrivals, etc. This complexity is driven by the fact that some of these limitations are related to environmental practices (e.g., noise pollution) rather than resources at the airports.

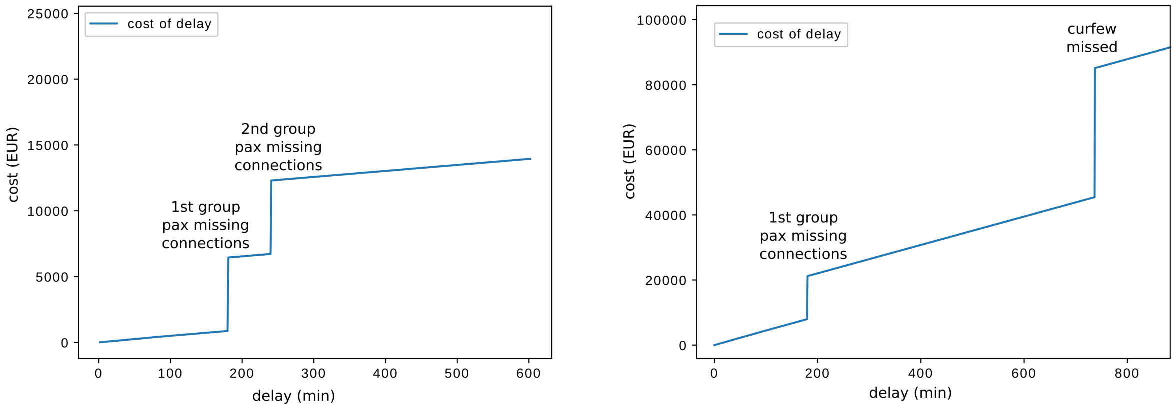

Figure 3: Examples of cost function used in the model. The cost for a given flight increases slowly with the delay (quadratically). It jumps when some passengers miss their connection (different groups of passengers may miss different flights). At the end of the day, a flight may miss a curfew, which is particularly expensive for the airline, as seen on the right hand side.

In Mercury, a simplified version of a curfew model is implemented. Only curfews which prevent airlines from filling a flight plan to the destination are considered. Curfews are enforced at arrival only and at airports in the ECAC region which were provided by EUROCONTROL. These airports have been considered as airports with curfews as part of the validation activities of UDPP performed by EUROCONTROL. This reduces the total number of airports which enforce curfews to 14. If the expected arrival time to the airport, at the moment of submitting the flight plan, is after the curfew time, the flight plan will be rejected by the Network Manager and if no alternatives are available (e.g., using a different flight plan with an earlier arrival time, which could be possible with a different route or cost index), the flight will be cancelled. If the flight breaches the curfew and no alternative is possible, it would be cancelled, but subsequent flights with the same aircraft frame are considered as possible to be operated, if they do not breach a curfew themselves. This cancellation approach has two consequences for the model:

- the Network Manager agent will reject flight plans for which the flight would arrive to the destination after the curfew threshold;
- the Airline Operating Centre agent will consider the potential infringement of the curfew when estimating the expected cost of a given amount of delay, even if that flight is not the one which would incur in the infringement. This might change the behaviour of the AOC, for example trying to recover delay in early flights to avoid a potential curfew in the evening.

The breach of a curfew has a costly impact on the operations. In Mercury this cost has been defined as follows:

- 40 000€ for a light or medium aircraft, as flagged in their wake turbulence category,
- 80 000€ for a heavy or a jumbo jet.

These values are based on EUROCONTROL's estimate for a similar cost function model (used in particular for the User-Driven Prioritisation Process (UDPP)).

As mentioned, curfews do not only have an impact on the final flight of the day, airlines are aware that delays early in the day can translate into reactionary delays, which might eventually breach a curfew. Figure 4 presents how the 'curfew buffers' are estimated in the model for each flight. In this example there are five consecutive flights that will be performed by the same aircraft frame through the day. As depicted, flight 2 and flight 4 have as destination an airport with a curfew set at 23h00. Between each rotation there is an estimated Minimum Turnaround Time (MTT). With this information, the buffers to propagate delay between the flights (turnaround buffer) and to breach the different curfews are computed. This is presented in the middle image of Figure 4, for example, flight 2 should be delayed by

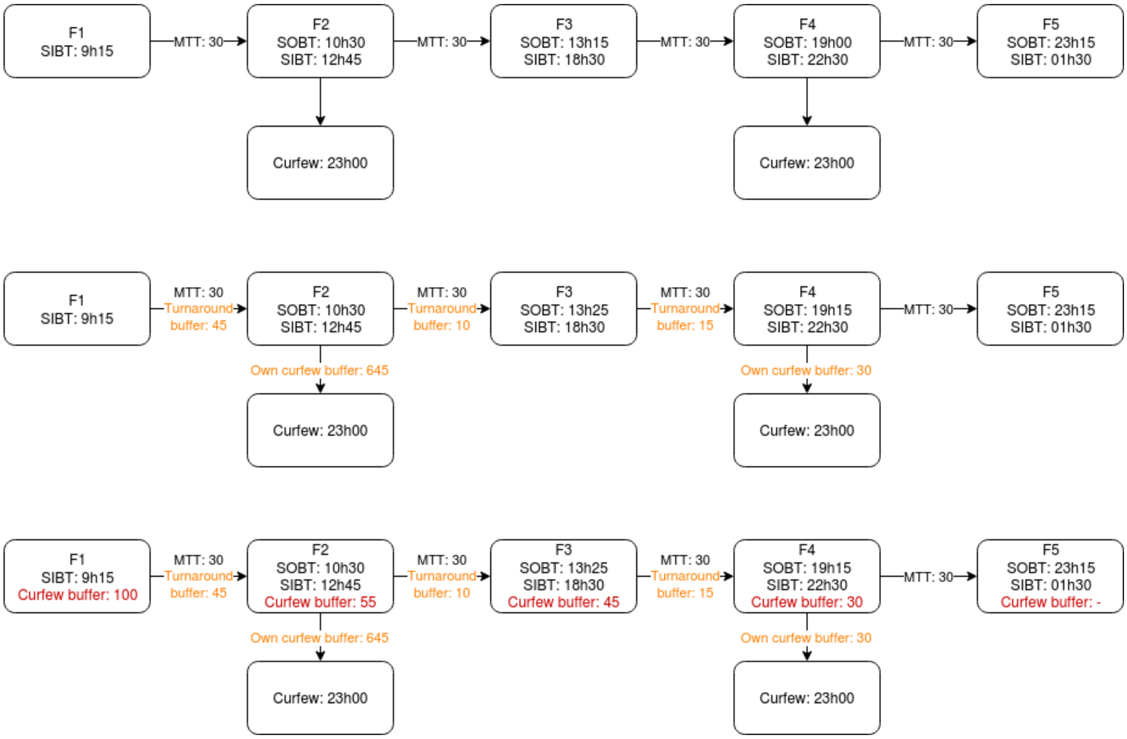

Figure 4: Examples of maximum delay before curfew estimation for one aircraft thought the day.

more than 645 minutes for the flight to breach the curfew, as its SIBT is at 12h45 and the curfew at destination starts at 23h00. Then, finally, it is possible to compute the 'curfew buffer' for the different flights considering from which moment flights might propagate enough delay to potentially breach a curfew through the day. For example, flight 3 has a 'curfew buffer' of 45 minutes as if the flight is delayed over 45 minutes, it will propagate over 30 minutes to flight 4 which will then breach a curfew. In this example, even if flight 2 has 645 minutes of delay until it breaches the curfew in its own flight, its 'curfew buffer' is only 55 minutes, as after that delay it could potentially propagate enough delay for flight 4 to breach its curfew later in the day.

In Mercury, these buffers are re-computed dynamically using the most up-to-date available information during the simulation, such as the updated EIBT of downstream flights.

A missing feature in the model when it comes to AOC operations is the crew management and explicit cost estimation due to disruptions. In case of disruption, potential solutions are in reality evaluated against the feasibility (and cost) of having the right crew for the right aircraft, complying with various regulations (e.g., maximum number of hours on duty), and considering costs such as misplacement of the crew at the end of the day leading to accommodation expenses (Clausen et al., 2010; Zhang et al., 2015). This crew management cost is captured by the "cost of crew" in Table 5 and is thus represented at an aggregated level in the model. Considering the true contextual cost is particularly important if one wishes to fully consider airline disruption management strategies. However, a major issue preventing us from including a crew management module in the model has been the access to relevant data. Crew rosters are typically confidential. The crew management is also highly specific to each airline, so a small sample might not be representative of the model as a whole.

4.1.2. General flights-related processes

The AOC reacts to several events related to its fleet, and it is also able to provide services for other airlines, particularly the ones part of their own alliance.

First, it listens to the 'FP submission' event, created at model initialisation and triggered 3 hours before SOBT by the AOC itself. When the event is triggered, the AOC chooses a Flight Plan (FP) for the flight from a pool of pre-computed flight plans (see Section 5 for more details), based on the origin-destination pair needed and the type of aircraft operated. This pre-computation is done to reduce the computational load of the simulation. The selection of FP is an iterative process in which the AOC, using a logit decision function, selects different flight plans for the flight considering their expected cost (including fuel costs, en-route charges and cost of delay), submits them to the Network Manager, and if ATFM delay is issued, alternative flight plans might be considered, with the subsequent re-submission.

Unless the flight plan breaches a curfew, the FP will be accepted by the NM. In some cases, the AOC might cancel the flight, either with a certain exogeneous probability, or if all possible flight plans infringe the curfew. When a flight is cancelled, for simplicity, the model assumes that the aircraft frame is available for the subsequent flight on the rotation plan. An alternative to this rule is to cancel all subsequent flights, but this might generate non-realistic high cancellation rates.

The second event the AOC reacts to is the ‘Delay estimation’, which is triggered one hour before EOBT. This is where the AOC gets notified of non-ATFM delay (if any) and reassess its estimated departure time. If the flight had an ATFM slot which will be missed, a new one is requested from the NM. If the delay is greater than 30 minutes, a fully new flight plan (considering the different alternatives) is recomputed. The delay estimation process can be also triggered prior a flight if reactionary delay is generated by a previous flight.

The third event that the AOC listens to is the ‘Passenger check’ event. This event is used within the ‘4D trajectory adjustments’ mechanism. It computes the expected arrival of the connecting passengers that are supposed to board this flight and decides whether to wait for them, using rules of thumb or in an advance implementation relying on cost estimates. If the flight decides to wait for passengers, it requests a departure reassessment to ensure that if ATFM slot has been assigned to this flight, this is respected.

