D3.1 Architecture definition

Deliverable 3.1

Domino

Grant:	783206
Call:	H2020-SESAR-2016-2
Topic:	SESAR-ER3-06-2016 ATM Operations, Architecture, Performance and Validation
Consortium coordinator:	University of Westminster
Edition date:	29 June 2018
Edition:	01.00.00

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Luis Delgado / University of Westminster	Project member	28 June 2018
Gérald Gurtner / University of Westminster	Project member	28 June 2018
Damir Valput / Innaxis	Project member	28 June 2018
Nadine Pilon / EUROCONTROL	Project member	28 June 2018
Laurent Guichard / EUROCONTROL	Project member	28 June 2018

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Andrew Cook / University of Westminster	Project coordinator	29 June 2018
Graham Tanner / University of Westminster	Project member	29 June 2018

Approved for submission to the SJU By — Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Andrew Cook / University of Westminster	Project coordinator	29 June 2018

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
N/A		

Document History

Edition	Date	Status	Author	Justification
01.00.00	29 June 2018	Release	Domino Consortium	New document for review by the SJU

Domino

NOVEL TOOLS TO EVALUATE ATM SYSTEMS COUPLING UNDER FUTURE DEPLOYMENT SCENARIOS

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 783206 under European Union's Horizon 2020 research and innovation programme.

Abstract

This deliverable presents the concept of operation of Domino. It includes a description of the systems, subsystems and processes that will be taken into account in the model, as well as the general scope of the model. For each of the mechanisms suggested to be modelled in the project, the deliverable provides a set of possible operational concepts and uptake/scope to be deployed.

The opinions expressed herein reflect the authors' views only. Under no circumstances shall the SESAR Joint Undertaking be responsible for any use that may be made of the information contained herein.

Table of Contents

List of tables

List of figures

Executive summary

The Domino project is developing a set of tools that can be used to analyse the architecture of the air transportation system, and to study certain effects therein, such as delay propagation. Its goal is to provide a methodology for such an analysis and to demonstrate its potential with selected case studies. Domino's toolbox aims to answer challenging questions regarding the potential implementations of new concepts in ATM.

The model's temporal scope is one operational day, encompassing the whole ECAC area. The model will be a holistic, stochastic, agent-based model heavily based on different levels of networks (e.g. passengers and flights). The systems and subsystems modelled will be agents of the model, which include: flights, airport terminals, AMAN, E-AMAN, the network manager, flight operations centres, UDPP processors, etc. Domino's model aims at being a complement to EATMA. It provides an executable implementation for prediction. Compared with EATMA, it includes stochasticity, idiosyncratic agent rules, several instances of the same subsystems, and a holistic view of all the processes at the same time. It is more high-level than EATMA.

Domino's model will be tested on several scenarios, described in D3.2. The scenarios are based on the three core mechanisms:

- 4D trajectory adjustments – including dynamic cost indexing, re-routing, lower flight levels, and wait-for-passenger rules;
- flight prioritisation – including UDPP and other associated, advanced rules;
- flight arrival coordination – including current E-AMAN and other advanced, associated rules.

Furthermore, three levels of implementation are considered per mechanism. Those levels differ by complexity of their implementation, but do not align necessarily with specific temporal developments. Level 0 aims to represent the current operations; Level 1 implies some further capabilities from technological and operational improvements; whilst Level 2 is the most advanced case which might require further research and it is, by definition, more prospective.

1 Introduction

The Domino project is developing a set of tools that can be used to analyse the architecture of the air transportation system, and to study certain effects therein, such as delay propagation. Its goal is to provide a methodology for such an analysis and to demonstrate its potential with selected case studies. The exact scenarios modelled by Domino are described in D3.2. In this deliverable, the scope of the future model is described. This includes its general point of view, the main systems and subsystems modelled, as well as the mechanisms that will be tested.

Domino draws on the general framework of complex sciences to build its methodology. This includes:

- a holistic approach: since several heterogeneous systems interact strongly with each other, they need to be taken into account bottom-up, simultaneously;
- an emergent phenomenon approach: since the sum of individual behaviours is not the behaviour of a sum of individual parts, many instances of subsystems need to be modelled at the same time;
- an agent-based approach: since systems can have arbitrarily complex rules with non-perfect behaviours and information access, the model needs to be able to include complex and potentially stochastic rules for its parts;
- a network approach: since the main goal is to be able to capture ‘tightness’ or ‘slackness’ in the system, a network framework provides natural tools for it.

Domino’s perspective is high-level. Its natural level of description is at the level of flights, airports (potentially subdivided into terminals, AMAN, etc.), the network manager, etc. Compared to EATMA, it is roughly at the same level, with less described interactions (messages etc), and, more importantly, no prescription on how the interactions actually take place. On the contrary of EATMA, it features several instances of the same subsystems and is executable, i.e. in particular it can be used for as a prediction tool in ‘what-if’ scenarios. A more detailed comparison of Domino and EATMA is included in Section 2.

Domino’s model will be tested on several test cases. In Section 3 the different mechanisms that will be used for these test cases are presented. Their concepts are described, explaining their current implementation and their likely ones for the future. We present their implementations in three levels. The first one corresponds roughly to what was done in the baseline (2017), the second level corresponds to an implementation where very little would need to be done to reach it if it was decided to implement it today. Finally, the last level is more prospective and less consensual. It does not represent the most likely scenario for the future, but incorporates ideas from different documents, including the (2015) Master Plan. In Domino we consider three mechanisms:

- 4D trajectory adjustments: including dynamic cost indexing (DCI), lower flight levels, re-routing, and wait-for-passenger rules;
- Flight prioritisation: including the current User-Driven Prioritisation Process (UDPP) and more advanced prioritisation rules;
- Flight arrival coordination: including Extended Arrival MANager (E-AMAN) and more advanced coordination rules.

These mechanisms have been selected for the diversity of the stakeholders involved in their implementation and deployment and for their synergies on the evolution towards 4D trajectories. Deliverable D3.2 will describe more in detail how these mechanisms are combined to produce scenarios executable by the model. We conclude by sketching out future work to be done based on this deliverable.

Throughout this deliverable, some boxes have been added to highlight the feedback that the team received from the consultation performed by WP6 among experts. The detailed description and output of the consultation is available in Deliverable D6.2.

Note: in this deliverable, the terms “tactical” and “execution” are used practically synonymously to designate the off-blocks phase. “Pre-tactical” is used to denote the “planning” period during the day of operations for the flight, including pre-departure (i.e. but not extending as far back as “Day -6”, as defined in the Network Operations Portal). Although there is a lack of uniform clarity on these definitions across the corresponding documentation and sources, these terms will be further refined in D3.2, wherein we will also consider the mapping to the trajectory-based operations context, taking into account the ICAO TBO concept (ICAO, 2015), to the extent that is required for the scoping of the Domino scenarios.

2 Domino's concept of operations

Domino's toolbox aims to answer challenging questions regarding the potential implementations of new concepts in ATM. Domino defines three mechanisms: 4D trajectory adjustment, flight prioritisation and flight arrival coordination, which will be tested at different levels of implementation, conjointly or independently, as defined in D3.2. Domino is primarily interested in demonstrating the changes of interdependencies between the different subsystems resulting from their implementation. This section looks at the main subsystems that will be modelled in Domino, how do they currently work and how they could work differently in given future scenarios. The specification of the subsystems and their relationships is the basis of the different scenarios potentially considered by Domino.

2.1 Geographical and temporal scope

When choosing the temporal or geographical scope, one has to ask the question: "How much does what is *not* included in the scope, influence what *is* included?", or in other words: "Is the whole bigger than the sum of the parts?" Domino aims to capture some effects which arise only when a high number of systems interact with each other. This is in contrast with the usual reductionist approach followed by enterprise architecture tools, like the one used in EATMA and its back-end MEGA. A typical example outside the ATM field is of a system modelling a traffic jam, where the analysis of a single car/driver system cannot predict the appearance of a traffic jam. The congestion is a product of a bigger system composed of many cars, which individually have no rules implying the existence of the jam.

Following these lines, Domino takes a holistic approach to capture these effects. This includes the necessity to have a range of system types (flight, AMAN, etc.), but also to have multiple instances of each one. The instances have interactions with other instances in the same system, as well as with instances of other systems. More detailed examples are given below, such as the interaction of flights with their operations centre.

