EXPLORATORY RESEARCH

sesar JOINT UNDERTAKING

D4.1 TAPAS Integrated Prototype

Deliverable ID:	D4.1
Dissemination Level:	PU
Project Acronym:	TAPAS
Grant:	892358
Call:	H2020-SESAR-2019-2
Topic:	SESAR-ER4-01-2019 Digitalisation and Automation principles for ATM
Consortium Coordinator:	CRIDA
Edition Date:	07 June 2022
Edition:	00.04.01
Template Edition:	02.00.02

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
G. Vouros / **UPRC**	WP4 Leader	07.06.2022
G. Papadopoulos/ **UPRC**	Researcher	07.06.2022
G. Santipantakis / **UPRC**	Researcher	07.06.2022
A. Bastas / **UPRC**	Researcher	07.06.2022
T. Kravaris / **UPRC**	Researcher	07.06.2022
G. Andrienko / **FRHF**	Researcher	07.06.2022

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
R. Rodríguez / CRIDA	Project member	08.06.2022
N. Valle / CRIDA	Project member	08.06.2022
F. Lema / CRIDA	Project member	08.06.2022
E. Iglesias / CRIDA	Project member	08.06.2022
N. Andrienko / FRHF	Project member	08.06.2022

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

Name/Beneficiary	Position/Title	Date
Rodríguez, Rubén / **CRIDA**	Project Manager	09.06.2022
Cordero, Jose Manuel / **CRIDA**	Project Coordinator	09.06.2022
Scarlatti, David / **BRTE**	Researcher	09.06.2022
Crook, Ian / **ISA**	Researcher	09.06.2022
Andrienko, Gennady / **FRHF**	Researcher	09.06.2022
Vouros, George / **UPRC**	Researcher	09.06.2022

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
N/A		

Document History

Edition	Date	Status	Author	Justification

00.00.01	10/03/2021	Draft	G.Vouros	TOC
00.00.03	17/03/2021	Draft	G.Santipantakis	Input data sources
00.00.05	26/03 2021	Draft	T.Kravaris	AI/ML specifications
00.00.07	03/04/2021	Draft	K.Lentzos	Explainability components specifications
00.00.08	07/04/2021	Draft	G.Santipantakis	Technical details on integration
00.00.09	13/04/2021	Draft	G.Andrienko	Visual Analytics specifications
00.01.00	16/04/2021	Completed Draft	G.Vouros	
00.01.03	20/04/2021	Completed Draft	K.Lentzos	Change of explainability components output specifications
00.01.05	26/04/2021	Final	G.Vouros	Final Review
00.02.00	26/07/2021	Final	G.Vouros	Revised taking into account SJU comments
00.02.02	03/03/2022	Draft	G.Vouros	Added ToC and description of the CD&R use case
00.02.05	08/03/2022	Draft	G.Santipantakis	Added content on the integration of CD&R components
00.02.08	17/03/2022	Draft	G.Papadopoulos, A. Bastas and G.Santipantakis	Added description of CD&R prototype components' functionality
00.02.09	18/03/2022	Draft	G.Andrienko , N.Andrienko	Added content for the functionality of the Vis&UI CD&R prototype component
00.03.00	28/03/2022	Final	G.Vouros	Final amendments according to reviewers' comments.
00.04.00	23/05/2022		G.Vouros	Major revisions and restructuring to improve readability and address the comments of SJU
00.04.01	07/06/2022		G.Vouros	Revisions to address the round-2 comments of SJU

TAPAS

TOWARDS AN AUTOMATED AND EXPLAINABLE ATM SYSTEM

This document is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892358 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document provides descriptions of the prototype systems that has been implemented in the context of TAPAS, addressing the requirements of the ATFCM and CD&R use cases. Specifically, the document describes the ATFCM and the CD&R use cases and the specific problem that is considered, together with the data sets that are exploited by the system, per use case. The document describes the overall architecture of the ATFCM and CD&R prototype systems, and the functionality of components, specifying in detail the input and output of each of the components. Finally, the document describes how components are integrated in a prototype system.

Table of Contents

List of Figures

List of Tables

1 Executive Summary

This document is the final report that serves as a reference for the integrated prototype systems addressing the requirements towards explainable Artificial Intelligence methods for (a) the ATFCM use case and (b) the CD&R use case, in the context of the TAPAS project. These prototype systems have been used during the TAPAS validation exercises performed in the ATFCM and in the CD&R use cases, providing solutions and explainability functionality to operators.

First, the document describes in detail the ATFCM use case and the roadmap of the functionalities decided for this use case, up to realizing the needs for implementing automation level 3. Then it describes the overall ATFCM system architecture, functionality of individual components and their interactions.

Then, it proceeds to describe in detail the CD&R use case and the roadmap of the functionalities decided, up to realizing the needs for implementing automation level 3. Then it describes the overall CD&R system architecture, and interactions among components.

The Appendix of this document, in the first part provides details on the specific format of files required as input and provided as output per system component. In the second part it provides details concerning the integration of components via the exchange of messages, as implemented.

1.1 Intended Readership

This is a PU document.

1.2 Terminology and Acronyms

Term	Definition
AI	**Artificial Intelligence**
ATC	Air Traffic Control
ATCo	Air Traffic Controller
ATFCM	Air Traffic Flow and Capacity Management
ATM	Air Traffic Management
CD&R	Conflicts detection and Resolution
CP	Conflict point
CPA	Closest Point of Approach
DCB	Demand and Capacity Balancing
DL	Deep Learning
EC	European Commission
HEC	Hourly Entry Counts
LTM	Local Traffic Manager
MDV	Multivariate Data Visualization
ML	Machine Learning
NM	Network Manager
NOP	Network Operations Plan
OCC	Occupancy Counts
SESAR	Single European Sky ATM Research

EUROPEAN PARTNERSHIP

Co-funded by the European Union

TAPAS	Towards an Automated and Explainable ATM System
VA	Visual Analytics
XAI	Explainable AI

2 ATFCM Use Case

2.1 ATFCM Use Case Specification

The scope of this use case is limited to the detection, declaration and resolution of imbalances during the pre-tactical phase of operations, and particularly, focused on D-1 (day before operations). This includes not only imbalances induced by existing differences between demand and capacity per sector, but also induced by the opportunity to improve performance levels.

The primary actor is the Local Traffic Manager (LTM).

The supporting actors are the Air Traffic Service Unit Supervisor and the Network Manager.

Traffic demand from D-7 (seven days before operation) to D-1 (day before operations), that is pre-tactical phase, is available, allowing the Imbalance Monitoring and Prediction Service to calculate all the imbalance outputs.

Capacity figures, expressed for each possible configuration are also available, expressed in terms of entry count capacities.

Once a demand-capacity imbalance is detected, the appropriate kind of spot is declared, and the required DCB measures decided by the ATFCM AI/ML component (restricted to ground delays and level capping measures) at the pre-tactical phase (up to D-1) are provided per automation level, with explanations.

2.2 ATFCM Problem Specification

The DCB problem (or process) considers two important types of objects in the ATM system: *trajectories* and *airspace sectors*.

The *capacity of sectors* is of utmost importance: It determines the maximum number of flights flying within a sector during a specific time interval.

Let there be N trajectories in a set **T** that must be executed over the airspace in a total time period of duration **H** (in hours: typically in 24 hours). Time can be divided in intervals of duration Δ_t specifying the time steps at which decisions are taken. The set of sectors in all possible airspace configurations is denoted by **S**. For active sectors in open airspace configurations (these change in **H**), the demand evolution is measured in consecutive time periods using the Hourly Entry Count measure.

Each trajectory is a sequence of timed positions in airspace. This sequence can be exploited to compute the series of active sectors that each flight crosses- depending on the open airspace configurations, together with the entry and exit time for each of these sectors. Given a specific spatial region (in our case LETOT covering the Iberian Peninsula) the first (last) sector of the flight is where the departure (resp. arrival) airport resides, the entry (resp. exit) time is the departure (resp. arrival) time. However, there may exist flights that cross the airspace but do not depart and/or arrive in any of the sectors of the airspace of interest: In that case we only consider the entry and exit time of sectors within the airspace of interest.

Thus, a trajectory T in the set of trajectories considered, **T,** is a time series of elements of the form:

T=[($sector_1$, $entry_{t1}$, $exit_{t1}$) .... ($sector_m$, $entry_{tm}$, $exit_{tm}$))],

where $sector_i$ is in **S**, $i=1,\ldots m$.

Airspace sectorization changes frequently during **H**, given different operational conditions, and this should be considered when a flight expands in temporal periods with different sectorizations and/or when applying measures to flights to resolve DCB problems: It must be noticed that given different measures imposed to a trajectory, sectors crossed by the same flight may differ, due to the changing sector configurations. This may result to a number of alternative representations of a single trajectory (each representation crossing a different set of sectors) one for each possible measure: Ground delays and re-routings have a direct effect on sectors crossed.

This information per trajectory suffices to measure the demand $D_{s,p}$ for each of the sectors s in the airspace in any period p of duration Δ_t.

Trajectories that co-occur in sector s and time period p and contribute to hotspots, are defined to be *interacting trajectories* for the period p and the sector s.

Given the demand per sector, the DCB problem consists of these cases where the demand exceeds capacity:

Each sector s in **S** has a specific capacity C_s. Imbalances of sectors' demand and capacity occur when $D_{s,p} > C_s$, for any period p in **H**. These cases result to *hotspots* in the airspace.

In case of capacity violation for a period p and sector s, the interacting trajectories are defined as *hotspot-constituting trajectories*: Measures must be applied to one or more of these trajectories to resolve the imbalance in s and p.

Imposing measures to trajectories may propagate hotspots to subsequent time periods for the same and/or other sectors crossed by that trajectory, or may direct trajectories to other sectors (e.g. due to level capping): In any case, the sets of interacting trajectories in different periods and sectors may change due to the application of measures, and thus, hotspot-constituting trajectories may change as well. This can be done in many ways, either via imposing delays or by re-routing the trajectory to different sectors.

Towards the agent-based formulation of the problem, we consider the following:

Each agent A_i is specified to be the aircraft performing a specific trajectory in a specific date and time. Thus, we consider that **agents** and trajectories coincide in our case and we may interchangeably speak of agents A_i, trajectories T_i, or agents A_i executing trajectories T_i. Agents, as it will be specified, have own interests and preferences, although they are assumed to be collaborative, and take autonomous decisions on their measures: It must be noted that agents do not have communication and monitoring constraints, given that imbalances are resolved at the pre-tactical phase, rather than during operation.

Therefore, agents have to learn *joint* measures (ground delays and/or re-routings) to be imposed to their trajectories w.r.t. the operational constraints concerning the capacity of sectors crossed by their trajectories.

It must be noted that agents –although considered collaborative- have conflicting preferences, towards minimizing individual costs, while also jointly executing their planned trajectories safely and efficiently.

Agents with interacting trajectories have to jointly decide on their measures: The decision of one of them affects the others.

The options available in the inventory **Ac**$_i$ of any agent A_i for contributing to the resolution of hotspots may differ between agents: These, for agent A_i are the ground delay options **D**$_i$ from 0 to $MaxDelay_i$, in conjunction to the re-routing options **R**$_i$ at any time step. We consider that (a) delays may be combined with re-routings to provide joint measures for a single agent, thus **Ac**$_i$**=** **D**$_i$ x **R**$_i$, and (b) possible measures may be ordered by the preference of agent A_i to any such option, according to a function $\gamma(i)$: **Ac**$_i$ $\rightarrow\Re$. We do not assume that agents other than A_i have any information about $\gamma(i)$. This represents the situation where airlines set own options and preferences for measures even in different individual own flights, depending on operational circumstances, goals and constraints. However, as far as ground delays are concerned, we expect that the order of preferences should be decreasing from 0 to $MaxDelay_i$, although, with a different pace for different agents.

Problem statement: Considering any two interacting agents A_i, and A_j, these agents must select measures among the sets of available options in **Ac**$_i$ x **Ac**$_j$, so as to resolve hotspots w.r.t. their preferences on options $\gamma(i)$ and $\gamma(j)$.

According to the problem formulation stated above, and using the model of collaborative multi-agent MDP framework we assume:

- The *society of agents* **A** corresponding to trajectories.
- A *time step* $t=t_0, t_1, t_2, t_3,..., t_{max}$, where $t_{max}- t_0=$**H and** $t_{i+1} - t_i$ is constant for any i=0,...max and defines the simulation time step Δ_t.
- A *local state* per agent A_i at time t, comprising state variables that correspond to (a) the measures applied to that agent$_i$ up to time point t, ranging to the sets of options assumed by A_i, and (b) the number of hotspots in which A_i is involved in (for any of the sectors it crosses). Other parameters that contribute to problem awareness and provision of explanations on agent's decision-making logic are included, as these are also indicated by transparency/ explainability requirements.

  Such a local state is denoted s^t_i.

  The *joint state* $\mathbf{s}^t_{Ag}$ of a set of agents A_g at time t is the tuple of the state variables for all agents in A_g. A *global (joint) state* $\mathbf{s}^t$ at time t is the tuple of all agents' local states.

  The set of all joint states for any subset A_g of **A** is denoted $\mathbf{State}_{Ag}$, and the set of joint society states is denoted by **State**.
- The *local strategy* for agent A_i at time t, denoted by str^t_i is the action that A_i performs at that specific time point: Such an action, in case the agent is still on ground, is to add some nimutes of delay until the next time step, according to simulation time step Δ_t. The agent may decide, also in conjunction to delay, on possible re-routings (i.e. crossing different sectors instead of those initially intended to cross).

  The *joint strategy of a subset of agents* A_g of **A** executing their trajectories (for instance of **N**(A_i)) at time t, is a tuple of local strategies, denoted by $\mathbf{str}^t_{Ag}$ (e.g. $\mathbf{str}^t_{N(Ai)}$).

  The *joint strategy* for all agents **A** at any time instant t is denoted $\mathbf{str}^t$.

  The set of all joint strategies for any subset A_g of **A** is denoted $\mathbf{Strategy}_{Ag}$, and the set of joint society strategies is denoted by **Strategy.**
- The *state transition function Tr* gives the transition to the joint state $\mathbf{s}^{t+1}$ based on the joint strategy $\mathbf{str}^t$ taken in joint state $\mathbf{s}^t$. Formally

  Tr: **State** $\times$ **Strategy** $\rightarrow$ **State**.

 Co-funded by the European Union

It must be noticed that although this transition function may be deterministic in settings with perfect knowledge about society dynamics, the state transition per agent is stochastic, given that no agent has a global view of the society, of the decisions of others, and/or of changing sector configurations, while the interacting agents change. Thus, no agent can predict how its own state will be affected in the next time step. Thus, for agent A_i this transition function is actually

Tr: $\mathbf{State}_{Ai} \times \mathbf{Strategy}_{Ai} \times \mathbf{State}_{Ai} \rightarrow [0,1]$, denoting the transition probability $p(s^{t+1}_i | s^t_i, str^t_i)$.

- The *local reward* of an agent A_i, denoted Rwd_{Ai}, is the reward that the agent gets by executing its own trajectory in a specific joint state with other interacting agents, according to the sectors' capacities, and the joint strategy of agents. The *joint reward*, denoted by Rwd_{Ag}, for a set of interacting agents A_g specifies the reward received by agents in A_g by executing their actions in their joint state, according to their joint strategy.

  The reward Rwd_{Ag} for and subset *Ag* of **A** depends on the participation (contribution) of agents in hotspots while executing their trajectories according to their joint strategy $\mathbf{str}^t_{Ag}$ in their joint state $\mathbf{s}^t_{Ag}$, i.e. according to their decided measures: Any agent shall get higher rewards for actions (corresponding to hotpots resolution measures) that contribute to resolving hotspots.

- *A (local) policy* of an agent A_i is a function π_i: $\mathbf{State}_{Ai} \rightarrow \mathbf{Strategy}_{Ai}$ that returns local strategies for any given local state, for A_i to execute its trajectory. The objective for any agent in the society is to find an optimal policy π_i that maximises the expected discounted future return

$$V_{Ai}\ (s) = max_{\pi i} E\left[\sum_{t=0}^{\infty} \gamma^t * Rwd_{Ai}\left(s^t_{Ai}, \pi_i\left(s^t_{Ai}\right)\right) | \pi_i\right]$$

  for each state s^t_{Ai}, while A_i executes its trajectory, given a set of measures. The discount factor γ ranges in [0,1].

This model assumes the Markov property, assuming that rewards and transition probabilities are independent of time.

2.3 ATFCM Functional Roadmap

Based on the TAPAS ATFCM Use Case description, the following functions / tasks have been identified. For each function / task, the description is provided, as well as the input, output and step of the ATM Master Plan Automation Levels Task Breakdown.

1. Traffic Demand Monitoring (pre-tactical phase, focused on D-1)

This function refers to the continuous traffic demand monitoring. This includes the monitoring of demand indicators.

2. Identification of imbalances

Once the traffic demand exceeds the established traffic monitoring values or thresholds, an imbalance between the traffic demand and the available capacity is identified.

3. Analysis of the imbalances detected

Co-funded by the European Union

This task is devoted to the analysis of the characteristics of each one of the imbalances detected based on the violation of traffic demand indicators thresholds. The final objective is to determine if the imbalance is safety-critical or not.

4. Identification of hotspots

Based on the nature of the imbalance, this function refers to the identification of the appropriate kind of spot, i.e. hotspot or optispot.

5. Declaration of hotspots / optispots

Once a hotspot / optispot is identified, the LTM will declare it to the Network Manager.

6. Preparation of DCB measures to solve the hotspot

This task refers to the preparation of the required DCB measures to be taken in order to solve the hotspot.

This preparation task also includes the selection of the candidate flights to be impacted by the DCB measures considered and the analysis of the DCB measure impact (What-If).

7. Decision on the DCB measure and flights impacted

Once the preparation of the DCB measure is finished, local criteria are considered to make the final decision on which measure should be applied and which flights are going to be impacted.

8. Implementation of the DCB measures

This function refers to the implementation, by the corresponding actors (Network Manager, Airspace Users, etc.) of the DCB measures decided to be applied.

9. Hotspot Resolution monitoring

For the different hotspots declared and the multiple DCB measures taken, the status of the hotspot, that is, if it is resolved or not, should be monitored in order to detect any possible deviation due to the volatility of the air traffic demand.

Considering the functions identified, the following table depicts the allocation of tasks between the human and the machine for the ATFCM Use Case, following the Automation Levels considered by the European ATM Master Plan.

