Analysis & report, Sim 1

Deliverable ID:	D5.2
Dissemination Level:	PU
Project Acronym:	MAHALO
Grant:	892970
Call:	H2020-SESAR-2019-2
Topic:	SESAR-ER4-01-2019
Consortium Coordinator:	DBL
Edition date:	15 February 2022
Edition:	00.02.00
Template Edition:	02.00.02

Founding Members
EUROPEAN UNION
EUROCONTROL
SESAR JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Erik-Jan van Kampen/DUT	WP5 Leader	10/11/2021
Tiago Monteiro Nunes/DUT	WP5 Researcher	15/11/2021
Magnus Bång/LIU	WP5 Researcher	17/11/2021
Sami Yahia/LIU	WP5 Researcher	22/11/2021

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Carl Westin (LiU)	WP6 Leader	22/11/2021
Brian Hilburn (CHPR)	WP2 Leader	22/11/2021
Clark Borst (TUD)	WP4 Leader	23/11/2021
Martin Christiansson (LFV)	Project member	24/11/2021
Matteo Cocchioni (DBL)	Project Contributor	26/11/2021
Stefano Bonelli (DBL)	Project Coordinator	26/11/2021
Stefano Bonelli (DBL)	Project Coordinator	15/02/2022

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

Name/Beneficiary	Position/Title	Date
Stefano Bonelli (DBL)	Project Coordinator	26/11/2021
Stefano Bonelli (DBL)	Project Coordinator	15/02/2022

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
00.00.01	10/11/2021	Draft	Erik-Jan van Kampen	Doc created
00.00.02	12/11/2021	Draft and internal review	Erik-Jan van Kampen	Deliverable improvement
00.00.03	15/11/2021	Draft and internal review	Tiago Monteiro Nunes	Deliverable improvement
00.00.04	17/11/2021	Draft and internal review	Magnus Bång	Deliverable improvement
00.00.05	22/11/2021	Draft and internal review	Sami Yahia	Deliverable improvement
00.00.06	26/11/2021	Internal release	Matteo Cocchioni	Deliverable to MAHALO management
00.01.00	26/11/2021	Final Release	Stefano Bonelli	Deliverable approved for submission
00.01.01	02/02/2022	Internal release	Erik-Jan van Kampen	Added section 5.2 based on update request from SJU
00.02.00	15/02/2022	Final Release	Stefano Bonelli	Deliverable approved for submission

Copyright Statement

© – 2022 – MAHALO Consortium. All rights reserved. Licensed to the SESAR Joint Undertaking under conditions.

MAHALO

MODERN ATM VIA HUMAN / AUTOMATION LEARNING OPTIMISATION

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892970 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document describes the SIM1 experiment that has been carried out in MAHALO WP5: Integration. The purpose of SIM1 is to validate the integrated system, composed of machine learning models and human machine interfaces that will be used for the main MAHALO experiments that are carried out in WP6.

Table of Contents

List of Tables

List of Figures

1 Introduction

The MAHALO project asks a simple but profound question: in the emerging age of Machine Learning (ML), should we be developing automation that matches human behaviour (i.e., conformal), or automation that is understandable to the human (i.e., transparent)? Further, what trade-offs exist, in terms of controller trust, acceptance, and performance?

To answer these questions, two types of Machine Learning (ML) models have been developed in WP3: a conformal/personalized prediction model based on Supervised Learning (SL) and an optimized prediction model based on Reinforcement Learning (RL).

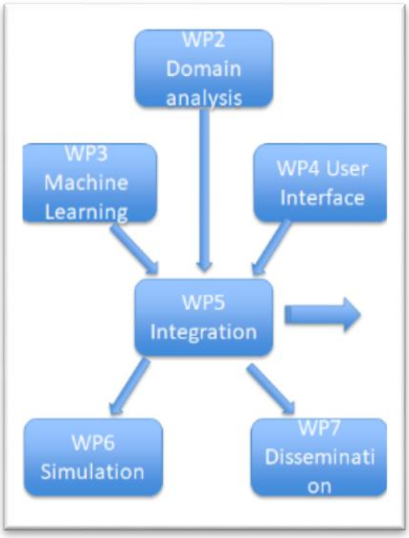

Figure 1: MAHALO Work Packages

In WP5 the machine learning models from WP3 and the User Interfaces from WP4 are integrated, such that the combined system can be used for experiments with real air traffic controllers in WP6.

In order to validate the integration process of WP5, a preliminary experiment or test of the system has been envisaged. This test is called SIM1 (the real experiments in WP6 are called SIM2a and SIM2b).

The deliverable D5.1 describes in detail how the integration of machine learning models and user interfaces has been achieved. The current documents, D5.2, reports upon the setup and results from SIM1. It will, thus, contain a description of the following items:

- Refined scenario settings
- Refined machine learning model settings
- Analysis of Simulation 1 results
- Data collection and processing protocols

For more detailed descriptions of the ML systems and the interface used one should refer to deliverables D3.1 and D4.1, respectively.

1.1 Objective

The objective of D5.2 is to report on SIM1 outcomes and to reflect on the integration process of WP5 in general. It should become clear from this report what has been achieved (and how), but also what elements have found to be still lacking in maturity and that will need some further work leading up to the experiments in WP6.

1.2 Scope

As also explained in MAHALO deliverable D5.1, MAHALO WP5 is a mainly technical work package that deals with integration aspects. It is not meant to address the science behind the main research questions directly. Of course it should enable answering of these questions in other work packages.

With limited resources in terms of real air traffic controllers, SIM1 has been tested with non-experts. This means that the data generated from SIM1 should not be used to draw conclusions about the research questions related to acceptance of automation. Instead, the results of SIM1 should only be used to validate if the integration process of WP5 has been successfully completed, in preparation for the experiments with real air traffic controllers in MAHALO WP6.