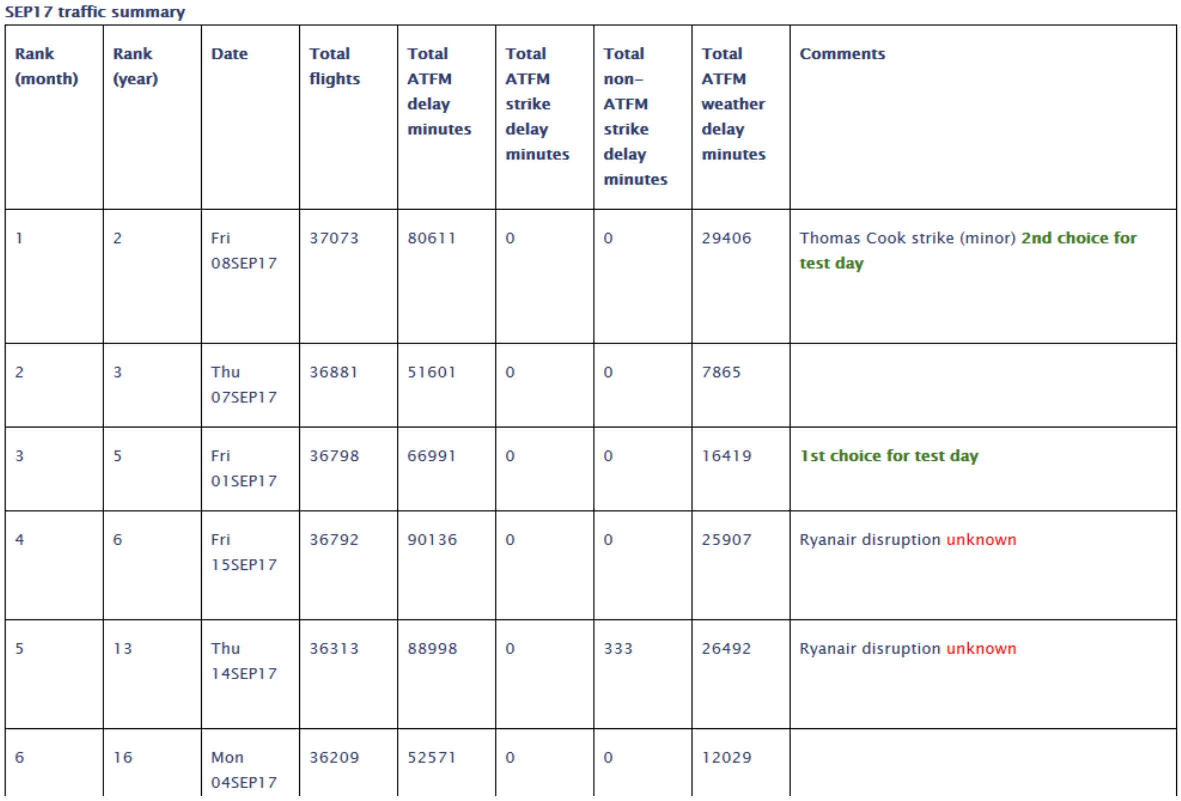
As a consequence of the need of these iterations between elements, the geographical scope has to be wide. Domino will use the full ECAC space for obtaining the results of the final model. More specifically, we will take into account all the flights crossing the ECAC space at some point of their journey. For testing purposes, the team might use some smaller geographical areas or subsets of flights. Some good candidates are UK, Italy, France, and Spain FIRs, since they are big enough for interesting effects and with a structure well known by different members of the consortium. Some flights might be filtered out from the historical sets of flights, since in Domino we are focusing on commercial passenger aviation (scheduled and non-scheduled operations). Therefore, flights such as very light aircraft, VFR or military flights will not be considered. Other flights, such as cargo flights, will be maintained in the dataset as they have an impact on capacity and demand imbalances.

For the temporal scope, Domino will focus on the operations of a typical day in 2017, to which the different mechanisms will be added to create the case studies of interest. Contrary to the geographical scope, the temporal scope does not need to be very wide, because the different days of operations are largely independent from each other (for instance delays do not tend to propagate from one day to the next).

In order to compare the consequences of the implementation of the mechanisms with a known situation, Domino selected a historical day of operation. The team analysed different days in September 2017 (for which the consortium already had most of the data). The days were assessed based on:

- the traffic on that day;
- the ATFM regulations;
- the various disturbances caused by strikes etc.;
- their proximity to holidays, summer season, etc.

The aim was to select a 'typical' busy, i.e., a day which can be reasonably considered as a good base to represent a challenging day from a management point of view, but without massive random disturbances. Figure 1 shows the part of the table built to help the decision making.



SEP17 traffic summary

Rank (month)	Rank (year)	Date	Total flights	Total ATFM delay minutes	Total ATFM strike delay minutes	Total non–ATFM strike delay minutes	Total ATFM weather delay minutes	Comments
1	2	Fri 08SEP17	37073	80611	0	0	29406	Thomas Cook strike (minor) **2nd choice for test day**
2	3	Thu 07SEP17	36881	51601	0	0	7865	
3	5	Fri 01SEP17	36798	66991	0	0	16419	**1st choice for test day**
4	6	Fri 15SEP17	36792	90136	0	0	25907	Ryanair disruption unknown
5	13	Thu 14SEP17	36313	88998	0	333	26492	Ryanair disruption unknown
6	16	Mon 04SEP17	36209	52571	0	0	12029	

Figure 1: Ranking of the different days of operations

Using this method, the team has selected two days:

- 1st choice: Friday 01 September 2017 – ranked #3 (traffic-wise) in September 2017; ranked #5 in 2017; total ATFM delay quite high (but lower than 08 September 2017).

- 2nd choice: Friday 08 September 2017 – ranked #1 (traffic-wise) in September 2017; ranked #2 in 2017; total ATFM.

The first choice is the main day to be modelled in Domino. It has almost no strikes and a high traffic volume. The second day has been selected in case the team needs another one to simulate, for instance to check that the results obtained are not too specific to the day of operation, or if the first choice happens to have any issue unforeseen at the present time.

Note that Domino toolbox is a 'what-if' scenario generator, i.e., it does not try to predict the future state of the air transportation system. Instead, it aims to predict what would have happened if on the day of operation (01 September 2017) some mechanisms were implemented. For the same reason, Domino does not evaluate the cost of the implementation of the mechanisms. Whether that is done through incentives, subsidy or another mechanism does not impact the results of the implementation itself. This does not preclude the use of the Domino methodology and toolbox to analyse scenarios set in the future.

2.2 Systems and subsystems

The definition of the system and subsystems is driven by the level of details required in the model, itself driven by the objectives of the models. A *system* can be abstracted as an entity on which other systems act, and which returns a response based on these inputs. A system can be fully deterministic, i.e., the response is unambiguously specified by the inputs, producing the same output every time the same combination of inputs is presented to it, or stochastic, i.e., the response can be different for the same inputs. Systems are telescopic: they are themselves composed of subsystems, and participate in the response of a bigger system. The modelling exercise thus starts with a specification of the individual subsystems.

Domino is interested in macro-effects arising in the air transportation system during the pre-tactical and tactical phases arising from changes of management rules derived from the implementation of mechanisms. In the air transport system, the main entity on which other subsystems act is the aircraft. From a physical point of view, almost all of the actions performed on an aircraft come from the pilot, as least during the flight. The pilot receives different types of information – notably from the flight operations centre and the air traffic controller, but also from instrument readings, etc. – and acts on the controls to change how the aircraft behaves. The aircraft is a system in itself, composed of many different subsystems, but at a higher level, it follows the will of the pilot, except in case of a failure. As a result, Domino considers them as a single entity from a system point of view, that we call a 'flight'.

Insights from consultation

It is still an open question whether or not the flight should be a pro-active agent, with a dedicated objective function, or a simple 'processor' guided mechanically by other agents. The feedback from the consultation is also divided on the subject, even if the balance seems to be in favour of a passive flight, but at least points out that in the current situation, it is very dependent on the airline culture. The model will thus likely include the possibility for the flight to have a more complex objective function, but we will first start with a simple entity.

The flight interacts with several other systems. First, it has frequent exchanges with a system that captures the airline operations centre behaviour. Broadly speaking, the flight operations centre (FOC) supplies information and directives to the flights regarding their wider environment. For instance, it provides information on other delayed flights arriving at the same airport, which can be fed into the cost index system. The use of dynamic cost indexing (DCI), i.e., adjusting the cost index based on updated delay and cost information, can be considered as a subsystem which can be distributed, i.e., included in the flight (in which case, the FOC only sends the relevant information to the flights) or centralised at the operation centre (in which case, the operations centre sends directives to the flights). The difference can be crucial, as a flight-based DCI system would take sub-optimal decisions compared to a centralised system which can run a global optimiser and then send its directives to all the concerned flights. The toy model will provide the team with insights as to which option should be preferred.

The flight is also in relationship with departure and arrival managers. These managers are considered to be irreducible subsystems in Domino, i.e., not differentiating between the automated and the human elements. Exchanges of messages and directives take place between them and the flight. Moreover, the managers interact with another subsystem: the runway. The runway interacts with the flights by accepting them and releasing them, thus creating a measure of congestion which is used by the arrival and departure managers to create queues. The limitation of the runway leads to the need of a flight arrival prioritisation that could be done by the extended arrival manager (E-AMAN). The E-AMAN will interact with several flights at the same time, perform certain optimisation procedures, and send directives to the flights. It is also foreseen that the E-AMAN will likely be a distinct entity from the arrival manager itself, the latter being only in charge of the queue at the final approach, while the E-AMAN will be in charge of planning activities. These interactions will be clarified in D4.1 (Initial model design).