ATFCM Functions / Tasks	Automation Level 1	Automation Level 2	Automation Level 3
Traffic Demand Monitoring	Machine	Machine	Machine
Identification of imbalances	Machine	Machine	Machine
Analysis of the imbalances detected	Human	Machine	Machine
Identification of hotspots / optispots	Human	Machine	Machine
Declaration of hotspots / optispots	Human	Human	Machine
Preparation of DCB measures to solve the hotspot • Capacity Measures (sector configuration) • ATFCM Scenarios • Trajectory Measures • ATFCM Regulation • Selection of candidate flights • Analysis of the DCB measure impact – What-if	Human	Machine	Machine

ATFCM Functions / Tasks	Automation Level 1	Automation Level 2	Automation Level 3
Decision on the DCB measure and flights impacted	Human	Human	Machine
Implementation of DCB measures	Human	Human	Machine
Hotspot resolution monitoring	Human	Machine	Machine

Table 1. TAPAS ATFCM Functional Roadmap

The rationale behind the allocation of tasks between the human and the machine illustrated in **¡Error! No se encuentra el origen de la referencia.** 1 is highly based in TAPAS Advisory Board and operational expert meetings. The following justifications for the TAPAS ATFCM Roadmap proposal have been considered:

1. Automation Level 1 reflects the current general situation at ECAC level in terms of ATFCM, although some ANSPs might be already in Automation Level 2 for some tasks.
2. The allocation of tasks for Automation Level 2 was proposed as the logic follow up of Automation Level 1, supporting the human in action implementation for the tasks for which a further step in automation was considered as needed and beneficial.
3. The distribution of tasks between human and machine for Automation Level 3 was done considering that all the tasks addressed in the TAPAS ATFCM use case could be initiated by automation in nominal cases, leaving Automation Level 4 and Automation Level 5 out of the scope of TAPAS with the consideration of non-nominal cases, as explained below.

Regardless the automation level and the environment conditions (nominal or non-nominal), the human might be able to take control of the system at any time. In addition, for non-nominal situations, the human will continue to have automated support for executing the different tasks, but the control will rely on the human and not in the machine.

2.4 Data sets

Data sets exploited in the ATFCM use case include the Flight Plan and Sector Configuration data sets, briefly presented in the next paragraphs. Details on the structure of files that each dataset comprises are provided in Appendix A.

2.4.1 ALL\FT+ (Traffic) Files

- Spatial Coverage: Worldwide
- Temporal Coverage: 18th July to 14th August 2019 (28 days)
- Format: CSV files compressed to 7z. Each file corresponds to a separate day (each uncompressed file is approximately 1.9GB). Total number of records (in all files) are 1,074,111.

This data source is described in ``DDR2 Reference Manual For General Users 2.9.7'' (pg. 134). The DDR2 reference manual reports 189 fields for the version 6 of ALLFT+ format while the data set used here contains 181 fields in all records.

Figure 1 illustrates the initial/indented trajectories found in the data set.

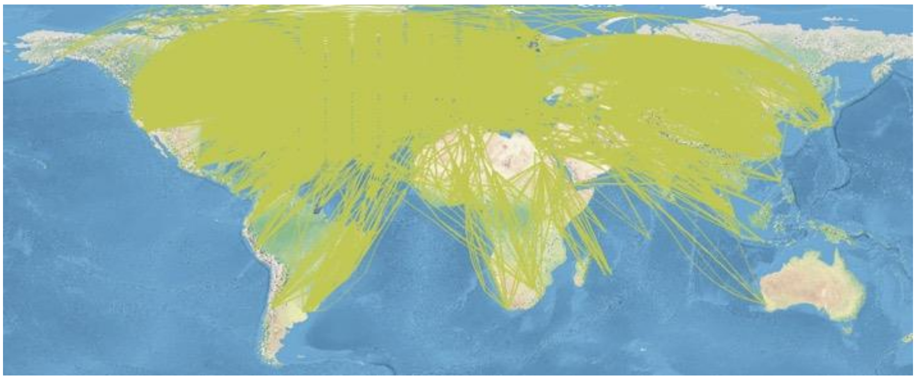


Figure 1. The trajectories in the ATFCM data set

The trajectories found in the flight plans data set are categorized according to their spatial relation to the geometry of LETOT (the part that covers Iberian Peninsula as follows:

a. if the geometry of a trajectory is completely covered by the LETOT geometry, the trajectory is classified as "**closed**" (i.e. both departure and destination airports are within Spanish airspace);
b. the trajectory is classified as "**disjoint"** if no common point between the two geometries exists and it is classified as
c. "**incoming"** or "**outgoing**", if the geometry of the trajectory is partially covered by LETOT and the trajectory ends in LETOT or originates from LETOT, respectively;
d. The trajectory is classified as "**overflight**" when the geometry of the trajectory is at least partially covered by LETOT but neither the trajectory ends in LETOT, nor it originates from LETOT; and finally,
e. a subset of incoming trajectories are distinguished as being "**close to the Spanish airspace**", if they penetrate LETOT geometry during their take off phase.

Any trajectories not satisfying the criteria of any of the above categories, are classified as "other".

Table 2 reports the number of flights in each category found in the sample data set of 18th July to 14th August 2019 (28 days).

Date	disjoint	closed	overflight	outgoing	closeToSpanishAirspac	incoming	other	Total
20190717	1843	1	102	107	4	74	0	2131
20190718	29640	944	1785	2011	27	1989	48	36444
20190719	30019	913	1652	2052	16	2079	37	36768
20190720	25803	673	1919	2236	20	2209	28	32888
20190721	27210	755	1770	2156	16	2112	40	34059
20190722	28882	902	1755	1998	28	1979	43	35587
20190723	28536	849	1663	2054	22	2018	48	35190
20190724	28534	885	1696	2001	31	1984	39	35170
20190725	28990	942	1746	2008	26	2009	33	35754
20190726	29396	890	1690	2052	20	2048	38	36134
20190727	25595	652	1883	2160	17	2147	23	32477
20190728	27100	765	1760	2139	18	2101	38	33921
20190729	28889	892	1779	2017	27	1999	42	35645
20190730	28226	816	1715	2038	12	2019	39	34865
20190731	28727	882	1703	2036	7	2006	39	35400
20190801	29137	854	1788	2044	11	2036	44	35914
20190802	29231	853	1692	2061	11	2070	45	35963
20190803	25755	669	1910	2228	18	2207	33	32820
20190804	27114	728	1749	2150	20	2122	42	33925
20190805	28141	811	1782	2010	18	1998	45	34805
20190806	27556	808	1716	2022	12	2002	41	34157
20190807	27649	800	1677	1986	16	1978	37	34143
20190808	28692	848	1743	2005	19	2004	44	35355
20190809	29359	840	1688	2061	15	2056	40	36059
20190810	25635	680	1897	2246	13	2189	28	32688
20190811	27114	736	1713	2129	18	2101	44	33855
20190812	28342	821	1741	1967	22	1964	42	34899
20190813	27937	808	1681	2028	17	2008	35	34514
20190814	26626	831	1635	1876	16	1877	37	32898
Total	785678	22848	49030	57878	517	57385	1092	**974428**

Table 2. Flights per category for AIRAC1908 data set

2.4.2 Sector Configuration

- Spatial Coverage: Worldwide (more detailed in European airspace)
- Temporal Coverage: (AIRAC 1908) Wednesday, July 17, 2019 9:00:00 PM UTC to Wednesday, August 14, 2019 8:59:00 PM UTC
- Format: A set of CSV files.

The information provided covers the entire globe, as Figure 2 illustrates.

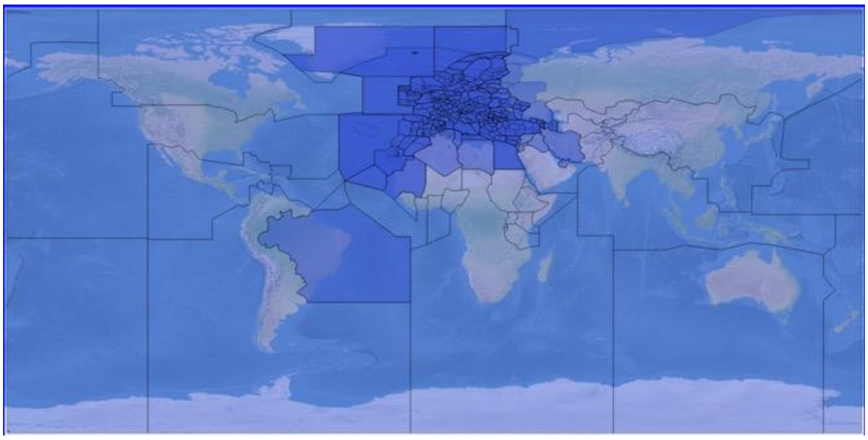


Figure 2. Spatial Coverage of Sector Configuration ATFCM data set.

The Sector and Volumetry information in Demand Data Repository (DDR) is organized in the following files:

- Configurations file (*.cfg): This file contains the information related to airspace configuration.
- Opening scheme file (*.cos): This file contains information of the airspace configuration applied in a certain period of time (active sectors)
- Capacity file (*.ncap): This file contains information of the airspace capacity in a certain period of time.
- (Volumetry) Airspace file (*.spc): This file contains information of the airspace, its id, the number of sectors, and the name of each of the sectors that compose it.
- (Volumetry) Sector Gasel level file (*.gsl/sls): This file contains information of the airspace sector configuration, its id, the number of airblocks, the name of each of the airblocks that compose it and its boundary levels.
- (Volumetry) Airblock file (.gar/are): This file contains information of the airblock configuration, its id, the number of waypoints, the name of each of the waypoints that compose it and its latitude and longitude.

The exact format of these files is provided in Appendix A.

The Sector Configuration data set contains information about 50551 sectors.

Since we address the ATFCM use case in the Spanish airspace only, we need only the sectors that their 2D geometry is spatially covered by the interior and the perimeter of the geometry of LETOT. The 397 sectors that satisfy this spatial filter, are illustrated in Figure 3.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

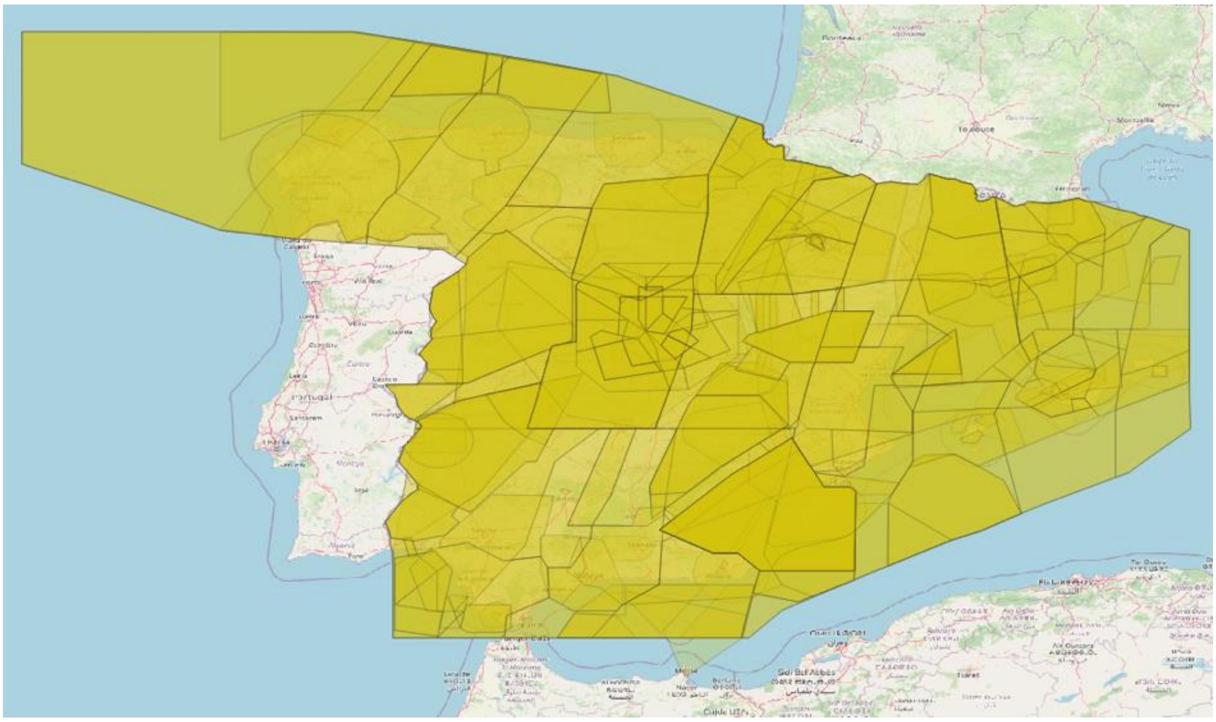


Figure 3. Sectors used in the trajectory enrichment task (ATFCM use case)

The capacity of an airspace sector is evaluated only during the activation intervals of the sector. Formally, the capacity C is defined for a time interval I for an airspace sector S, if S is active for some time interval that intersects with I. The sectors that satisfy the LETOT spatial filter and have a capacity value defined for at least one of their activation intervals are 132 for AIRAC1908.

2.4.3 Data Pre-processing

Data pre-processing functionality is implemented in the data preparation system component, specified in section 2.5.1.1.

2.5 Overall ATFCM Prototype

Figure 4 provides the overall architecture of the ATFCM prototype system, including, data preparation, AI/ML, Explainability components, and the Explainability Logic as part of the Visual Analytics component.

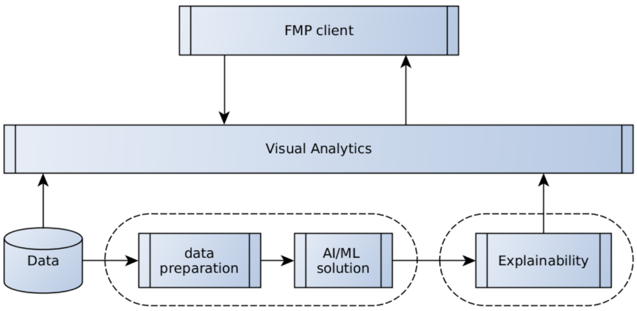


Figure 4. Overall ATFCM Integrated Prototype and interaction with FMP

Given the **data sources** (as these are specified in section 2.4), these are exploited by

(a) The **Visual Analytics** component, that provides data exploration facilities, and

(b) The **data preparation component** for clearing and linking data sets to provide the representations of flights' enriched trajectories, as these are specified in the ATFCM problem specification (Section 2.2) and in the exact form required from the AI/ML module.

The **AI/ML** algorithms provide the solutions for the DCB problems identified.

It must be noted that in Automation level 2 the system may provide more than one alternative solutions with appropriate explanations, while in automation level 3 the system provides the solution that it implements.

The **explainability components** provide explanations' content to the explanation logic.

The **explanation logic** realizes surface representations and renders textual and visual information in appropriate displays: together with **visual analytics** it provide advanced facilities for exploring, transforming, comparing, filtering and/or projecting explanation data, in conjunction to data from the data sources.

Appendix B provides technical information on the integration of components into a functional whole.

2.5.1 Functional specification of components

The next paragraphs present for each of the functional components of the ATFCM integrated prototype, the input files, a description of functionality implemented and the output files.

The detailed structure of each file is defined in Appendix A, for each of the components, and according to their order in this section.

2.5.1.1 Data Preparation

2.5.1.1.1 Input

Flight plans and sector configurations, as specified in Section 2.4.

2.5.1.1.2 Functionality

This component produces alternative flight plans for any potential ground delay that may be imposed to a flight with/without any level capping measure. The output comprises enriched flight plans, as explained below.

The building blocks for the ATFCM use case, are aircraft trajectories compiled from the point profile of their flight plans (FTFM), and the airspace configurations.

Airspace configurations define the geometrical boundaries of airspace sectors, the activity intervals (i.e. temporal intervals for which the airspace is active), as well as their capacity.

It is essential that the points of any trajectory to be associated to the corresponding active airspace sectors in which they occur, at the time of their occurrence. We call this process as **(trajectory) data enrichment**, since it associates each trajectory position with contextual information.

The data enrichment, in addition to the above, involves the computation of entry/exit points for each crossed active sector, and the corresponding entry/exit time instances. The computed points are also added in the trajectory and annotated as "inferred" points, to indicate the transitions between sectors.

The inferred points are used (a) for computing the demand for the active sectors, and identifying hotspots, and (b) for identifying the trajectory segments affected by level capping measures. The application of DCB measures may result into new such points.

2.5.1.1.3 Output

The data preparation component produces, given a flight plan, alternative flight plans for any potential ground delay and/or level capping measure.

The files generated for each flight are stored in a folder named ADEP-ADES-Callsign-LOBT given the corresponding data of the flight. Each such folder is stored in a folder that corresponds to the date of flight.

For example, the files generated for the flight from LEMD to LEJR, with callsign GES321S and LOBT 20190801051500, will be stored in the folder LEMD-LEJR-GES321S-20190801051500, under the folder 20190801. Flights having their take-off and landing at different dates (date changes during flight) are replicated in the corresponding date folders.

The generated files for each flight are as follows:

1. delay: this file is generated by temporally shifting all the positions reported in the FTFM point profile (FTFM trajectory).

It is a 3-column TSV file, reporting the flight plan key, the ground delay (minutes) and the trajectory as a list of positions, where latitude/longitude for each position are provided in WGS84 CRS, altitude in feet, timestamp in unixtime. Each position is also enriched with the ID of the active sector (CS or ES) that contains the position. The file reports only changes on the original flight plan (i.e. if the crossed sectors do not change given some minutes of ground delay, the trajectory is not reported in the file).

2. delay.raw: contains the FTFM point profile and crossed sectors, temporally shifted by the delay minutes. This file is generated for validation and debugging purposes.

3. fp: contains the FTFM point profile data as provided in the source, for validation and debugging purposes.

4. traj.raw: this file is the same as the file delay.raw, but also includes a flag identifying each position as either asserted (retrieved from the FTFM point profile) or inferred (interpolated).

5. levelCapping.txt: This file specifies trajectories that result due to the application of DCB measures to the original trajectory. It is a 3-column TSV file where the first column reports the ground delay, the second column reports the combination of sectors regulated due to level capping measures (the "empty" combination is included)

6. ranking.txt: reports values estimating the quality of each alternative route computed, for each segment of the trajectory crossing a sector and which is different than the one originally planned, and a total value for the entire trajectory.

7. sectors.xai: reports for each potential delay regulation and combination of potential level capping measures, the changes made on the trajectory w.r.t. the crossed sectors. For example, the record:

0 [LECBLVU]   LECBLVU->[LECBGO2, LECLVVLC, LEBLDDS]

reports that at delay 0 and a level capping regulation on sector LECBLVU, the trajectory has to cross the sectors [LECBGO2, LECLVVLC, LEBLDDS]

2.5.1.2 AI/ML Solutions

2.5.1.2.1 Input

As described in the Output of the “Data Preparation” component in Section 2.5.1.1.3.

2.5.1.2.2 Functionality

This component implements the following functionality regarding the ATFCM prototype:

- Traffic Demand Monitoring (pre-tactical phase, focused on D-1)
- Identification of imbalances
- Analysis of the imbalances detected
- Identification of hotspots
- Declaration of hotspots
- Preparation of DCB measures to solve the hotspot
- Decision on the DCB measure and flights impacted

Actually, it prescribes the DCB measures to be taken towards resolving hotspots in critical cases. A critical case is defined to occur when the demand, measured by means of the Hourly Entry Count measure, exceeds 110% of sector capacity at a specific period.