1.3 Report structure

Chapter 2 contains information on the scenario settings that were used for SIM1. It also describes the scenario design process, which turned out to be more challenging than initially expected. In Chapter 3 the settings for the machine learning models in SIM1 are provided. Chapter 4 discussed data collection protocols, while Chapter 0 analyses the results of SIM1 and summarizes some of the lessons learned during this phase of the project. This report is concluded in Chapter 6

2 Scenario settings

This chapter present the scenario settings that are used for SIM1. The scenario development process is discussed first, after which the specific settings for the scenarios in SIM1 will be presented.

2.1 A few words on scenario realism

The scenarios are designed to resemble the scenarios to be used in the final experiments SIM2a/2b. Special care has been taken to ensure that the scenarios themselves are realistic, i.e. contain a realistic air traffic flow that is complex enough to ensure immersion. In order to remove the influence of bias from interacting with previously known sectors, a virtual sector is created that contains all the elements one would expect from a normal sector such as aircraft tracks, boundaries, set waypoints and dominant air traffic flows. This virtual sector does not exist in real-life and, thus, using it reduces the effect of ATCO biases and preconceptions.

Since the ML agents considered only solve conflicts using horizontal resolutions, from the ML perspective the altitude of the aircraft is not relevant as long as there is no altitude conflict (which the automation is not capable of dealing with in its current implementation). Considering all aircraft at the same altitude would, thus, not affect the automation. It would, however, greatly decrease the realism of the scenarios and the sector which would affect the behaviour of the ATCO's towards it. Scenarios are, thus, designed such that aircraft go through the sector at different Flight Levels but conflicts are limited to the horizontal direction. This means that aircraft from different flight levels only have to be de-conflicted with aircraft of the same flight level. Doing this, optimizes data collection since all scenarios collected during the experiment can then be used for training of the agents.

2.1.1 Data quantity and variety

As will be expanded upon later on in this report, one large issue whenever AI agents are used is the amount of data needed. As a rule of thumb, ML, in general, performs better whenever more data is available. Even if the scenario design is optimized for the maximum efficiency when it comes to the amount of data that can be usable for training, it is likely that the data will still be insufficient. This is a known possible issue, some mitigation strategies will be discussed further ahead in the data collection chapter. Another issue that arises from this is that, by restricting scenario design in this way, scenarios are necessarily more "simplistic" and there is a concern regarding whether there will be much difference between the ATCOs in terms of solutions chosen. Usually, ATCOs come up with and follow a plan. Complex scenarios lead to different ATCOs choosing different plans. If the scenario is too simplistic it might be that most ATCOs choose the same, somewhat simplistic, plan which might impact results. All of these considerations have been taken into account. Unfortunately, regarding the fact that ATCOs might choose the same solutions, there is little that MAHALO can do to mitigate the problem. Ideally, there would be time to expose the ATCOs to many different, complex, scenarios such that a rich and varied data set could be collected. Due to logistical limitations and limited ATCO availability this is simply not possible.

2.2 Scenario development

The scenario development starts by generating a sector from a set of boundary points, see Figure 2. The sector has a size similar to a MUAC (Maastricht Upper Area Control) sector and is shaped in such a way that when the sector or mirrored or rotated, the shape does not appear to be very different (unlike a more strongly elongated sector, which would appear different under rotation).

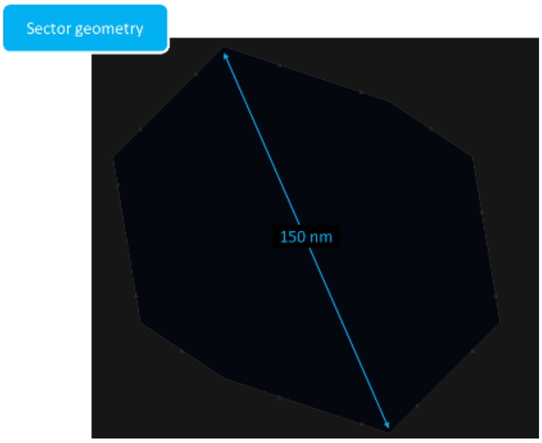

Figure 2: Scenario development: Sector definition

The next step in scenario development is to introduce the conflict that will have to be solved by the ML models (or by the human air traffic controller when generating training data for the SL model). Conflicts are defined in terms of Conflict Angle (CA), time to closest point of approach (tCPA) and the distance at closest point of approach (dCPA). Figure 3 shows an example of a conflict generation with a CA of 105 degrees, a tCPA of 300.0 s and a dCPA of 2.0 nm.

After conflict generation, the scenario is made more realistic by introducing other aircraft. These aircraft are placed in such a way that they do not create additional conflicts, either because of different headings/position or by having a different altitude. Also, to try to force the controllers to create a resolution in the heading mode, which is what the ML models are trained for, some traffic is placed above and below the conflicting aircraft pair, such that solving the conflict by altitude changes is discouraged. Figure 4 shows an example of a scenario where additional non-conflicting traffic is added.

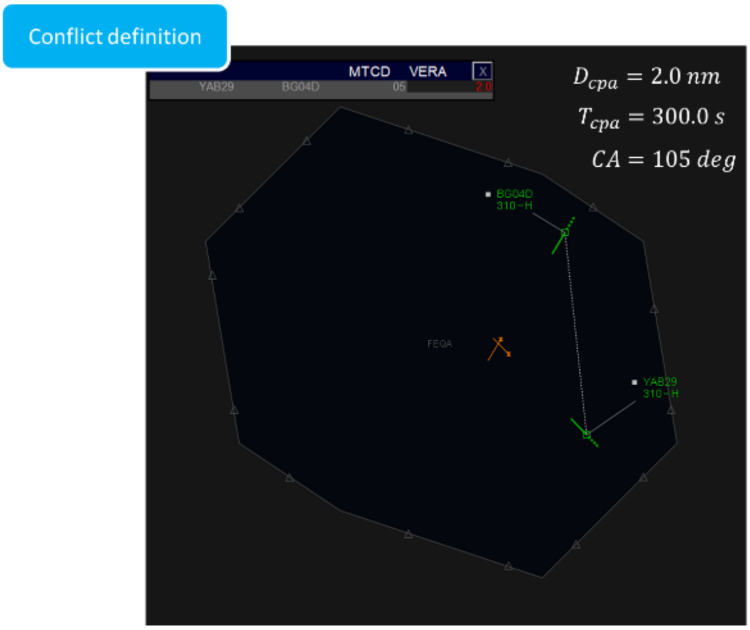

Figure 3: Scenario development: Conflict generation

The traffic is generated as a Free Route Airspace, but with some main flight directions. This concept is show in Figure 5.