Once the flight is ready for push-back, the corresponding event (‘Pushback ready’) will be triggered. The AOC does not have any tasks related to this event, it is the flight that will directly request a departing slot to the DMAN (see Section 4.2). Congestion at the departure might produce some delay which will be accrued at the gate. Note that delayed connecting passengers might still board this flight until the actual push-back time. When the actual push-back arrives, the AOC checks which passengers are actually on-board, and computes some of the passengers metrics such as departing delay, type of delay, etc. For the passengers who have missed a connection to this flight, but are already at the airport, i.e., they have landed but their connecting time did not allow them to get to the gate on time, it triggers a reallocation process. To do this, it considers all the possible new itineraries and prioritises:

- Direct itineraries with the same airline,
- Indirect (only 2 legs) itineraries formed of the airline’s flights,
- Direct itineraries with flights belonging to other airlines within the same alliance,
- Indirect itineraries with flights belonging to other airlines within the same alliance (or a combination of its own flights with the other airlines’),
- Direct itineraries with other airlines, but only for ‘flex’ passengers, and for a fee.

Passengers which are not able to be reallocated with the previous process (e.g., there is follow up flights which provides a possibility to reach their final destination, or all flights are already full) are marked as ‘arrived’ and compensated (see Section 4.1.4).

The AOC reacts to flights arriving to their destination (when they reach the gate, i.e., when ‘Flight Arrival’ is triggered by the flight). When this happens, the AOC launches two parallel processes: the aircraft turnaround and the passenger processing. The first one is handled by the ground airport, see 4.3, and might trigger the computation of reactionary delay due to late aircraft arrival. Passengers go through the following processes:

- If passengers have reached their final destination, they are marked as ‘arrived’ and compensated if needed (see Section 4.1.4).
- If passengers are connecting on a flight on the same airline, the AOC computes whether they can make the connection or they missed the flight, i.e, if their connecting flight has already departed or been cancelled. In the latter case, the AOC initiates the reallocation process, as described above.
- If passengers are connecting on a flight from another airline, the AOC delegates the connecting process to the corresponding AOC agent.

4.1.3. Trajectory-related processes: 4DTA

As described in Section 3.1, the '4D trajectory adjustment' is the core mechanism used by the AOC to manage the delay of its flights. 4DTA is composed of two sub-mechanisms: Dynamic Cost Indexing (DCI), to adjust flight speed, and Waiting for (connecting) Passengers (WfP), to actively delay flights to wait for connecting passengers. In this article we present two different implementations of this mechanism, a basic and an advanced one. The basic implementation of 4DTA aims at capturing current practices in the airline industry, which often rely on rule of thumbs and heuristics, and has been validated with a number of stakeholders through various consultations and workshops in which many airline representatives participated. These consultations were performed as part of previous projects that actively developed Mercury and related concepts over the years, such as CASSIOPEIA (Molina et al., 2014), DCI-4HD2D (Cassiopeia project, 2016), POEM (Cook et al., 2012), or ComplexityCosts (Cook et al., 2016). We refer the reader to the section 2 for more information on those projects relate to the development of the concepts implemented in Mercury, and in particular the basic 4DTA mechanism. On the other hand, the advanced version of 4DTA ventures outside of those current industry practices so to try to explore novel and creative solutions to the problem of delay management. The techniques implemented in this mechanism were designed and tested during the SESAR exploratory research project Domino Mazzarisi et al. (2020); Zaoli et al. (2020).

Basic implementation

In its base level implementation, 4DTA will apply simple rules of thumb that serve as an approximation of the current practices in the airline industry for the tactical management of flight delay and waiting for passengers at the hub:

- at 'Passenger check', the flight will wait for any passenger with a flexible ticket whose at-gate time is estimated to be at most 15 minutes after the flight's expected push-back time, taking into account their expected minimum connecting time.
- at 'Pushback', the cost index is calculated before the take-off and it is fixed throughout the flight. The attempted delay recovery is performed according to the probability distribution shown in Figure 5. Notice that departure delays below 15 minutes are never recovered, as many airlines consider it economically non-beneficial.

  The model accounts for the fact that the maximum delay that can be considered for recovery is limited by the amount of extra fuel that would be required to perform this recovery, and it is additionally capped at 70% of the total amount of additional fuel available (besides the required for the flight plan). Moreover, in order to make the application of this rule more aligned with current practices, the flight never speeds up to the maximum possible speed; rather, the speeding up is capped at 90% of the maximum velocity. Finally, if after applying all of these constraints, the amount of delay that can be recovered is lower than 5 minutes, no recovery is performed. This behaviour was decided after consultation with the experts due to the fact that the recovery of the delays lower than 5 minutes is seldom performed.

The possibility of aircraft frame swapping are not yet considered in the model, i.e., it is considered that an aircraft frame will operate the flights that are planned for it at the beginning of the simulation. The sequence of flights on the rotation plan will be maintained.

Advanced implementation

Advanced versions of the 4DTA mechanism were already explored in previous research such as Cassiopeia project (2016) or Delgado et al. (2016), where operations at a hub were optimised for a given airline relying on a coupled DCI and WfP approach and corresponding decisions are based on expected costs. The effects of a simplified 4DTA mechanism at a network level, from a cost resilience perspective, were analysed in Cook et al. (2016). In this paper the network implications of an advanced implementation, as described below, are further explored.

As explained in the previous section, in the baseline implementation, DCI and WfP decisions are decoupled and taken considering only basic rules of thumbs while the flight is still on the ground, and they are not reassessed. In the advanced version used in this article:

- When the 'Passenger check' event occurs, the AOC performs a joint assessment of departure delay recovery and wait for passenger options. In this decision-making process, AOC estimates the cost of delay and recovery options by speeding up, as well as costs of waiting vs. not waiting for missing passenger groups. Finally, it

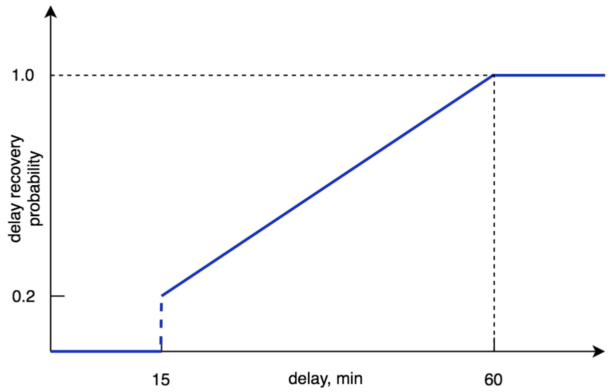

Figure 5: Probability function based on which the decision on the cost index (delay recovery) before departure is made, base level.

makes the decision that minimises the total cost and according to it cost index might be changed (speeding up) and a number of passenger groups waited for.

- Then, at Top Of Climb (TOC), the flight considers its expected arrival delay by comparing the current estimates of EIBT and SIBT. According to that estimate, the flight performs a potential delay recovery by looking at fuel and time costs. Additionally, the flight can decide to slow down at the top of the climb in case it estimates to arrive more than 30 minutes before the expected arrival time. In that case, the flight reduces its speed so to plan for the arrival at exactly 30 minutes before its Estimated Time of Arrival (ETA). This has been done in order to save fuel and prevent potential holding times which can likely occur in case of such very early arrivals. Fuel costs cover the cost of the extra fuel that would be needed in order to recover a (part of) delay, or saved if slower down, while time costs cover a number of costs that come with unrecovered delay (maintenance, crew, and passenger related costs).

Note that even in the advanced implementation of this mechanism, the decision for modifying the cost index and/or waiting for passengers are performed by the AOC without coordination with other agents in the system. The model assumes that ATC will grant the adjustment of the cruising speed if tactically modified, but does not coordinate which slot at arrival will be assigned to the flight by the E-AMAN. In some cases, this might lead to sub-optimal decisions, such as speeding up only to be affected by congestion at arrival.

4.1.4. Passenger costs

Passenger groups are modelled as simple placeholders which contains information on the passenger itineraries and their characteristics (flight sequence, type of passengers (‘standard’, ‘flex’), fare, number of passengers in itinerary group), which are handled by other agents, notably the AOC.

Costs associated to passenger delay are composed of three components:

- duty of care,
- compensation,
- soft costs.

These costs are briefly described in Table 5. First, duty of care is computed based on the departure delay of a given flight. For this, we used average values paid by companies from (Cook and Tanner, 2015). Duty of care are paid irrelevantly of the type of delay flights experience.

Second, compensations are given to passengers following Regulation 261 from the EU (European Commission, 2004). The rules of compensation are very specific. A passenger is entitled to compensation only if:

- the main reason for their delay is due to the turnaround, cancellation or other issues associated to the airline processes, or

	120 min $\leq \delta t < 180$ min	180 min $\leq \delta t < 240$ min	$\delta t > 240$ min
Short haul – $d \leq 1500$ km	0€	250€	250€
Medium haul – 1500 km < $d \leq 3500$ km	0€	400€	400€
Long haul – $d \geq 3500$ km	0€	300€	600€

Table 6
Compensation entitlement for passengers, based on the great circle distance between origin and destination and the delay at arrival.

- the main reason for their delay is reactionary delay as long as this reactionary delay is not generated due to prior ATFM regulations (except if the reason for those regulations were weather related, in which case the entitlement will prevail).

Delays due to capacity (ATFM delay) do not trigger compensation. The exact amount of compensation is a function of the arrival delay to the final destination and the distance between origin and final destination, as explained in Table 6. Importantly, not all passengers claim this compensation. Over the past there has been a significant increment of the claiming rate (now around 70%), but in 2014 (the baseline used for our model, see Section 5), only 11% of the passengers were claiming compensations. This is reflected in the model by multiplying the values in the table by 0.11.

4.2. Flight Agent

The Flight Agent models the processes related to the operation of a flight. When the 'Pushback ready' event is triggered, the flight request from the Departure Manager a departing slot and considering an estimated taxi-out time, provided by the Ground Airport agent, computes the push-back time.

At push-back, the Flight request the actual taxi-out time from the Ground Airport and with that information calculates the take-off time. Then the Flight agent integrates the trajectory of the flight using BADA 4 performances and considers the flight plan and uncertainties linked with wind and distances (e.g., considering that the route might be shortened or increased due to ATC intervention, such as granting short-cuts). When the flight reaches pre-defined waypoints the agent triggers 'Flight Crossing Point' events so that the Radar agent can notify interested parties (see Section 4.5).

If the arrival airport operates an E-AMAN, when the flight enters its planning horizon, the E-AMAN will be notified by the Radar, and subsequently request information from the flight (expected arrival time) and assigns a slot. The flight then updates its cruising speed to absorb delay, if possible. Once the flight reaches the execution horizon of the E-AMAN (or the Arrival Manager (AMAN) for airports without the extended management of arrivals), the E-AMAN agent is notified and a final slot assigned to the flight, the required delay to be performed as holding will be communicated to the flight who will estimate the fuel required to perform this holding.

Note that the flight plan that the flight is operating can be updated with information from the AOC. For example, the estimation of a new cost index can be performed at the top of climb as a part of the "4D trajectory adjustements" mechanism in its advanced implementation (see Section 6).

The Flight agents record the information related to the performance of the flights so that indicators such as flight time or fuel consumption can later be recorded as part of the output of the model.

Finally, when the flight lands, a taxi-in time is requested to the Ground Airport and, when the flight reaches the arrival gate, the Arrival event is triggered launching the aircraft turnaround and the passenger processes explained in the AOC Agent section (Section 4.1).