To do that, this component implements an agent-based simulation, where agents (corresponding to flights) decide on additional minutes of ground delay to be taken at every time step of the simulation according also to the ATFCM problem specification specified in Section 2.2. Currently, the time step of the simulation is 10 minutes and the decision of agents for additional delay at every time step ranges in [0,10]. Agents at any time step may increase their delay by adding additional minutes, until they reach their *MaxDelay.*

Specifically, a specific scenario of 1439 minutes is simulated (**H**=24 hours), given a simulation time step set to 10 minutes. Each episode comprises a series of rounds. Each round corresponds to a simulation time step. At each round every agent takes a concrete decision for [0..10] minutes of delay: This is according to the methodology implemented, where flights participating in hotspots are not regulated with a number of minutes at once (e.g. 14 min of delay), but by adding to their existing delay at each time step a number of minutes up to MaxDelay, according to needs.

Agents, to decide on additional delay, and at every time step, they calculate the demand per active sector in the airspace, identify the imbalances and hostpots, prepare the types of measures to be taken to resolve any hotspot, and finally, they individually decide on the DCB measures to be taken.

The types of solutions considered by the agents correspond to the three different types of measures decided by this component:

- Solution type 0: These are solutions where agents consider only ground delay.
- Solution type 1: In these solutions agents decide on level capping measures.

- Solution type 2: In this case agents consider combinations of level capping and ground delay measures.

2.5.1.2.3 Output

The output from the AI/ML component concerns both, the solutions computed and hotspots identified, as well as explanations regarding decisions taken.

Given that alternative solutions can be provided, we identify each solution with a SolutionID.

While each "solution" describes the measures applied to a flight plan (FTFM), the "decisions" elaborate more on the solution and provide details for each decision taken, per agent (flight) and (in case of delays) at any time step of the simulation.

In the case of solutions with only level capping regulations, no simulation takes place. The method takes into account the existing daily traffic, given all the initial flight plans and each flight (agent) decides whether to have a set of level capping measures in specific sectors (mostly those to which hostpots exist), depending on the number of hotspots it participates. Thus, the decision of level capping measures is an "one-shot" decision per agent (i.e. not a sequential decision task), given the initial state resulting from FTFM and active sectorizations.

2.5.1.2.3.1 Solution

File name: scenario_*_exp**_solution.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the flight plans with the specific measures decided by the AI/ML method, as well as Flight IDs and measures applied.

2.5.1.2.3.2 Baseline Flight Plans

File name: scenario_*_exp**_baseline_flight_plans.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the flight plans of the scenario. In the case of level capping measures combined with ground delays (solution type 2) this file contains the resulted flight plans after applying the level capping measures decided.

2.5.1.2.3.3 Baseline HotSpots

File name: scenario_*_exp**_baseline_hotspots.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the hotspots that are identified at the beginning of the scenario. In the case of level capping regulations combined with ground delays (solution type 2) this file contains the hotspots detected after the application of the level capping measures.

2.5.1.2.3.4 Baseline HotSpots Flights

File name: scenario_*_exp**_baseline_hotspots_flights.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the flights that participate in the baseline hotspots (i.e. the hotspots that occur at TimeStep=0). In the case of level capping regulations combined with ground delays (solution type 2) this file contains the flights that participate in hotspots after the application of the level capping measures.

2.5.1.2.3.5 Decisions

File name: scenario_*_exp**_decisions.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the decisions of each agent at each TimeStep of the simulation, until all hotspots are resolved, or until it is decided that no further delay should be applied.

2.5.1.2.3.6 Demand

File name: scenario_*_exp**_demand.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the demand of all sectors at all periods, throughout the simulation, taking the measures by the agents into account.

2.5.1.2.3.7 HotSpots

File name: scenario_*_exp**_hotspots.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the hotspots at every time period. At first (TimeStep=0) it provides the initial hotspots of the scenario. It includes entries up to the TimeStep when all hotspots have been resolved, or when TimeStep=1439, i.e. when the simulation ends.

2.5.1.2.3.8 HotSpots Flights

File name: scenario_*_exp**_hotspots_flights.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the flights that participate in hotspots at any time period during simulation. The key to cross the different sources is the tuple <SolutionID, SectorID, TimeStep, TimePeriod >.

2.5.1.2.3.9 Snapshots

File name: scenario_*_exp**_snapshots.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the location of the flight at any given minute of the day, according to simulation. This file does not exist for Solution type 1.

2.5.1.2.3.10 Output of AI/ML to XAI component

2.5.1.2.3.10.1 SGT Train

File name: kl_train_scenario_*.csv

- *Year, Month, Day of the scenario, for example 20190801

Overview: Provides the agents' states for a number of episodes with the trained AI/ML model interacting with the environment for the purpose of training the explainability model. To gather as much information as possible, agents use an exploration scheme according to which exploration diminishes as the episodes progress.

2.5.1.2.3.10.2 SGT Test

File name: kl_test_scenario_*.csv

- *Year, Month, Day of the scenario, for example 20190801

Overview: Provides the states of a single episode, where the trained model interacts with the environment.

2.5.1.2.3.11 Flight ID Map

File name: scenario_*_ flight_id_map.csv

- *Year, Month, Day of the scenario, for example 20190801

Overview: Provides a translation between the numeric representation of flights and their actual id.

2.5.1.2.3.12 Baseline Flight Plans

File name: scenario_*_exp**_baseline_flight_plans.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Overview: Provides the flight plans of the scenario. In the case of level capping measures combined with ground delays (solution type 2) this file contains the flight plans after the level capping measures decided.

2.5.1.3 AI/ML Explainability Component

2.5.1.3.1 Input

As described in the Output of the “AI/ML Solutions” component in Section 2.5.1.2.3

2.5.1.3.2 Functionality

In case of solution type with delays only (Solution type 0), the explainability component provides per agent and per simulation timestep, arguments regarding agents’ individual decisions to add minutes of delay to their ground delay. This explains every decision (i.e. additional minutes of delay) that each agent forms during the simulation, until the last timestep. In case of level capping measures, explanations are provided only for agents’ decisions to apply a level capping measure (this, as already said, is one-shot decision and does not involve sequential decisions taken in subsequent time steps).

In any case, the provided explanation arguments specify:

- Why the agent has taken a specific decision.
- What the agent considers as counter arguments against the decision taken (this is contrastive explanation given the above, providing counter arguments for agent’s actual decision).

Specifically, for each decision taken by an agent (flight) at a given state occurring at a time step, an explanation is formed.

For each possible decision (i.e. 0/1 for level capping, and 0,1,2,…10, for additional minutes of delay) a mimicking Stochastic Gradient Tree (SGT) model is trained in order to provide the explanations, realizing a regression process on the action values provided by the well-trained AI/ML component. We have as many trees as the number of different options per solution type.

Thus, in case the agent decides on additional delay at any time step, there are 11 trees corresponding to 1,2,…,10 minutes of additional delay per simulation time step (providing action values Q1, Q2,…,Q10), and one that explains why the agent considers it should not take any delay (providing the action value Q0). The final decision 0, 1, 2, … or 10 from the SGT is taken by the tree predicting the maximum action value among the Q0, Q1,…,Q10.

In case of level capping measures SGT uses 2 trees corresponding to decision 0 for **not** changing the original flight plan, and decision 1 for applying a level capping measure: Q0 corresponds to action 0 (level capping measures do not apply) and Q1 corresponds to action 1 (level capping measures apply).

In both cases, counterarguments for a decision are provided by the SGT model corresponding to any action, which is different to what it has been decided.

Each SGT model is a decision tree: A non-leaf node specifies a split attribute defined by a state feature and a split value for that attribute, defining an argument for a decision. The explanation for the decision taken at a specific state consists of all arguments in all nodes in the SGT path from the root to a leaf node. That path shows the specific state features considered and conditions tested to reach a decision on a DCB measure.

All state features and their values are provided in State Information (Section 2.5.1.3.3.1). The corresponding SGT nodes with the split attributes and values are provided in Explanations Q & Explanations Q Thresholds (Sections 2.5.1.3.3.2, 2.5.1.3.3.3), respectively.
In particular, explanations comprise arguments for or against a decision, of the form

SplitAttribute:Sector:StateValue:Interval

The SplitAttribute is a state feature. If this feature is the sector of a hotspot, or the duration of the agent in a sector, then the Sector specifies the sector id that corresponds to this feature, otherwise it is "null". The StateValue is the value of this feature in the state. The Interval is a string of the form SplitValue;+inf or -inf;SplitValue, where the SplitValue is the threshold specifying the criterion for the split regarding this SplitAttribute at a particular SGT tree node.

Additionally, explanations on all possible decisions an agent can take are provided in separate files: one file per decision and solution type. Specifically, for each possible action a file (specified in 2.5.1.3.3.4) that contains the SGT explanations of the considered action is created. The order of features appearing in an explanation shows how important a feature is for the considered action.

2.5.1.3.3 Output

This subsection provides the form of the output from the AI/ML Explainability component.

2.5.1.3.3.1 State Information

File name: states_scenario_*.csv

- *Year, Month, Day of the scenario, for example 20190801

Overview: List of state features with values, showing the state of the agent prior to the decision, as well as the action values predicted from the AI/ML and XAI components, and the resulting decision.

2.5.1.3.3.2 Explanations Q

File name: explanationsQX*.csv

- X: for solutions of type 0 and 2, X is in [0,10],and for solutions of type 1 X is in {0,1}.
  *Year, Month, Day of the scenario, for example 20190801

Overview: Each row provides the list of state features considered as more important for deciding the respective action. The number of columns equals the path length in the SGT corresponding to the action and varies across solutions with different solution types.

2.5.1.3.3.3 Explanations Q Thresholds

File name: explanationsQXSplit_*.csv

- X: for solutions of type 0 and 2, X is in [0,10], and for solutions of type 1 X is in {0,1}.
  *Year, Month, Day of the scenario, for example 20190801

Overview: Each row provides a list of threshold values corresponding to the list of state features. Rows correspond to rows in Explanations Q files. The number of columns equals to the path length in the respective SGT.

2.5.1.3.3.4 Extracted Explanations

File name: xaiQX*.csv

- X: for solutions of type 0 and 2, X is in [0,10], and for solutions of type 1 X is in {0,1}.
- *Year, Month, Day of the scenario, for example 20190801

Overview: Each row provides the list of state features considered as more important for the respective decision on action X.

2.5.1.4 Visual Analytics

2.5.1.4.1 Input

The Visual Analytics (VA) module takes the output of the AI/ML module as input. Specifically: capacities of sectors, variants of flight plans according to the DCB measures decided by the AI/ML component (with and without level capping measures and/or ground delays), with explanations.

2.5.1.4.2 Functionality

VA module enables exploration of the input data, providing

- visual summaries for the variants of solutions;
- the evolution of the solutions over iteration steps of the simulation process;
- details for each hotspot, sector, and time interval, including aggregated information about the flight delays;
- on demand: features that justified delay decisions for selected flights, sectors, and time periods;
- further relevant information, to be selected in collaboration with project partners.

The VA module is supposed to provide tools for comparing solutions in overall and, specifically, for different aspects (sectors, time intervals etc.)

2.5.1.4.3 Output

Currently, no outputs of VA to be consumed by other components is foreseen. At later stages of the project, the VA module may output specifications of constraints (e.g. no delays allowed for flights passing sector X in interval T, delays less then 15min for flights departing from airport A, etc.) to be considered by the AI/ML module. Potentially useful and feasible to support constraints are to be defined later.

2.5.1.5 Output to FMP Client

The output to the FMP client comprises the following per automation level:

- Automation Level 2: Traffic demand.
- Automation Level 3: Traffic demand, DCB measures (level capping, ground delay), hotspots and flights impacted.

Specifically, the output comprises:

- Specification of hotspots, as this is described in Section 2.5.1.2.3.7.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

  - Baseline hotspots are provided as specified in Section 2.5.1.2.3.3
  - Flights participating in hotspots are specified according to Sections 2.5.1.2.3.4(baseline) and 2.5.1.2.3.8.
- Measures are specified as agents' decisions, as described in Section 2.5.1.2.3.5.
- Traffic demand is specified as described in Section 2.5.1.2.3.6

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3 Conflicts Detection and Resolution (CD&R) Use Case

3.1 CD&R Use Case Specification

This use case concerns the conflict detection and resolution process in the Executive Controller timeframe, as part of the tactical separation assurance process.

The primary actor is the Executive Controller and the supporting actors are the executive controllers of the upstream and downstream sectors.

It is assumed that the conflict detection and resolution during the planner timeframe for action has been performed. Separation requirements are considered to be independent from the region being analysed (area of responsibility (AoR)) and stable. Flights are able to traverse the AoR and are handed-over to the downstream sector free of conflicts, w.r.t. specific exit/entry sector conditions.

The objective is to perform conflict detection and resolution during the tactical separation assurance process, identify and resolve all the potential tactical conflicts.

3.2 CD&R Problem Specification

The CD&R task involves a number of flights in a spatial area of responsibility (AoR). The task is about detecting at any time point *t* the conflicts that may occur between any pair of flights, and for any such conflict, decide whether and what resolution actions should be applied to any, or both, of the conflicting flights.

Casting this problem into a multiagent problem, we consider that each agent *i* represents one of the flights in the AoR, or in any potential downstream sector. We consider the set of Relevant AoRs (RAoRs)} as the union of AoR with any potential downstream sector.

Given the trajectory T_i of agent *i* within the RAoRs, we define the set of neighbouring agents to be the set of conflicting trajectories to T_i in RAoRs at a specific time point *t*. These are denoted by *Neigh*(*i,AoR,t*).

Agent *i* has to react and resolve all conflicts with *Neigh*(*i,AoR,t*), deciding whether it will apply any resolution action at *t*+1, and what this action should be.

Specifically, we consider the following potential actions:

(a) *Flight Level change*, where the agent changes its current flight level with vertical speed +17 or -17 feet/s for ascending/descending course;

(b) *Course change*, where the available changes of agent's course are 10, -10, 20, -20 degrees;

(c) *Horizontal speed change*, where the available changes of agent's horizontal speed are -3.6008 or 3.6008 m/s;

(d) *Direct to waypoint*, where the agent can choose one of the next flight plan waypoints; and

(e) *No action*, where the agent continues its current course without any change.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

The problem is formulated as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), where at each timestep *t* each agent *i* receives a local observation o^t_i, takes an action a^t_i, and gets an individual reward r^t_i. The objective is to maximize the sum of all agents' expected returns by resolving conflicts.

A local observation of an agent is a vector comprising the following features:

- $Nalt = alt/max_{alt}$, where *alt* is the agent's current altitude in feet and max_{alt} is a normalization factor,

- *cos x* and *sin x*, where *x* is the bearing of the aircraft, i.e. the angle of the agent's course w.r.t North, in degrees,

- $Nh_{speed}=(h_{speed}-min_{hspeed}) / (max_{hspeed}-min_{hspeed})$, where h_{speed} is the magnitude of the agent's horizontal speed in m/s, max_{hspeed} and min_{hspeed} are mormalization factors,

- cos(x-ψ)$ and sin(x-ψ), where ψ is the relative bearing of the agent w.r.t. the AoI exit point,

- NdistExitPoint= d_{exit}/D_{exit}, where d_{exit} is the horizontal distance of the agent w.r.t. the AoR exit point in meters, and D_{exit} is a normalization factor,

- NaltDiffExitPoint=$|alt - alt_{Exit\ Point}| / |max\ |alt - alt_{Exit\ Point}|$, where $|alt - alt_{Exit\ Point}|$ is the absolute difference in feet between the agent's altitude at the AoR exit point and the filed altitude at the AoR exit point,

- $\cos(dcourse_{wp})$ and $\sin(dcourse_{wp})$ for each one of the next four waypoints wp=1,2,3... according to the most recent flight plan, where $dcourse_{wp}$ is the angle of the current agent's course w.r.t. the course that the agent must follow to reach the corresponding waypoint,

- NdistWaypoint= h_{d_wp}/HD for each one of the next four waypoints according to the most recent flight plan, where h_{d_wp} is the horizontal distance in meters between the current agent's position and the position of the corresponding waypoint. HD is a normalized factor.

- NaltDiffWaypoint=v_{d_wp}/VD for each one of the next four waypoints, where v_{d_wp} is the vertical distance in feet between the agent's altitude at the waypoint and the filed altitude at the waypoint, and VD is a normalized factor.

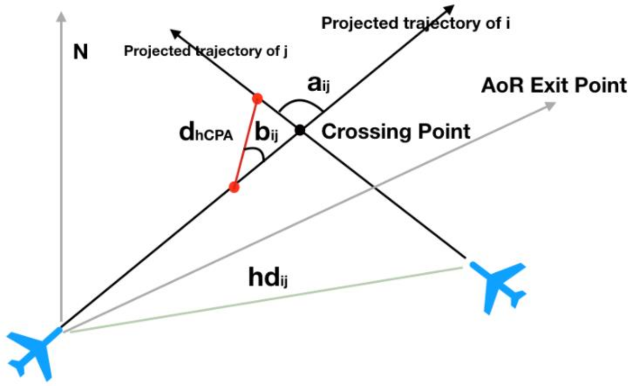


Figure 5. CPA Geometry

In addition to these observations, each agent i maintains a vector e_{ij} with any agent j in *Neigh*(*i,AoR,t*), comprising *ij* edge features that depend on the Closest Point of Approach (CPA): The point in which agents *i* and *j* are estimated to be (or have been) closer using trajectories projections for a time horizon of t_h minutes, as shown in Figure 5:

- $Nt_{CPA}=t_{CPA}/T_{CPA}$, where t_{CPA} is the time required in seconds for agent i to reach the CPA with agent j and T_{CPA} is a normalization factor,

- $N_{d\_h\_CPA}=d_{h\_CPA}/D_{h\_CPA}$, where d_{h_CPA} is the horizontal distance in meters between agents i and j at the CPA and D_{h_CPA} is a normalization factor,

- cos $a_{i,j}$ and sin $a_{i,j}$, where $a_{i,j}$ is the intersection angle in degrees between agents i and j,

- cos $b_{i,j}$ and sin $b_{i,j}$, where $b_{i,}$ is the relative bearing of agent i w.r.t. to agent j at the CPA,

- $Nd_{v\_CPA}=v_{d\_CPA}/V_{d\_CPA}$, where $\$v_{d\_CPA}$ is the vertical distance in feet between agents i and j at the CPA, and V_{d_CPA} is a normalization factor,

- $Nd_{cp}=d_{cp}/D_{cp}$, where d_{cp} is the distance in meters between agents i and j when any of them passes the crossing point first, and D_{cp} is a normalization factor,

- $Nt_{cp}=t_{cp}/T_{cp}$, where t_{cp} is the time required in seconds for any of agents i and j to pass the crossing point first, and T_{cp} is a normalization factor,

- $Nd_{h(i,j)}=hd_{i,j}/HD$, where $h_{d\_i,j}$ is the current horizontal distance in meters between agents i and j,

- $Nd_{v(i,j)}=vd_{i,j}/VD$, where $vd_{i,j}$ is the current vertical distance in feet between agents i and j.

3.3 CD&R Functional Roadmap

Based on the TAPAS CD&R use case description, the following functions / tasks have been identified for the CD&R use case. For each function / task, the description is provided, as well as the input, output and the ATM Master Plan Automation Levels Task Breakdown.

1. Assessment of the planned and desired trajectory profile

This task refers to the evaluation of each aircraft desired profile (flight plan filled and radiofrequency or datalink requests) against the current trajectory to comply as much practicable as possible with the Airspace Users' preferences.