The last step in the scenario development process is to create 'different' scenarios in an easy way by mirroring the sector, rotating the sector, randomizing call signs and changing flight levels. This will create new scenarios that are in essence the same, i.e. the underlying conflict has the same CA, tCPA and dCPA, but to a human air traffic controller it would seem like a different scenario. An example of this process is shown in Figure 6, which is a modified scenario derived from the one in Figure 4. The underlying conflict in these two figures is exactly the same, even though the scenarios seem different. This allows us to investigate the consistency of human air traffic controllers, which plays a role in how well the SL models can be trained to create conformal automation. If the training data is not consistent, i.e. if the human air traffic controller provides different resolutions for the same conflict, the SL model cannot any more consistent than the air traffic controller it is mimicking.

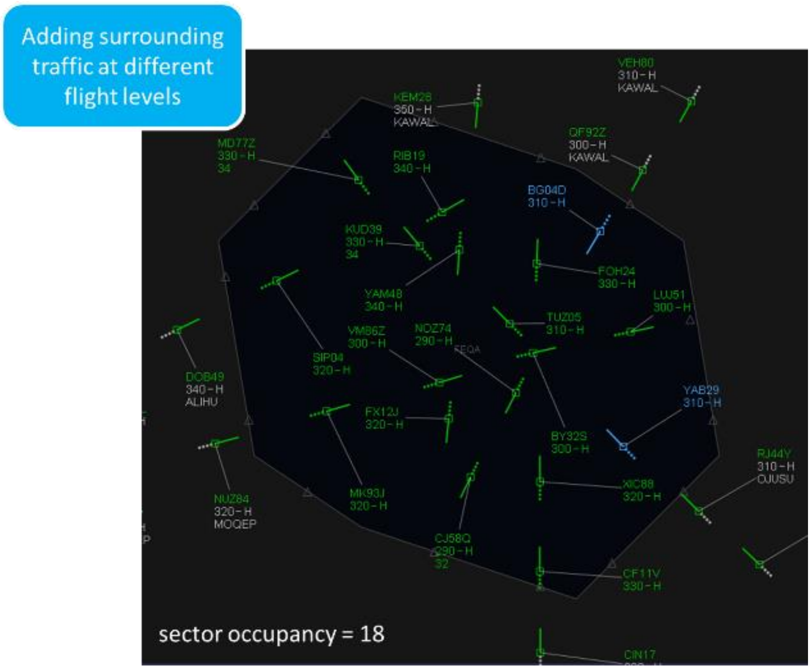

Figure 4: Scenario development: Adding additional non-conflicting traffic

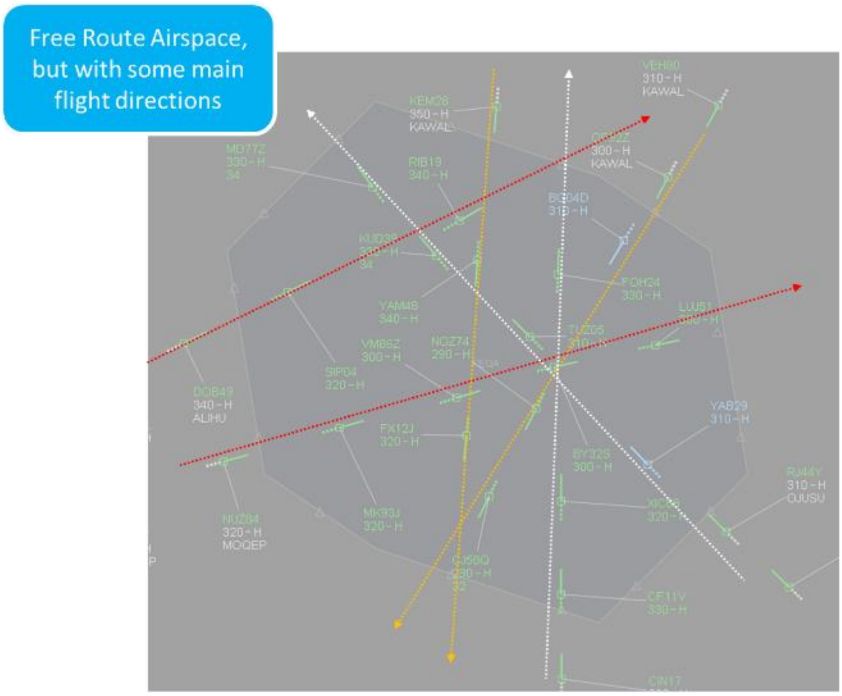

Figure 5: Scenario development: Traffic patterns

In order to increase efficiency of the scenario design process. Scenarios were also flipped alongside a randomization of both aircraft call signs and waypoint names. This can be seen in **Error! Reference source not found.**.