Apart from the runway, other systems will be included to model the airport functioning. One system, that we call 'terminal', takes care of the connecting passengers and possibly also of the passengers coming to the airport. This means that the terminal provides some measures of the time required and expended on connecting between flights. Since for computational reasons including each passenger as a distinct system would be too demanding, the terminal interacts directly with the landing flight(s) and the departing ones by moving the passenger flows between them. Departing flights receive the connecting times from the terminal, which will be modelled by a stochastic process. Another subsystem which could be included at the airport is the ground movements, which would provide stochastic taxi-in and taxi-out operation times.

Another relevant system that the models will include is the Network Manager (NM). Indeed, the flights are dependent on the NM to get the ATFM slots if required. Flights interact with the NM by providing a flight plan (including departure time, route and possibly a trajectory) and requesting a slot. The NM takes into account the traffic load on different sectors, and if a regulation is in place, allocates the required slots.

The last important system taken into account in the simulation is the flight prioritisation processor. Depending on how the mechanism is implemented in a given scenario, there could be one instance of the system per airline, or a centralised entity. It directly interacts with the Network Manager when flights are swapped by the airspace users. The processor will probably also interact with different agents in different scenarios. For instance, the E-AMAN system(s) could interact with the flight prioritisation one(s) to better build the arrival queue. It could also interact with the airports to check airport slots availability before taking actions on the flights.

Insights from consultation

Some answers in the consultation pointed out that the ATCOs should be more present in the model. In the previous description, they are just considered as creating *ad hoc* distributions of delay (positive and negative) for the en-route phase. It is out of the scope of Domino to consider detailed controllers able to separate planes. However, Domino can include some high-level agents (for instance at the NAS level) computing their own dynamic load and modifying the probability for the flights to have a direct for instance, or to have their changes of trajectories such as speed variations cleared.

Similarly, systems such as CDM have been suggested as relevant for the model. For now, Domino will consider CDM within the flight prioritisation and arrival coordinator but assume it is implemented as A-CDM at the airports and not considered as an explicit system in the model.

2.3 Summary of foreseen systems’ behaviours

To provide a summary of how the system will be modelled, we show a schematic view in Figure 2 of the main elements and we present in Table 1 a list of possible interactions foreseen so far. This schema will be slightly different in various scenarios, and, in any case, it is subject to minor changes (see the ‘Open questions’ in Section 4.2, for instance). Note that an open question is whether or not AMAN should be integrated with E-AMAN, or with the runway. Those architecture specifics will be refined in D4.1, when the model is described in more detail.

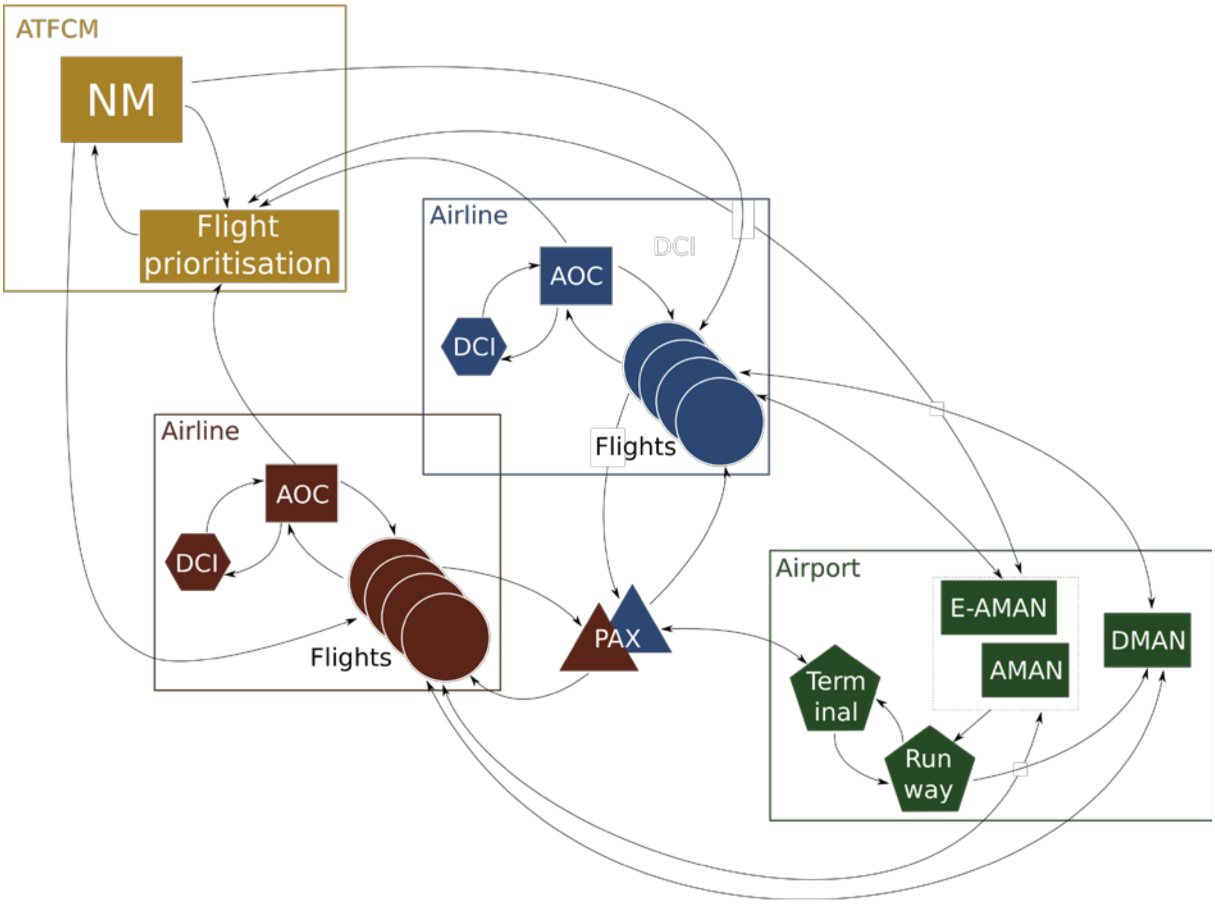


Figure 2: main systems to be considered in Domino and their relationships

Table 1. Main interactions between subsystems

Subsystem	Possible behaviours/interactions
Flights	• Before departure: o Gets information on delay connecting flights (from FOC), possibly waits for passengers o Chooses cost index o Requests new slots from DMAN and NM if needed • During flight: o If dynamic cost indexing is implemented on-board: gets information on delay connecting flights, etc.; changes CI o Requests priority in E-AMAN queue (if E-AMAN allows these interactions) • After arrival: o Handed over to runway o Pax disembark o Aircraft prepared for the next flight
FOC	• Gets information from NM, AMAN, and DMAN on available slots planned departure • Communicates slot requests to NM • Communicates prioritisation/slot swapping/protection to Flight Prioritisation • Computes best cost index for flights (independently or jointly) • Computes requested 4D trajectory, which might be modified if the flight gets delayed due to airspace congestion (e.g., re-routing or lower flight levels) • Relocates passengers to flights if they missed their connections
NM	• Gets information from the FOCs on slots request • Uses first-planned first-served (FPFS) to compute slot allocation (respecting capacity constraints) • Interacts with Flight Prioritisation processor to get swapped slots • Asks DMAN and AMAN at airports for slot constraints
E-AMAN	• Gets information on flights entering its time horizon • Gets user preferences from Flight Prioritisation or FOC if implemented • Builds and updates queue, • Sends entry target times to flights • Hands over flights to AMAN
AMAN	• Gets flights from E-AMAN • Enforces time/space separation • Hands over flights to runway
DMAN	• Manages slot request changes from flights. • Interacts with E-AMAN and AMAN to generate required departing slots • Hands flights over to runway

Subsystem	Possible behaviours/interactions
Runway	• Gets flights from DMAN and AMAN • Processes flights with time constraints between them (probably without interaction between departing and arriving flights) • Hands over flights to terminal
Terminal	• Gets flights from runway • Processes connecting passengers • Potentially adds non-connecting departing passenger delays
Pax	• Passive system(s), handed over from flight to flight

2.4 Network point of view

Domino adopts a point of view strongly inspired by complexity science, from which the holistic approach is derived. A typical object used in this field to abstract the relationships between entities is a network. The main objective of the model in Domino is to estimate the degree of 'tightness' or 'slackness' that exists in the system under different conditions. A network perspective is particularly well suited to do this.

Specific metrics will be used and developed during the project, in particular in WP5, to precisely measure the tightness. Here we highlight the main issues related to this concept. The degree of tightness is indeed qualitatively linked to several related ideas:

- the potential long-range effects of a change in the system (a.k.a 'cascade effect') e.g. the change of departure of a flight changing the operations of other flights, themselves triggering further changes;
- the controllability of the system, i.e. the degree to which one change brings the system to a given functioning point;
- the stability of the system, i.e. how much one can disturb a part without having a total collapse;
- the resilience of the system, which includes its stability but also the possibility to bring it back to a former working point;
- the degree of flexibility given to the operating agents;
- the degree of predictability of the system overall.

