Air Traffic Control automation: integration with a simulation environment for the evaluation of a Decision Support Tool

João Pereira
Instituto Superior Técnico
Lisbon, Portugal
jpmp9@hotmail.com

Rodrigo Ventura
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
rodrigo.ventura@isr.ist.utl.pt

***Abstract*—Because of the significative air traffic increase that has been occurring in the past decades, automation of Air Traffic Control (ATC) operations can be considered in a long-term as a method to cope with this problem. The development of Decision Support Tools (DST) to help ATC Operators (ATCO) diminish their workload is an intermediate step between a possible future of having a predominantly automated system and a mostly manual, human-controlled one, as in the present.**

In the sequel of a previous MSc thesis at IST, a DST for controllers of enroute positions was made. The DST receives a flight scenario and finds an optimal solution that prevents conflicts until the end of the aircraft trajectories within a given Flight Information Region (FIR). Simulations were conducted with various scenario files with different traffic density and complexity and the DST proved its robustness.

In order to transform a concept into a DST that is fully integrated with the ATCO system, there are numerous stages. This thesis aims at looking into two of those stages: transform the previously created DST into a real-time, integrated with an air traffic simulation environment DST; test it with ATCOs to evaluate its usability. A plug-in was made in order to make the bridge between the DST and the air traffic simulator and the specific problem of oceanic airspace was taken into account as a case study.

***Index Terms*—Air Traffic Control Automation; Decision Support Tool; Oceanic Control; Plug-in.**

I. Introduction

If air traffic demand grows in the next decades as predicted in [1], the capacity for managing considerable higher traffic than today needs to be provided. The upper bound capacity is directly correlated with the Air Traffic Control Officer's (ATCO) workload and to keep it at an acceptable level, automation in the form of Decision Support Tools (DSTs) need to take a more active role in Air Traffic Management (ATM).

The current strategies for automation differ on the level of involvement that they require from controllers. The best ones in near-term are still human-in-the-loop approaches, where the DST includes the ATCO in the decision loop, thus making the whole process more trustworthy in the ATCO's eyes. A study on ATCOs behavior [2] shows that by completely removing the human from the control loop, entrusting him with the role of passive monitor only, automation can hinder the operator's ability to maintain an adequate mental representation of the system, as well as inhibit his ability to quickly reassure manual system in case of emergency – this phenomenon has been referred to as *out-of-the-loop unfamiliarity* problem [3].

An interactive DST inserted in the human-in-the-loop automation strategy, supervising the cruise phase of flights was done as part of a thesis in Instituto Superior Técnico [4] in 2013. It detects potential conflicts inside a given Flight Information Rules (FIR) and proposes Air Traffic Control (ATC) instructions to the operator that prevent the referred conflicts. The given instructions are only in the form of Flight Level (FL) changes and delivered at report fixes only. The DST was developed using the Python language and its validity was assessed through experimental data. Relevant work on the interactive, human-in-the-loop approach can be found in [5] and [6], [7], these last ones with partnership of EUROCONTROL and NASA respectively.

The ATM system is a 24-hour, 7-day-a-week operation and any technological improvements need to be fully scrutinized before implemented. Thus, the stages and time behind the transformation of an idea, a concept into an actual DST validated to be used by real controllers in real ATM are often very numerous and lengthy. A description of the stages for validation of a new decision support system is made in [8] and the conclusion is that the validation of the early stages is of extreme importance since it enables the study of how a new system of people and machines will behave, before reaching a situation when it would be very costly to go back or to make changes.

This paper proposes to take the previously made DST into the next steps of the validation process, necessary for every DST present in the ATM system, focusing on two of those important stages: first evolve it from the offline and limited DST that was previously created into an online and fully integrated – in the form of a plug-in – with a simulated ATM environment DST; then, test it with ATCOs to attest for its functionality.

Regarding the plug-in making phase, some consid-

erations must be taken into account. It was shown – again in the study in [2] – that even in human-in-the-loop approaches, where humans are involved in the control loop, special attention when designing the DST is needed in order to truly benefit the ATCO's workload. If not, the controller, for lack of trusting and understanding about the DST, will run a simultaneous self-derived solution and a subsequent comparison with the automation's solution which will increase the mental workload. It was demonstrated in [9] that an automation aid can go unused if the cost of initiating the aid, considering its advice, generating one's own solution and comparing solutions becomes too high. More studies on the human side of automation and on the effects to performance and mental workload of ATCOs can be found in [10], [11], [12] while the role that controllers shall play in the Next Generation Air Transportation System (NextGen) is discussed in [13].

Thus, attention was paid when building the plug-in: making it the maximum intuitive as possible and adding some pre-simulation high level explanation about the DST's way of finding solutions. This way ATCOs, by working with a cognitive tool and having some insight about the DST, are more likely to trust the DST's provided solutions and to find them transparent, i.e. easy to determine if they are correct or not.