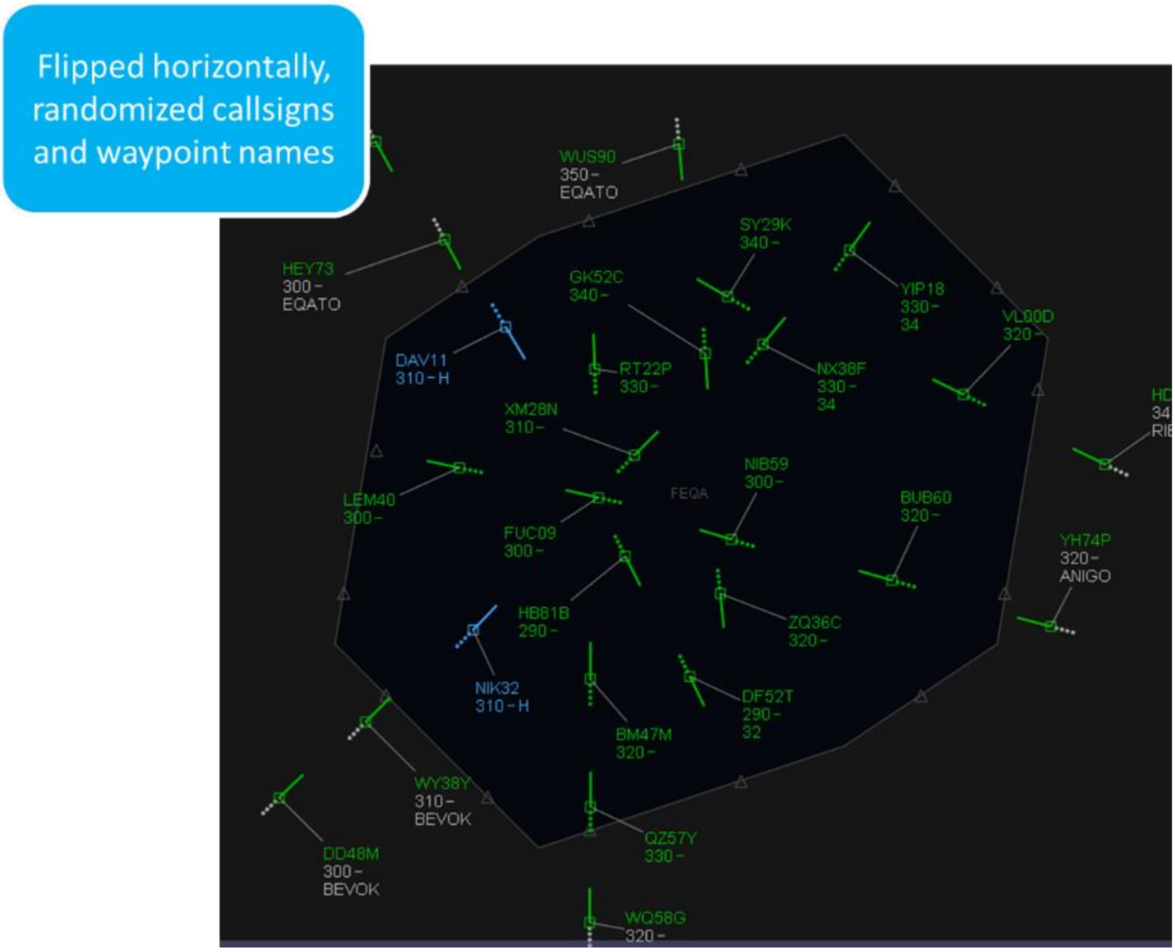

Figure 6: Scenario development: creating new scenario by manipulating a previously created scenario

2.3 Scenario settings for training of the RL agent

Whereas the Supervised Learning (SL) Conformal agent system only be trained once operator data is available, the Reinforcement Learning (RL) ‘Optimal’ agent can be trained using virtual data since it is only concerned with the ‘optimality’ of the solution and not operator preferences. Therefore, virtual data was generated based on scenarios similar to those used in the simulation. This virtual data is generated by using a Modified Voltage Potential (MVP) deterministic algorithm. MVP is used since it provides decent solutions that can be used as demonstrations for the DQFD agent. The MVP-Q-Learning agent is, obviously, also tied to the MVP algorithm so it is run using it.

The scenarios used for the training are set as a conflicting pair of aircraft in a generic sector. The controlled aircraft is initialized at a given set of coordinates and the intruding aircraft or aircrafts are initialized such that a conflict is guaranteed to happen. The conflict angles and dcpas are then varied to obtain a varied set of data. The tlos is also varied to guarantee a set of short-term vs medium-term conflicts being available in the data.

For the Q-learning agent, the different conditions considered can be found in Table 1. The different conflict angles are split up into three types, each one is used to train a separate agent to reduce training time and facilitate parallel learning. Since the main concern of Q-learning is with learning the Q-table, this is equivalent to splitting up the Q-table in 3 parts and allowing the agents to learn them independently. For the final implementation the tables are then joined so that the agent can infer upon the correct value depending on the state. The agent with the crossing-track conflicts will learn half of the total Q-table, the other two agents will each learn a quarter.

Table 1 Parameters of the Q-Learning states

Parameter (states/actions)	Min value	Max Value	Var	N
Initial Dcpa [nm]	0 [nm]	2 [nm]	0.5 [nm]	5 (3 : 0/1/2)
Initial Tcpa [s]	500 [s]	600 [s]	50 [s]	3 (1:500)
Conflict angle Head-on [deg]	135 [deg]	215 [deg]	20 [deg]	5
Conflict angle Crossing	45/235	125/315	20	10
Conflict angle Overtaking	315	45	20	5
Lookahead time [s]	150 [s]	450 [s]	120 [s]	3
Safety factor [-]	1 [-]	2 [-]	0.5 [-]	3
Action index	-	-	-	3x3=9
Q-table Size	-	-	-	5 * 3 * (5+10+5) * (3*3) * (3*3) = 24 300

The reasoning behind the variation between states being so high (for example, only considering conflict angles every 20degrees instead of 5) has to do with the amount of cases to be trained on. For 4 times more conflict angles, which would make the agent have a resolution of 5degrees, there would be 97 200 different possible scenarios. At the current computational speed that corresponds to 10-15 seconds depending on each episode, this would take about 27 hours to run every scenario. Consider that each scenario should be sampled more than once as the Q-learning agent will each time update its Q-value estimate for that scenario to be closer to its actual real Q-value. This means that each scenario should be visited more than once and that would lead to a too-large computation time.