Many factors need to be taken into account to tackle all these issues, including the agents' behaviours and the 'rules of the game', i.e. the different regulations in place. To give a better idea of the kind of practical issues that the model will be able to tackle, here is a non-exhaustive list of how tightness could change due to different factors:

- smaller buffers for flights: any delay is propagated to the next flight;
- increased traffic density: more conflicts increase the interaction between one flight behaviour and the others;

- increased time horizon for prioritisation: in a highly concurrent world, increased time horizon means more flights to prioritise at the same time in different places;
- increased predictability: better predictions can translate into more 'gaming effects', where airlines for instance take more decisions based on the other airlines, which leads to different effects like 'herding' etc.;
- increased flexibility: leaving more latitude to the agents in the system can lead to more interactions among them, in the same way than the previous point;
- increased collaborative decisions: for the same reasons.

In summary, tightness can be understood mainly in three interrelated domains and their corresponding efficiencies:

- time: how close are flights (or other agents, or processes) to each other?
- space: how close are the flights (or other agents, or processes) to each other?
- relationship: how much does one decision depends on another?

Domino will look at these three dimensions at the same time.

2.5 Relationship between Domino and EATMA

One of the objectives of Domino is to develop a tool and a methodology that could be used in conjunction with other existing models of the ATM system. EATMA is the most complete and the most important of them, as it acts as a central repository for SESAR.

The European ATM Enterprise Architecture (EATMA) is an architectural model of the processes taking place in the ATM system. It is designed to describe the current situation, but also to assess the future changes of procedures envisaged by the different SESAR operational improvements. The model is a finely detailed description of the type of exchanges, messages, and decisions that the different actors of the ATM system take in different situations. The model is a logical one, with (typically) Boolean decision nodes leading to different decision branches.

Interesting for the Domino project is that EATMA is organised mainly around nodes, activities, and information exchanges. Activities like "Prepare DCB solution" represent logical processes, specified independently of how the processes are actually carried out. The activities exchange messages through three types of objects:

- "Information exchange" objects, e.g. "Acknowledge TOBT, raise alert to relevant stakeholders", which represent the fact that two Activities exchange something.
- "Information element" objects, e.g. "4D Trajectory Situation Awareness", which represent the (meta-)content of the message.
- "Information entities" objects, e.g. "Airport Demand-Capacity Balancing", which broadly represent the type of an operational item.

Finally, nodes represent a logical entity which performs an activity. For instance, 'ATCO' is a node, which represents the entity responsible for carrying out activities such as 'Adapt sector configurations'. Nodes and activities have also hierarchical relationships with other nodes, to represent respectively the level at which each decision is taken, and the task breakdown of any activity. All the above elements are specified independently from their physical implementations. A more technical aspect is present in EATMA when dealing with the implementation itself. Note that other views of EATMA are also available, in particular the 'Performance View' to monitor the advances of SESAR, a service view, and planning view, which is more oriented towards the future SESAR solutions.

EATMA is partially available through the eATM portal, through static images related to the different elements (see for instance Figure 3). However, the 'real' model, more up to date and more interactive, is stored in a MEGA repository which is accessible only on demand. Domino considers the information present in the eATM portal to be the appropriate level of detail and does not require access to the MEGA repository.

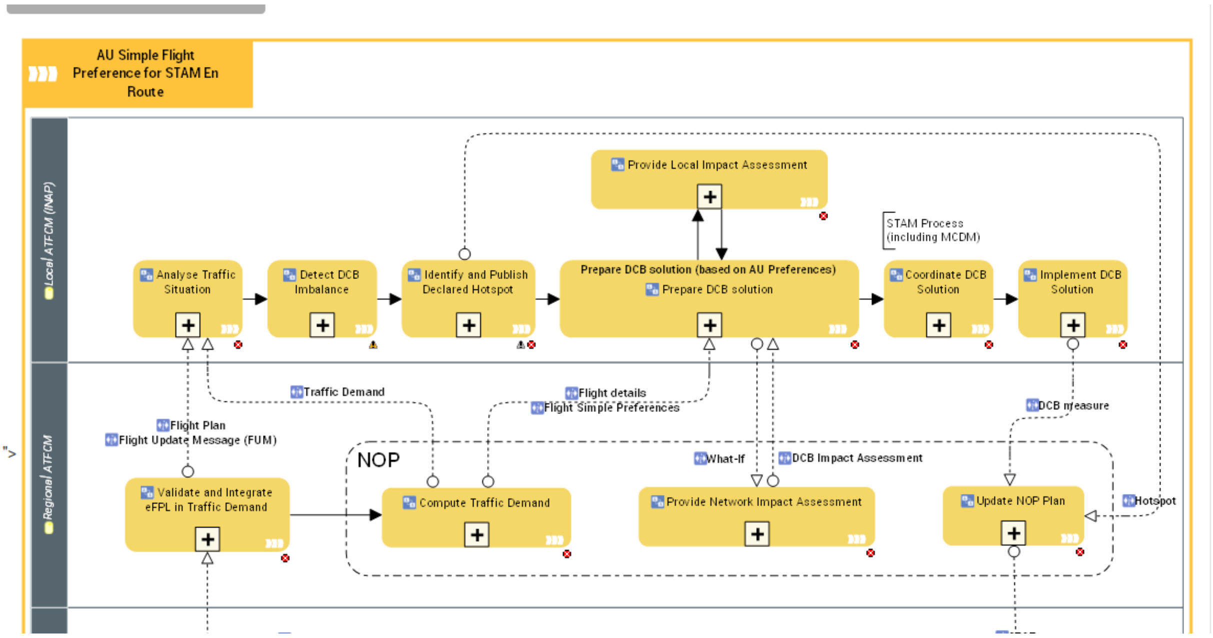

AU Simple Flight Preference for STAM En Route
Local ATFCM (INAP)
Regional ATFCM
Provide Local Impact Assessment
STAM Process (including MCDM)
Analyse Traffic Situation
Detect DCB Imbalance
Identify and Publish Declared Hotspot
Prepare DCB solution (based on AU Preferences)
Prepare DCB solution
Coordinate DCB Solution
Implement DCB Solution
Traffic Demand
Flight details
Flight Simple Preferences
DCB measure
Flight Plan
Flight Update Message (FUM)
NOP
What-If
DCB Impact Assessment
Hotspot
Validate and Integrate eFPL in Traffic Demand
Compute Traffic Demand
Provide Network Impact Assessment
Update NOP Plan

Figure 3: Example of process schema from EATMA

With respect to the model developed by Domino, EATMA has a few similarities and differences, summarised in Table 2. Note that their objectives are different. EATMA aims to be a repository for SESAR knowledge with a very fine level of detail and the possibility to link the qualitative changes envisioned in the Master Plan to actual operational changes. EATMA represents a set of logical models to be explored by humans but do not, as far as we know, have a numerical implementation leading to actual prediction. On the contrary, Domino's method aims to quantitatively estimate the impact of new operational improvements with a numerical model. In particular, Domino features an integrated, holistic model including behavioural rules.

Table 2: Comparison between EATMA and Domino

EATMA	Domino
Isolated logical models	Holistic model of the ECAC space
Deterministic	Stochastic
Does not include behavioural rules	Includes behavioural rules
Few number of instances of systems (e.g. one airport): cannot capture systemic architecture issues	Large number of instances: can capture systemic, emergent issues
Models are focused on detailed processes	Model captures high-level phenomena
No numerical implementation for predictions	Model implemented for quantitative predictions
Main objects are processes	Main objects are agents
Processes interact through messages via pre-determined rules	Agents interact between them based on idiosyncratic objectives functions

3 Mechanisms

3.1 Summary of mechanisms

Domino considers three mechanisms for their implementation and assessment, as summarised in Table 3. These mechanisms tackle different operational improvements which are aligned with a development of trajectory-based operations (TBO). (This relationship with TBO will be discussed further in D3.2, as flagged earlier.) The mechanisms have also been selected as they are developed by different stakeholders and implemented in different operational concepts. The three mechanisms will be applied on the historical traffic data considered and described in D2.1. Their names have been selected to reflect their general approach to different technical solutions, and to avoid confusion with existing SESAR concepts (e.g. “this does not accurately reflect UDPP plans”).

Three levels of implementation are considered per mechanism. Those levels differ by complexity of their implementation, but do not align necessarily with specific temporal developments. Level 0 aims to represent the current operations; Level 1 implies some further capabilities from technological and operational improvements; whilst Level 2 is the most advanced case which might require further research and it is, by definition, more prospective.

For each mechanism, different implementation scopes are also considered covering both a geographical and uptake approach.

The mechanisms can be implemented in isolation or conjointly. In the latter case, specific interactions on their implementation should be considered.

The following table summarises the characteristics of the mechanisms considered in Domino.