- **Input:** Flight plan, trajectory (radar track) information and AUs' preferences
- **Output:** Gap between flight desired profile and flight trajectory
- **ATM Master Plan Automation Levels Task Breakdown:** Information acquisition and information analysis

2. Identification of potential conflicts

This task is devoted to identifying all the potential conflicts between the aircraft inside the Executive Controller's Area of responsibility.

- **Input:** Flight trajectories (track) and flight plan for the aircraft within Executive Controller's AoR

- **Output:** Potential conflicts within Executive Controller's AoR
- **ATM Master Plan Automation Levels Task Breakdown:** Information acquisition and information analysis

3. Identification of conflict resolution strategies and clearances proposal

This task includes the proposal of conflict resolution strategies (e.g. flight level change, speed restriction, vectoring, direct to, etc.) for all the conflicts identified in the Executive Controller's Area of Responsibility.

This task is also devoted to the proposal of the most appropriate ATC clearance to be given to the Cabin Crew in order to comply with the proposed resolution strategies and flight sector exit conditions.

This task should consider the management of every planned constraint (flight level, coordination, speed restriction, target time over certain points, etc.).

- **Input:** Potential conflicts within Executive Controller's AoR and flight's planned constraints
- **Output:** Conflict resolution strategies and ATC clearances
- **ATM Master Plan Automation Levels Task Breakdown:** Information analysis, decision and action selection

4. Clearances implementation

This task refers to the implementation of the selected ATC clearances in the corresponding ATC control system.

- **Input:** Conflict resolution strategies and ATC clearances
- **Output:** De-conflicted trajectories for the aircraft within the Executive Controller's AoR
- **ATM Master Plan Automation Levels Task Breakdown:** Action implementation

5. Conformance Monitoring

This function allows the identification of any possible deviation of each aircraft trajectory with regards to a given ATC clearance.

- **Input:** Flight trajectories (radar track) and ATC clearances
- **Output:** Conformance monitoring alerts
- **ATM Master Plan Automation Levels Task Breakdown:** Information acquisition and information analysis

Taking into account the functions identified the following table depicts the allocation of tasks between the human and the machine for the CD&R (Executive Controller) Use Case.

CDR (Executive) Functions / Tasks	Automation Level 1	Automation Level 2	Automation Level 3
Assessment of the planned and desired trajectory profile	Human	Machine	Machine
Identification of potential conflicts	Machine	Machine	Machine
Identification of conflict resolution strategies and clearances proposal	Human	Machine	Machine
Clearances implementation	Human	Human	Machine
Conformance Monitoring	Machine	Machine	Machine

Table 3. TAPAS CD&R (Executive) Initial Functional Roadmap

Overall, the functional requirements identified for the CD&R use case are as follows:

- The executive controller shall be provided with potential encounters between the aircraft within his area of responsibility for a 7-10-minute look-ahead timeframe.
- The executive controller shall be provided with conflict resolution strategies for the encounters identified between the aircraft within his area of responsibility.
- The executive controller shall be provided with ATC clearances implementation options to solve the identified conflicts.
- The executive controller shall be provided with the ATC clearances required to comply with the agreed exit sector conditions.
- The executive controller shall be able to implement the ATC clearances given to the flight Cabin Crew.
- The executive controller shall be provided with conformance monitoring alerts, indicating deviations of the flight trajectory compared to the given ATC clearances.
- The executive controller shall be provided with information about the flight desired profile (including radiofrequency and datalink requests), flight constraints and flight actual trajectory.

3.4 Data Sets

This section describes the data sets exploited for the CD&R case. In the next subsections, we briefly describe the data sets, the data pre-processing process and the generated data sets.

Detailed specifications of files' structure is provided in Appendix A, in the order in which data sets are presented here.

3.4.1 Flight Plans

Temporal coverage: entire 2019
Spatial coverage: Iberian Peninsula

Size on disk: 33.4GB
Distinct FPKeys: 1328640
Distinct RTKeys: 1328848
minAlt: 100.0
maxAlt: 55000.0

This data set reports the flight plans submitted for the region over Iberian Peninsula during 2019. Each record provides information about the flight and information per waypoint of the trajectory. The version of the flight plan is identified by the field "flightPlanInstant". The data is provided in a CSV format.

3.4.2 Radar Tracks

Temporal coverage: entire 2019
Spatial coverage: Iberian Peninsula and Canaria Islands (spatially covers the entire Flight Plans data set)
Size on disk: 183.2GB
Distinct FPKeys: 892027
Distinct RTKeys: 892323

This data set provides information about flights during 2019 over the Iberian Peninsula. Each record reports the position (longitude, latitude, altitude and time) as well as other information (departure, destination, callsign, etc) for each flight. The data is provided in a CSV format.

3.4.3 ATCo Events

Temporal coverage: entire 2019
Spatial coverage: Iberian Peninsula
Size on disk: 343 MB

The data set provides ATCo instructions to particular flights to resolve conflicts. This data set is provided in a CSV format.

3.4.4 Training and testing data sets.

For the purpose of training and testing the algorithms, datasets have been split into scenarios. Each training or test scenario comprises flights crossing an AoR and downstream sectors in a specific time period. The computation of these scenarios, except the data pre-processing step, is not part of the integrated system prototype pipelines, since data to be used by the system is provided by an external operational platform (SACTA).

The process that generates the training and testing data sets per scenario comprises the following phases:

First, the **data pre-processing step** filters the input data by the region and time period specified, and enriches trajectories with spatial and temporal information regarding sectorizations. This step organizes data appropriately, so that the time demanding operations in subsequent stages is reduced.

The second and third steps involve the **generation of training and testing data sets**. These steps take as input the results of the pre-processing step, and they can run independently or concurrently, if needed.

3.4.5 Data Pre-processing

This step exploits information regarding sector configuration for the scenario time period, to compute spatio-temporal relations between recorded trajectory positions and sectors. Specifically, it uses the sector configuration to enrich a trajectory position with the sector identification covering the position at the reported time.

The objective of this step is twofold: a) to reduce the on-disk reading operations and processing time, given that any sector configuration for an AIRAC involves several files that need to be joined before any spatio-temporal relation is evaluated, b) to derive adjacent sectors directly from the enriched trajectory positions without evaluating spatio-temporal relations between the complex 3D geometries of sectors.

Finally, this step computes the entry/exit time points for each sector crossed by each flight. Doing so, it reduces the processing time required for detecting flights that cross the same sector concurrently and identifies their downstream sectors.

The above process results to a file structure where the date (year and month) names the root folder which contains one folder for each Radar Track Key (RTKey). In the RTKey folder, a file is generated for each version of the flight plan reported for the specific flight (identified by the RTKey). We keep track of the versions of the flight plans by adding the timestamp of the *Flight Plan instant* in the name of each generated file, concatenated with the Flight Plan key.

The same method applies on the records of Radar Tracks data set, to include in each folder named by an RTKey, in the above-mentioned file structure, a file that contains updates of aircraft positions reported in the Radar Tracks data set.

Finally, from the given data sets we construct a table which relates the Flight Plan Key and Radar Track Key with the callsign and date (reported with the callsign).

3.5 Overall CD&R Prototype

Figure 6 provides the overall CD&R prototype.

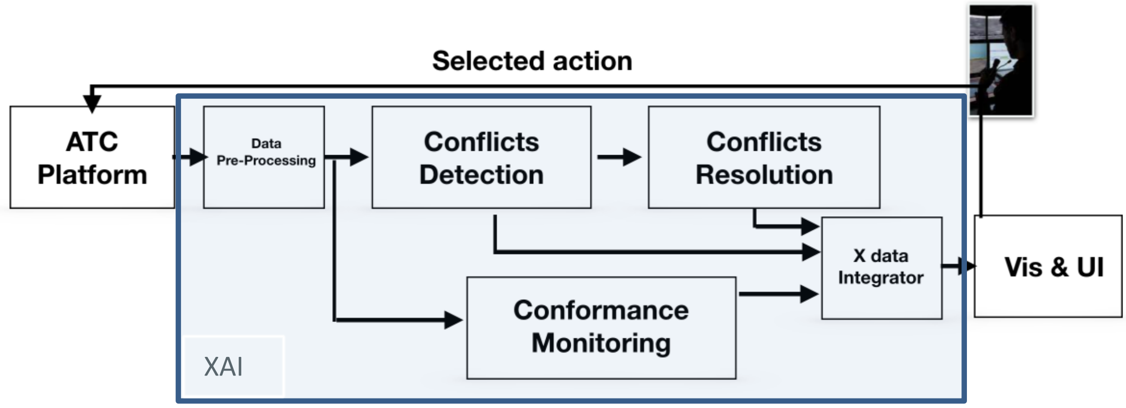


Figure 6. Overall CD&R Prototype System

The CD&R integrated prototype comprises the operational ATC platform SACTA and the XAI system, together with the visualization and user interface (Vis&UI) component, all integrated into a whole. Technical details on the integration of components is provided in Appendix B.

Specifically, the ATC platform provides in operational mode updates of radar tracks and flight plans for flights within an AoR. These data updates are provided at every P seconds (currently set to P=30sec), which is a system configuration parameter, together with the AoR and all related downstream sectors. The platform receives as input the air-traffic controllers' instructions for the resolution of conflicts.

Data provided (radar tracks and flight plans) are according to the formats specified for the data sources in Section 3.4. This data is processed at the data pre-processing component and are forwarded to the conflicts detection and conformance monitoring component. These two components are jointly referred as traffic simulation component. Parameters describing the conflicts detected and non-conformance events, together with data describing the conflicts and non-conformance, are provided in the data integrator component. The conflicts resolution component is the XAI/ML component that decides on the ATC clearances (instructions) for the resolution of conflicts detected. These instructions, together with their effects, as assessed by the system, are provided to the data integrator. The data integrator gathers output data from these three components and provides this to the Vis & UI component.

It must be noted that in Automation level 2 the system provides per conflict detected all potential instructions per flight (including the "no action") ranked according to their potential (as assessed by the system) to resolve the conflict. At this level of automation, the air traffic controller may choose one of these actions to instruct a flight. At automation level 3 the system instructs the flights to execute actions with no human intervention, on its own initiative.

It must be noted that the CD&R prototype does not have any distinct explainability component for providing explanations of instructions decided by the system. Thus, here we do follow a different paradigm from the one followed in the ATFCM case: All relevant parameters that drive system's decisions are provided, offering more transparency on decision making (i.e., making transparent the situations the system considers and foresees), focusing on operational concerns, rather than explainability on **how** decisions are taken from an AI/ML model, driven from purely technical/engineering concerns.

Data gathered by the data integrator component and provided to the Vis&UI component are specified in Figure 7 and are described in detail in the following subsections, w.r.t. the system component that creates any data portion. Figure 7 specifies the components generating the data. All the component specifications in the following subsections, refer to Figure 7, specifying each of the components' input/output.

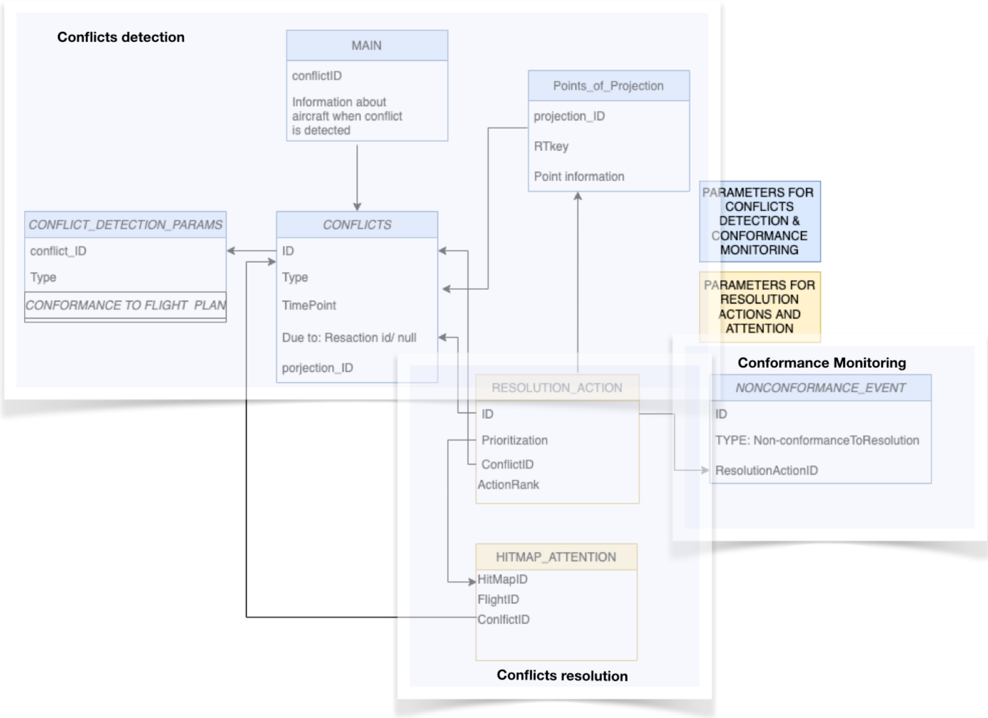


Figure 7. Data provided by the XAI (CD&R Use case)

Appendix B provides technical information on the integration of components into a functional whole.

Subsequent paragraphs provide an overview of the data specified in Figure 7, before delving into the details of that data, w.r.t. the functional specification of components generating it.

Overall, the main entity is the conflict (specified in CONFLICTS). This is related to the specific aircraft involved in that conflict (as specified in MAIN), as well as to (a) the parameters involved for the detection of that conflict (specified in CONFLICT_DETECTION_PARAMS) and to (b) the specific future projections of flight trajectories that drive the detection of the conflict (specified in Points_of_Projection). These are provided by the conflicts detection component.

In case a conflict is assessed to be a side-effect of a potential resolution action (not yet instructed to any of the flights), this conflict is also related to the resolution action that may cause it.

Resolution actions are ranked according to their assessed potential to resolve a conflict. This and all specific parameters (specified in RESOLUTION_ACTION) related to the conflict to be resolved, as well as future projections of flight trajectories (specified in Points_of_Projection), are provided for all conflicting flights. Any conflict resolution action is related to a hitmap that specifies the attention of the corresponding agent(flight) to all other agents in its neighbourhood (i.e. the flights with which is in conflict). This data is provided by the conflicts resolution component.

Finally, given that a resolution action is instructed to a flight, the system provides an alert for any non-conformance event (non-conformance to the resolution action) for the instructed flight (this is specified in the NONCONFORMACE_EVENT). Non-conformance events are provided by the conformance monitoring component.

3.5.1 Functional Specification of components

The subsections that follow specify per component, the input, the functionality implemented and the output. The structure of the input and output files is described in detail in Appendix A, per component and in the order presented here.

3.5.1.1 Data Pre-processing

The pre-processing component prepares the data that is used by the conflicts detection and conflicts resolution components.

3.5.1.1.1 Input

The pre-processing component receives the following files for each scenario and time instant:

3.5.1.1.1.1 Output Scenario

File name: outputSc.csv

Overview: It contains the information about the sectors crossed according to the flight plan of an aircraft, as well as the entry/exit time per sector, departure/destination airports and callsign, flight plan instant, and flight plan key.

3.5.1.1.1.2 Output Flight

File name: outputWp.csv

Overview: it specifies the waypoints crossed according to the flight plan, including the altitude, estimated time of flying over the waypoint, as well as the waypoint name and coordinates (longitude, latitude). The flight plan key and the flight plan instant are also provided.

3.5.1.1.1.3 Radar Tracks

File name: raderTracks.csv

Overview: it contains the reported positions of each flight according to the radar tracks. This file provides the flight plan key, the coordinates and altitude, as well as the timestamp for each reported position of a flight.

3.5.1.1.1.4 Sector Geometries

File name: sectorGeometries.csv

Overview: it provides the three-dimensional geometries of the active sectors.

3.5.1.1.1.5 Flight Plans

File name: fpList.csv

Overview: provides the list of flights (callsign, departure, destination airport and flight plan key) that are considered.

3.5.1.1.2 Functionality

The functionality of the pre-processing component is specified in Section 3.4.5.

3.5.1.1.3 Output

The pre-processing component generates files in a folder named after the scenario ID, the sector of interest (AoR) and the scenario time instance. Within each folder, and for each flight involved in the scenario, the following files are generated:

3.5.1.1.3.1 Entry/Exit

File name: *.entry_exit

Overview: these files report the entry/exit time instances for each one of the crossed sectors per flight.

3.5.1.1.3.2 Flight Plan

File name: *.fp

Overview: these files contain the waypoints and estimated time of flying over each waypoint for each flight, according to the flight plan.

3.5.1.1.3.3 Flight Phases

File name: *.phases

Overview: these files report the time intervals of each of the three phases for each flight: *climbing*, *cruising* and *landing* phase.

3.5.1.1.3.4 Radar track updates

File name: radar

Overview: this is a single file for each scenario and contains the latest reported updates of aircraft positions for all the flights involved in the scenario.

3.5.1.2 Conflicts Detection Component

3.5.1.2.1 Input

The input for the conflicts detection component comprises the files provided by the pre-processing component.

3.5.1.2.2 Functionality

The functionality of this component fulfils the requirement that the executive controller shall be provided with potential encounters between the aircraft within his area of responsibility for a 7-10-minute look-ahead timeframe. It computes and exploits the CPA geometry between pairs of flights (shown in Figure 5) to detect conflicts.

3.5.1.2.3 Output

The output of the conflict detection component provides information about the following events: conflicts, alerts and losses of separation. As losses of separation we consider the violation of the separation minima between a pair of aircraft. Conflicts are predicted losses of separation for a 10-minute look-ahead timeframe, and alerts are conflicts where the loss of separation is predicted to occur in less than a 2 minutes timeframe.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Specifically, the conflicts detection component generates the following files containing information about detected events (i.e. first and last point of conflict, aircraft participating in conflict, CPA geometry etc.) and how these events are detected (i.e. the projection of the aircraft positions into the future to detect the conflicts).

3.5.1.2.3.1 Main

File name: main.csv

Overview: This file includes data for the MAIN entity specified in Figure 7.

3.5.1.2.3.2 Conflicts

File name: conflicts.csv.

Overview: This file includes data for the CONFLICTS entity specified in Figure 7.

3.5.1.2.3.3 Conflicts Parameters

File name: conflict_params.csv.

Overview: This file includes data for the CONFLICT_DETECTION_PARAMS entity specified in Figure 7.

3.5.1.2.3.4 Trajectory Projection

File name: points_of_projection.csv.

Overview: This file includes data for the Points_of_Projection entity specified in Figure 7.

3.5.1.3 Conformance Monitoring Component

3.5.1.3.1 Input

The input for the conformance monitoring component comprises the files provided by the external platform (updates on radar tracks and flight plans), the positional information of the conflicting aircraft and the ATCO instructions to the conflicting flights.

3.5.1.3.2 Functionality

The functionality of this component fulfils the requirements that (a) the executive controller shall be provided with conformance monitoring events, indicating deviations of the flight trajectory compared to the given ATC clearances, and (b) the executive controller shall be provided with information about the flight desired profile, flight constraints and flight actual trajectory. It must be noted that conformance monitoring is performed per flight and resolution action provided, and only for the update of radar tracks and flight plans given from the external platform at the time point next to a resolution action for each flight.