Moreover, the Q-learning agent is only in charge of increasing the performance of the MVP algorithm, which is already good. This means that the resolution does not matter as much in terms of performance so computation time can be saved by reducing the total number of states. For states that are not contained in the Q-table as is, an interpolation can be performed to determine the correct parameters.

2.4 Scenario settings for SIM1

Table 2 provides the details of the scenarios that were used in SIM1, in terms of the conflict setup (distance to closest point of approach (dcpa), time to closest point of approach (tcpa), and conflict angle(ca)). It also contains information on the callsigns that are involved in the conflict and details on how the scenario was created by rotation from another scenario. There are two 'base' scenarios in terms of conflict parameters. The other scenarios are transformed by altering: flight level, callsigns, waypoints, rotation angles.

Table 2: Scenario settings for SIM1

	name	dcpa [nm]	tcpa [s]	ca [deg]	FL	callsign A	callsign B	rotation [deg]	min occupancy	max occupancy
crossing	scene1A	2.0	300	105	310	YAB29 (A321)	BG04D (A332)	0	17	20
	scene1B	2.0	300	105	320	XAN57 (A321)	CAK17 (A332)	10	17	20
	scene1C	2.0	300	105	340	LUD37 (A321)	WUL38 (A332)	20	17	20
	scene1D	2.0	300	105	360	BOD57 (A321)	NUD67 (A332)	30	17	20
	scene1E	2.0	300	105	330	QOQ88 (A321)	DET45 (A332)	40	17	20
	scene1F	2.0	300	105	350	YR42S (A321)	XT91G (A332)	-10	17	20
crossing	scene2A	0.0	340	90	340	KUR02 (A321)	DM14J (A342)	0	16	20
	scene2B	0.0	340	90	330	PV47X	XB40G	10	16	20
	scene2C	0.0	340	90	350	KIJ60	PM74V	20	16	20
	scene2D	0.0	340	90	320	HON97	DEN20	30	16	20
	scene2E	0.0	340	90	310	SUJ31	TEN15	40	16	20
	scene2F	0.0	340	90	340	SW54Z	ZZ56C	-10	16	20

3 Machine Learning Models settings

3.1 Reinforcement Learning agents

The RL agents tested in SIM1 are set up so that they provide heading conflict resolution advisories depending on the states of the system. For a detailed explanation of the models themselves and how they are trained refer to MAHALO deliverable D3.1.

The different hyper parameters of the Reinforcement learning agents are chosen through a careful analysis where the performance of the system is evaluated such that the best settings are found. Unfortunately, no method currently exists to determine what the most optimal parameters and settings are. Thus, this analysis and evaluation involves a lot of trial and error.

The Q-learning agent outputs a series of parameters to be used by the MVP algorithm whereas the DQFD agent outputs a resolution directly. This means that the Q-learning agent is being built on top of an algorithm that already achieves decent performance thus making it less sensitive to the changing hyper-parameters. The DQFD agent is more sensitive to these since: it has to provide a direct action command (which gives it more possibilities to be wrong) and it has to consider more information than the Q-learning agent.

For the purposes of SIM1 these parameters and settings are not as important as testing the entire pipeline of the MAHALO project. Throughout the time between SIM 1 and SIMs 2a/b there will be constant virtual trials being run to try to improve performance and find even more optimal settings for the algorithms. Within SIM1, the parameters and settings used were those found to be the best performing so far.

The RL agents also present the benefit of being able to be trained on purely virtual data and interaction with the ATM simulator. This makes it easier to test the algorithms and also makes data requirements looser than when compared to the SL system.

3.2 Supervised Learning system

The SL agent, at this stage, shall provide a heading, the time (scenario time) at which the heading change was made by the agent, and the aircraft which the SSD displayed belongs to. A suitable prediction time, that is the time in which the SL model provide a resolution advisory, is currently examined in the consortium, and will be fine-tuned when more data has been collected from ATCOs. A suitable information for an ATCO is an advisory preceding the conflict and follow flight rules and time constraints given speeds. To achieve this, since each SL agent is trained by SSD images specific to each ATCO, one approach could be to tune and label these images earlier than usual.

The hyper parameters while training and testing these SL models are decided after a series of runs while varying these parameters. Trying different values for epochs (number of passes), dataset ratio, images shape, loss function etc. and then checking the accuracy these parameters yield. A positive aspect with the SL agents in comparison to the RL agents, is that trial and error, during development, is minimized because all models are trained on the same type of input. This input being the SSD images

that only change depending on the ATCOs input that generate these images (scenarios). Hence, once the best Convolutional Neural Network (CNN) model, analysing a set of consistent and evenly sampled data, is generated, the performance of each SL agent depends on the skill of the ATCO.

The pipeline for producing a Supervised Learning agent starts with a series of pre-processing scenario files generated by an air traffic controller and extracting the SSD images in connection to the decisions an ATCO makes.