4.3. Ground Airport Agent

The Ground Airport agent is tasked with estimating and computing the turnaround, taxi-out and taxi-in times. For each of these processes, it provides the following two services:

- provide an estimate of the process time,
- compute an actual time for the process.

The first one is used by the AOC and the Flight when estimating departure or arrival time. For instance, the Flight agent requests an estimated taxi-out time before departure to schedule its push-back time. Then, at push-back, the

Flight requests an actual taxi-time to compute the time of arrival to the runway and the potential taxi delay experienced. These times are all drawn from probability distributions, based on the type of aircraft, the airport and the type of airlines operating the flight.

The agent has an additional service, which is to provide connecting times for passengers. In this case, the Ground Airport provides the actual time that the passenger needs to reach the connecting gate. It does not require to provide estimates as the AOC agent uses directly static minimum connecting times values (MCT), which are fixed per airport and connection type (domestic-domestic, international-domestic, international-international), when estimating if passengers might miss a connection.

4.4. DMAN and E-AMAN Agents

The model features one instance of DMAN and one instance of E-AMAN per airport (around 2×800 in total). Both use a queue system equivalent to the ATFM regulation queue described in 4.6.

DMAN is the simplest agent. Nominal values of capacity which incorporate the average effects of mixed operations and aircraft sizes are considered. The agent receives messages from the Radar (to be notified of flight plans) and from the Flight (to request departure slots). If flights are cancelled, it is also notified of it by the Radar and its queue is updated accordingly, i.e., freeing a slot.

The E-AMAN is more complex, but builds on the same principles. All airports have an E-AMAN agent which has a tactical horizon. In the tactical horizon flights are assigned a final arrival slots to adjust the landing sequence. If delay is required it will be performed as holding by the flight.

Additionally, some airports have an extended horizon for planning the arrivals. In this case, when the flight enters the planning horizon, the E-AMAN requests information from the flight on its status. Different levels of implementation are available in Mercury when it comes to performing optimisation of arrivals. The flight will provide the E-AMAN information on the earliest time when it can reach the arrival runway. The E-AMAN uses this information to pre-assign a slot to the flight following a first-come-first-served principle.

Regardless of how the slots are assigned in the (extended) planning horizon, the flight will receive an estimated delay which will allow some fuel saving strategies to absorb delay to be implemented. The Flight agent will then use this information to update its cruising speed, and if possible save fuel during the final part of the cruise and the descent, reducing the duration of the required holding.

Finally, note that for the E-AMAN queues, the capacities are defined based on the information on airport capacities and they might be adjusted if explicit ATFM regulations are issued at those airports during the duration of the regulation (see Section 4.6).

4.5. Radar Agent

The Radar is a central agent (with one instance in the simulation) which notionally represents the process of tracking a flight during its execution and share the information on aircraft reaching some milestones during the flight. To do this it relies on a number of dedicated events.

During the flight plan submission, the Radar agent receives the flight plan from the NM. At this stage, the flight plan is a set of segments, mainly composed a climbing phase, a cruise phase (which could include climb steps), and a descent phase. However, some agents may be interested in being notified when the flight reaches significant points. For instance, the E-AMAN needs to know when a flight enters its tactical or strategic horizon (see Section 4.4). This information is not directly available from the flight plan as there might not be a waypoint which marks this point (and this information is not known by the AOC that produces the flight plan).

The Radar agent thus creates an *augmented* flight plan. First, during the initialisation of the simulation, interested parties can register to the Radar and ask to be notified when a flight with certain characteristics (e.g., arriving to a given airport) verifies a given condition during execution (e.g., reaching a point in distance before landing). When a new flight plan having these characteristics arrives to the Radar, the Radar creates new waypoints in the trajectory corresponding to the condition verification. It also creates a new event ('Flight Crossing Point'), which is stored within the flight plan. When the flight reaches this point (i.e., finishing the previous segment), the Flight agent triggers the event (as explained in Section 4.2). The Radar, which listens to the event, can then propagate the information by sending messages to the interested parties.

This setup is very flexible. It enables various agents to subscribe to the service, allowing them to track the progress of the flight at key stages. It also decouples the success of events from the flights and the agents which are interested on being notified. Only the Radar agent is listening to these events which are triggered by the flights.

The Radar agent thus listens only to one type of event ('Flight Crossing Point'), of which there are many instances. It receives messages from the NM for new flight plan and flight plan cancellation and sends messages to subscribers when points are reached, or flight plans cancelled.

4.6. Network Manager (NM) Agent

The Network Manager (NM), which has only a single instance in the simulation, has a simplified view of the European airspace. It does not have an explicit knowledge of sectors and traffic, and thus is not an explicit modeller of the full ATFM network. Instead, the NM uses:

- random en-route ATFM delays, based on empirical data (see Section 5),
- and explicit regulations at airports, sampled from empirical data.

In the first case, there is a probability of a flight being regulated due to going through some regulation issued by airspace elements crossed by its flight plan. If this is the case, we sample a cause for the most penalising regulation based on empirical data (e.g., weather, capacity), and based on this type of regulation, we use a tailored distribution to draw some delay. The type of regulation that issues the delay to the flight is stored so that if the ATFM slot is missed and a re-submission is requested, the same distributions are used, maintaining a good inner consistency.

In the second case, when regulations are explicitly modelled at airports, we sample directly empirical regulation at airports, using starting/ending timestamps as well as capacity constraints to generate the regulation. Then, an explicit 'queue' of slots for the regulation is modelled. The implementation allows for any kind of timestamps and capacity combinations within the same regulation. For instance, an airport can be constraints to 12 arrivals in the period between 10:00 and 10:30, then 20 between 10:30 and 11:15, etc. This matches the level of precision available in the data where capacity adjust its expected evolution. Flights which would infringe the capacity regulation are assigned to slots in the queue using ration by schedule. This produces a delay which is the ATFM delay. If flights are cancelled or delayed (and slots missed) the initially assigned slot will be freed.

Due to potential race conditions between flights trying to access the queue at the same time, we use a system of resources to manage the regulation queues. An agent (an AOC) needs first to book this resource (called a 'booker') to access the queue. The booker allows only one agent to have access to the queue at the same time. Hence, an AOC typically tries various flight plan and obtain some ATFM delay based on a *fixed state* of the queue. Once it has reached a decision, i.e., submitted a flight plan, it releases its booking and the booker can give the hand to the next agent (another AOC). The NM is in charge of managing the bookers. Note that this is just an implementation detail to avoid concurrence which might produce inconsistencies on the availability of slots in the queue.

The NM reacts only to messages, not events. First, the NM processes the flight plan submissions. The NM will check if the flight breaches a curfew, in which case the flight plan will be rejected. Otherwise, the NM will request the dissemination of the flight plan, thanks to the Radar (see Section 4.5). The acceptance of the flight plan will then be communicated to the AOC. Second, the NM answers to ATFM delay requests. This is when the NM checks the state of the queue at the destination airport, or draws some en-route ATFM delay randomly based on the underlying probability distribution (if the flight is not already in an airport regulation). Finally, the NM manages the cancellation of flight plans, as required by the AOC, by disseminating the cancellation to the Radar and updating the ATFM regulation resources (updating the available slots at explicit regulations at airports) if needed.

4.7. Flight Swapper Agent

The Flight Swapper is inspired by the SESAR User-Driven Prioritisation Process (UDPP) (Pilon, 2016; SESAR, 2018). In practice, UDPP allows to protect flights by various mechanisms by exchanging ATFM regulation slots among flights arriving at the same regulated airport. Important flights can thus be protected and have small delays, whereas other flights bear the higher delays.

In the model, the AOC delegates the computation of swapping possibilities to another agent (as in reality sometimes airlines delegate it to EUROCONTROL): the Flight Swapper. The Flight Swapper provides the AOC with the best possibility when important flights with ATFM delay need to be protected. The process is the following.

- When an AOC has some flights falling into a regulation at their arrival airport, it sends the list of flights and the associated cost functions to the central Flight Swapper agent.
- The Flight Swapper agent finds the best slot allocation for the airline, i.e., the one with the minimum the total cost (found by brute force), and sends it back to the AOC.

- The AOC makes the final decision (using the best allocation) and sends a swapping request to the Network Manager agent. The latter updates the relevant flights.

The current deployed implementation of UDPP is available only to flights belonging to the same company, under regulation at the same arrival airport. Other processes are under development, in particular to enhance the benefit to low-volume users, given that the current implementation implies high volume of traffic at the regulated airport is required to fully benefit from the mechanism.

The decision to keep the Flight Swapper as a separate entity, rather than part of the AOC or the NM, is driven by the advanced implementation of the swapping mechanism implemented during Domino. Indeed, during the project we developed the possibility for airlines to swap flights between them, and not only for their own flights. For this, they communicate their true cost to the Flight Swapper[6], which finds the best swapping option. This is also in line with further UDPP development planned by Eurocontrol, which could play the role of our Flight Swapper in the future.

5. Model calibration

In this section, we present the calibration and validation methodology used, together with the input data used to perform those tasks and their outcome. Since Mercury is employed to model the operations of a one concrete nominal day, in the calibration process we parameterised the model considering the historical values reported for the day in question (see Table 8) and compared its output against the historically reported performance indicators (see Figure 6), as obtained through consultation with experts. This way the output of the model is validated against the historical observations.

5.1. Model input

Mercury requires a significant amount of data to define in detail the parameters needed to execute a simulation. The input is composed of:

- information related to the scenario,
- BADA 4 performance data,
- flight schedules,
- probabilities of ATFM delay per ANSP for en-route ATFM regulations,
- sample of ATFM regulations issued at airports,
- distributions of turn-around times, connecting times, taxi-times, non-ATFM-delay for each airport,
- flight plan pool,
- information on curfews at airports,
- airport data, such as runway capacity,
- passenger itineraries,
- general information on the airlines (e.g., alliance),
- information on the presence of arrival managers at airports,
- various scenario-independent parameters (e.g., fuel),
- various scenario-dependent parameters (e.g., cost index management implementation).

Most of this input is fixed in the present article, the rest is specified for each scenario.

[6]This is equivalent to assuming that there is some trading mechanism in the background allowing to reveal the true costs of the airlines.

Data source	Main usage	Reference
DDR2	Used to get the set of flights, origin-destination, routes, aircraft type, estimated cruise wind, distributions on climb and descent profiles, requested nominal cruise speeds and flight levels, companies, alliances, airspace structure, ATFM regulations	(EUROCONTROL, 2015b)
Cost of delay report	Used to compute cost of delay function	(Cook and Tanner, 2015)
IATA Summer Season 2010 from CODA	taxi times	(Cassiopeia project, 2016)
DDR2	minimum turnaround times, minimum connecting times	(ComplexityCosts project, 2016)
CODA	non-ATFM delays	(EUROCONTROL, 2015a)
Paxis, GDS	For passenger itineraries, including fares and class	(ComplexityCosts project, 2016)
Innovata (Cirium)	Flight schedules	–

Table 7
List of data sources used within the project.