3.5.1.3.3 Output

3.5.1.3.3.1 Conformance Events

File name: conformance.csv

Overview: Each record in this file specifies a non-conformance event detected.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.5.1.4 Conflicts Resolution Component

3.5.1.4.1 Input

The input for the conflicts resolution component comprises the files provided by the conflicts detection component.

3.5.1.4.2 Functionality

The functionality of this component fulfils the following requirements: (a) The executive controller shall be provided with ATC clearances implementation options to solve the identified conflicts. (b) The executive controller shall be provided with the ATC clearances required to comply with the agreed exit sector conditions. (c) The executive controller shall be provided with conflict resolution strategies for the encounters identified between the aircraft within his area of responsibility.

3.5.1.4.3 Output

3.5.1.4.3.1 Resolution Actions

File name: resolution_actions_episode_1.csv

> Overview: Each record of this file provides information about a resolution action proposed by the AI/ML model for any conflicting flight. It includes data for the RESOLUTION_ACTION entity specified in Figure 7.

3.5.1.4.3.2 HITMAPS

File name: conflict_hitmaps_episode_1.csv

> Overview: Each record of this file contains information about the attention hitmap of a flight at a specific timepoint w.r.t to the other flights in which it is in conflict. It includes data for the HITMAP_ATTENTION entity specified in Figure 7.

3.5.1.5 Visualization (explanation logic) & User Interface Component

This component will be henceforth referred to as TAPAS CDR UI. The purpose of TAPAS CDR UI is to inform the users in real time about conflicts between flights that have been predicted by the XAI component and present the conflict resolution actions proposed by the XAI component.

3.5.1.5.1 Input

The input of this module comprises all files provided by the conflicts detection, conflicts resolution and conformance monitoring components. TAPAS CDR UI arranges the information extracted from the files into several complementary views, which show

- a list of recently detected conflicts with their components and attributes,
- a graphical representation of the geometry of a single selected conflict,
- a list of conflict resolution actions (for a single selected conflict) proposed by the XAI component.

The views are put together in the main window of the TAPAS CDR UI; an example is shown in Fig. 15. The components of the TAPAS CDR UI are described in detail in the deliverable D4.3 “Visualisations and Visual Analytics Methods”.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.5.1.5.2 Functionality

The functionality of this component fulfils the following requirements: (a) The executive controller shall be able to implement the ATC clearances given to the flight Cabin Crew. (b)The executive controller shall be provided with conformance monitoring conflicts and alerts, indicating deviations of the flight trajectory compared to the given ATC clearances. (c) The executive controller shall be provided with information about the flight desired profile, flight constraints and flight actual trajectory.

TAPAS CDR UI consists of a main window, which is demonstrated in Fig. 15, and two additional windows, which appear on demand. The top part of the main window contains a table with the list of conflicts, the graphical view is in the lower left part, and on the right of it is a table with the list of actions.

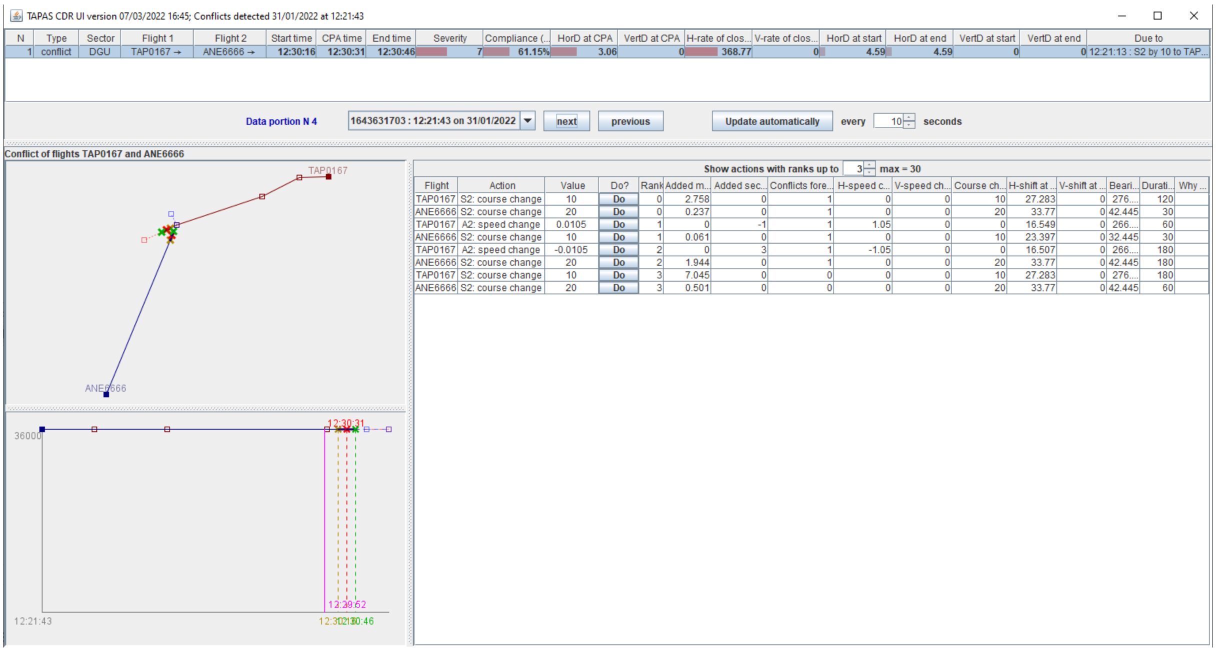


Figure 8. User interface of the VA component for the CD&R use case.

In the conflict description table (Figure 8, top), each row describes one conflict. When no conflicts have been detected, the table is empty, and a message about the absence of conflicts is displayed below the table. For each detected conflict, the following data are shown in the table columns:

- Event type: conflict, alert, or loss (i.e., loss of separation)
- Sector in which the conflict is expected to happen
- Callsigns of the two flights involved in the event followed by one of the symbols ↗, →, or ↘ indicating, respectively, the climbing, cruising, and descending flight phases.
- Start time of the conflict.
- Time of the CPA (Closest Point of Approach).
- End time of the conflict.
- *Compliance* measure, i.e., the extent (expressed in %) to which the situation complies with the separation requirement, so that 100% means full compliance, i.e., a safe situation, and 0 indicates the most critical situation.
- Horizontal and vertical distances between the flights at the CPA. The horizontal distance is measured in nautical miles (NM) and the vertical distance in feet.

- Horizontal rate of closure, i.e., the relative speed with which the horizontal distance between the flights decreases. This rate is measured in knots, i.e., nautical miles per hour.
- Vertical rate of closure, i.e., the relative speed with which the vertical distance between the flights decreases. This rate is measured in feet per minute.
- *Severity* score, a value from 0 (not critical) to 15 (most critical). It is based on potential safety infringement classification, considering the minimal distance reached and the duration of the separation loss. The severity score is derived as a sum of two scores, one of which is based on the Measure of Compliance (MOC) and the other on the horizontal or vertical rate of closure (more details on the calculation of the severity score are provided in TAPAS deliverable D4.3 "Visualizations and Visual Analytics Methods").
- Horizontal and vertical distances (in nautical miles and feet, respectively) at the start and end time moments of the conflict event.
- The column "Due to" indicates whether the conflict is going to emerge because of non-conformance of one of the flights to a resolution action that they were required to fulfil according to an earlier taken decision of the air traffic controller. In this case, the column contains information about the action that has been violated.

A sub-window with a graphical illustration of a conflict is positioned on the left below the table with conflict descriptions. The graphical view represents the conflict corresponding to the "active" row of the conflict description table, i.e., the row that is currently highlighted in the table of conflict descriptions. The graphical view (Figure 8, lower left) consists of two parts separated by a horizontal movable divider. The upper part represents a lateral projection, in which the display dimensions correspond to the geographic longitude and latitude. Dot symbols (small squares) and lines represent flight positions and paths. Cross marks represent the positions of the flights at the moments of conflict start (these crosses are coloured in dark yellow, or brownish), CPA (red), and conflict end (green). The lower part of the graphical view contains an altitude plot, where the horizontal axis represents time and the vertical axis the flight altitude. The changes of the flight altitudes over time are represented by lines. When the flight altitude does not change, the corresponding line is horizontal. The graphical encoding is consistent with the encoding in the lateral view.

The sub-window on the right below the conflict description table contains a table describing the conflict resolution actions proposed by the XAI component. The XAI component usually proposes multiple possible actions, which are ranked by preference, so that rank 0 indicates the most preferable action while higher values correspond to less preferable actions. Each action refers to one of the two flights involved in the conflict. The actions are ranked separately for each flight; therefore, there are pairs of actions having the same rank. A text field above the table can be used to set the desired maximal rank of the shown actions, which means that actions with larger ranks, if any, will be hidden.

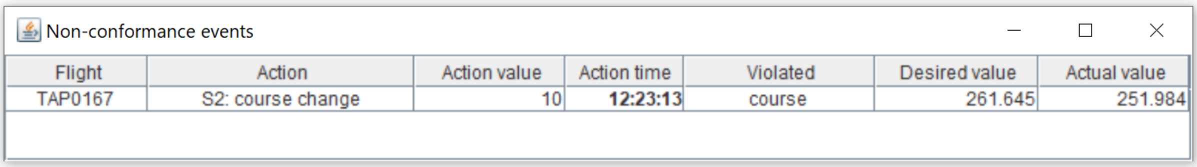
The columns of the action table present the following information:

- Callsign of the flight that will need to fulfil the action.
- Action type.
- Action value, the meaning of which depends on the action type.
- Rank: 0, 1, 2, and so on.
- Added miles: change of the flown distance due to the action.
- Added seconds: change of the flight duration due to the action.
- Conflicts foreseen: number of conflicts that will remain or emerge after applying the action.

- H-speed change and V-speed change: changes of the horizontal and vertical speeds, the former represented as Mach number and the latter in feet per minute.
- Course change: angle with respect to the current course, in degrees.
- H-shift at exit and V-shift at exit: the relative position of the point of exit from the sector with respect to the previously planned point of exit. The horizontal shift is measured in metres and the vertical shift in feet.
- Bearing: the bearing of the flight after fulfilling the action.
- Duration: duration of the resolution action, in seconds.
- Why not: when some action is theoretically possible (according to the AI model, which has been derived from historical data) but cannot be performed in the current situation, this field contains a description of the reason, for example, “Vertical speed cannot be increased because it will exceed 60.0 feet/s”.

Apart from the information columns, there is a special column containing buttons with the text “Do”. Pressing on such a button means selecting of the corresponding action to be fulfilled. In response, the VA component displays a dialog window asking for a confirmation that the chosen action is going to be applied to the corresponding flight. If the user confirms the decision, a message about the chosen action is sent to the XAI component.

When the field “Conflicts foreseen” in a row of the table with the resolution actions contains a value greater than zero, clicking on this row creates an additional window showing information about the secondary conflict(s) that will remain or appear after the action is fulfilled. This additional window is similar to the main window but does not include a table with resolution actions.


Non-conformance events

Flight	Action	Action value	Action time	Violated	Desired value	Actual value
TAP0167	S2: course change	10	12:23:13	course	261.645	251.984

Figure 16. An example of a window with information about non-conformance events.

A window containing a table with non-conformance events (Figure 16) appears when such events are detected by the XAI component. A non-conformance event means that a flight does not fulfil some conflict resolution action that the air traffic controller has chosen earlier. In this case, the table shows the flight that does not conform to the controller’s decision, the action that is not fulfilled, the time when it was supposed to be done, what parameter is violated, i.e., the value differs from the required (e.g., "horizontal speed", "vertical speed", "course"), the desired value of this parameter, and the actual value. The window disappears when no non-conformance events are detected in the next step of the monitoring process.

Hence, the functionality of TAPAS CDR UI is to present the outputs of the XAI component to the user and allow the user to select one of the proposed resolution actions to be fulfilled. When the user does this, TAPAS CDR UI sends a message about the action selection to the XAI component.

3.5.1.5.3 Output

The output of TAPAS CDR UI consists of messages that are sent to the XAI component when the user takes a decision to apply one of proposed conflict resolution actions. A message is a text consisting of the following fields:

- Time of receiving the last input from the XAI, specified in Unix seconds.

- Identifier of the chosen action.
- Identifier (callsign) of the flight to which the action is applied.
- Action type.
- Action value.
- Time of the last input transformed to calendar date and time.

4 Conclusions

This document is a complete technical reference for the integrated prototype systems addressing the requirements towards explainable Artificial Intelligence methods for (a) the ATFCM use case and (b) the CD&R use case, in the context of the TAPAS project.

These prototype systems have been used during the TAPAS validation exercises performed in the ATFCM and in the CD&R use cases, providing solutions and explainability functionality to operators, with respect to the functionality and explainability requirements specified in WPs 2 and 3. Validation experiments provide evidence to the functionality of these prototypes, as implemented and described here.

For each of the TAPAS use cases, the document describes in detail the use case and the roadmap of the functionalities decided for this use case, up to realizing the needs for implementing automation level 3. Then it describes the overall integrated system architecture per use case and specifies in detail the functionality of individual components and components' interactions.

While the main part of the document describes the TAPAS integrated prototypes at a first level of detail, the Appendix A of this document provides more details on the specific format of files required as input and provided as output per system component.

Appendix B provides technical details concerning the integration of components via the exchange of messages, as it is implemented.

EUROPEAN PARTNERSHIP

Co-funded by
the European Union

Appendix A

In this section we present in detail the data sets involved in the use cases. We focus on the fields of each data set that are necessary for the tasks to be carried out in the context of TAPAS. Considering the volume of information transferred between the modules of the ATFCM and CD&R prototypes, we specify the minimum necessary data retrieved from any data source.

ATFCM Use Case Data Sets

Flight Plans

This data set provides the data about intended (FTFM), regulated (RTFM), and actual (CTFM) flight trajectories. It has worldwide spatial coverage for a complete year (2019). Data is provided in a compressed archive (7z compression) for each month (varying between 3 to 5GB per month). The compressed data set for entire 2019 is approximately 48.5GB. Each compressed archive contains a set of CSV files, one for each day of the archived month. The CSV files are in the ALLFT+ format, described in “DDR2 Reference Manual For General Users 2.9.7”.

The fields of this data source that are involved in the data enrichment task, are the following:

Departure airport (ADEP) at field 1
Destination airport (ADES) at field 2
Aircraft Callsign at field 3
Aircraft model at field 5
Last off-block time (LOBT) at field 19
Flight plan state at field 20
FTFM point profile at field 86
FTFM crossed sectors at field 88

The combination of fields 1,2,3,19 uniquely identify each flight. We use these fields to construct a key for each flight for joining with other data sets if needed and for future reference. The field 20 is used for filtering the submitted flight plans, so as to process only those that are marked to be final (i.e. value “TE”). Finally, field 86 contains the data for constructing the geometry of trajectories and computing the spatio-temporal relations with active sectors (e.g. points of entering/exiting an active sector), while field 88 is useful for validating the enriched trajectory (i.e. when no regulation on the FTFM profile is applied).

In addition to the above, we need also to further identify the trajectory segments of type one of {“take-off”, “en-route” and “landing”} per trajectory, according to the altitude changes between consecutive positions of the trajectory. The identification of trajectory segments, enables different measures to be applied w.r.t. the trajectory segment type crossing each sector. This information is not provided in the provided datasets, and it is something that is computed at the data pre-processing stage.

Sector Configuration

This data set describes the airspace configuration, i.e. the airspace volumes composed by 3D “airblocks”, as well as their capacities, and activation intervals. The data set spatially covers the whole world, however it is more detailed in the European airspace. The first sample of the data source that have been provided, temporally covers AIRAC 1908, from Wednesday, July 17, 2019 9:00:00 PM UTC to Wednesday, August 14, 2019 8:59:00 PM UTC.

This data source comprises a set of CSV files containing information about airspace configurations for specific time intervals, sectors and volumetry information, sector flight levels and information about the airspace capacity in a certain period of time. All this information in DDR is organised in the following files:

- **Configurations file (*.cfg):** This file contains the information related to airspace configuration and its filename extension is .CFG. It comprises the following fields:
    - **Field 1:** ID name of the group within which all sectors of any configuration must be bounded, generally an ACC
    - **Field 2:** name of the configuration, generally a number or a string with a number indicating the number of sectors of the configuration
    - **Field 3:** ID name of elementary sector or collapse sector can be used here

```
LEMDTMA;CNF10N;LEMDAPN
LEMDTMA;CNF10N;LEMDDEN
LEMDTMA;CNF10N;LEMDDWN
LEMDTMA;CNF10N;LEMDENN
LEMDTMA;CNF10N;LEMDESN
LEMDTMA;CNF10N;LEMDREN
LEMDTMA;CNF10N;LEMDRWN
LEMDTMA;CNF10N;LEMDWNN
LEMDTMA;CNF10N;LEMDWSN
LEMDTMA;CNF10S;LEMDAPS
LEMDTMA;CNF10S;LEMDDES
LEMDTMA;CNF10S;LEMDDWS
LEMDTMA;CNF10S;LEMDENS
LEMDTMA;CNF10S;LEMDESS
LEMDTMA;CNF10S;LEMDRES
LEMDTMA;CNF10S;LEMDRWS
LEMDTMA;CNF10S;LEMDWNS
LEMDTMA;CNF10S;LEMDWSS
```

Figure 9. ATFCM use case: Sector configurations file . Data sample.

- **Opening scheme file (*.cos):** This file contains information of the airspace configuration applied in a certain period of time (active sectors) with the following fields:
    - **Field 1:** Date as DD/MM/YYYY
    - **Field 2:** ID name of the group within which all sectors of any configuration must be bounded, generally an ACC
    - **Field 3:** "HH:MM", configuration activation time beginning
    - **Field 4:** "HH:MM", configuration activation time ending
    - **Field 5**: name of the configuration opened, generally a number or a string with a number indicating the number of sectors of the configuration
    - **Field 6:** source of info.

```
15/05/2004;LEBLAPP;00:00;05:29;CONF1;E
15/05/2004;LEBLAPP;05:30;19:29;CONF4W;E
15/05/2004;LEBLAPP;19:30;23:59;CONF1;E
15/05/2004;LECBACC;00:00;05:29;CNF2;E
15/05/2004;LECBACC;05:30;12:29;CNF10H;E
15/05/2004;LECBACC;12:30;19:29;CNF9D;E
15/05/2004;LECBACC;19:30;20:29;CNF8J;E
15/05/2004;LECBACC;20:30;21:59;CNF5;E
15/05/2004;LECBACC;22:00;23:59;CNF2;E
15/05/2004;LECLAPP;00:00;04:59;CONF1;E
15/05/2004;LECLAPP;05:00;07:29;CONF2A;E
15/05/2004;LECLAPP;07:30;19:29;CONF3A;E
15/05/2004;LECLAPP;19:30;21:59;CONF2A;E
15/05/2004;LECLAPP;22:00;23:59;CONF1;E
```

Figure 10. ATFCM use case: Opening scheme file. Data sample.