Table 3: Summary of mechanisms

Mechanism	Rationale	Implementing stakeholder	Implementation	Scope
4D trajectory adjustment	Model delay management strategies from the airline operations point of view. This includes trajectory management (e.g., re-routing/lower flight levels, dynamic cost indexing) and hub management (e.g., waiting for connecting passengers)	Airspace users	Level 0: • Basic delay recovery rules linked to amount of delay incurred; cost index computed before departure; limited waiting for passengers; basic re-routing and level changes used. Level 1: • Dynamic cost indexing for flights, with a simplified in-flight delay and cost estimation based on heuristics and general cost of delay rules • Wait-for-passenger rules used more widely but with limited rules only • Estimation of possibility of re-routing and level changes with respect to dynamic cost indexing Level 2: • Advanced estimation of expected costs due to connecting delayed flights. • Optimisation of actions considering the three available techniques (dynamic cost indexing, trajectory modification and wait-for-passenger rules).	Airspace users implementing it: • FSC • LCC • CHT Operations types using the mechanism: • To-from hub • Flight with a high load factor • Flight with a high number of connecting passengers • All operations

Mechanism	Rationale	Implementing stakeholder	Implementation	Scope
Flight prioritisation	Collaborative decision making considering airlines prioritisation of their flights to minimise their expected cost	Airspace users Network manager Airports	Level 0: • No pre-departure prioritisation of flights by the AU, use of first-planned first-served (FPFS) when assigning ATFM delay; no slot swapping takes place Level 1: • Pre-departure prioritisation of flights, allowing AUs to reorder several flights in the same constraint among their own slots when they are delayed (UDPP principles); includes Enhanced Slot Swapping (SESAR1) Level 2: • As per Level 1, plus a 'credit' system so that airlines can prioritise flights at a given airport using credits previously earned at the same airport.	'Hotspots' at major airports when congested 'Hotspots' en-route (airspace regulations)

Mechanism	Rationale	Implementing stakeholder	Implementation	Scope
Flight arrival coordination	Tactical management of arrival at airports doing sequencing and merging with planning capabilities.	Airspace users Airport operators ANSP	Level 0: • Current principles applied on E-AMAN systems; the flight arrival coordinator tries to minimise the amount of holding delay that will be carried out at the TMA by providing speed advisories for flights during their en-route phase; the AMAN in the TMA is focused on the maximisation of throughput at the runway; no information from the airlines is taken into account when applying this mechanism. Level 1: • The flight arrival coordinator tries to minimise the expected reactionary ('knock-on') delay, considering information from the airport on the expected turnaround of flights. Level 2: • The airspace user provides the prioritisation of their flights when entering the arrival coordinator's planning horizon or the priorities are defined pre-tactically (i.e. flight prioritisation (e.g. UDPP) choices are relayed to the flight arrival coordinator and taken into account when creating the landing sequence).	Current implementation at 13 airports At airports reported in PCP as using E-AMAN

The following sections describe each of the mechanisms in more detail and the impact of their simultaneous implementation.

3.2 4D trajectory adjustments

3.2.1 Description

During the daily operations, airlines face the need of managing delay incurred by their flights. When a flight is delayed, different options are considered by the airspace users depending, among other factors, on the reason for the delay. For example, if a flight has been assigned some ATFM delay due to airspace congestion, the airspace user can consider the possibility of submitting a new flight plan avoiding the regulated area by re-routing or lower flight levels (if possible). There is always the possibility to increase the flight cost index (on-board fuel permitting) to recover part of the delay.

Note that these decisions are performed trying to minimise the expected (perceived) costs which should consider among other things the cost of extra fuel, the cost of passenger delay (e.g., due to missed connections, passenger provision schemes (hard costs) or impact of delay on airline reputation (soft costs)) taking into account potential downstream effects (e.g., propagation of delay or reaching a curfew at the end of the day). Crew management is out of scope of Domino, and therefore only flight and reactionary delay and passenger-associated delay costs are considered.

The different strategies can be implemented tactically (during the flight) or pre-tactically (a few hours before the flight) and they can be decentralised (each flight decides the best approach to manage their delay) or centralised by each airline operations centre (AOC). The AOC has a view of the situation of all their flights and their statuses, along with their passengers and their connectivities. Therefore, they can try to optimise the strategy of each of their flights. Besides delay recovery, airlines might use active wait-for-passenger rules at hubs to minimise passenger connection disruptions.

The different options considered in Domino to manage delay from the airline/flight perspective include:

- dynamic cost indexing: changing the cost index of flight to recover delay by means of speed and trajectory variations (maintaining the route);
- trajectory modification to avoid ATFM delay: re-routing, increasing the total flight distance, or lower flight levels using sub-optimal cruising altitudes, but trading the extra fuel and potential flying time required for en-route ATFM delay;
- wait-for-passenger rules: actively delaying outbound flights to wait for delayed inbound passengers so that they don't miss their connections.

In order to select the option which gives a lower expected cost, the airline will have to assess the downstream effects of that flight through the day (reactionary delay of the aircraft) plus through the rest of their network (i.e., with connecting passengers). This can be done with different heuristics or with a more advanced computation.

3.2.1.1 Dynamic cost indexing

The cost index (CI) is a number used in the Flight Management System (FMS) to optimise the aircraft's trajectory. It gives the ratio between the time (delay) and fuel costs. The generic equation to express cost index is:

$$\mathrm{CI} = \frac{\text{time costs [\$/hr]}}{\text{fuel costs [cents/hr]}}$$

Observing the formula, it follows that a lower CI results in a flight trajectory that minimises the fuel consumption, giving an overall slower flight. In contrast, a higher CI reduces the flight time at the expense of using more fuel, in order to decrease other operational costs (and consequently, increasing the fuel cost for that flight). Basically, the general objective is to minimise the cost of the flight.

Time-related costs might include various factors, ranging over flight crew wages with an hourly cost associated to them, marginal depreciation or aircraft leasing costs for extra flying per hour, hourly maintenance costs, and similar. In addition to these costs, additional ones can emerge from passenger dissatisfaction (soft costs), missed connections, re-booking or accommodation of passengers (in case of overnight stays), compensation policies, etc. These costs are airline specific and it is important to be able have good estimates in order to integrate them adequately into the delay managing schemes. It is worth noting that airlines can have different policies for different passengers, flagging certain passengers as 'premium', in which case, a different policy may apply. All of the above mentioned factors play an important role in building a tool able to continuously update the estimate of the current cost of the unrecovered delay.

In the denominator of the CI ratio is the total estimated fuel cost for the observed flight. The calculation of this cost is certainly much more straightforward than the calculation of time related costs; however, certain factors such as variable fuel prices can make this calculation more complicated.

By balancing time and fuel related costs, the FMS can select the best climb, cruise, and descent speeds, i.e. so-called ECON speed which minimises the total cost.

The cost index could be adjusted by the aircraft crew as their own initiative or as requested by the AOC dynamically considering the situation of the flight and/or the flights of the airline. This adjustment could be done before take-off or during the flight. Prior to take off, the airline almost always has incomplete data, and consequently, a higher degree of uncertainty to deal with. Allowing the cost index to be dynamically updated during the flight (by incorporating static flight data with the dynamic passenger data required for the calculation of the dynamic CI, and sending information updates on the CI to the aircraft crew) has the potential to save unnecessary fuel burn, recover significant delay costs for airlines and/or result in a positive environmental impact by reducing CO_2 and NO_x emissions. One should note that the decision about updating the CI is more impactful if made earlier in the flight.

3.2.1.2 Trajectory modification

When ATFM delay is assigned to a flight, the airline might decide to resubmit a flight plan to avoid the congestion which is imposing that delay. There are only some situations when this is possible, notably the limiting regulation must be in the airspace (it is not possible to avoid regulation at an arrival airport by changing the trajectory), and the airspaces around the congested airspace should not be congested in their turn. The whole route could be modified by re-routing around the congested region leading to a longer route or a different (lower) flight level can be selected during the congested region to avoid entering it. As noted, reduced flight levels lead to extra fuel consumption, while re-routing will require a longer flight distance to cover.

This trajectory modification can in turn be applied jointly with the dynamic cost indexing. For example, flying a longer route but selecting a higher cost index to minimise the total incurred delay. This will however, lead to even higher fuel costs that will need to be traded against the expected cost of delay.

3.2.1.3 Wait-for-passenger rules

In some cases, airlines might decide to actively delay outbound flights from their hub to wait for inbound connecting passengers. This option is not frequently used by airlines as it impacts the on-time performance of the outbound flights which might not be delayed and, in some cases, waiting for passengers might lead to an outbound flight being regulated. However, when the optimal solution to minimise the cost of delay is sought then this might be a relevant strategy. This is particularly important for the last flights of the day, where if passengers do not make their connection, they need to be rebooked on next day flights, leading to significant hard costs for the airline.

As with the trajectory modification, wait-for-passenger rules could be implemented with dynamic cost indexing. In this manner, an outbound flight might wait for inbound passengers and then select a higher cost index to recover part of the delay.

3.2.2 Operational concept levels

Given the previous discussion, Domino defines the following levels of implementation for the 4D trajectory adjustment mechanism.

3.2.2.1 Level 0

Level 0 implementation tries to replicate current practices done by airspace users to manage their delayed flights. In general, they will consist of a set of basic delay recovery rules, some use of trajectory modification and infrequent use of wait-for-passenger rules.

3.2.2.2 Level 1

Level 1 implementation considers the use of dynamic cost indexing for flights, but with a simplified downstream delay, and cost estimation based on heuristics and general cost of delay rules. Wait-for-passenger rules could be used more widely but also with a set of limiting rules.

3.2.2.3 Level 2

Finally, Level 2 implementation considers an advanced estimation of expected costs due to delay flights and an optimisation of actions considering the three available techniques (DCI, trajectory modification and wait-for-passenger rules).