5.2. Data

Agent-based models like Mercury need to be calibrated to produce reliable results. So even if the models are able to simulate any particular input, the quality results will be dependent on the calibration of some distribution parameters. In this article, we present an initial calibration of the model, using a few target metrics from data and some input data from a particular day. For simplicity, we used the 12$^{\text{th}}$ of September 2014, a day which has already been curated in past projects (ComplexityCosts project, 2016; Delgado et al., 2020). This day represents a busy day with no major disruptions.

For this, we used various data sources, summarised in Table 7. First, the datasets are restricted to only commercial flights landing or taking-off in Europe. DDR2 data are used to obtain for all flights' characteristic relevant for the model, including possible routes or aircraft rotations. Schedule data are used to extract the SOBT and SIBT of these flights. Passenger data are used to get individual itineraries and merge them with the flight dataset. Other data sources are used to estimate certain parameters in the model.

DDR2 data are also used for sampling by various agents in the model. In particular, initial flight plans are prepared based on the flight plans observed in the data. This is done by performing a clustering of possible routes between each origin and destination, and using the information from DDR2 to estimate requested flight levels and speeds. BADA 4 aircraft performances are used to integrate the flight plan estimating expected fuel consumption and times. Wind between regions is also estimated based on historical recorded DDR2 data. Likewise, these datasets are used to sample regulation types and regulation delays. In the case of regulations a longer period of several AIRACs has been used to calibrate the probabilities of regulations and magnitude of ATFM delays.

AOCs are grouped in alliances obtained from data (see Table 7). These alliances allow the AOCs to reallocate passengers within the alliance if there are no available seats for them on one of their own flights. In practice, besides the main alliances (e.g., Star Alliance), there exists a vast spectrum of possible bilateral agreements between airlines. To keep alliances simple, we decided to group together all airlines which had flights in the same passenger itinerary. In other word, if an itinerary is composed of two flights belonging to two different companies, we consider these companies to be in the same alliance. This definition overestimates the possible itineraries for the airlines but ensures that passengers are re-accommodated, if needed, considering their original itineraries. Note that the original set of itineraries do not contain self transferred passengers (i.e., passenger itineraries built by passengers using two separate individual tickets in two distinct airlines).

5.3. Processes to be calibrated

Many processes in the model use various statistical distributions. For example, the average wind encountered by flights during the en-route phase, or the amount of ATFM delay assigned to a flight due to an en-route regulation.

Process	Distribution	Based on
Taxi-in/out	LogNormal distribution with modified mean, standard deviation for different scenarios	IATA Summer Season 2010 from CODA
Climb uncertainty	Normal distribution (minutes)	Analysis DDR2 difference between planned and executed trajectories (m2, m3) from DCI-4HD2D Project (Cassiopeia project, 2016)
Cruise	Normal distribution (Nautical Miles)	Analysis DDR2 difference between planned and executed trajectories (M2, M3) from DCI-4HD2D Project (Cassiopeia project, 2016)
Wind	Empirical probability distribution for planned wind during the cruise. Used percentile of wind between regions. No noise added on execution.	For each ANSP to ANSP origin and destination airport consider the difference between requested speed and observed average ground speed for cruise segments from DDR2 analysis (AIRAC1409).
Turnaround time	Exponential distribution based on minimum turnaround time based on airport size, aircraft wake turbulence category and type of airline. Distributions modified based on scenario	Analysis of turnaround times performed in POEM project and used in ComplexityCosts project (ComplexityCosts project, 2016)
Airport ATFM delay	Airport regulations are sampled from a historical day. The day is selected based on their percentile ranked by number of regulations at airport in the day. Airspace regulation delays are based on two distributions, one for weather and one all other types of regulations	Based on analysis of DDR2 (AIRAC1313-1413 excluding days with industrial actions)
Airspace ATFM delay	Empirical probability distribution function for regulations due to weather and regulations for other reasons.	Based on analysis of DDR2 (AIRAC1313-1413 excluding days with industrial actions)
Non-ATFM delay	Exponential distribution with parameters modified based on scenarios	-
Connecting times	Log-normal distribution based on minimum connecting times per airport and type of connection (national-national, international-international and national-international)	Based on analysis of minimum connecting times at ECAC airports originally performed in POEM project (Cook et al., 2012)
Variation of cruise length due to DCI	Normal distribution (Nautical Miles)	Analysis of Performance using Airbus PEP (Cassiopeia project, 2016)

Table 8
Calibration parameters in the model.

Table 8 presents some of the key processes that are modelled in Mercury, and how their distributions have been adjusted.

5.4. Validation

First, while the calibration is performed on average values in general, one can look at various distributions to understand how the model replicates the empirical data. Figure 6 shows examples of these distributions for arrival and departure delay. In this case, arrival delay distributions are quite similar, even if empirical delays seem to have a longer tail. On the contrary, departure delays differ between the simulations and in historical data. This is due to the fact that in the model we assumed no flight can depart before its scheduled departure time, whereas it seems that in practice a significant number of flights are departing before their schedule (down to 20 minutes). The impact on the simulations is likely to lie in the delay generation. More specifically, departing early triggers less reactionary delay, since flights arrive earlier at their destination and thus have more buffer if turnaround times are higher than expected. The exact impact is unknown, but if 30% of the flights depart early, they usually depart only a few minutes in advance, and thus this will have a mild impact on the system. However, it may be important for small companies, which have very little buffer otherwise. This will be corrected in future versions of the model.

Due to the fairly low-level nature of the model, various other metrics can be computed for validation. The average

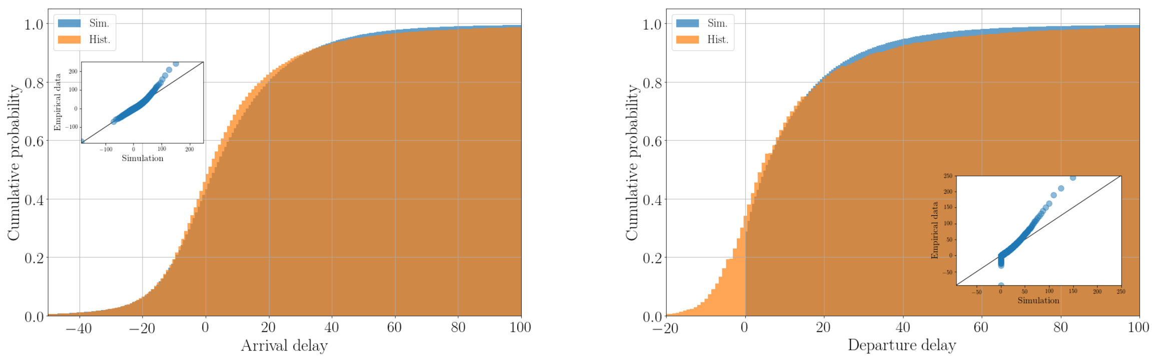

Figure 6: Cumulative distribution for arrival (left) and departure (right) delays, obtained from the model (blue) and the empirical data (orange) with corresponding QQ-plots in inset.

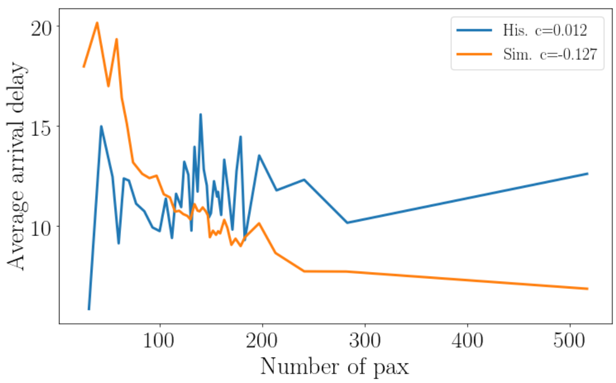

Figure 7: Arrival delay against number of passenger in the flight, averaged over quantiles. For historical data (blue) and simulations (orange).

delay as a function of the number of passengers in the plane is particularly interesting. It is expected that airlines will have different policies depending on the number of passengers carried due to different cost of delay. In Figure 7, we show the evolution of arrival delay (restricted to positive values) with the number of passengers, binned per quantile. Interestingly, there is no correlation between these two variables in the data (Pearson correlation coefficient $c = 0.012$, not significant to a 1% level). However, in the simulation results we observed a negative correlation ($c = -0.127$, significant).

This negative correlation is most likely the result of the fact that airlines in the model care about and proactively react to the cost incurred due to passengers delay. This difference could also stem from some model limitations, e.g., an overestimation of the passenger costs. It could also come from the fact that the airlines may use other instrumental goals that may conflict with their profit-driven paradigm. An obvious candidate for that is the On-Time Performance (OTP), an important measure for airlines, which is flight-based, and not passenger-based (limiting in some cases the waiting for passengers).

5.5. Model output

The raw output of the model consists of a series of detailed metrics for each flight and each passenger. Aggregated values (KPIs) are then computed based on this output. The output includes:

- flight information: scheduled, controlled (after ATFM), and actual off and in block times, ATFM/ Non-ATFM/ Reactionary delay, main reason for ATFM delay, taxi-times, information on fuel consumption, climb/ cruise/ descent/ holding phase parameters (times, distances, fuel), speeds (selected, ground, average wind), etc;

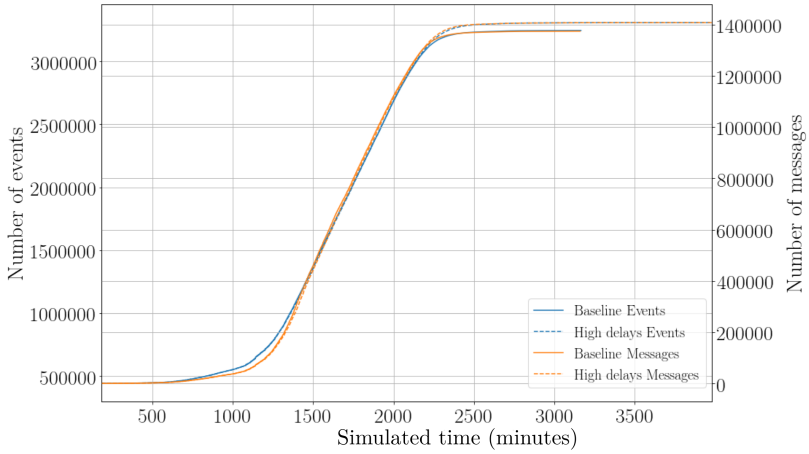

Figure 8: Number of messages created and exchanged and number of events created/destroyed/triggered during the simulation for two scenarios.

- passenger information: actual departure and arrival times, missed connections, new reallocated flights, compensation due to Regulation 261, etc;
- specific information by the airlines on the use of the different mechanisms: dynamic cost indexing, user-driven prioritisation processes, etc.

5.6. Computational aspects, model's behaviour, and emergence

We use test scenarios in order to evaluate computational aspects of the model (time and memory usage, messages and events in the simulation), as well as to present an emergence of complex behaviour.