- **Capacity file (*.ncap):** This file contains information of the airspace capacity in a certain period of time with the following fields:
  - **Field 1:** Date as DD/MM/YYYY
  - **Field 2:** Element on which the capacity is applied.
  - **Field 3:** start time, format is HH:MM
  - **Field 4:** ending time, format is HH:MM
  - **Field 5**: capacity of the element. Meaning “999” that it has not a defined capacity
  - **Field 6:** Hourly 3 capacity, not used, always “_”
  - **Field 7**: type of the element:
    - AD = Aerodrome
    - AS = Airspace
    - TV = Traffic Volume
    - SP = Significant Point
    - AZ = Set of aerodromes
  - **Field 8:** category of the element
    - G = Global, for AS, TV and SP
    - A = Arrival for AD and AZ
    - D = Departure for AD and AZ
  - **Field 9:** Source table
    - B = Body/default
    - P = Permanent
    - T = Temporary
    - N = Tact

```
31/03/2016;LEJR;00:00;23:59;999;_;AD;D;B
31/03/2016;LEJRALL;00:00;23:59;12;_;TV;G;B
31/03/2016;LEJRARR;00:00;23:59;8;_;TV;G;B
31/03/2016;LEJRDEP;00:00;23:59;8;_;TV;G;B
31/03/2016;LEJU;00:00;23:59;999;_;AD;G;B
31/03/2016;LELA;00:00;23:59;999;_;AD;G;B
31/03/2016;LELC;00:00;23:59;4;_;AD;G;B
31/03/2016;LELC;00:00;23:59;4;_;AD;A;B
31/03/2016;LELCARR;00:00;23:59;0;_;TV;G;B
31/03/2016;LELL;00:00;23:59;999;_;AD;G;B
31/03/2016;LELM;00:00;23:59;999;_;AD;G;B
31/03/2016;LEMD;00:00;19:59;78;_;TV;G;B
31/03/2016;LEMD;00:00;23:59;999;_;AD;D;B
31/03/2016;LEMD;00:00;23:59;999;_;AD;G;B
31/03/2016;LEMD;00:00;23:59;999;_;AD;A;B
```

Figure 11. ATFCM use case: Capacity file. Data sample.

- **(Volumetry) Airspace file (*.spc):** This file contains information of the airspace, its id, the number of sectors, and the name of each of the sectors that compose it. It describes how elementary sectors can be collapsed, and can also contain description of other bigger airspace (like AUA, etc). Information is provided using the following fields:
    - **Header line:**
        - **Field 1**: type of element, must be “A” from Airspace
        - **Field 2:** name of the airspace, ID.
        - **Field 3:** free text describing the airspace, can be empty
        - **Field 4:** type of Airspace, can be CS for collapsed sector, AUA, AUAG, RSA…
        - **Field 5:** number of items composing the airspace (number of body lines)
    - **Body**:
        - **Field 1:** type of element, must be S for Sector
        - **Field 2:** name of the airspace, ID
        - **Field 3:** type of airspace, ES for elementary sector, could be AUA or other

```
A;ENLECP;SPAIN PALMA;AUAG;4;_
S;LECPCTA;AUA
S;LEIBCTR;AUA
S;LEMHTMA;AUA
S;LEPATMA;AUA
```

Figure 12. ATFCM use case: Airspace file. Data sample.

- **(Volumetry) Sector Gasel level file (*.gsl/sls):** This file contains information of the airspace sector configuration, its id, the number of airblocks, the name of each of the airblocks that compose it and its boundary levels.
    - **Header line:**
        - **Field 1:** type of element, S from Sector
        - **Field 2:** Sector ID
        - **Field 3:** Sector name
        - **Field 4:** Number of airblock (number of body lines)
        - **Field 5:** Airspace category
        - **Field 6:** Sector type.
            - FIR = Flight Information Region
            - ERSA = Elementary Restricted Airspace

            - ES = Elementary Sector
            - ERAS = Regulated AS category “FRA” or “DCT”
    - **Body:**
        - **Field 1:** type of element, A from Airblock
        - **Field 2:** Airblock name
        - **Field 3:** Operation, always +
        - **Field 4:** Lower FL
        - **Field 5:** Upper FL

```
S;LECMASL;MADRID SECTOR ASTURIAS LOWER;11;_;ES
A;001LP;+;105;345
A;160LE;+;0;285
A;201LE;+;245;345
A;221LE;+;155;345
A;260LE;+;0;345
A;261LE;+;0;345
A;262LE;+;145;345
A;269LE;+;145;345
A;270LE;+;0;345
A;550LP;+;245;345
```

Figure 13. ATFCM use case: Sector Gasel level file. Data sample.

- **(Volumetry) Airblock file (*.gar/are):** This file contains information of the airblock configuration, its id, the number of waypoints, the name of each of the waypoints that compose it, specifyin their latitude and longitude.
    - **Header line:**
        - **Field 1:** type of element, must be A from Airblock
        - **Field 2:** name of the airblock, ID
        - **Field 3:** number of items (vertices) composing the airblock (number of body lines)
    - **Body:**
        - **Field 1:** must be P from Point
        - **Field 2:** latitude, in degrees decimals
        - **Field 3:** longitude, in degrees decimals

```
A;001LP;6
P;41.9416666666667;-6.625
P;41.9333333333333;-6.56666666666667
P;41.7697222222222;-6.49944444444444
P;41.9363888888889;-7.11694444444444
P;41.9366666666667;-7.11638888888889
P;41.9416666666667;-6.625
```

Figure 14. ATFCM use case: Airblock file. Data sample

Since the lower/upper flight levels vary within a sector, it is required to also have the set of 3D airblocks composing each sector. Ideally, we can reduce the volume of data transferred if each collapsed sector (CS) is directly associated with the airblocks it contains, i.e. without considering any intermediate elementary sectors to retrieve the building blocks of a collapsed sector.
The above data can be provided in 4 CSV files as follows:

a) The airblock file reporting only airblocks within the area of responsibility (LETOT),

b) The Sector Gasel Level file, reporting only Elementary Sectors within the area of responsibility
c) The Airspace.spc file, reporting only Collapsed Sectors within the area of responsibility. Sectors in this file can be directly associated to the airblocks they contain, instead of the intermediate Elementary Sectors.
d) Capacity intervals for Collapsed and Elementary sectors provided in the above files. The capacity intervals provided, have to overlap with activity intervals of the sectors. This filter will guarantee that only capacities of sectors that can be used will be provided.

ATFCM Use Case Prototype File Structures

AI/ML Solutions Component Output

Solution

File name: scenario_*_exp**_solution.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- Delays: The minutes of delay decided by our methods (starting from zero, up to maximum delay, 100 for the time being). (This column does not exist for SolutionID 1)
- FlightCapping: The level capping regulation decided by our methods. NULL if no capping regulation was applied.
- Sectors: A series of columns starting from Sector_0 and continuing up to Sector_15. Each column contains the sector IDs crossed by the final flight plan (in the order it is crossed). The number of columns is subject to change, if a scenario includes a flight that crosses more than 16 sectors.
- EntryTimes: A series of columns starting from EntryTime_0 and continuing up to EntryTime_16. Each column contains the times of entry in the corresponding sector column. The number of columns is subject to change, if a scenario includes a flight plan that crosses more than 16 sectors. Please note that these series contain an extra column, signifying the time of landing or exit from the final sector in LETOT.

Baseline Flight Plans

File name: scenario_*_exp**_baseline_flight_plans.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- Delays: 0-maximum delay, all possible flight plans. (This column does not exist for SolutionID 1)
- FlightCapping: The level capping regulation decided by our methods, in the form of the sector(s) which the flight is prohibited to enter. NULL if no capping regulation was applied. In the case a regulation is applied it is described by a series of sectors separated by " ".
- Sectors: A series of columns starting from Sector_0 and continuing up to Sector_17. Each column contains the sector IDs crossed by the final flight plan (in the order it is crossed). The number of columns is subject to change, if a scenario includes a flight that crosses more than 18 sectors.
- EntryTimes: A series of columns starting from EntryTime_0 and continuing up to EntryTime_18. Each column contains the times of entry in the corresponding sector column. The number of columns is subject to change, if a scenario includes a flight plan that crosses more than 18 sectors. Please note that these series contains an extra column, signifying the time of landing or exit from the final sector in LETOT.

Baseline HotSpots

File name: scenario_*_exp**_baseline_hotspots.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- SectorID
- TimeStep: 0, since the file contains only the baseline hotspots. (This column does not exist for SolutionID 1)
- TimePeriod: The time when the period of the hotspot starts.
- Demand: The number of flights that enter the sector during this period.
- Capacity: Sector's capacity.
- SafetyCritical: "MEDIUM" when demand exceeds 110% of capacity.
- Day: Day of the hotspot and of the scenario.
- Duration: Fixed to 60 minutes.
- StartDate: Start date and time of the hotspot in format "yyyy-mm-dd hh:mm".
- EndDate: End date and time of the hotspot in format "yyyy-mm-dd hh:mm".

Baseline HotSpots Flights

File name: scenario_*_exp**_baseline_hotspots_flights.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- SectorID
- TimeStep: 0, since the file contains only flights from the baseline hotspots. (This column does not exist for SolutionID 1)
- TimePeriod: The time when the period of the hotspot starts.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- Duration: The number of minutes the flight stays in the hotspot.
- Flight Phase: The flight phase of the flight (take-off, en-route, landing), placeholder for the time being.
- FlightCapping: NULL, as all the flight plans are unregulated.
- Delay: 0, as all the flight plans are unregulated.

Decisions

File name: scenario_*_exp**_decisions.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- TimeStep: The minute of day in our simulation. Starting from 0 ending to 1439. (This column does not exist for SolutionID 1)
- Decision: 0 for no new action, n for adding n minutes of delay. 0 for no regulation, 1 for level capping regulation in the case of SolutionID 1.
- Greedy: Used for internal purposes. Binary flag showing if the action was greedy. (This column does not exist for SolutionID 1)
- WasAllowedToTakeDelay: Used for internal purposes. Binary flag showing if the action was allowed to add more delay. Flights that do not participate in any hotspots, have already taken off, or have already reached the maximum allowed delay are not permitted to add more. (This column does not exist for SolutionID 1)
- SectionID: The sector responsible for the regulation.
- StartDate: Start time of the hotspot for which the measure is being applied in format "yyyy-mm-dd hh:mm".
- EndDate: End time of the hotspot for which the measure is being applied in format "yyyy-mm-dd hh:mm".

- TotalDelay: Total amount of delay accumulated up to this point, in minutes.
- ADEP
- ADES
- EOBT: Given the total amount of delay accumulated up to this point.
- RegulatedSectors: Only appearing in SolutionID 1. The sectors avoided by the flight due to level capping regulation. Always include the sector of column 7, but can include more sectors.

Demand

File name: scenario_*_exp**_demand.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- TimeStep: The minute of day in our simulation, when the hotspot is created. Starting from 0 ending to 1439. In the case of SolutionID 1 this column is replaced by 'Initial', a binary flag that shows whether the demand concerns the initial situation, or the final one, after level capping regulations are applied.
- SectorID
- TimePeriod: The time when the period starts.
- Demand: The number of flights that enter the sector during this period.

HotSpots

File name: scenario_*_exp**_hotspots.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- SectorID
- TimeStep: The minute of day in our simulation, when the hotspot is created. Starting from 0 ending to 1439. (This column does not exist for SolutionID 1)
- TimePeriod: The time when the period of the hotspot starts.
- Demand: The number of flights that enter the sector during this period.
- Capacity: Sector's capacity.
- SafetyCritical: "MEDIUM" when demand exceeds 110% of capacity.
- Day: Day of the hotspot and of the scenario.
- Duration: Fixed to 60 minutes.
- StartDate: Start date and time of the hotspot in format "yyyy-mm-dd hh:mm".
- EndDate: End date and time of the hotspot in format "yyyy-mm-dd hh:mm".

HotSpots Flights

File name: scenario_*_exp**_hotspots_flights.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- SectorID
- TimeStep: The minute of day in our simulation, when the hotspot is created. Starting from 0 ending to 1439. (This column does not exist for SolutionID 1, where only level capping regulations apply)
- TimePeriod: The time when the period of the hotspot starts.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- Duration: The number of minutes the flight stays in the hotspot.
- Flight Phase: The flight phase of the flight (take-off, en-route, landing), placeholder for the time being.
- FlightCapping: The level capping regulation decided by our methods, in the form of the sector(s) which the flight is prohibited to enter. NULL if no capping regulation was applied. In the case a regulation is applied it is described by a series of sectors separated by “ ”.
- Delay: The minutes of delay decided by our methods (starting from zero, up to maximum delay, 100 for the time being). (This column does not exist for SolutionID 1)

Snapshots

File name: scenario_*_exp**_snapshots.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 2 for combination of both.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- TimeStep: The minute of day in our simulation. Starting from 0 ending to 1439.
- Delay: Total amount of delay accumulated up to this point, in minutes.
- SectorID: The sector the flight is in at the specific timestep. NULL if the flight currently is outside LETOT.
- Departure: Estimated take-off, given the current delay.
- Arrival: Estimated landing, given the current delay.
- prevSectorID
- nextSectorID

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Output of AI to XAI component

SGT Train
File name: kl_train_scenario_*.csv

- *Year, Month, Day of the scenario, for example 20190801

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- AgentID (this corresponds to a flight ID as described in section 5.11)
- State:
  - Delay the corresponding flight has accumulated up to this point
  - Total number of hotspots the corresponding flight participates in
  - A number of columns with the hotspots the corresponding flight participates in. This number is currently 18, but can change to accommodate different scenarios. Sectors here are presented with an id subjective to the flight, corresponding to the sequence of sectors crossed by the trajectory. Specifically, number 0 in this list means that this particular flight participates in a hotspot on the first sector it crosses, 1 at the second sector of its flight plan etc. A specific file is provided to map these ids to the actual sector ids (Section 5.2). The columns without hotspot to show are padded with -1.
  - A number of columns with the periods in which the corresponding flight participates in hotspots. This number is currently 18, but can change to accommodate different scenarios. These columns correspond to the hostpots (point (d), above), meaning that the first number here indicates the period in which the first sector that appears in the previous series has the hotspot. The columns without hotspot are padded with -1.
  - The minute of day the flight takes off given the delay (CTOT).
  - A number of columns indicating the time period in minutes the flight remains in each sector it crosses, with respect to the sequence of sectors crossed by the trajectory. This number is currently 16, but can change to accommodate different scenarios.
- Q0: The prediction of the trained model of the discounted expected reward if no additional delay is added to this state.
- Q1-10: The prediction of the trained model of the discounted expected reward if one to ten minutes of additional delay is added to this state.
- Action: Action decided by the model. Currently a number (0-10) indicating the minutes of additional delay attributed to the flight.

SGT Test
File name: kl_test_scenario_*.csv

- *Year, Month, Day of the scenario, for example 20190801

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- Step: The minute of day in our simulation, when the hotspot is created. Starting from 0 ending to 1439. (This column does not exist in SolutionID 1)
- Greedy: Binary flag showing if the action was greedy.
- WasAllowedToTakeDelay: Binary flag showing if the action was allowed to add more delay.
- AgentID (this corresponds to a flight ID as described in section 5.11)
- State:
    - Delay the corresponding flight has accumulated up to this point
    - Total number of hotspots the corresponding flight participates in
    - A number of columns indicating the hotspots the corresponding flight participates. This number is currently 18, but can change to accommodate different scenarios. Sectors here are presented with an id subjective to the flight, corresponding to the sequence of sectors crossed by the trajectory. Specifically, number 0 in this list means that this particular flight participates in a hotspot on the first sector it crosses, 1 at the second sector of its flight plan etc. A specific file is provided to map these ids to the actual sector ids (see Section 5.10). The columns without hotspot to show are padded with -1.
    - A number of columns with the periods in which the corresponding flight participates in hotspots. This number is currently 18, but can change to accommodate different scenarios. These columns correspond to the hostpots (point (d), above), meaning that the first number here indicates the period in which the first sector that appears in the previous series has the hotspot. The columns without hotspot are padded with -1.
    - The minute of day the flight takes off given the delay (CTOT).
    - A number of columns indicating the time period in minutes the flight remains in each sector it crosses, with respect to the sequence of sectors crossed by the trajectory. This number is currently 16, but can change to accommodate different scenarios.
- Q0: The prediction of the trained model of the discounted expected reward if no additional delay is added to this state.
- Q1-10: The prediction of the trained model of the discounted expected reward if one to ten minutes of additional delay is added to this state.
- Action: Action decided by the model. Currently a number indicating the minutes of additional delay attributed to the flight.

Flight ID Map

File name: scenario_*_ flight_id_map.csv

- *Year, Month, Day of the scenario, for example 20190801

Columns:

- NumericID
- FlightID

Baseline Flight Plans

File name: scenario_*_exp**_baseline_flight_plans.csv

- *Year, Month, Day of the scenario, for example 20190801
- ** integer starting from 0, the experiment ID

Columns:

- SolutionID: same as ** in file name, the experiment ID. 0 for delays, 1 for level capping, 2 for combination of both.
- FlightID: an identifier of each flight composed of the ICAO of origin and destination airport, the aircraft call sign and last off-block time (YYYYMMDDHHmmSS).
- Delays: 0-maximum delay, all possible flight plans. (This column does not exist for SolutionID 1)
- FlightCapping: The level capping regulation decided by our methods, in the form of the sector(s) which the flight is prohibited to enter. NULL if no capping regulation was applied. In the case a regulation is applied it is described by a series of sectors separated by “ ”.
- Sectors: A series of columns starting from Sector_0 and continuing up to Sector_17. Each column contains the sector IDs crossed by the final flight plan (in the order it is crossed). The number of columns is subject to change, if a scenario includes a flight that crosses more than 18 sectors.
- EntryTimes: A series of columns starting from EntryTime_0 and continuing up to EntryTime_18. Each column contains the times of entry in the corresponding sector column. The number of columns is subject to change, if a scenario includes a flight plan that crosses more than 18 sectors. Please note that these series contains an extra column, signifying the time of landing or exit from the final sector in LETOT.

AI/ML Explainability Component Output

State Information

File name: states_scenario_*.csv

- *Year, Month, Day of the scenario, for example 20190801

Columns:

- Solution ID: 0 for delays, 1 for level capping, 2 for combination of both
- Step: The minute of day in our simulation, in cases of solution types 0 and 2. This column does not exist for solution type 1.
- Agent ID (this corresponds to a flight ID as described in section 4.1.2.4)
- State:
  - Delay the corresponding flight has accumulated up to this point. This column does not exist for solution type 1.

  - Total number of hotspots the corresponding flight participates in
  - A number of columns indicating the hotspots the corresponding flight participates. This number is currently 18, but can change to accommodate different scenarios. Sectors here are presented with an id subjective to the flight, corresponding to the sequence of sectors crossed by the trajectory. Specifically, number 0 in this list means that this particular flight participates in a hotspot on the first sector it crosses, 1 at the second sector of its flight plan etc. A specific file is provided to map these ids to the actual sector ids (see Section 4.1.2.5). The columns without hotspot are padded with -1.
  - A number of columns with the periods in which the corresponding flight participates in hotspots. This number is currently 18, but can change to accommodate different scenarios. These columns correspond to the hostpots (point (c), above. The columns corresponding to not hotspot are padded with -1.
  - The minute of day the flight takes off given the delay (CTOT).
  - A number of columns indicating the time period in minutes the flight remains in each sector it crosses, with respect to the sequence of sectors crossed by the trajectory. This number is currently 18, but can change to accommodate different scenarios. A specific file is provided to map the sector of each column to the actual sector ids (see Section 4.1.2.5). Namely, each sector id corresponds to the sector id minus 1 of the respective flight in Section 4.1.2.5.
- A number of columns indicating the prediction of the AI/ML model regarding the Q value of any possible action (regulation to be taken).
- Action: Action decided by the AI/ML model (according to the solution).
- A number of columns indicating the Q values of any possible action (regulation to be taken) predicted from the SGT.
- Action_real: The action the AI/ML model would had given, if only the Q values were considered. This is used only to validate the accuracy of the predictions made by the SGTs.
- Action_pred: The action as predicted by the SGTs. This is used only to validate the accuracy of the predictions made by the SGTs.