Insights from consultation

One expert has reminded us that the airlines usually have very comfortable buffers, particularly on long-haul flights, which means that they are able to recover some delay in such cases. In fact, buffers introduce some slack in the system by allowing elasticity in the arrival times, as explained in Section 2. As a consequence, we expect that 4D trajectory adjustments are performed more often and have more impact when the buffers are smaller. The same expert also pointed out that the 4D adjustments are mainly short- and medium-term solutions, but that ANSPs work on longer-term solutions whereby for instance the departure times are updated when updates on weather are available.

Other experts mentioned the importance of curfews and crew. Domino cannot take crew factors into account, since it has no data available to calibrate the model. Concerning curfew, there might be a possibility for Domino to get some data on this, but most probably the model will feature a simplified approach with a fixed curfew for every airport in Europe. This will be considered regarding the expected cost of delay at the end of the day.

One expert pointed out that re-routing is a complex operation because of route charges. Since Domino will not feature a geographical model *per se*, we must take care of not introducing a strong bias one way or another (towards smaller route charges or reduced times).

3.2.3 Scope of implementation

All airlines are affected by delay and need to manage it. Therefore, delay management strategies will be applied by all airspace users. However, the complexity of the downstream effects varies across airlines and hence some flights have a higher criticality with respect to the management of their delay. For those reasons, different implementation scopes can be considered considering:

- which airspace users are implementing it:
  - FSC (scheduled, ‘traditional’ companies): these companies usually have high costs and operate hub-based operations,
  - LCC (low-cost): these companies usually have low costs and operate in a more point-to-point environment,
  - CHT (charter),
- in which type of operations the mechanism is used:
  - to/from hub,
  - flight with a high load factor,
  - flight with a high number of connecting passengers,
  - all operations.

Note that from a modelling point of view, it is not necessary to write explicit behaviours in all these cases. Indeed, the decisions taken by the airlines should ideally be the result of the same objective function (maximisation of expected profit) and different initial conditions (i.e. the airline network, the passenger itineraries, etc.).

3.3 Flight prioritisation

3.3.1 Description

In order to maintain safety in the air traffic management (ATM) system, European air traffic flow management (ATFM) imposes airport or en-route capacity-driven delays on certain flights before departure. It is well known that ATFM delay causes operational irregularities with significant costs to the airspace users (AUs), airports and passengers. Profitability in the air transport industry is very sensitive to cost variations (profit margins might be as low as 1-2%), thus AUs would like further flexibility, i.e., the ability of the ATM system to accommodate AUs' changing business priorities, to reduce the cost of delay during irregular operations.

Delay is used today as a key performance indicator (KPI) of ATM capacity (capacity to maintain safety in operations), and thus most of the KPIs steering the ATFM demand and capacity balancing (DCB) function are based on average delay per flight, while DCB targets are strongly oriented towards a 'no-delay paradigm'. As a consequence, in the event of a demand-capacity imbalance (a.k.a., 'hotspot'), the flow management position (FMP) in charge will most likely find a solution that decreases the overall system delay first, and whenever possible also reduce the impact of delay on AUs.

However, the impact of delay on AU's operations, which is highly important information being only known by the AUs, cannot be fully taken into account by DCB. Whereas, for ATFM, all flights are equal, for AUs, every flight is unique, due to:

- passenger service;
- airport/crew/aircraft limitations;
- schedule integrity.

If AU's priorities could be considered during the DCB decision-making processes, it could have a large positive impact on the efficiency and predictability of ATM operations. Airspace users' participation in ATM and airport collaborative processes is therefore essential to minimise the impacts of deteriorated operations on all such stakeholders, thus giving strong arguments for the application of de-centralised decision-making (i.e., a user-driven approach) as the best solution to achieve efficiency in ATFM slot/delay allocation.

SESAR envisioned the development of the User-Driven Prioritisation Process (UDPP) to achieve additional flexibility for AUs to adapt their operations in a more cost efficient manner. The UDPP concept is today under development and new features are being progressively incorporated aiming to fulfil different operational requirements and implementation constraints. Some of these features have been already proposed and validated with different levels of maturity, such as Enhanced Slot Swapping validated in 2015, and deployment starting in May 2017.

In today's operations, a few hours before a potential demand-capacity imbalance is foreseen with a certain level of confidence, the ATFM authority activates a regulation scenario and issues 'ATFM slots', which will apply a tactical time-based separation between flights to ease the safe and smooth management of air traffic flows and sector/airport capacities during tactical and flight execution operations. Those ATFM slots are then allocated to the flights involved in the regulation, thus changing their times of departure with respect to the original slots scheduled for those flights, and thus causing delay on flights as a consequence.

The ATFM slots are not allocated on an arbitrary basis, but rather the process typically follows a transparent set of rules and policies previously agreed and accepted by all the relevant ATM stakeholders, including the AUs. The most common policy used today to allocate delay – when no other more stringent rule or operational policy applies – is the first-planned first-served (FPFS) principle, which sorts the flights by the estimated time of arrival or over (ETA/ETO) present in the filed flight plans, and assigning the slots in such order.

FPFS is widely accepted by AUs because it preserves the original sequence of flights (considered fair), and it is well accepted today in ATFM operations because it minimises the total delay in a hotspot. FPFS policy does not take into account that delay is allocated differently to the flights and that each flight may have different impact of delay in their operational costs.

3.3.2 Cost of delay and prioritisation

Flight prioritisation could reduce the cost of delay to AU by reallocating the slots. AUs are very heterogeneous in their size, form and business strategies, and thus, they often have very different operational needs, in particular regarding the flights subject to ATFM regulations, but in general it has been recognised by AUs that flights often have some tolerance to delay (i.e., margins). Although delay usually has a cost, this cost can be often considered marginal in practice if the delay is not impacting more stringent operational margins. The concept of operations which allows this is UDPP.

Figure 4 shows the cost model that is being developed in the context of UDPP together with the AUs' participation. Each flight has its own particular complex cost structure, only known by the AU. The cost structure of a flight is typically not linear, due to the presence of different milestones and time constraints for each flight, such as crew out-of-hours constraints, maintenance slot requirements (such as a ramp check), passenger missed connection costs, high-yield passenger business retention (soft costs), or a missed airport curfew, etc. If a flight is delayed so that these important milestones or constraints cannot be fulfilled, then a large negative impact on AUs operational costs might be typically the consequence. To mitigate such impacts, the AUs would like whenever possible more flexibility to prioritise their flights to redistribute delay on the basis of the consequences on operations and costs.

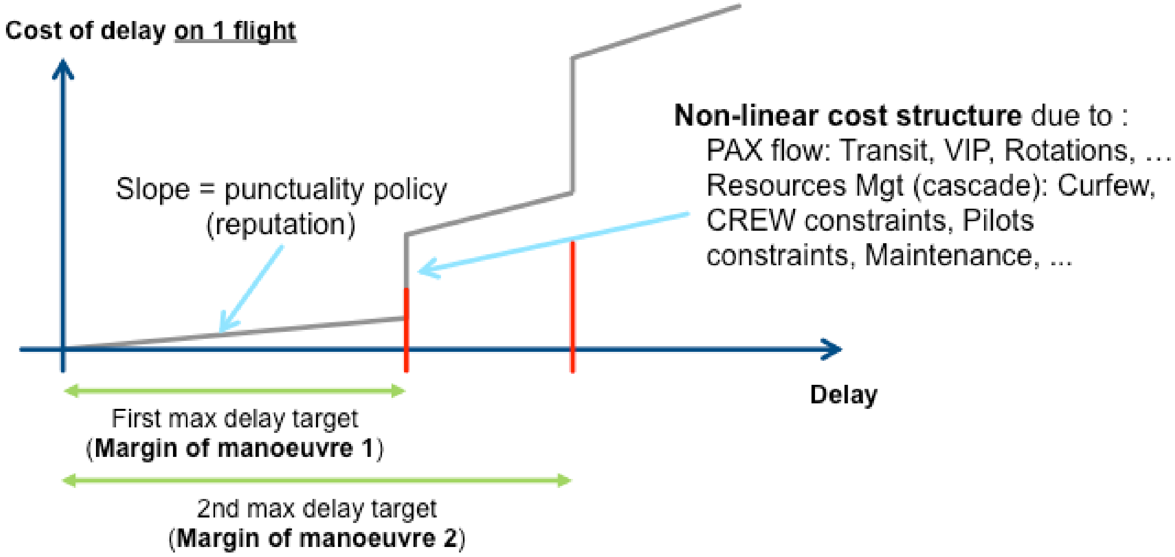

Cost of delay on 1 flight
Slope = punctuality policy (reputation)
Non-linear cost structure due to :
PAX flow: Transit, VIP, Rotations, …
Resources Mgt (cascade): Curfew,
CREW constraints, Pilots
constraints, Maintenance, ...
Delay
First max delay target
(Margin of manoeuvre 1)
2nd max delay target
(Margin of manoeuvre 2)

Figure 4: Typical cost structure model per flight

Figure 5 shows three flights of a same AU that are impacted differently by delay, since each flight has a different position in a sequence as well as different cost structures. Note that each flight has a very different cost structure shape, either in the size of their delay margins and/or in the magnitude of the impact of delay. In the example, flight FL001 has little delay and little impact of delay, FL002 has medium delay but a relatively large impact, and FL003 has the largest delay but relatively small impact in comparison with FL002. It must be pointed out that the impact of delay for a single flight (e.g., FL002) might also include the costs associated with the potential knock-on (reactionary) effect caused by a certain amount of delay allocated to that flight.