5.6.1. Scenario description

These scenarios run a simple setup, with an increasing number of airports (and thus flights) in the simulation. This allows us to study how the model behave internally when the interactions between agents increase. To build these scenarios, we look at the traffic at every airport in our dataset (around 800) and we rank them by decreasing traffic. We then choose 10 values, from 1 to the maximum number of airports, and define one scenarios for each. In each scenario, only the flights taking off or landing at these airports are kept. Because the distribution of traffic among airports is highly unbalanced, we use a logarithmic law to select the 10 values, which gives a roughly linear progression in the number of flights included in the simulation. For each value of the number of airports, we simulate two sub-scenarios. The first one is the baseline, in which we are using the calibrated model as is. The second one is a scenario where we artificially increased the level of delay in the system, by modifying various distributions of delay (including turnaround). This increase is quite drastic: the average delay in the system is roughly three times the delay in the baseline for the whole system. This allows us to study how the system answers to major disruptions.

5.6.2. Simulation indicators results and emergence

First, we start by looking at the number of events and messages created and exchanged during *one* simulation, where all airports are included, i.e, including all flights in a whole day of operations in Europe. In Figure 8, we show these metrics as a function of the simulation step. As expected, the number of messages/events starts by being very low, corresponding to early morning in simulated time, to increase roughly linearly during the simulated day and saturates towards the end of the simulation, waiting for the very last flights to land. We show on the figure the metrics for the baseline and the severely disrupted scenario. The progression in terms of number of messages and events is almost the same, until the end where they increase in the high delay scenarios. This is due to the fact that airlines are more active with the last flights of the day in order to avoid breaching curfews.

Figure 9 presents the simulation time and memory reserved for the simulation per flight as the size of the scenario increases. As shown, both time and memory per flight decrease with the number of flights simulated[7], showing good

[7] This is mainly due a big overhead, as the model needs to read data and prepare the model before executing it.

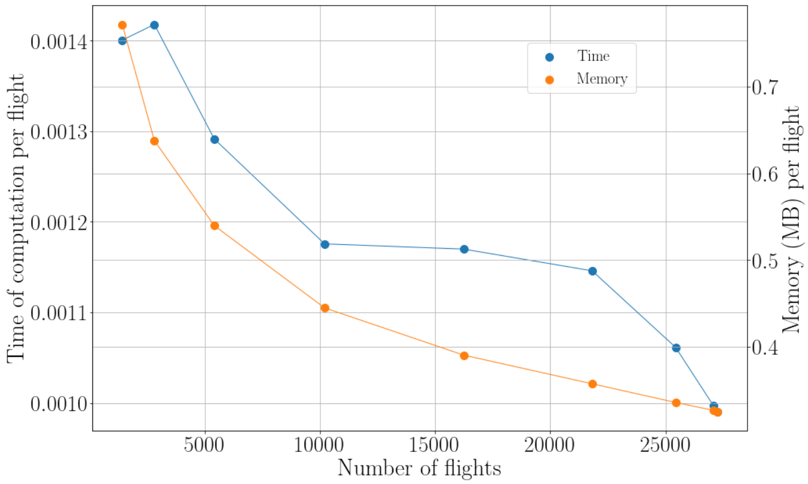

Figure 9: Average time of computation per flight, in minutes (on a CPU@3.2GHz), and average memory of process per flight, in MB

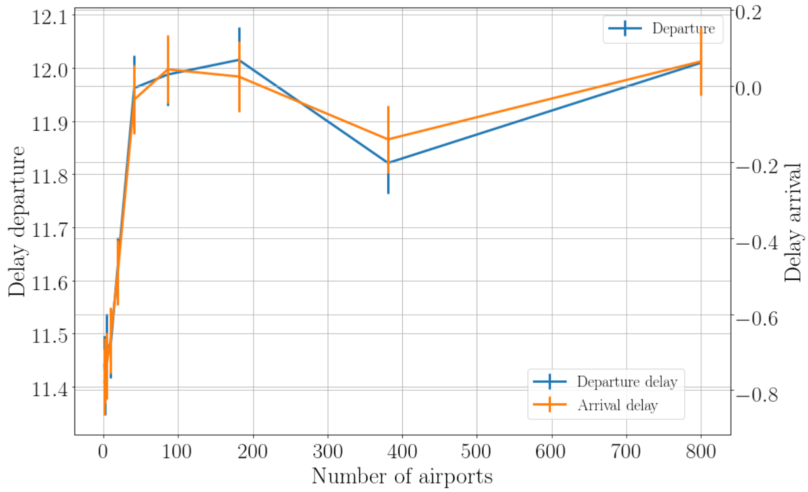

Figure 10: Average flight departure and arrival delay at Heathrow depending on the number of airports included in the simulations. Error bars are standard errors.

scalability. Note that the code has only been loosely optimised at this stage, and could gain from a more thorough round of optimisation.

Finally, Figure 10 shows the evolution of two metrics (departure and arrival delay) computed only on flights departing/arriving at Heathrow, as a function of the number of airports in the simulation. Overall, the variation are of the order of 0.5 minutes, around 3-4%. Since these metrics are computed on the same set of flights, the variations are due purely to the interaction with the rest of the system. It is interesting to notice that the delay increases with the inclusion of more airports at first, but seems to decrease again later. That might be because the smaller airports act as buffers for the system and delay is computed for all flights in the simulation, i.e., flights operating to/from those smaller airports experience lower delays.

6. Results

In this section we present some results obtained with the model on a specific scenario, featuring basic and advanced 4DTA capabilities, in order to assess the impact of this mechanism. The objective of this scenario is to present how

Mercury is able to capture complex behaviour and emergence in a full operational assessment when evaluating the impact of introducing a given mechanism.

6.1. Scenario description

To evaluate the performance of the model capturing the behaviour of stakeholders and KPIs in the European-wide ATM system when 4DTA is implemented in its advanced version, a case study disruption at three hubs has been implemented. The performance of the system under the implementations of the advanced and basic 4DTA mechanism is compared through that case study, with results as shown on Figures 11 and 12.

In this scenario, ATFM regulations are modelled at London Heathrow (EGLL), Amsterdam Schiphol (EHAM) and Paris Charles de Gaulle (LFPG). These regulations are set in the morning (06:00 - 14:00 local time), and reducing the capacity at the airport to half their nominal arrival rate (EGLL with 54 arrivals/hour, HEAM with 45 arrivals/hour and LFPG with 44 arrivals/hour).

In addition to these manually defined ATFM regulations, the rest of the delay in the system is set as default (i.e., nominal day of operations) and ATFM regulations are defined at other airports based on a randomly selected nominal day.

Note that even if the manually set disruptions are only defined for the three hubs, the 4DTA mechanism is implemented for all the airlines everywhere in the system, and the whole network is simulated.

6.2. Agents' behaviour

Figure 11 on the left panel presents the characteristics of flights at different flight phases and the decision they are making for the baseline 4DTA mechanism. Note that in the baseline mechanism, the flight only decides to either wait for passengers prior departure if passengers that were connecting to this flight were delayed, and then prior departure decide if maintain the speed or speed up. Note how most flights decide not to wait for passengers even though some of the flights delayed at departure consider it as an option. Most flights will also decide not to speed up and maintain the nominal speed. This is in line with the fact that no delay is recovered if at least five minutes cannot be recovered.

In the right panel of Figure 11, the speed selected for the flights speeding up is presented as a quantile between maximum range speed (MRC) and maximum speed (MMO). As shown, only 2.46% of the flights decide to speed up. There is, as expected, a relationship between delayed flights and the probability of trying to recover delay since in the baseline mechanism the probability of trying to recover delay just depends on the initial departure delay. As it can be observed in the left panel, as the speed is increased significantly, some flights manage to arrive on time and the mechanism could be considered effective for delay recovery, even if at a high fuel consumption cost.

As observed, flights are deciding to speed up prior to departure considering their departure delay and trying to recover delay without assessing the expected total cost (just ensuring that no more than 70% of the total extra fuel available is expected to be used and that at least 5 minutes are expected to be recovered). As the amount of delay that can be potentially recovered by just speeding up is relatively low for most flights, this leads to selection of very high-speeds for around 2.46% of the flights with very large speeds. This behaviour will produce relatively large amounts of fuel burn (and cost) (see Section 6.3).

In the advanced modelling of the 4DTA mechanism, a richer behaviour is captured by the model. As shown in the left panel of Figure 12, reassessing the speeding up decision taken prior to departure once the flight is airborne (TOC) and introducing the possibility of slowing down produces a more complex behaviour.

The coupling of waiting for passengers and speeding up while assessing the expected total cost prior to departure leads to seldom decision to wait for passengers, and also all flights which wait for passengers in their turn decide to try to recover some delay. However, as shown in the right panel of the figure, only 1.94% of the flights consider speeding up prior to departure, and as far as fuel cost is concerned, there is a range of potential speeds that could be selected, albeit higher speeds are still the norm. In other words, waiting for passenger followed by speeding up with almost the maximum speed seems to be the most common decision taken prior to departure.

When the flights reach the TOC they reassess their delay. It is important to note that in this case the expected arrival delay is used and that costs prior to departure have already been accrued. This leads to the fact that most flights which decided to speed up prior to departure maintain that decision and some additional ones also decide to speed up, increasing the number of flights with higher cost index to 3.46% of the flights (note that this is higher than the 2.46% of flights speeding up in the baseline 4DTA mechanism implementation). However, as clearly shown in Figure 12 right panel, the selected speeds are in fact more moderate (in the 0.6-0.7 range between MRC and MMO) when compared

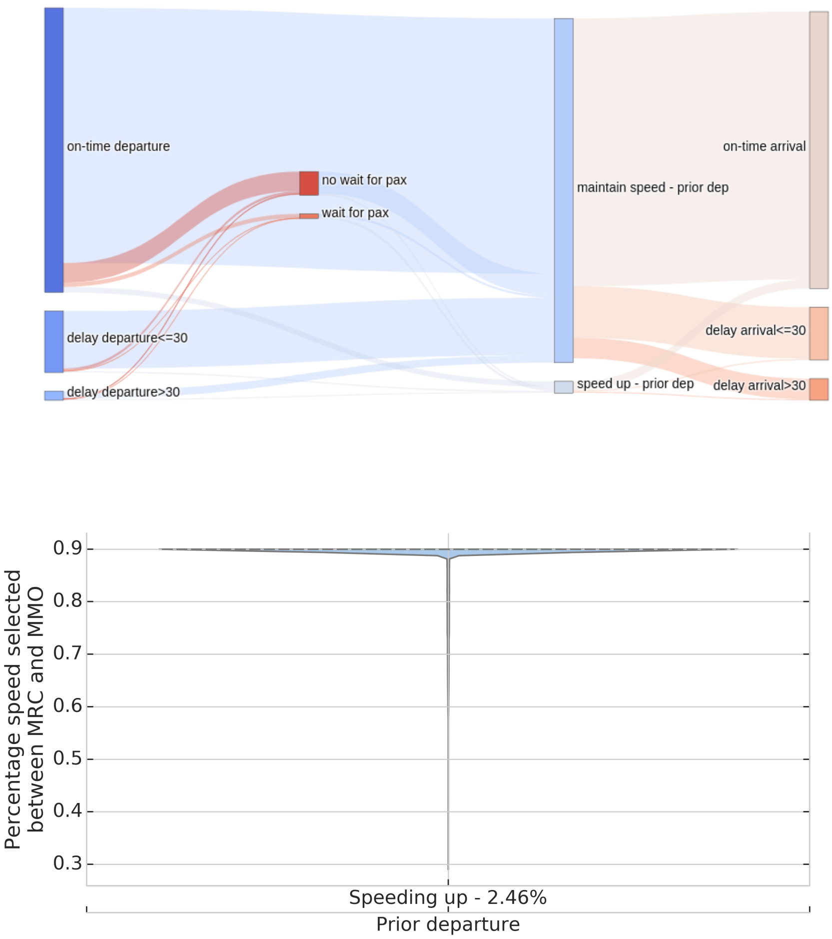

Figure 11: Speed selection and behaviour on baseline 4DTA scenario. Left: Decision of actions by flight phase, Right: Speed selected by flights at different phases with percentage of flights performing speed changes with respect to total number of flights in simulation.

to the higher speeds considered prior departure (in the 0.9-1 range). These speeds are also lower than in the baseline case. This shows the importance of considering the cost of fuel and the actual expected benefit in terms of delay cost.