Explanations Q

File name: explanationsQX*.csv

X: for solutions of type 0 and 2, X is in [0,10],and for solutions of type 1 X is in {0,1}.
*Year, Month, Day of the scenario, for example 20190801

Columns:

A number of columns which is equal to the path length followed in the corresponding SGT corresponding to an action 0 given the corresponding state (specified **in the same row** in State Information as described in 4.1.3.3.1). Each cell specifies a state feature (i.e. a feature of the State, as these features are provided by the State Information in Section 4.1.3.3.1).

Co-funded by the European Union

Corresponding state features' thresholds are specified in 4.1.3.3.3.

Explanations Q Thresholds

File name: explanationsQXSplit_*.csv

- X: for solutions of type 0 and 2, X is in [0,10], and for solutions of type 1 X is in {0,1}. *Year, Month, Day of the scenario, for example 20190801

Columns:

A number of cells with numeric values considered as thresholds for the features in Explanations Q.

Extracted Explanations

File name: xaiQX*.csv

- X: for solutions of type 0 and 2, X is in [0,10], and for solutions of type 1 X is in {0,1}. *Year, Month, Day of the scenario, for example 20190801

Columns:

- Solution ID: 0 for delays, 1 for level capping, 2 for combination of both
- Step: The minute of day in our simulation, in cases of solution types 0 and 2. This column does not exist for solution type 1.
- Agent ID (this corresponds to a flight ID as described in section 4.1.2.4)
- Flight ID
- Action: Action decided by the AI/ML model (according to the solution).
- Q: The Q value predicted by the SGT for the action X.
- A number of columns containing the explanations as described in Extracted Explanations (Section 4.1.3.2).

CD&R Use Case Data Sets

Flight Plans

This data set reports the flight plans submitted for the region over Iberian Peninsula during 2019. The data is provided in a CSV format, comprising the following columns:

- RTkey: Radar Track Key (unique for each flight)
- FPkey: Flight Plan Key (unique for each fligt)
- numberOfTotalEmissions: Number of different flight plans for each flight
- flightPlanInstant: DateTime for the flight plan
- callsign: Flight identifier (not unique for each flight)
- airline: Airline name
- airlineCode: Airline code
- departureAerodrome: ICAO code for origin aerodrome
- arrivalAerodrome: ICAO code for destination aerodrome
- previousAerodrome: ICAO code for previous aerodrome
- aircraftRegistration: Aircraft registration (unique for each aircraft, not flight)
- flightType: Flight Type (GENERAL, MILITARY, NON REGULAR, OTHERS, REGULAR)
- aircraftType: Aircraft type code
- aircraftWake: Aircraft wake turbulence (L/M/H/J)
- cruiseSpeed: Cruise Speed
- rfl: Requested Flight Level
- iobt: Initial Off-Block Time
- eobt: Estimated Off-Block Time
- aobt: Actual Off-Block Time
- etot: Estimated Take-Off Time
- ctot: Calculated Off-Block Time
- atot: Actual Take-Off Time
- eldt: Estimated Landing Time
- totalNumberOfWaypointsInFlightPlanEvent: Number of the waypoints of the flight plan
- waypointOrder: waWaypoint order
- wpName: Name of the waypoint
- wpLatitude: Latitud of the waypoint (not unique for same waypoint in different flight plans)
- wpLongitude: Longitude of the waypoint (not unique for same waypoint in different flight plans)
- wpETO: Estimated time over the waypoint
- wpAltitudeInFeets: Estimated Altitude over the waypoint (feets)

Radar Tracks

This data set provides information about flights during 2019 over the Iberian Peninsula. The data is provided in a CSV format, comprising the following columns:

- RTkey: Radar Track Key (unique for each flight)

- FPKey: Flight plan Key (unique for each flight)
- Callsign: Flight identifier (not unique for each flight)
- departureAerodrome: ICAO code for origin aerodrome
- arrivalAerodrome: ICAO code for destination aerodrome
- cruiseAltitudeInFeets: Cruise altitude (feets)
- cruiseSpeed: Cruise speed (knots)
- flightType: Flight Type (GENERAL, MILITARY, NON REGULAR, OTHERS, REGULAR)
- aircraftType: Aircraft type code
- aircraftRegistration: Aircraft registration (unique for each aircraft, not flight)
- numberOfEngines: Aircraft number of engines
- aircraftWake: Aircraft wake turbulence (L/M/H/J)
- trackInstant: Datetime point instant
- latitude: Latitude point instant
- longitude: Longitude point instant
- altitudeInFeets: Altitude point distance in feets
- heading: Heading point in degrees
- moduleSpeedInKnots: Horizontal speed module in knots
- xSpeedInKnots: Axis x speed in knots
- ySpeedInKnots: Axis y speed in knots
- verticalSpeedInKnots: Vertical speed in knots

ATCo Events

The data is provided in a CSV format, comprising the following columns:

- Event name: The type of the event done by the controller. We want to focus on the ones related to the conflict resolution, which are:
    - A1 – Flight level clearance due to traffic.
    - A2 – Speed adjustment due to traffic.
    - A3 – Direct to fixpoint clearance due to traffic.
    - S2 – Radar vectoring clearance due to traffic (heading).
- Controller: The ATCo responsible of the event. In most cases, tactical controller.
- Time: Timestamp of the event.
- Day: Date of the event.
- Flights: Not relevant information.
- State: Three different states:
    - S. “Sobrevuelo”, which means overflight.
    - D. Departure. Flights that are in the climbing phase.
    - A. Arrival. Flights that are in the descent phase.
- Callsign: Identification of the flight that “suffers” the event.
- Sector: Sector where the event occurs.

CD&R Prototype File Structures

Data Pre-processing

Output Scenario

File name: outputSc.csv

Columns:

- RTkey: Radar track identifier
- FPkey: Flight Plan identifier
- flightPlanInstant: timestamp of flight plan update
- callsign: Callsign of the flight
- airline: Airline name
- airlineCode: Airline identification
- departureAerodrome: ICAO code of the departure airport
- arrivalAerodrome: ICAO code of the destination airport
- flightType: Type of the flight (e.g. Regular)
- aircraftType: Type of the aircraft
- aircraftWake
- cruiseSpeed
- rfl: Requested flight level
- iobt: Initial Estimated Off-block time
- eobt: Estimated Off-block time
- eldt: Estimated Landing Time
- totalNumberOfSectorsInFlightPlanEvent
- sectorOrder
- sectorName: DDR sector name
- sectorName2: alternative sector name
- sectorEntryLevel
- sectorEntryTime
- sectorExitLevel
- sectorExitTime

Output Flight

File name: outputWp.csv

Columns:

- RTkey: Radar Track identifier
- FPkey: Flight Plan identifier
- flightPlanInstant: timestamp of flight plan update
- callsign: Callsign of the flight
- airline: Airline name
- airlineCode: Airline identification
- departureAerodrome: ICAO code of the departure airport
- arrivalAerodrome: ICAO code of the destination airport
- flightType: Type of the flight (e.g. Regular)
- aircraftType: Type of the aircraft
- aircraftWake

- cruiseSpeed
- rfl: Requested flight level
- iobt: Initial Estimated Off-block time
- eobt: Estimated Off-block time
- eldt: Estimated Landing Time
- totalNumberOfWaypointsInFlightPlanEvent
- waypointOrder
- wpName: Code name of the waypoint
- wpLatitude: latitude of the waypoint
- wpLongitude: longitude of the waypoint
- wpETO: Estimated time of arrival to the waypoint
- wpAltitudeInFeets: Altitude (ft) of the aircraft at the waypoint

Radar Tracks

File name: raderTracks.csv

Columns:

- RTkey: Radar Track identifier
- Rtkey2: Flight Plan identifier
- Callsign: Callsign of the flight
- trackInstant: date/time (UTC) of the reported position
- latitude: latitude of the reported position
- longitude: longitude of the reported position
- altitudeInFeets: altitude of the reported position
- heading: heading at the reported position
- moduleSpeedInKnots: magnitude of speed
- xSpeedInKnots: magnitude of speed projected on longitude
- ySpeedInKnots: magnitude of speed projected on latitude
- verticalSpeedInKnots

Sector Geometries

File name: sectorGeometries.csv

Columns:

- SectorID: name of sector
- VolumeID: name of volume. One or more airspace volumes defined by a 2D geometry and a min-max flight level, comprise a sector
- FLmin: minimum flight level for the airspace volume
- FLmax: maximum flight level for the airspace volume
- NbVertices: Number of vertices of the 2D geometry
- Latitude: latitude of the geometry vertex (Degrees, Minutes, Seconds format)
- Longitude: longitude of the geometry vertex (Degrees, Minutes, Seconds format)
- DGU Adjacent: flag (empty or “1”) to indicate that current sector is adjacent to DGU
- TLUN Adjacent: lag (empty or “1”) to indicate that current sector is adjacent to TLUN

EUROPEAN PARTNERSHIP
Co-funded by the European Union

- Comments
- Latitude: latitude of the geometry vertex (Decimal Degrees format)
- Longitude: longitude of the geometry vertex (Decimal Degrees format)

Flight Plans

File name: fpList.csv

Columns:

- Callsign
- Departure airport
- Destination airport
- Flight Plan key

Entry/Exit

File name: *.entry_exit

Columns:

- RTKey: Radar track identifier
- FPKey: flight plan identifier
- TimeInstant: time instant of flight plan update
- sectorID: ID of current sector
- EntryLat: latitude (Decimal Degrees) of entry position to current sector
- EntryLon: longitude (Decimal Degrees) of entry position to current sector
- EntryAlt: altitude of entry position to current sector
- EntryTime: timestamp of entry position to current sector
- ExitLat: latitude (Decimal Degrees) of exit position from current sector
- ExitLon: longitude (Decimal Degrees) of exit position from current sector
- ExitAlt: altitude of exit position from current sector
- ExitTime: timestamp of exit position from current sector

Flight Plan

File name: *.fp

Columns:

- RTKey: Radar track identifier
- Lat: latitude (Decimal Degrees) of reported waypoint
- Lon: longitude (Decimal Degrees) of reported waypoint
- Time: timestamp of reported waypoint
- Alt: altitude of reported waypoint
- Sector: sector ID that covers the reported waypoint
- WPorder
- cruiseSpeed
- RequestedFlightLevel
- Callsign: callsign of the flight

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- ADEP: Departure airport
- ADES: destination airport
- wpName: waypoint name

Flight Phases

File name: *.phases

- phase: one of {climbing, cruising, landing}
- timeStart: time start of the phase reported
- timeEnd: time end of the phase reported

Radar track updates

File name: radar

Columns:

- RTKey: Radar track identifier
- Lat: reported latitude (decimal degrees)
- Lon: reported longitude (decimal degrees)
- Time: timestamp of reported position
- Alt: reported altitude
- sectorID: current sector (for reported position)
- cruiseSpeed: cruise speed
- moduleSpeedInKnots
- xSpeedInKnots
- ySpeedInKnots
- verticalSpeedInKnots
- callsign: callsign of the flight
- ADES: destination airport

Conflicts Detection Component

Main

File name: main.csv

Columns:

- Event_ID: This ID has the following form: TimePoint_RTkey1_RTkey2, where RTkeyx corresponds to a flight in conflict/loss.
- Type: The type of the event. This can be Conflict, Alert, Loss of Separation.
- TimePoint: Time point in the form of a timestamp at which an event is detected.
- RTkey1: Radar Track Key (integer).
- Callsign1: Callsign corresponding to RTkey1 (Alphanumeric).
- DestinationAirport1: Destination airport code corresponding to RTkey1 (Alphanumeric).
- RTkey2: Radar Track Key (integer).
- Callsign2: Callsign corresponding to RTkey2 (Alphanumeric).

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- DestinationAirport2: Destination airport corresponding to RTkey2 (Alphanumeric).
- fp_projection_flag1: Boolean flag denoting if flight plan is used to project the aircraft's position into the future. Corresponds to RTkey1.
- fp_id1: The id of the flight plan used to project the aircraft's position into the future. This has the following format FPKey_timestamp, where timestamp is the time instant that the flight plan was filed. Corresponds to RTkey1.
- fp_projection_flag2: Boolean flag denoting if flight plan is used to project the aircraft's position into the future. Corresponds to RTkey2.
- fp_id2: The id of the flight plan used to project the aircraft's position into the future. This has the following format FPKey_timestamp, where timestamp is the time instant that the flight plan was filed. Corresponds to RTkey2.
- course1: Aircraft's course. Angle (degrees) from the North. Corresponds to RTkey 1.
- course2: Aircraft's course. Angle (degrees) from the North. Corresponds to RTkey 2.
- speed_h1: Aircraft's horizontal speed (m/s). Corresponds to RTkey 1.
- speed_h2: Aircraft's horizontal speed (m/s). Corresponds to RTkey 2.
- speed_v1: Aircraft's vertical speed (feet/s). Corresponds to RTkey 1.
- speed_v2: Aircraft's vertical speed (feet/s). Corresponds to RTkey 2.
- lon1: Aircraft's longitude. Corresponds to RTkey 1.
- lat1: Aircraft's latitude. Corresponds to RTkey 1.
- alt1: Aircraft's altitude (feet). Corresponds to RTkey 1.
- lon2: Aircraft's longitude. Corresponds to RTkey 2.
- lat2: Aircraft's latitude. Corresponds to RTkey 2.
- alt2: Aircraft's altitude (feet). Corresponds to RTkey 2.
- flight_phase_1: The flight phase of the aircraft ("climbing", "cruising" or "descending"). Corresponds to RTkey1.
- flight_phase_2: The flight phase of the aircraft ("climbing", "cruising" or "descending"). Corresponds to RTkey2.
- projection_time_horizon: The time horizon of the projection (seconds).

Conflicts

File name: conflicts.csv.

Columns:

- conflict_ID: This ID has the following form: TimePoint_RTkey1_RTkey2. Where RTkeyx corresponds to a conflicting flight.
- RTkey: Radar Track key (Integer).
- Callsign: Callsign of the flight (Alphanumeric).
- DestinationAirport: Code of the destination airport (Alphanumeric).
- conflict_lon: Longitude at the closest point of approach.
- conflict_lat: Latitude at the closest point of approach.
- conflict_alt: Altitude (feet) at the closest point of approach.
- time_to_conflict: Time (seconds) to closest point of approach.
- h_distance_at_conflict: Horizontal distance (meters) at closest point of approach.
- v_distance_at_conflict: Vertical distance (feet) at closest point of approach.
- first_conflict_lon: Longitude at first point of conflict.
- first_conflict_lat: Latitude at first point of conflict.
- first_conflict_alt: Altitude (feet) at first point of conflict.

- time_to_first_conflic: Time (seconds) to first point of conflict.
- h_distance_at_first_conflict: Horizontal distance (meters) at first point of conflict.
- v_distance_at_first_conflict: Vertical distance (feet) at first point of conflict.
- last_conflict_lon: Longitude at last point of conflict.
- last_conflict_lat : Latitude at last point of conflict.
- last_conflict_alt: Altitude (feet) at last point of conflict
- time_to_last_conflict: Time (seconds) to last point of conflict.
- h_distance_at_last_conflict: Horizontal distance (meters) at last point of conflict.
- v_distance_at_last_conflict: Vertical distance (feet) at last point of conflict.
- crossing_point_lon: Longitude at crossing point.
- crossing_point_lat: Latitude at crossing point.
- t_to_crossing_point: Time (seconds) to crossing point.
- d_h_cp: Horizontal distance (meters) at crossing point.
- d_v_cp: Vertical distance (feet) at crossing point.
- Due_to_flight_1: ResolutionActionID (if it is detected as an effect of a resolution action) or null (if it is detected due to other reasons).
- Due_to_flight_2: ResolutionActionID (if it is detected as an effect of a resolution action) or null (if it is detected due to other reasons).
- command_category: issued/foreseen. Denotes if the conflict is caused by a resolution action that was issued or by a resolution action foreseen (one of the top-3 suggested by the model but not issued yet).
- Sector: SectorID denoting at which sector the conflict has been detected.
- projection_ID: This ID has the following form: TimePoint_RTkey_resolutionActionType_ResolutionActionValue

Conflicts Parameters

File name: conflict_params.csv.

Columns:

- ConflictID This ID has the following form: TimePoint_RTkey1_RTkey2.
  1. Where RTKeyx corresponds to a conflicting flight.
- RTkey: Radar Track key (Integer).
- FP_Track_Cross_Point_Long: Longitude of Point where the current track of the aircraft intersects with the flight plan.
- FP_Track_Cross_Point_Lat: Latitude of Point where the current track of the aircraft intersects with the flight plan.
- FP_Track_Cross_Point_Alt: Altitude (feet) of Point where the current track of the aircraft intersects with the flight plan.
- Track_Course: Course (degrees from North) of the aircraft at the at the FP_Track_Cross_Point according to its current track.
- Flight_Plan_Course: Course (degrees from North) of the aircraft at the FP_Track_Cross_Point according to the flight plan.
- Flight_Plan_Alt: Altitude (feet) of the aircraft at the FP_Track_Cross_Point according to the flight plan.
- Flight_Plan_VSpeed: Vertical speed (feet/s) of the aircraft at the FP_Track_Cross_Point according to the flight plan.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- Flight_Plan_HSpeed: Horizontal speed (m/s) of the aircraft at the FP_Track_Cross_Point according to the flight plan.
- Due_to_flight_1: ResolutionActionID (if it is detected as an effect of a resolution action) or null (if it is detected due to other reasons).
- Due_to_flight_2: ResolutionActionID (if it is detected as an effect of a resolution action) or null (if it is detected due to other reasons)
- command_category: issued/foreseen. Denotes if the conflict is caused by a resolution action that was issued or by a resolution action foreseen (one of the top-3 suggested by the model but not issued yet).

Trajectory Projection

File name: points_of_projection.csv.