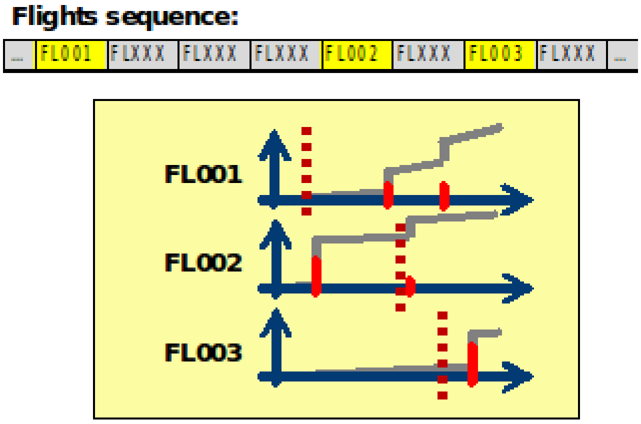


Figure 5: Different cost structures for each flight

In Figure 6 is shown the global cost of delay for the AU of the example taking into consideration its three flights. The initial situation in the baseline sequence (e.g., FPFS sequence) is shown in the left part of the figure. The right part of the figure shows the benefits of giving flexibility to AU to transfer delay between its flights. For instance, by exchanging the positions of FL001 and FL002 (UDPP Slot Swapping) the delay D1 initially allocated to FL001 is transferred to FL002, and delay D2 to FL001. A large cost reduction might be possible for the AU in terms of just by changing that position.

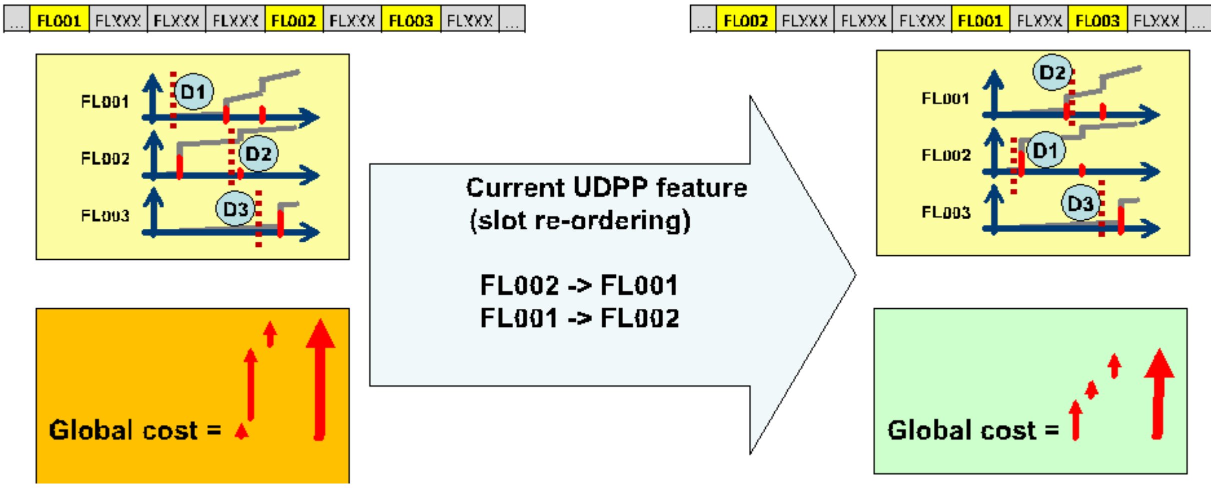


Figure 6: Contribution of flexibility to global cost optimisation

The flight cost curve shown in Figure 5 drives prioritisation decisions for each flight impacted by delay. This curve represents the impact of the delay on the operational cost of the flight. Amongst the many operational factors included, the main ones are linked to passenger flows (transfers, reaccommodation, VIPs, …) and to airports (curfew, missed stand and gates slot). This is where airports' and airlines' interest converge; it is a hidden dependency.

Flight prioritisation allows airspace users to indicate which flights are more relevant in term of cost of delay, helping to minimise the economic impact of delay on their operations.

3.3.3 Operational concept levels

The UDPP is under development and several improvements to the current system are foreseen in the future. Given these facts, Domino will consider the following levels of implementation for the flight prioritisation mechanism.

3.3.3.1 Level 0

In this level of implementation, when there is a mismatch between capacity and demand impacting safety, an ATFM regulation is issued and the slots are assigned to flights following an FPFS protocol. In Level 0, no input of prioritisation from the airspace user will be considered.

3.3.3.2 Level 1

In level 1 implementation, when there is a mismatch between capacity and demand, instead of creating an ATFM regulation, a UDPP measure could be set, allowing airspace users to reorder and protect their affected flights. It is important to note that the AUs can only swap flights from their own company in this level.

The implementation will broadly follow these lines, which represent the most up to date state of UDPP:

1. The Network/Airport 'FMP' detects a hotspot.
2. It decides to let AUs prioritise their flights and triggers UDPP until a cut-off time.
3. The UDPP function continuously receives a snapshot of the Network situation and updates the slot list according to the input given by airspace users. If airspace users don't give inputs, the final situation is the same as if a regulation is put instead of a UDPP measure (corresponding to the level 0 implementation).
4. AUs may reorder/protect their own flights with using a 'what-if' on the snapshot network. They can use a combination of the following methods, based on equity rules: prioritisations of an AU on its flights don't negatively impact other AUs' flights:

   a) FDR: Fleet Delay Reordering

   - ordinal priority values to reorder AU's flights
   - keep the baseline delay (delay same as without UDPP)
   - 'UDPP-suspend' a flight: put it last in a constraint (flight probably then subject to cancellation)

   b) SFP: Selective Flight Protection

   - Protect flights (keep flights on time: allowed only if other flight can be delayed after the protected flights)

   c) margins of manoeuvre

   - time margins can be given to each flight: AU given time constraints to the UDPP function: 'time not before' / 'time not after', and, if necessary, an ordinal priority value to select the order of the flight to manage if no possibility to have all the margins observed
   - the system optimises reordering

5. When satisfied, each AU independently published the new order of its flights, which is sent to the network. It may happen that some flights don't get what was expected because the actual Network situation has changed meanwhile or the airport cannot accommodate the proposal. This AU publication updates the Network situation.
6. After the cut-off time, the AUs can't prioritise any more. The Network/airport FMP use the new Network situation to accommodate: the UDPP solution has taken account of other constraints on prioritised flights in the Network.

3.3.3.3 Level 2

In Level 2 implementation, Level 1 can be applied with added principles where airlines can more freely exchange their flights. This includes two independent features:

- the possibility to exchange flights even when one of them is not regulated, potentially across several days;
- the possibility to exchange slots between airlines.

Instead of single 1-to-1 exchanges, another possibility is to use credits. This allows, for instance, small airlines (or airlines with few flights) in an imbalanced problem to exchange delay (get extra delay) for some flights in order to accumulate credits that can be used in other constraint situation (by exchanging delay again). The degree to which these credits are persistent and exchangeable against real money in a central secondary market, for instance, is an open question. It is out of scope of Domino to implement a full financial market for slots. Moreover, the limited temporal scope (one day of operation) of the model makes such an implementation irrelevant, since long-term strategies obviously play a strong role in such markets. The model will thus take a relatively simple approach, allowing simple trades between most delayed flights, potentially with virtual flights departing on other days. Domino will monitor the typical imbalance in terms of hypothetical credits that each airline gets at the end of the simulated day.

Insights from consultation

Some experts pointed out during the consultation that a centralised market with real money, or other kinds of credits, is definitely a road to explore. Others have pointed out how averse some actors are to any kind of monetisation of the slots. Domino will try to shed some light on the potential benefits and drawbacks of more market-oriented solutions. In any case, several experts have pointed out how much the equity among airlines is important and should be preserved.

3.3.4 Scopes of implementation

The prioritisation of flights could be performed at:

- ‘Hotspots’ at major airports when congested on arrival or departure by creating a UDPP measure;
- ‘Hotspots’ en-route by the creation of a UDPP measure instead of the current airspace regulations, but applicability is lower than at airports due to the diversity of another possible measure: short-term ATFM measures (STAMs).

3.4 Flight arrival coordination

3.4.1 Description

Even if flows are smoothed by the application of ATFM regulations, coordination on arrivals at airports are needed to provide the landing sequencing and the merging of flows. These activities ensure that the runway throughput is maximised while reducing the amount of expensive holding in TMAs. The ATM Master Plan introduces the concept of E-AMAN as a planning tool to carry out these activities. Domino will explore current implementations of flight arrival coordination and the possibilities of including some AU inputs (priorities) into the sequencing optimisation, along with different techniques. The main objective of these planning tools is to smooth the arrival traffic so that the AMAN can create the final landing sequence minimising holding at the TMA.