Finally, it is worth noticing how the majority of the flights that decide to slow down are flights which were on-time on departure, but in some cases even flights delayed on departure decide to slow down. In total they represent 1.78% of flights. These flights will save fuel while increasing their flying time. As explained in Subsection 6.1, flights have the possibility to decide to slow down to save fuel if they are ahead of their expected arrival schedule. This highlights the importance of considering buffers. We will expect that small delays might increase when this mechanism is introduced (as slack in the buffers is reduced, i.e., used tactically to save fuel) but fuel will be saved leading to an overall cost reduction.

6.3. Impact on KPIs

The model output is very detailed in terms of metrics, since we can measure any variable attributed to any of the agents, as presented in Section 5.5. In this section we focus on two important indicators for airlines and passengers: delay and cost. We consider departure and arrival delay, and we further break down the indicator by considering the number of flights delayed by more than 15, 60, and 180 minutes. We also record the number of cancellations in each

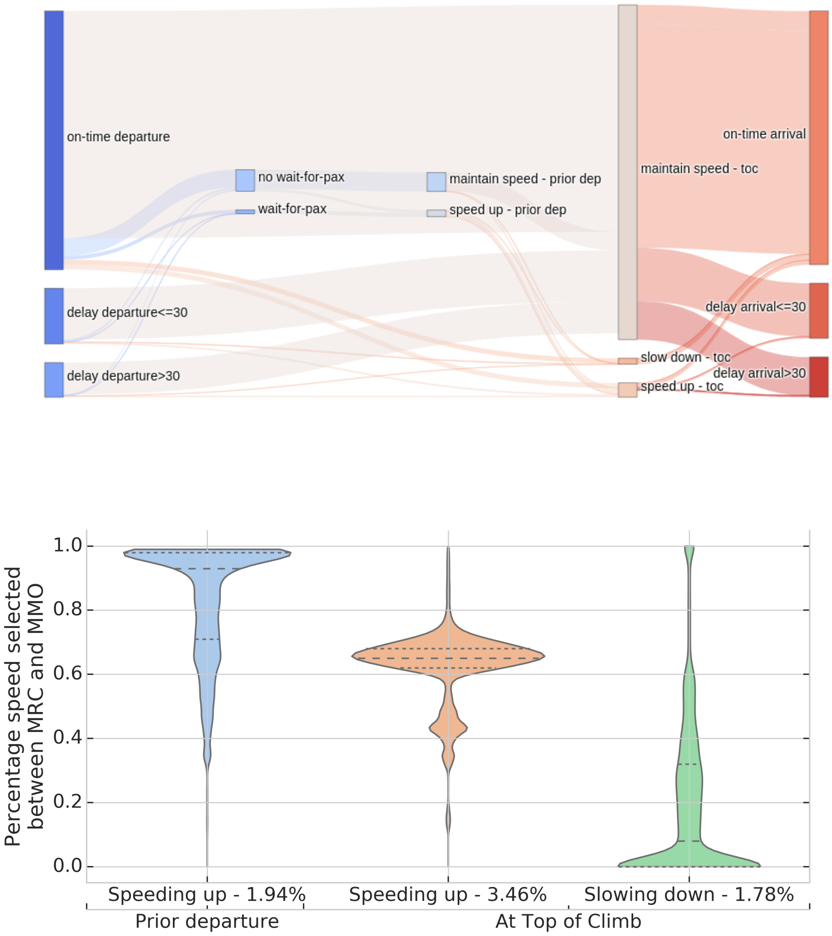

Figure 12: Speed selection and behaviour on advanced 4DTA scenario. Left: Decision of actions by flight phase, Right: Speed selected by flights at different phases with percentage of flights performing speed changes with respect to total number of flights in simulation.

simulation. When it comes to costs, we compute each of them independently: duty of care, fuel cost, non-pax cost, soft cost, and transfer cost.

We compute those metrics for all flights, as well as on the subsets of flights which depart/arrive at the disrupted hubs (EGLL, EHAM and LFPG). Finally, we further divide the results considering the types of airlines: Full Service Carrier (FSC), Low-Cost Carrier (LCC), Charter Carrier (CHT), and Regional Carrier (REG).

The results are presented in Figure 13 and 14 as percentage changes with respect to the baseline, indicating the standard error with error bars.

When computing indicators on all flights (Figure 13), the results across the airline types differ slightly. First, it seems that airlines either see the fuel cost decrease slightly (which has a large economic impact due to the high absolute value of fuel costs) or none at all. Non-passenger costs tend to decrease as well for all the airlines, whereas soft costs increase by more than 12%. This suggests a deterioration of the passenger experience, as discussed below. Departure delays are decreasing or they are constant for all types of airlines.

Average arrival delays increase largely for charter airlines, and to a lesser extent for FSC too. This seems to be driven partly by an increase in delays bigger than 15 minutes for the former. For the LCC and REG airlines, arrival delays experience a large reduction (3-6%), but it seems to be driven mainly by small delays (< 15).

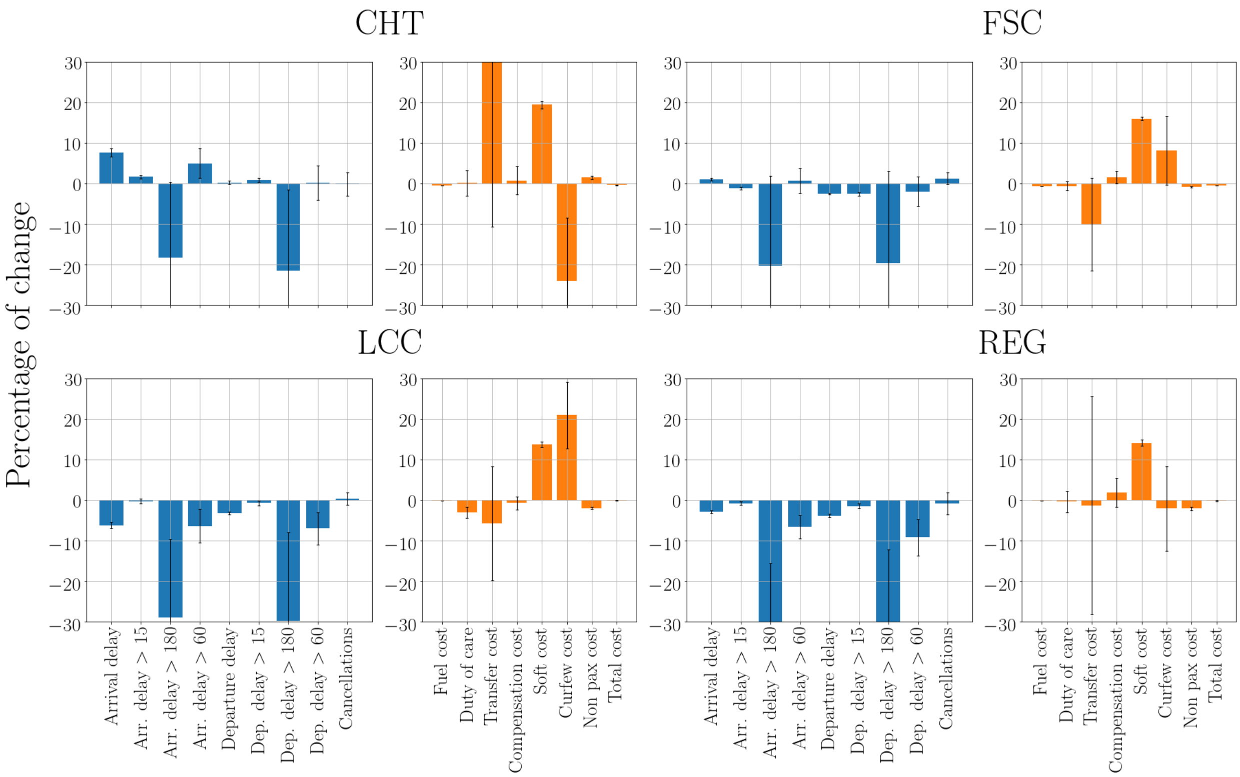

Figure 13: Percentage of change in KPIs between baseline and scenario with 4DTA, computed on flights from all airports.

Comparing these results to the indicators computed on the flights departing and arriving from the disrupted hubs (Figure 14) is also informative. First, soft costs are rising in this case too. Arrival delays seem to follow a similar but stronger pattern than above: Charter airlines and FSC airlines have higher arrival delays (> 20 %), mainly from small delays, while LCC and REG airlines do not have this issue. Fuel cost are also decreasing overall, by a small amount. Finally, non-passenger costs barely change in this case. It is worth noticing how in both cases the total cost remains either similar or shows a small reduction with respect to the baseline.

As the same apparent situation can be very different for passengers and flights, we also compute passengers indicators (see Figure 15). This time we focus only on delays, and we use the same thresholds of 15, 60, and 180 minutes. We also consider different passenger groups. First, we differentiate the passengers who are going through one of the disrupted hubs against all passengers. Second, we consider the difference between passengers with one leg itineraries (point-to-point, “P2P”) and passengers with multi-leg itineraries (connecting, “con.”).

Overall, it seems that the situation deteriorates in average for passengers (top chart on Figure 15), but improves for passengers going through the disrupted hubs. Indeed, most delays increase (or remain similar) in the first case, and decrease in the second one. However, on further inspection, there is a difference between P2P passengers and connecting ones. For connecting passengers, delays are increasing or staying the same, and negative delays seem to increase (in absolute values) for all passengers, driving the average down in this case. On the contrary, point-to-point passengers who are going through the disrupted hubs see their delays decrease, but increase in average when all airports are considered.