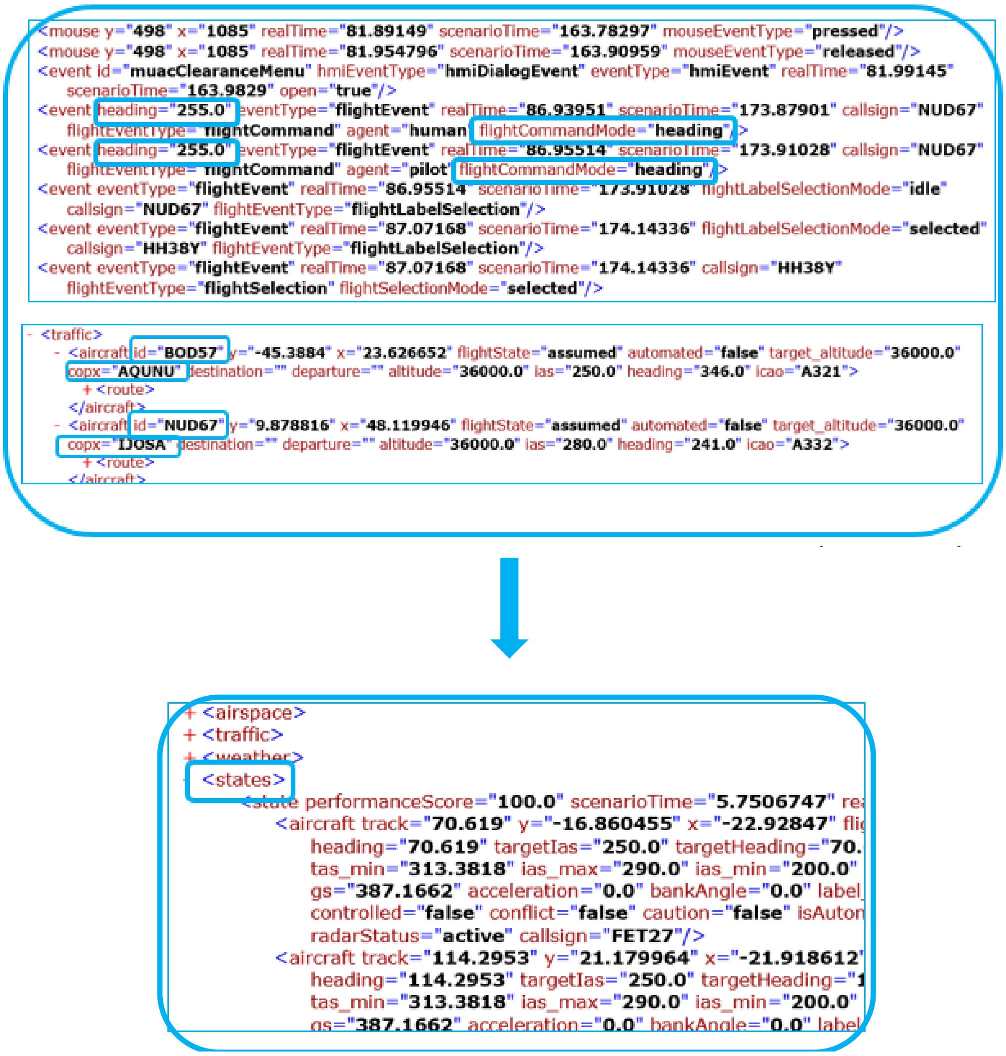

Figure 7: Extraction from an event and state log of a simulated scenario

Figure 7 above shows what the pipeline focuses on extracting from the event logs of each scenario at this stage. It goes through these logs and extracts the ‘heading’ events made by an ATCO and the information about the aircrafts involved in the collision to be avoided. The next parts of the pre-processing are going through the state logs to pull out more information of the surroundings in a sector at the time of interest. Eventually the pipeline ignores scenarios where an ATCO makes no heading changes.

Based on the above processing steps of the scenarios, an SSD image is created showing the model when the ATCO is seeing at the instant said ATCO made the vital decision to avoid the conflict. This is done by a python script and libraries such as *Pillow* and *OpenCV* in addition to calculations made to find the distance between aircrafts, exit bearing etc. Each SSD image is associated with a label describing the heading change made by the air traffic controller.

Figure 8 below shows a flowchart of the pipeline for generating and testing our SL model based on ATCO's data input. The input to the either training or using the model are SSD images, while the output at this stage is a text file displaying the Time of prediction, Callsign, and the Heading change suggested by the SL agent. The idea is to integrate these variables with SectorX to efficiently show the model's assistance to the ATCO. Since SectorX is developed in the Java framework, these outputs from the agents can be illustrated in real time with suitable warnings or animations to make sure the ATCO sees this information.