Generally speaking, extended arrival managers have two horizons of activities: AMAN planning horizon (which can be as large as 500 NM as in the case of Heathrow) and AMAN advisory horizon (set at 350 NM for Heathrow). When a flight enters the planning horizon it is taken into account by the E-AMAN system and, once it reaches the outer bound of the advisory horizon, a speed adjustment is reported to the controller so that the flight can be slowed down to smooth the arrival traffic and start building the arrival sequence. The flights within the scope of the E-AMAN are monitored continuously and, if required, updates on their sequencing are performed. All these are done *via* advisories sent to the ATCOs. Once the flights enter the TMA, the AMAN advises the controllers of the final sequencing to minimise the use of the holding stacks while maintaining a high runway usage.

Flights that depart within the scope of the E-AMAN have a higher uncertainty on their actual need for a slot. Therefore, they are considered during the optimisation process by allocating some runway slots for them that will be fine-tuned once they are airborne and join the rest of the flights in the landing sequencing. The arrival manager generates the gaps required by the departure manager on the landing sequence in mixed-mode operations.

3.4.2 Operational concepts levels

3.4.2.1 Level 0

Level 0 implementation, the flight arrival coordination is performed to minimise the need for a holding stack and moving delay from the TMA to the en-route phase. The arrival manager within the TMA is maximising the arrival throughput.

3.4.2.2 Level 1

At Level 1 the flight arrival coordinator is not only minimising the need for holding at the TMA, it is also considering other operational constraints relayed by the airport to minimise the reactionary delay at the airport. In this approach, airlines are prioritised on their arrival based on what the airport considers is the most efficient.

3.4.2.3 Level 2

Finally, Level 2 consider the introduction of airlines priorities in the sequencing. One of the current limitations of the system is that airlines feel that they have less control on their operations as the arrival manager is deciding for them the landing sequence. With this approach, airlines can prioritise their flights and send that information to the sequencing system. The airlines report the priorities of their flights at the planning horizon and following an approach similar to the flight prioritisation mechanism, these priorities are considered when creating the landing sequence.

3.4.3 Scope of implementation

The scope of implementation of the arrival coordinator varies by location (which airports implement the system) and their planning scope:

- Geographical implementation
  - Current implementation of E-AMAN systems: London Heathrow, London Gatwick, Oslo, Zurich, Rome, Milano, Istanbul, Paris Charles de Gaulle, Paris Orly, Copenhagen, Helsinki, Frankfurt and Munich
  - Full list of airports as considered in the PCP for their implementation of E-AMAN
- Planning scope:
  - 500 NM for LHR as currently implemented, other airports ranging from 180 – 350 NM depending on the demand and location of the airport

Insights from consultation

For Level 2, some experts have mentioned the fact that priorities should be drawn at least from cost minimisation, and ideally from the pre-tactical ones given to the (extended) UDPP system plus some updates given by the airlines during the flight. It seems indeed logical for the airlines to prioritise their flights in general and that this same priority is used throughout the execution phase (even if it can slightly evolve). This idea will be explored in one of the scenarios (see D3.2), whereby when the flight prioritisation mechanism is implemented, both will interact and thus be in synergy. There might be some unforeseen issues with joint implementations, however. In order to do this, Domino will need to align the concept of 'priority' in both mechanisms. Also, in E-AMAN the airlines have to prioritise all their flights, whereas with UDPP, they do not have to put any priority if there is no regulation. The model will need to be able to derive an overall priority for each flight, which is not a trivial task.

It is worth noting that one expert has also mentioned a simple 'stick to plan' objective for the prioritisation. In other words, the AUs is charged with asking for the best flight plan for itself before departure and the E-AMAN simply tries to stick to it. Another expert has also pointed out that the focus should be more on the ground process rather than on the sequence of arrival to avoid reactionary delay, for instance.

Several experts have noted that focusing on the reactionary delay at Level 1 should be enough. One of them noted that: "This may work out of hub only, i.e., in the hub, carrier may use tail swap to minimise disruption and allocate 'on-time' departures for the right flights".

4 Next steps and look ahead

4.1 Next steps

This deliverable detailed what the architecture of Domino encompasses. In particular, it explained in detail the geographical, temporal, and operational scope of Domino.

The main systems and subsystems to be included in Domino's model have been listed, chosen for the level of granularity needed to achieve the objectives of the project.

The three mechanisms that will be used in Domino as case studies to demonstrate the validity of its methodology have been described. For each of them, different variations have been selected, which will be used to build the different scenarios run by the final model.

The scenarios are described in D3.2, together with the parameters associated with them.

The model itself will be described in more detail in D4.1, which will present the final agents/systems considered and their main relationships.

As described earlier, a consultation has been carried out for this deliverable and D3.2, and its results have been integrated into both deliverables.

(D6.2 will describe the consultation process itself and show questions were asked of the experts.)

4.2 Open questions

This section is a simple compilation of the open questions so far concerning the architecture of Domino's model and its implementation. These questions are more oriented towards the implementation itself and will be subsequently clarified when the architecture of the solution (the different agents and their interactions) will be refined through internal review (of, for example, the expected predictive power of the model, specific responsibilities of each subsystem, knowledge drawn from the toy-model), and presented in D4.1.

- The extent of integration of the four systems: E-AMAN, AMAN, runway management, and DMAN, with these options:
    - 4-system option (in which E-AMAN has AMAN responsibilities, i.e. carries out final sequencing)
    - 3-system option (E-AMAN, runway management, DMAN; in which runway (capacity) management considers AMAN and DMAN responsibilities, i.e. managing final arrivals and departures)
    - 2-system option (E-AMAN, runway management)

- What information should be used in DCI? How many legs can/should be taken into account when computing future expectations of delays? How feasible it is to use global optimiser v. local ones?
- What information should be used/processed by E-AMAN (e.g. time of arrival only)?
- Is it sufficient to model the ATCOs as random delay generators (i.e. they do not know about E-AMAN decisions)? Should the flight be able to change its trajectory to comply with E-AMAN targets?
- Does the model need a detailed geographical simulation (knowing the spatial position of each aircraft at any time)? Is a time-based model otherwise sufficient?
- Should flights be completely passive (taking 'commands' from E-AMAN, UDPP, the NM, etc.) or should they be pro-active, distinct systems (e.g. also considering the next rotation of the aircraft?
- What heuristics could we use for delay estimation (DCI)? How would the use of different heuristics affect the appropriateness of the model (e.g. learning a sub-optimal strategy that reduces short-term delay costs but worsens downstream impacts for the same airline)?
- What are the shared, c.f. unique, goals of agents in the model and their objective functions? How do they communicate with each other (i.e. through synchronous or asynchronous channels)?

5 References

Airbus, 1998. Getting to grips with the cost index, Issue II. Flight Operations Support & Line Assistance (STL), Customer Service Directorate. Blagnac.

Boeing, 2007. Fuel conservation strategies: cost index explained. AERO Quarterly (Quarter 2), Seattle.

Cook A, Tanner G, Williams V, Meise G, 2009. Dynamic cost indexing – Managing airline delay costs, Journal of Air Transport Management, 15(1), 26-35, ISSN 0969-6997. https://doi.org/10.1016/j.jairtraman.2008.07.001

ICAO, 2015. TBO Concept (V. 11.0). ICAO ATM Requirements and Performance Panel. https://www.sesarju.eu/sites/default/files/documents/reports/Appendix_A_to_WP652_TBOCD_3.0.pdf

6 Acronyms/glossary

Acronym	Full name
A-CDM	Airport Collaborative Decision Making
AMAN	Arrival MANager
ANSP	Air Navigation Service Provider
AOC	Airline Operations Centre
ATCO	Air Traffic COntroler
ATFCM	Air Traffic Flow Capacity Management
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
AU	Airspace User
CDM	Collaborative Decision-Making
CHT	Charter carrier
CI	Cost Index
DCB	Demand and Capacity Balancing function
DCI	Dynamic cost indexing
DMAN	Departure MANager
E-AMAN	Extended Arrival MANager
EATMA	European Air Traffic Management
EC	European Commission
ECAC	European Common Aviation Area
ETA	Estimated Time of Arrival
ETO	Estimated Time Over
FDR	Fleet Delay Reordering
FIR	Flight Information Region
FMP	Flight Management Position

Acronym	Full name
FMS	Flight Management System
FOC	Flight Operation Centre
FPFS	First-planned first served
FSC	Full-service carrier
IFR	Instrument Flight Rules
KPI	Key Performance Indicator
LCC	Low-cost carrier
NAS	National Airspace
NM	Network Manager
NOP	Network Operations Portal
PCP	Pilot Common Project
SESAR	Single European Sky ATM Research
SFP	Selective Flight Protection
SJU	SESAR Joint Undertaking
STAM	Short-Term ATFCM Measures
TBO	Trajectory-Based Operations
TMA	Terminal Manoeuvring Area
TOBT	Target Off-Block Time
UDPP	User-Driven Prioritisation Process
VFR	Visual Flight Rules

-END OF DOCUMENT-