These results are not easy to interpret, and the output may be the interplay between different mechanisms and costs. First, connecting passengers are important from a cost point of view for the airlines, because they are potentially very expensive if they miss their connection (producing high hard and soft costs). However, waiting for connecting passengers may be a very expensive strategy when delays are already high in the system. Indeed, P2P passengers represent the majority on some legs, and may dominate the cost because of the non-linearity in the cost of delay function. Importantly, also because of this non-linearity, it is usually better for airlines to reduce high delays, to the potential detriment of other passengers. This may explain why passengers globally might be worse off from some of

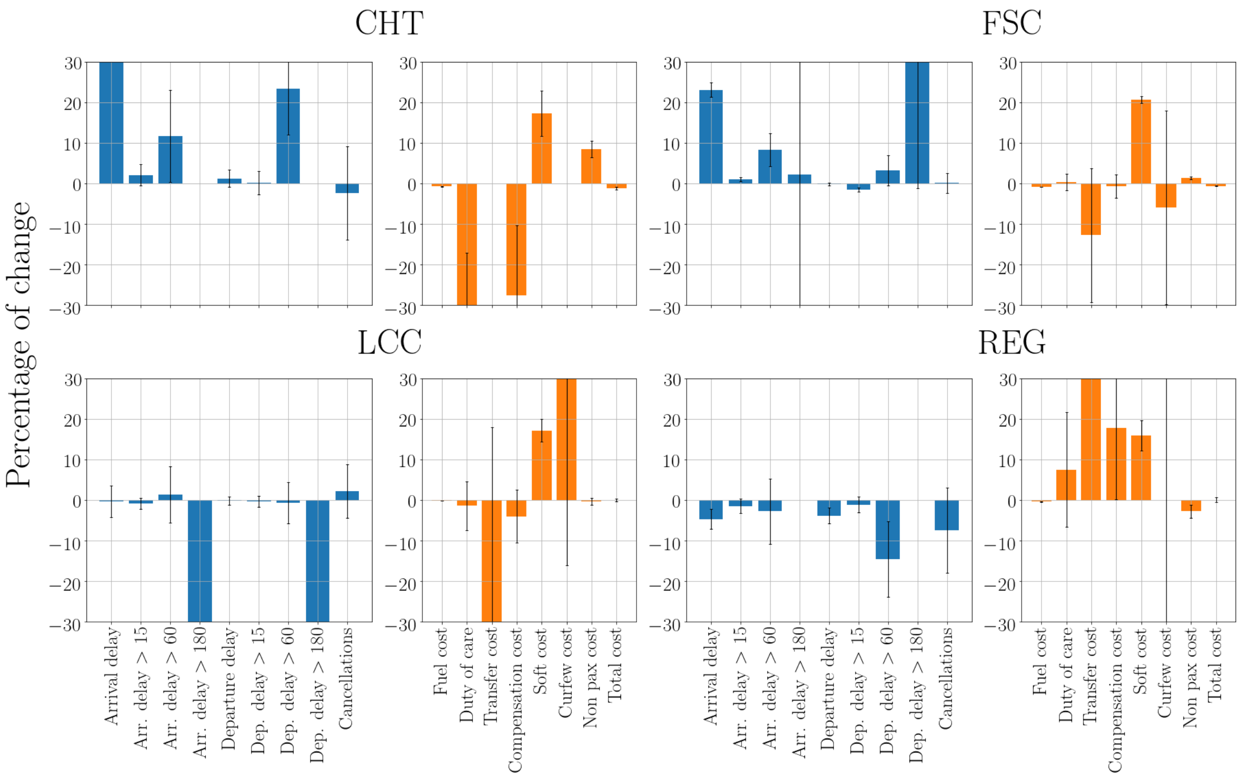

Figure 14: Percentage of change in KPIs between baseline and scenario with 4DTA, computed on flights only coming/going to disturbed hubs.

the airlines' actions, while the situation for passengers going through the disrupted hubs improve: they are the most impacted and drive the cost of the delay. Moreover, connecting passengers may be worse off because the effort to relieve their delay would in turn delay too many P2P passengers, who are already on the steep part of the cost of delay function.

Finally, it is worth noticing how by providing the possibility for flights to slow down in order to save fuel if they are expected to arrive earlier than scheduled (action decided by 1.78% of the flights as previously indicated), in some cases, might lead to an increment on small delays while saving fuel. Therefore, the overall cost for airlines will be reduced, as small delays will not trigger high penalisation in terms of delay cost, even if delays will overall increase.

7. Conclusions

The need for holistic air traffic models is strong due to the interconnectivity of the air transportation system, i.e., the complexity of air travel networks. Assessing new solutions in isolation is very difficult and usually sub-optimal as it does not account for possible emerging phenomena in such complex networks.

Therefore, an agent-based model such as Mercury, which allows for complex behaviour to emerge, as illustrated in section 5.6, is particularly suited to study delay management procedures, like the one presented in this article – combining waiting for passengers and dynamic cost indexing – where decision-making processes are decentralised and disseminated among various actors with complex objective functions.

As we show in section 6.2, through the comparison of the delay management mechanism 4DTA implemented on a baseline and advanced level, the strong (rational) cost-driven paradigm introduced on the advanced level is very beneficial in order to properly assess the actual expected benefit in terms of delay cost.

While beneficial in average, the impact assessment of the 4DTA mechanism shows some important trade-offs. First, a trade-offs between flight KPIs and passengers ones, already highlighted in the past with this kind of models, is clearly present. By focusing on cost reduction, airlines sometimes work against the interest of passengers. Going further, different passengers are not impacted in the same way. A direct consequence of regulation 261 for instance is

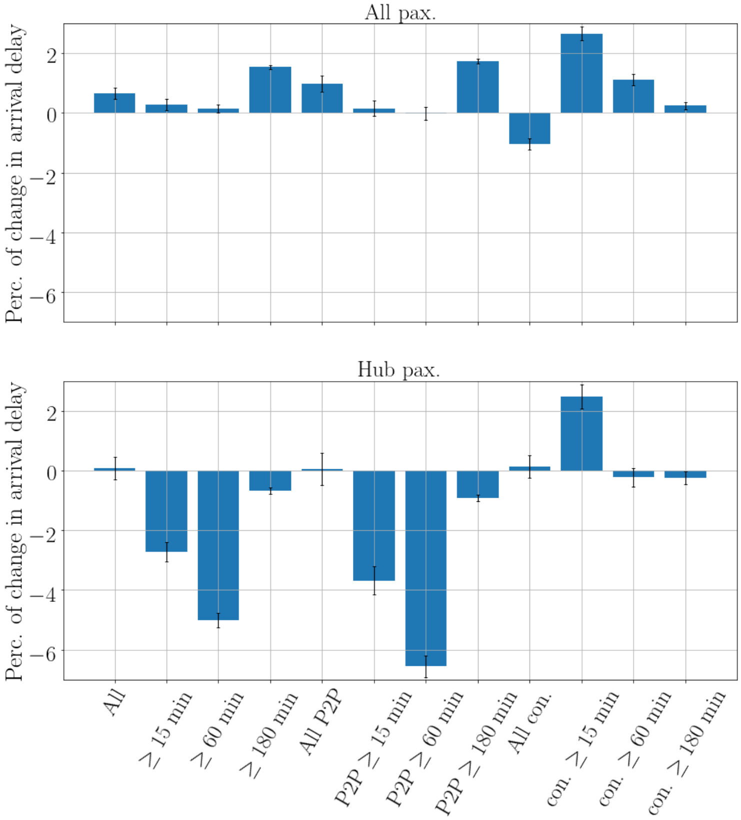

Figure 15: Percentage of change in the passenger arrival delay for all passengers (top) and only passengers going through the disrupted hubs (bottom).

that connecting passengers tend to be over-protected by airlines, to the detriment of point-to-point passengers. Even more interesting, it seems that the introduction of this mechanism has a negative impact on passengers that otherwise would not have been impacted by disruptions (in our scenarios, passengers not going through disrupted hubs). This is a typical network effect, and it is important to note that in this case the introduction of the mechanism **increases** the inter-dependencies among parts of the system.

Airlines are also impacting each other. Standard airlines are focused on their operations, waiting for passengers and modifying their speed, which may have an impact on others through capacity constraints. It seems overall that standard airlines gain from this particular mechanism a lot more than other types of airlines, in particular low-cost carriers.

These considerations pleads for a more nuanced approach in performance assessment schemes. Indeed, reducing the average delay (or event cost) may have unintended consequences, in particular on passengers, which, we remind, do not have the chance to be included in major performance schemes at the moment.

Finally, we acknowledge the fact that an important limitation of this kind of model is the difficulty to adequately calibrate them. As shown in this article, even after some careful calibration processes and usage of large data sets, some outputs may not match well with the empirical results, for instance due to a certain model hypothesis. Even though the model is able to simulate any given input, only the operational environment for which it has been calibrated will produce meaningful results.

Moreover, since the air transportation system is a socio-technical system, human actors often play crucial role in providing resilience to various disturbances. The lack of agents in the developed ABM that would model these human actors and thus allow us to capture the effects that emerge from their decision making processes is surely lacking in the current version of Mercury, and it would be a great addition to one future version of the model. One of the biggest obstacles to implementing this functionality is the difficulty in obtaining the data that would allow us to capture and model the behaviour of various human actors participating in the air transportation system.

Nevertheless, Mercury is particularly useful for performance assessment as it produces very fine-grained information about the system, which can then be aggregated in multiple ways (e.g., grouping results by airline, passenger type, airports, etc.), including passenger related indicators and airlines operational costs. Even though some actors from the real world system are not modelled or have limited actions in the current version of Mercury, high scalability and flexibility of Mercury that allows for adding new agents or implementing new agent rules, as intricate as modellers want or need them to be, makes the ABM paradigm of immense value for capturing different current and future real-world scenarios.

Acronyms

4DTA 4 Dimensional Trajectory Adjustments.

A-CDM Airport Collaborative Decision Making.

ABM Agent-Based Model.

AIRAC Aeronautical Information Regulation and Control.

AMAN Arrival Manager.

ANSP Air Navigation Service Provider.

AOBT Actual Off-Block Time.

AOC Airline Operating Centre.

API Application Programming Interface.

ATC Air Traffic Control.

ATFM Air Traffic Flow Management.

ATM Air Traffic Management.

BADA 4 Base of aircraft data version 4.

CHT Charter Carrier.

CODA Central Office Delay Analysis.

COVID Coronavirus Disease.

CRCO Central Route Charges Office.

DCI Dynamic Cost Indexing.

DDR2 Data Demand Repository version 2.

DMAN Departure Manager.

DSP Departure Slot Provider.

DSR Departure Slot Requester.

E-AMAN Extended Arrival Manager.

ECAC European Civil Aviation Conference.

EIBT Estimated In-Block Time.

EOBT Estimated Off-Block Time.

ETA Estimated Time of Arrival.

EU European Union.

FMS Flight Management System.

FP Flight Plan.

FSC Full Service Carrier.

GDS Global distribution System.

IATA International Air Transport Association.

KPI Key Performance Indicator.