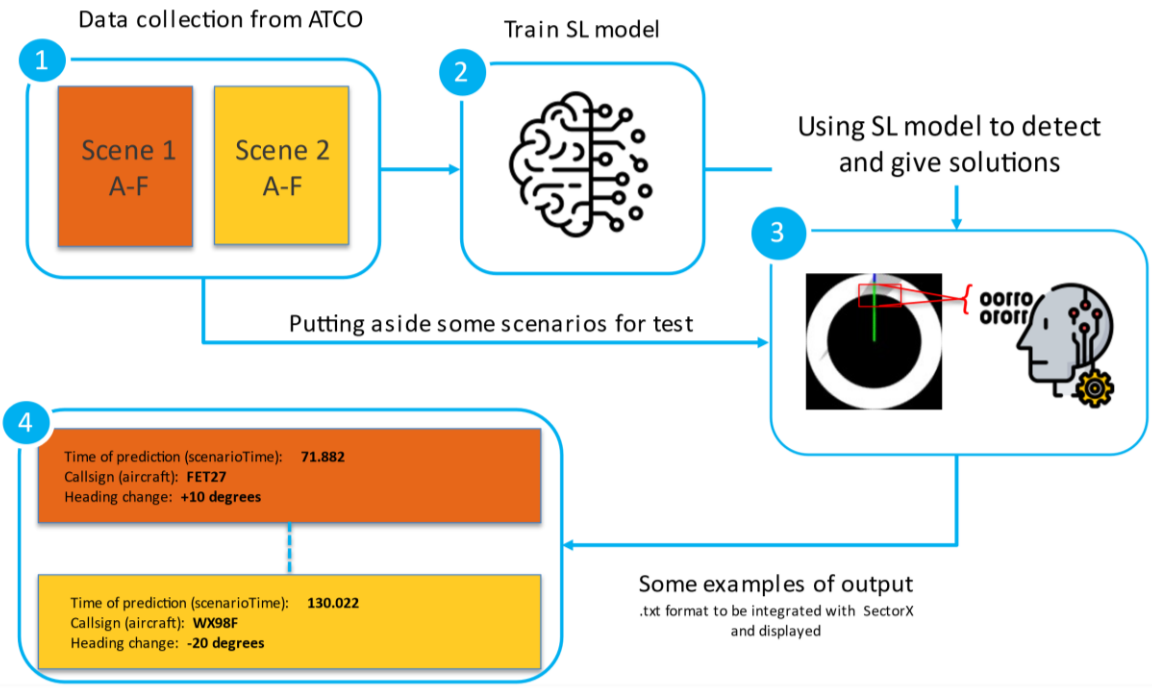

Figure 8: Sim1 SL agent generation, training, and usage pipeline

4 Data collection and processing

4.1 Data Collection

Throughout the simulations, the main data collected has to do with the ways the operators interact with the system. *SectorX* is able to keep track of both the aircraft and system states as well as the interactions of the operator with these systems. For example, *SectorX* is able to track and log all clicks the operator performs which helps in determining which aircraft the operator is looking at for further analysis. All this data is, thus, brought together into the two different logs mentioned earlier: the **StateLog** and the **EventLog**. Within these two files all the information about the simulation is contained and can then be processed. Details on the format of these two files can be found in MAHALO deliverable 5.1.

It should be mentioned that processing in SIM1 is necessary mainly for the conformal agent. The data from the operator's interactions with the system are converted into SSD information that can be fed into the conformal agent pipeline such that a conformal system can be properly trained.

Data is also collected on the level of agreement and acceptance of the operator towards the solutions proposed by the agent. This data helps measure the effects of transparency and conformance on these variables. Of course, only in Sim2a/b can these measurements be taken full advantage of since only then will the experiment be divided into a conformance pre-test and a final test, thus allowing for a conformal agent to be trained. However, for SIM 1 it is important to verify that this part of the data collection process is also functioning.

4.2 Privacy and data protection

Since the data collected in SIM 1 involves human participants, the data collecting and processing stages adhere to certain principles such that the proper privacy and data protection measures are in place. For a description of these measures please refer to D8.1

4.3 Augmenting data

A common problem with most ML algorithms is that they usually require a great deal of data to train. Considering that, within MAHALO, one of the goals is to train an agent to be conformal to a specific operator, it can be seen that having a lot of data is indeed important. Of course, ideally, we would have access to thousands or even tens of thousands of examples of resolutions given by each operator. This would allow the agent to learn more effectively and perhaps even pick up information on over-arching strategies.

Having thousands of data points is, however, extremely unrealistic. This would require hours upon hours of data collection experimentation for each operator. For any operator this would be unfeasible, especially so for ATCOs that already are quite busy and hard to access due to the critical work they do.

Therefore, one has to use the few data points collected as effectively as possible. This can be done by a process known as data augmentation. This augmentation usually takes the form of, for example for images, several rotation, scaling and inversion operation. Data augmentation allows for a more efficient use of data and is especially useful when one is faced with low data availability, as is the case in MAHALO.

5 Analysis of SIM1 results

The main purpose of SIM 1 is to provide a proof-of-concept experiment where MAHALO's approach can be proven to work. For this, all the different subsystems the MAHALO team has been developing, interfaces, converters, ML agents and ATM simulator, have to be tested separately and together.

Firstly, each of these systems is tested separately to verify if they function as required. Then they are integrated together in order to make up the full MAHALO approach. As in previous work-packages, most of the individual systems had been tested and validated, SIM1 is mostly concerned with testing and validating the integration of these systems.

The high-level success criterion for SIM1 is for the system to be able to perform a HITL experiment. This encompasses, in the case of MAHALO: interaction of the operator with the ATM simulator, collection of data from this interaction, training of the ML agents, ability to provide recommendations based on each of the ML agents to the operator and evaluation of the operator's performance and acceptance.

Several different factors will be looked at to determine the success of this HITL experiment. Namely, making sure that the different ML agents have adequate performance, that data can be shared and converted into the proper formats for the different systems and that the integration of all subsystems has been successfully carried out.

SIM 1's results will also provide a way to gauge performance of the overall system, in general, and of the ML agents, in particular, for a task involving a human operator. The results from this SIM1 will then be used to analyse and improve upon the approach used.

Section 5.1 will provide the outcomes and lessons learned from the SIM 1. In section 5.2 these outcomes will be summarized in a table that presents the success criteria and their outcomes.

5.1 Outcomes and lessons learned

Integrating all of these systems required developing quite a lot of algorithms to interface between and convert inputs/outputs into the different formats required by each system. Throughout this process quite a few challenges had to be overcome.

Scenario design is extremely important as a way to ensure that the proper data is collected. For example, since only horizontal conflict resolutions are learned by the ML systems, it is preferable to have situations in the scenarios displayed to the participants that only include horizontal conflicts or that can be more adequately solved through a heading change. This allows for more data efficiency, since if altitude commands are given throughout SIM1 these commands will not be usable by the different ML agents. Therefore, scenarios have to be carefully designed such that these requirements are fulfilled while maintaining an adequate level of realism.

One of the main inputs to SIM1 was the different discussions throughout the project within the project team itself and with outside partners. Namely, a workshop was held where several ideas on scenario and experiment design were put forward by people from the industry. These ideas were valuable to not only design SIM1's scenarios but also to fundament certain design decisions within the scenario design itself. Overall the response to the scenario design decisions was positive.