Columns:

- projection_ID: This ID has the following form: TimePoint_RTkey_resolutionActionType_ResolutionActionValue_Duration(_waypoint Name in case of a "direct to" resolution action.
- RTkey: Radar Track key (Integer).
- TimePoint: The time point (unix timestamp) at which the projection is made.
- resolution_action_type_value: String in the following format ResolutionActionType_ResolutionActionValue.
- sequence_number: Number of sequence (integer).
- Lon: Longitude of projection point.
- Lat: Latitude of projection point.
- Timestamp: Time (unix timestamp) at which the aircraft is estimated to be over this point.
- Altitude: Altitude (feet) at the projection point.
- Due_to_flight_1: ResolutionActionID (if the conflict detected with this projection is an effect of a resolution action) or null (otherwise)
- Due_to_flight_2: ResolutionActionID (if the conflict detected with this projection is an effect of a resolution action) or null (otherwise)
- command_category: issued/foreseen. Denotes if the conflict detected with this projection is caused by a resolution action that was issued or by a resolution action foreseen (one of the top-3 suggested by the model but not issued yet)

Conformance Monitoring Component

Conformance Events

File name: conformance.csv

Columns:

- ID:This ID has the form TimePoint_FlightID1, where FlightIDx is a conflicting flight,
- 2. Resolution Action ID: This ID has the form TimePoint_FlightID Where FlightID is a conflicting flight.
- Non_conformance_type: values range in {Course, Speed, Altitude, Time}

- Desired value: w.r.t. the non-conformance type this indicates the desired value (e.g. the value of course)
- ActualValue: w.r.t. the non-conformance type this indicates the actual value (e.g. the value of altitude).

Conflicts Resolution Component

Resolution Actions

File name: resolution_actions_episode_1.csv

Columns:

- ResolutionID: A string which corresponds to Resolution Action ID in the following form: *TimePoint_FlightID_X_Y_Z*, where
  - *FlightID* is a conflicting flight (and is equal to the *RTKey)*, *X* is the *ResolutionActionType* (e.g. A1), *Y* is the *ResolutionAction* value (e.g. 1) and *Z* is the *Duration* of the resolution action.

  In a special case, this ID will be in the form: *TimePoint_FlightID_no_resolution* This case is when a flight is executing an action and before the end of this action another conflict is detected. In this case, the field *ActionInProgress* will report the *resolutionID* of the action being performed.

- RTKey: This is the ID of the flight.
- ConflictID: The ID of a detected conflict in the following form *TimePoint_FlightID1_*FlightID2, where either *FlightID1* or *FlightID2* is equal to *RTkey*.
- ResolutionActionType: The possible values of this field are:
  - A1: flight level change,
  - A2: speed change,
  - A3: direct to waypoint,
  - A4: no action,
  - S2: course change,
  - CA: continue action.

  Note that *CA* is the action performed when a resolution action of type *A3* or *A1* is in progress.

- ResolutionAction: The corresponding value of the selected resolution action. Below, the possible values are presented grouped by the type of resolution action:
  - A1: [1,-1] (higher/lower flight level, thus it's a binary decision),
  - A2: [10, -10] (in knots),
  - A4: 0,
  - S2: [10, -10, 20, -20] (in degrees),
  - A3: [1, 2, 3, 4] (the selected waypoint, i.e., 1 is the first waypoint, 2 is the second waypoint, etc.),
  - CA : null.
- Q-Value: It's a float number indicating the output logit of the XAI model for the corresponding resolution action (i.e., the expected value of the resolution action given the current situation).

- ActionRank: This is an integer denoting the ranking of the current resolution action based on *Q-values*. The lowest *ActionRank* value (which is 0) corresponds to the highest *Q-value*.
- Duration: The duration of the selected resolution action in seconds. The possible values are:
  - 30,
  - 60,
  - 120,
  - 180,
  - null.

  null means that the duration is unknown (only for resolution actions of type *A1* and *A3*).

- AdditionalNauticalMiles: NMs added to the trajectory due to the resolution action.

A special case is when a waypoint is not available and thus the value of *AdditionalNauticalMiles* is null.

- FilteredOut: This field provides the reason of excluding a resolution action. The filtering is applied to the actions proposed by the XAI model in order to deterministically exclude actions that will result in the following conditions not being met:
  - 178.67 <= horizontal_speed <= 291,
  - -80 <= vertical_speed <= 60,
  - Flight phase is not climbing/descending,
  - The waypoint 1/2/3/4 is available.

  The possible values of this field are:

  - “Vertical speed cannot be increased because it will exceed 60.0 feet/s”,
  - “Vertical speed cannot be decreased because it will fall below -80.0 feet/s”,
  - “Horizontal speed cannot be increased because it will exceed 291 m/s”,
  - “Horizontal speed cannot be decreased because it will fall below 178.67 m/s”,
  - “Flight phase is climbing/descending”,
  - “The waypoint 1/2/3/4 is not available”.
- ActionInProgress: When a flight is executing an action and before the end of this action another conflict is detected, then the *ResolutionID* of the action being performed will be reported in this field. In any other case, the value of this field will be *null*.
- Prioritization: The ID of the corresponding attention heads’ heatmap included in the file *conflict_heatmaps_episode_1.csv*.
- VSpeedChange: The absolute difference (in feet/s) of vertical speed (VS) between VS after applying the resolution action and before that.
- HSpeedChange: The absolute difference (in m/s) of horizontal speed (HS) between HS after applying the resolution action and before that.
- CourseChange: The difference (in degrees) of the course of the corresponding aircraft between the course after applying the resolution action and before that.
- HShiftFromExitPoint: The horizontal shift from exit point (in meters) after applying the resolution action.

- VShiftFromExitPointQ The absolute Vertical shift from exit point (in feet) after applying the resolution action.
- Bearing: Angle of the agent's course w.r.t. North (χ), in degrees, after applying the resolution action.

HITMAPS

File name: conflict_hitmaps_episode_1.csv

Columns:

- HitMapID: This is an ID which corresponds to that of *Prioritization* field of the fie *resolution_actions_episode_1.csv*. Note that all resolution actions of a specific flight at a specific timepoint share the same attention heads' heatmap, and thus the value of the *HitMapID* field will be the same. This ID has the following form: *TimePoint_FlightID*
- RTKey: This is the ID of the flight in which the attention values are referred to.
- ConflictID: The ID of a detected conflict with the following form: *TimePoint_FlightID_FlightIDx* (or *TimePoint_FlightIDx_FlightID*) where *FlightID* corresponds to *RTKey* field of this file and *FlightIDx* is the ID of a flight from the total *N* flights (*N* is the total number of flights at a specific scenario).
- FlightID1: A float number (attention value) in the range [0,1] if there are attention values of the flight with ID=*RTkey* which (attention values) are referred to the flight with ID=*FlightID1*. Otherwise, the value of this field is *null*. The same is true for the rest *FlightIDx*.
- FlightID2: A float number if there are attention values of the flight with ID=*RTkey* which (attention values) are referred to the flight with ID=*FlightID2*.
- FlightIDN: A float number if there are attention values of the flight with ID=*RTkey* which (attention values) are referred to the flight with ID=*FlightIDN* (*N* is the total number of flights at a specific scenario).

Appendix B

Integration of Components

Components in the ATFCM and CD&R prototypes have been integrated using RabbitMQ.

RabbitMQ[1] is used as a broker for the message exchange between the components that each prototype comprises. RabbitMQ is a middleware that supports multiple protocols, and four message exchange types:

- Direct Exchange: A direct exchange delivers messages to queues based on the message routing key. A direct exchange is ideal for the unicast routing of messages (although they can be used for multicast routing as well).
- Fanout Exchange: A fanout exchange routes messages to all of the queues that are bound to it and the routing key is ignored. If N queues are bound to a fanout exchange, when a new message is published to that exchange then a copy of the message is delivered to all N queues. Fanout exchanges are ideal for the broadcast routing of messages.
- Topic Exchange: Topic exchanges route messages to one or many queues based on matching between a message routing key and the pattern that was used to bind a queue to an exchange. The topic exchange type is often used to implement various publish/subscribe pattern variations. Topic exchanges are commonly used for the multicast routing of messages.
- Headers Exchange: A headers exchange is designed for routing on multiple attributes that are more easily expressed as message headers than a routing key. Headers exchanges ignore the routing key attribute. Instead, the attributes used for routing are taken from the headers attribute.

RabbitMQ implements the message queues which share some properties with exchanges, in addition to the following properties:

- Name: the name of the queue
- Durable: the queue will survive a broker restart
- Exclusive: used by only one connection and the queue will be deleted when that connection closes
- Auto-delete: queue that has had at least one consumer is deleted when last consumer unsubscribes
- Other Arguments: (optional) used by plugins and broker-specific features such as message TTL (time-to-live), queue length limit, etc.

Before a queue can be used it has to be declared, which will cause it to be created if it does not already exist (the declaration will have no effect if the queue already exists).

Finally, a single broker can host multiple isolated "environments" (groups of users, exchanges, queues and so on). This is made possible by "virtual hosts": vhosts, which are similar to virtual hosts used by

[1] https://www.rabbitmq.com/

Web servers and provide completely isolated environments for RabbitMQ entities. The senders/consumers specify what vhosts will use during connection negotiation.

ATFCM Prototype: Integration of components

The backbone of ATFCM system comprises 4 basic components, as illustrated in Figure 15, the Data Preparation, AI/ML solution, Explainability, and Visual Analytics component. The system is invoked by and shares results with the FMP client.

A more detailed view of the ATFCM prototype system backbone is illustrated in Figure 11, where each component implements a sender (for sending messages), a queue for the received messages and a consumer which pops received messages from the queue.

It must be noted that in contrast to the overall system architecture in Figure 4, which specifies data exchange between components, Figure 15 specifies flow of messages for invoking and terminating the pipeline (i.e. computation of solutions and provision of explanations). Specifically, the whole pipeline is initiated with a message from the FPM client to the Visual Analytics component and then to the Data preparation component, and ends with a message from the Visual analytics component back to the FMP client.

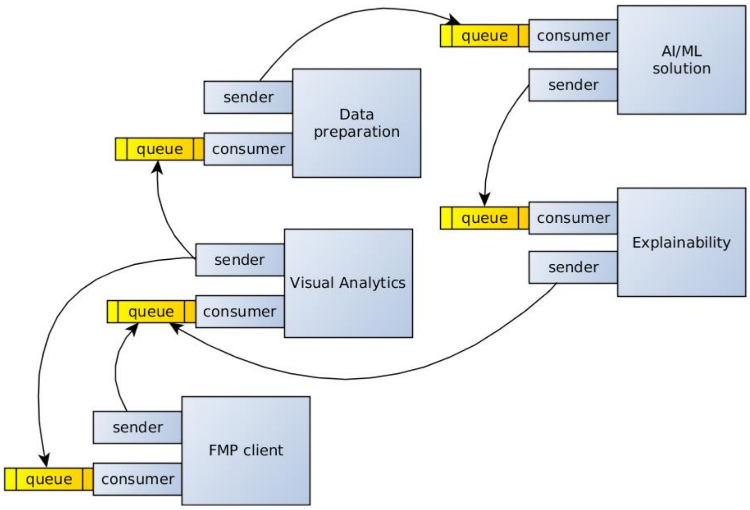


Figure 15. ATFCM Integrated Prototype: Message exchange overview

On a more technical point of view, each of the components, as Figure 15 shows, implements a sender and consumer classes. Herein we explain the implementation code. For the ATFCM system prototype, we deploy RabbitMQ on a machine accessible via the public IP 83.212.170.63. For simplicity, all components connect to the virtual host "tapas-vhost" using the same credentials (username/password). The code described in this section is based on the rabbitMQ-client 5.11.0 which can be imported using the following line in the dependencies section of gradle configuration in any project:

```
compile 'com.rabbitmq:amqp-client:5.11.0'
```

We assume per component at least one thread using the sender method (to transmit messages) and exactly one consumer object to consume messages.

Co-funded by the European Union

The sender class implements only one method (namely “sendMessage”), which accepts two arguments: a string specifying the queue where the message is to be sent, and the message itself. The connection negotiation is initialized only when a message is to be sent, and the connection closes when the message has been transmitted.

The class consumer implements the method listen() with a string argument queue, which specifies the name of the queue on RabbitMQ to listen. Message types and flow of messages in ATFCM prototype

The message types and the typical workflow, is the following:

- FMP client specifies to VA the corresponding **date** for the scenario to be evaluated
- VA sends to Data Preparation component the corresponding **date** for the scenario to be evaluated
- Data Preparation sends to AI/ML solution the **date** and **path** where the generated files are stored
- AI/ML solution sends the **date** and **path** where solutions are stored to Explainability component
- Explainability component sends the **date** and **path** for the output to VA analytics.
- (Optional) Any component (or any authorized client in general) can request the status of any other component, by sending the message “<myQueue>:status” to the component’s queue. The components should be able to estimate the progress and time left of the processing at any time and report to the queue specified in <myQueue>. For example, a message “vaQueue:status” from VA to Data preparation component, will trigger the function that estimates the progress of data processing in Data preparation component and it will report to the queue “vaQueue”.
- Finally, VA reports the results to FMP client

The date in messages is necessary as it enables tracking of the exchanged messages. For example, it is possible that while explainability component is processing an output, another request (for a different date) is initialized with a message from VA to Data Preparation.

Please notice that no data are exchanged directly between components since data transfer would increase the communication time and thus, the overall time needed to compute solutions. We assume that each component maintains/updates a copy of the data it needs for the processing in its local file system. Finally, each component processes the data on its own platform/system, providing to the architecture all the useful features of distributed systems, such as maintenance, openness, concurrency, scalability and transparency.

The generated data are stored in the filestore datacron1.ds.unipi.gr in the path specified in the transmitted messages. Components access the data through SFTP connections, and each component uses a separate folder as its own workspace (i.e., named after the component’s name). Generated files are stored under a folder named after the processed date in the component’s workspace.

CD&R Prototype: Integration of components

As in ATFCM prototype, the communication protocol uses brief messages between components to minimize the communication cost. Therefore the messages contain only the path where the processed/generated data are available and the corresponding scenario ID. We assume that each component has access to the path where data are stored. Each component upon receiving a message, automatically initiates its processing tasks.

On a more technical point of view, each of the components, as Figure 16 shows, implements a sender and consumer classes. Herein we explain the implementation.

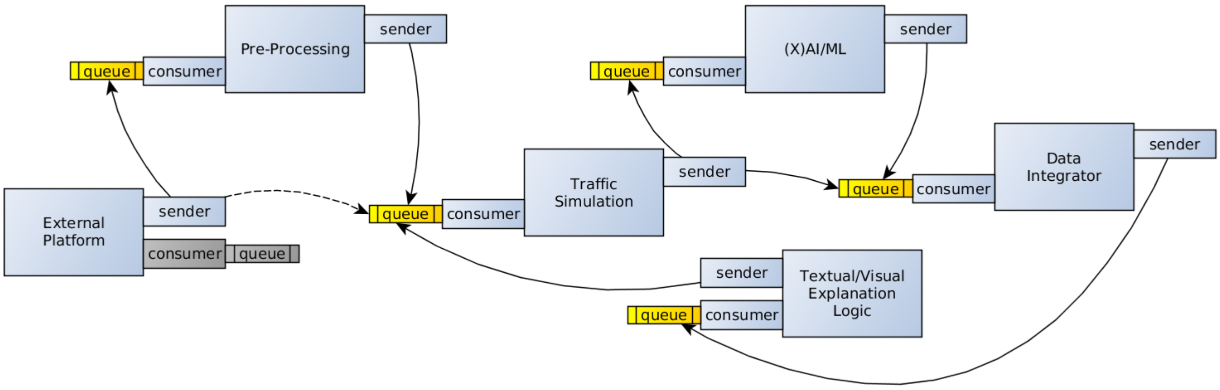


Figure 16. CD&R Integrated Prototype: The message exchange network.

We deploy RabbitMQ on a machine accessible via a public IP. For simplicity, all components connect to the virtual host "tapas-vhost" using the same credentials (username/password).

We assume per component at least one thread using the sender method (to transmit messages) and exactly one consumer to consume messages.

The class sender implements only one method (namely “sendMessage”), which accepts two arguments: a string specifying the queue where the message is to be sent, and the message itself. The connection negotiation is initialized only when a message is to be sent, and the connection closes when the message has been transmitted.

The class consumer implements the method listen() with a string argument queue, which specifies the name of the queue on rabbitMQ to listen. Message types

The sequence of messages exchanged between the components of the prototype is illustrated in Figure 16. The message types and the typical workflow, is the following:

- The external platform transmits to Pre-processing component the scenario ID, the path where the related data are stored and time instant to be processed (time step 30 seconds). Additionally, the external platform transmits to Traffic Simulation, the action taken for the specific scenario, for the previous time step.
- The Pre-processing component propagates the scenario ID, and the path to processed data, to Traffic Simulation.

- The Traffic Simulator transmits the path where the generated files are stored to Data Integrator and the path to detected conflicts to XAI/ML component, along with the scenario ID and the time step.
- The XAI/ML component transmits the path where the generated files are stored to Data Integrator, as well as the scenario ID and the time step.
- The Data Integrator propagates the messages received from both Traffic Simulator and XAI/ML for the same scenario ID and time step, to Textual/Visual Explanation Logic.
- Vis&UI component transmits the scenario ID, and the action taken at the previous time step (time t-30'').

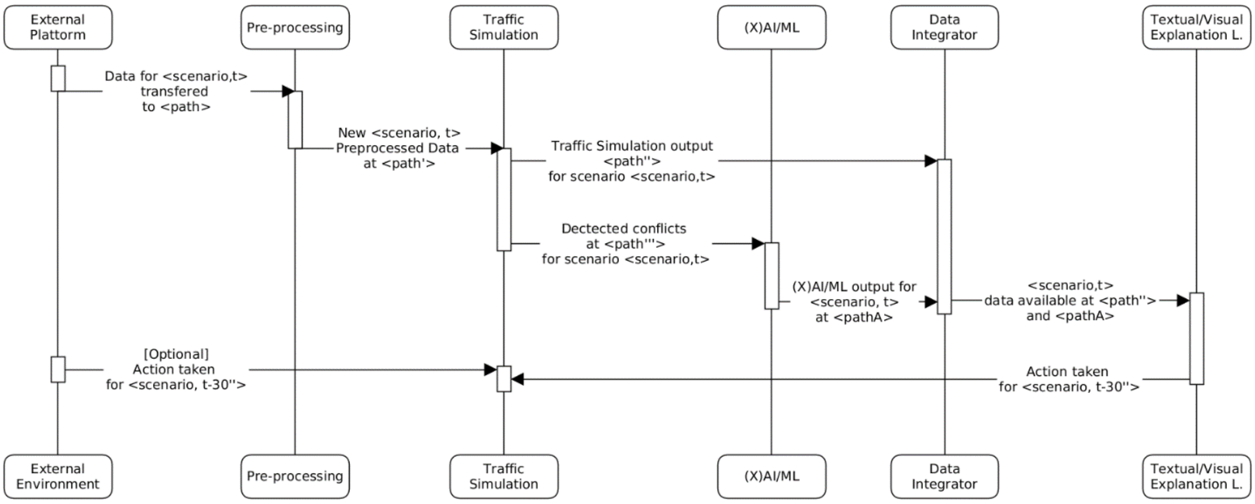


Figure 17. **CD&R Integrated Prototype: Sequence diagram for the message exchange between the components.**

Similarly to the ATFCM prototype, components do not exchange data: They maintain data in their own local file system and they access data provided by others via SFTP.

The system is also enabled to operate when an SFTP server is not present. Specifically, the system features configuration options that bypass the file transfer process of each component, enforcing the use of local file system. In this case, the file paths transmitted in messages between the components of the system, are the absolute paths in the local file system. In the following paragraphs we assume that the system is configured to use an SFTP server (generic solution).

Co-funded by the European Union