LCC Low-Cost Carrier.

MB Megabyte.

MMO Maximum Speed (Mach) Operative.

MRC Maximum Range Speed (Mach).

MTOW Maximum Take-Off Weight.

MTT Minimum Turnaround Time.

NM Network Manager.

OTP On-Time Performance.

P2P Point to Point.

PEP Performance Engineering Programme.

PRB Performance Review Body.

REG Regional Carrier.

SESAR Single European Sky & ATM Research.

SIBT Schedule In-Block Time.

SOBT Schedule Off-Block Time.

TOC Top Of Climb.

UDPP User-Driven Prioritisation Process.

WfP Waiting for (connecting) Passengers.

XMPP Extensible Messaging and Presence Protocol.

Acknowledgment

This work has been performed as part of the Domino project which has received funding from the SESAR Joint Undertaking under grant agreement No 783206 under European Union's Horizon 2020 research and innovation programme.

References

Blom, H. A. P. and Bakker, G. J. (2015). Safety evaluation of advanced self-separation under very high en route traffic demand. *Journal of Aerospace Information Systems*, 12:413–427.

Boeing (2017). Fuel conservation strategies: cost index explained. In *AERO quarterly*, number 26 in quarter 2, pages 26–28.

Boeing (2019). Airports with noise and emissions restrictions. https://www.boeing.com/commercial/noise/list.page.

Bouarfa, S., Blom, H. A., Curran, R., and Everdij, M. H. (2013). Agent-based modeling and simulation of emergent behavior in air transportation. *Complex Adaptive Systems Modeling*, 1(1):15.

Bouarfa, S., Blom, H. A. P., and Curran, R. (2016). Agent-based modeling and simulation of coordination by airline operations control. *IEEE Transactions on Emerging Topics in Computing*, 4:9–20.

Cassiopeia project (2016). DCI-4HD2D - D3.2 Final technical report. Technical report, Cassiopeia Consortium.

Chen, Y., Wu, C., Lau, P. L., and Yung Agnes Tang, N. (2018). Simulating passenger's shopping behavior at airport with a conceptual agent-based model. In *2018 Winter Simulation Conference (WSC)*, pages 2342–2353.

Cheng, L., Fookes, C., Reddy, V., and Yarlagadda, P. (2014). Analysis of passenger group behaviour and its impact on passenger flow using an agent-based model. In *4th International Conference on Simulation and Modeling*, pages 733–738.

Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industry—concepts, models and methods. *Computers & Operations Research*, 37(5):809 – 821. Disruption Management.

ComplexityCosts project (2016). D4.5 – Final Technical Report. Technical report, SESAR JU.

Cook, A., Delgado, L., Tanner, G., and Cristóbal, S. (2016). Measuring the cost of resilience. *Journal of Air Transport Management*, 56:38 – 47. Long-term and Innovative Research in ATM.

Cook, A. and Tanner, G. (2015). European airline delay cost reference values, updated and extended values. Technical report, University of Westminster.

Cook, A., Tanner, G., Cristóbal, S., and Zanin, M. (2012). Passenger-oriented enhanced metrics. In 2nd *SESAR Innovation Days*.

Cook, A., Tanner, G., Williams, V., and Meise, G. (2009). Dynamic cost indexing - managing airline delay costs. *Journal of Air Transport Management*, 15(1):26–35.

de Arruda, A. C., Weigang, L., and Milea, V. (2015). A new airport collaborative decision making algorithm based on deferred acceptance in a two-sided market. *Expert Systems with Applications*, 42(7):3539–3550.

Delgado, L., Gurtner, G., Cook, A., Martín, J., and Cristóbal, S. (2020). A multi-layer model for long-term KPI alignment forecasts for the air transportation system. *Journal of Air Transport Management*, 89:101905.

Delgado, L., Martín, J., Blanc, A., and Cristóbal, S. (2016). Hub operations delay recovery based on cost optimisation – dynamic cost indexing and waiting for passengers strategies. In *Proceedings of the 6th SESAR Innovation Days*.

Delgado, L., Martín, J., Blanch, A., and Cristobal, S. (2017). Agent based model for hub operations cost reduction. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 3–15.

Delgado, L. and Prats, X. (2012). En route speed reduction concept for absorbing air traffic flow management delays. *Journal of Aircraft*, 49(1).

Delgado, L., Prats, X., and Sridhar, B. (2013). Cruise speed reduction for ground delay programs: A case study for san francisco international airport arrivals. *Transportation Research Part C: Emerging Technologies*, 36:83–96.

Domino Consortium (2018). Deliverable D4.1: Initial model design. Technical report, Domino Consortium.

Enciso, J., Vargas, J., and Martínez, P. (2016). Modeling and simulation of passenger traffic in a national airport. In *The Fourteen LACCEI International Multi-Conference for Engineering, Education, and Technology: "Engineering Innovations for Global Sustainability"*.

EUROCONTROL (2015). BADA overview - Base of Aircraft Data (BADA) EUROCONTROL's aircraft performance model.

EUROCONTROL (2015a). CODA Digest: All-Causes Delay and Cancellations to Air Transport in Europe – 2014. Technical report, EUROCONTROL.

EUROCONTROL (2015b). DDR2 reference manual for generic users. Technical Report V. 2.1.2, EUROCONTROL.

EUROCONTROL (2018). CODA Digest: All-Causes Delay and Cancellations to Air Transport in Europe – 2017. Technical report, EUROCONTROL.

EUROCONTROL (2019a). CODA Digest: All-Causes Delay and Cancellations to Air Transport in Europe – 2018. Technical report, EUROCONTROL.

EUROCONTROL (2019b). CODA DIGEST Q1 2019. Technical report, EUROCONTROL.

EUROCONTROL (2020). CODA Digest: All-Causes Delay and Cancellations to Air Transport in Europe – 2019. Technical report, EUROCONTROL.

European Commission (2004). Regulation (EC) No 261/2004 of the European Parliament and of the Council of 11 February 2004 establishing common rules on compensation and assistance to passengers in the event of denied boarding and of cancellation or long delay of flights, and repealing Regulation (EEC) No 295/91.

European Commission (2014). Commission implementing regulation (EU) No 716/2014 of 27 June 2014 on the establishment of the Pilot Common Project supporting the implementation of the European Air Traffic Management Master Plan.

Foundation for Intelligent Physical Agents (2012). FIPA ACL Message Structure Specification. Technical report, Foundation for Intelligent Physical Agents.

Gurtner, G., Bongiorno, C., Ducci, M., and Miccichè, S. (2017). An empirically grounded agent based simulator for the air traffic management in the SESAR scenario. *Journal of Air Transport Management*, 59:26–43.

High Level Group on Aviation Research (2011). Flightpath 2050 – Europe's Vision for Aviation. Technical report, European Commission.

Jepessen (2020). Flitedeck advisor – prime and lite. Technical report, Jepseen a Boeing company.

Jones, J., Lovell, D., and Ball, M. (2018). Stochastic optimization models for transferring delay along flight trajectories to reduce fuel usage. *Transportation Science*, 52.

Kistan, T., Gardi, A., Sabatini, R., Ramasamy, S., and Batuwangala, E. (2017). An evolutionary outlook of air traffic flow management techniques. *Progress in Aerospace Sciences*, 88:15–42.

Kluge, U., Paul, A., Ureta, H., and Ploetner, K. O. (2018). Profiling future air transport passengers in Europe. In *Transport Research Arena (TRA) 2018*.

Kravaris, T., Spatharis, C., Blekas, K., Vouros, G. A., and Cordero Garcia, J. M. (2018). Multiagent reinforcement learning methods for resolving demand - capacity imbalances. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–10.

Mazzarisi, P., Zaoli, S., Lillo, F., Delgado, L., and Gurtner, G. (2020). New centrality and causality metrics assessing air traffic network interactions. *Journal of Air Transport Management*, 85:101801.

Molina, M., Carrasco, S., and Martin, J. (2014). Agent-based modeling and simulation for the design of the future european air traffic management system: The experience of cassiopeia. In Corchado, J. M., Bajo, J., Kozlak, J., Pawlewski, P., Molina, J. M., Gaudou, B., Julian, V., Unland, R., Lopes, F., Hallenborg, K., and García Teodoro, P., editors, *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, pages 22–33, Cham. Springer International Publishing.

PACE (2020). Fuel & operational efficiency management. Technical report, PACE a TXT company.

Performance Review Body (2018). PRB Monitoring Report 2018. Technical report, Performance Review Body of the Single European Sky.

Pilon, N. (2016). Improved Flexibility and Equity for Airspace User During Demand-capacity Imbalance: An introduction to the User Driven Prioritisation Process. In *SESAR Innovation Days 2016*.

Pilot3 Consortium (2020). Deliverable D1.1: Technical resouces and problem definition. Technical report, Pilot3 Consortium.

Saint-Andre, P. (2011). Extensible Messaging and Presence Protocol (XMPP): Core – RFC 6120. Technical report, RFC series.

Schultz, M. (2018). Implementation and application of a stochastic aircraft boarding model. *Transportation Research Part C Emerging Technologies*, 90:334–349.

Schultz, M. and Fricke, H. (2011). Managing passenger handling at airport terminals. In *9th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*.

Schultz, M., Kunze, T., and Fricke, H. (2013). Boarding on the critical path of the turnaround. In *Proceedings of the 10th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2013*.

SESAR (2018). PJ07-02 Deliverable D3.1.110, UDPP OSED Interim SPR-INTEROP-Part I. Technical report, SESAR.

SESAR (2020). European ATM Master Plan - Executive View, Ed. 2020. Technical report, SESAR JU.

Stroeve, S. H., Blom, H. A., and Bakker, G. B. (2013). Contrasting safety assessments of a runway incursion scenario: Event sequence analysis versus multi-agent dynamic risk modelling. *Reliability Engineering & System Safety*, 109:133–149.

Velaga, N. R., Rotstein, N. D., Oren, N., Nelson, J. D., Norman, T. J., and Wright, S. (2012). Development of an integrated flexible transport systems platform for rural areas using argumentation theory. *Research in Transportation Business & Management*, 3:62–70. Flexible Transport Services.

Wooldridge, M., Jennings, N. R., and Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3:285–312.

Xu, Y., Dalmau, R., Melgosa, M., Montlaur, A., and Prats, X. (2020). A framework for collaborative air traffic flow management minimizing costs for airspace users: Enabling trajectory options and flexible pre-tactical delay management. *Transportation Research Part B: Methodological*, 134:229–255.

Xu, Y., Dalmau, R., and Prats, X. (2017). Maximizing airborne delay at no extra fuel cost by means of linear holding. *Transportation Research Part C: Emerging Technologies*, 81:137–152.

Zaoli, S., Mazzarisi, P., Lillo, F., Delgado, L., and Gurtner, G. (2020). New centrality and causality metrics assessing air traffic network interactions. *Journal of Air Transport Management*, 85.

Zhang, D., Henry Lau, H., and Yu, C. (2015). A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. *Computers & Industrial Engineering*, 87:436 – 453.