One of the main challenges has to do with the different formatting of data from system to system. The SL system, for example, requires an SSD as input and outputs a given solution that is in the form of a relative heading and absolute speed command. This output has to be converted into **absolute** heading and speed command as this is the form used by SectorX. Dealing with the formatting of the data it has been found that one helpful lesson is to keep most of the data to a given standard. For example, the SL system was considered to also include a converter that makes sure the output command is in the standard form (absolute heading and speed, instead of relative heading and absolute speed), this makes it easier to exchange data between the different systems. Developing these standards is helped by the fact that both BlueSky and SectorX have relatively strict demands on the kinds of inputs they can receive, which already forces a certain conformity between software developed.

Another challenge has to do with using two different ATM simulators. Sometime was dedicated to performing simple test runs within BlueSky and SectorX such that the behaviour of the simulators can be compared. This is important because different ATM simulators can use different approaches to parameters such as for example: weather, aircraft performance models, atmospheric model used and the *dt* used for the update of the performance model itself. Since the standard BlueSky and SectorX performance models are fairly similar, the only alteration needed was to artificially cap the maximum bank angle of the aircraft within BlueSky to match that of SectorX. This is because, by default, BlueSky allows aircraft to achieve a higher bank angle which translates into different turn dynamics and a different behaviour of the aircraft. The lesson here being that, due to the availability of different parameters to be used within an ATM simulator, it is extremely important that, if one is to use different simulators, the behaviour of the aircraft is the same between them. It is clear that, if the RL agent is trained to be very optimal for an aircraft that behaves a certain way, that that behaviour will not be as good if, when the commands are translated into SectorX, it produces a different aircraft trajectory. These issues could be somewhat mitigated if, for example, BADA models were exclusively used throughout both simulators. This was not possible due to licensing limitations of these models. Therefore, within BlueSky the *OpenAP* models were used and within SectorX some included, simplified performance models were used.

Several modifications had to be made to SectorX to ensure proper functioning throughout the experiment. Most of these modifications have to do with the fact that the participants of both SIM1 and Sims 2a and 2b are not expected to be familiar with the software. This means that SectorX has to be able to handle errors the end-user might make while using the software. Most of these involve making the software more "rigid", for example, by not allowing users to change out of a scenario playlist once it has started.

5.2 Success criteria and outcomes

This section will summarize the outcome of SIM 1 in terms of a set of success criteria and their outcomes.

Table 3: SIM 1 success criteria and outcomes

#	Category	Success criterion	Outcome
1a	Scenario development	Scenarios can be created in the SectorX simulator	**Success**
1b		Scenarios can be converted to the Bluesky simulator	**Success** The conversion is not exactly one-to-one due to some differences in implementing reference frames in both simulators. However, the differences have been found to be so small that they will not affect the experiment.
1c		Scenarios are deemed realistic enough by Air Traffic Controllers	**Success** Although some simplifications had to be made for training the ML models, feedback from Air Traffic Controllers on the designed scenarios was positive.
2a	Interface ML/HMI	Scenarios can be played in the SectorX simulator	**Success** Playlists have been created that will automatically run sequence of predefined scenarios.
2b		Human Air Traffic Controllers can interact with the scenario, i.e. execute commands.	**Success**
2c		Data collected from the interaction with scenarios is logged for later use by the ML models	**Success**
3a	RL model training	RL Model constructed and is able to receive scenario files and provide commands	**Success**
3b		Model training curve converges and shows stable behaviour	**Success** Model is sensitive to hyperparameter tuning and required manual tuning
3c		Model is capable of outputting solutions to be	**Success**

		used for the Final Stages of the experiments	Model outputs commands in a format that can then be converted into *SectorX* eventLog
4a	SL model training	SL Model is constructed and is able to receive and process SSD data	**Success**
4b		SL Model training on virtual data and shown to converge to stable performance	**Success**
4c		SL model can provide solution advisories in the form required	**Success**
5a	Replay mode	Scenarios can be replayed to the human Air Traffic Controller with the ML resolutions activated	**Success**
5b		Three levels of transparency in presenting the ML resolution are available	**Partial success** At the time of SIM 1, the levels of transparency were not fully defined. There were the 'resolution only', the 'resolution with SSD' and the 'resolution with explanation' options, but the content of the last option was not defined. Note: between SIM 1 and SIM2a this has been resolved.
5c			
6a	Acceptance interface	An acceptance/agreement window should pop up after an ML resolution is presented	**Success**
6b		Data from the acceptance/agreement window must be logged	**Success**
6c		Additional questionnaires must be presented, between scenarios and at the end of the experiment	**Partial success** The technical options to present questionnaires were worked out, but the questions themselves were not defined at the time of SIM 1

6 Conclusions

This report has described SIM1, performed by the MAHALO project team as a validation of the work performed so far. It was found that the different pieces of software that have been individually developed throughout previous work packages work adequately together when using the different interfaces developed.

The use of two different ATM simulators created challenges regarding different simulator functioning that were adjust to ensure uniform behaviour.

Scenario design is a topic that should not be understated when it comes to its importance, it was also one of the main challenges faced during the lead-up to Sim 1. Sim 1 provided the first chance to test the scenarios designed throughout the project and provided guidance for Sim 2a and 2b. The feedback provided during the workshop on October 28th 2021 reinforced the idea that the scenario design was not only adequate for data collection but also realistic enough that it would not seem out of place to ATCOs. There are, of course, limitations to the scenario design that MAHALO is well aware of.

Sim1 was ultimately successful in that it provided MAHALO with proof that the approach chosen, from the technical perspective, works. Having all of the background pipeline properly working now provides the MAHALO team with the opportunity of focusing on Sim2a and 2b's data collection and analysis such that the approach can be validated from the human-machine interaction perspective.

The challenges faced throughout Sim 1 have also provided perspective on what topics should be focused on for improvement and modification ahead of the next simulations.

7 References

[1] MAHALO Deliverable D3.1 Machine Learning report

[2] MAHALO Deliverable D4.1 E-UI design doc & demonstrator

[3] MAHALO Deliverable D5.1 Integration report