The structure of this paper is as follows. Section II describes the functionalities of the DST and the framework used in its designing as well as the interaction of that same DST with the plug-in made as part of this paper. Some improvements made to the DST and the found limitations are also herein discussed. In section III the chosen simulation environment is presented and the plug-in's interface is described. The simulation parameters and the results of the experiments undergone with the ATCOs are presented in section IV while the main conclusions coupled with suggestions for future work may be found in section V.

II. System Architecture

A. DST

Some assumptions were made in order to define the simulation framework used in the DST's algorithms:

- **Airspace**: The special case of oceanic airspace, where conventional ATC is still used due to the lack of radar coverage, was used as a case study and the chosen FIR where the simulations were to take place was Santa Maria (the International Civil Air Navigation (ICAO) code for Santa Maria FIR is LPPO). As a simplification, LPPO's radar was ignored.
- **Minimum and maximum FL**: In LPPO, as in most of the world-wide ATC FIRs, the Reduced Vertical Separation Minimum (RVSM) rules are applied and so the minimum FL where the DST allocates aircraft was FL290 (29000 ft) while the maximum was FL410.
- **Tolerance for maximum FL**: In general, the RFLs in the FFP are respected by ATCOs and are not exceeded. However, in certain cases, the pilots may accept – or even request – flying at higher altitudes than the RFL e.g. when taken off weight is lower than expected. DST's way of dealing with this was to set a tolerance of FLs in which the aircraft could be allocated past its RFL. For example if the RFL of an aircraft is FL380 and the tolerance is set to 2, the maximum FL where the aircraft could be allocated would be FL400. The used tolerance was of 1 level.
- **Type of instructions**: Because of the lack of radar coverage in most oceanic airspace and so of the possibility to access the aircraft's position in real-time, horizontal and speed change maneuvers are rarely used by ATCOs as they are not effective when comparing to continental radar ATC. On the other hand, vertical maneuvers – backed with the modern aircraft autopilot's ability to measure and maintain a precise altitude – provide a safe way to ensure separation, even with some horizontal position uncertainty. Thus, most detected conflicts in oceanic airspace are solved by ATCOs resorting to vertical instructions and hence the developed DST utilizes vertical instructions only when solving conflicts.
- **Period for instruction issuing**: In oceanic ATC, besides the mandatory position reports that have to be done by pilots, the communications between ATC and pilots are infrequent. Thus, it is common practice in oceanic control to issue instructions to aircraft only when they pass route waypoints instead of at random route points. The DST was made so as to comply with that common practice.
- **Separation minima**: Since the chosen portion of the airspace was one where the RSVM rules apply, the chosen vertical separation minima used to detect conflicts in the CD&R algorithms in accordance to [14] and [15] were:
  - Vertical separation minimum: 1000 ft
  - Horizontal separation minimum: 50 NM
- **Aircraft models**: All aircraft were simulated using the Airbus A320 model.

As for the routines made for the DST, they can be divided into two categories: **trajectory prediction** and **conflict detection and resolution** (CD&R).

The purpose of the first ones was to detect the future possible conflicts between the multiple aircraft present in the simulation, through the use of built-in information from the Base of Aircraft Data (BADA). The algorithms use spherical coordinates with quaternion algebra to account for the curvature of the Earth and so that the aircraft constantly adjusts its heading angle in order to follow a Great Circle path between waypoints. When it comes to dealing with unpredictable factors, the trajectory prediction algorithms take into account the uncertainty that wind causes on the ground speed and therefore on the Estimated Time of Arrival (ETA) at waypoints, being

the wind modeled assuming a Gaussian distribution.

As for the second set of routines – the CD&R ones – their function was to find the global set of plans that solve a given traffic scenario i.e. that allow the completion of the simulation of a traffic scenario without any conflicts happening. The detection of conflicts is done by analysing at any given moment the horizontal and vertical separation minima for each pair of aircraft in the traffic scenario. If both are being violated, the aircraft-pair is assumed to be conflicting. In the end, every conflict-free plan is further scrutinized by running a robustness check consisting of a Monte Carlo simulation with random wind as the uncertainty model.

In order to evaluate and rank all found plans, and thus allow the best one (*optimal solution*) to be presented to the user, a cost function was constructed. Taking into account both the controller's workload and the concept of efficiency[1], the chosen criteria to try and lower were:

- **Number of instructions, *N*:** Including the total amount of instructions that are issued by an ATCO within a given time window, in the cost function serves the purpose of keeping the ATCO's workload at an acceptable level.
- **Deviation cost, *D*:** So as to minimize the distance flown by aircraft in other FL than the Requested Flight Level (RFL) and thus maxime efficiency, vertical deviation from the Filed Flight Plan (FFP) was measured and included in the cost function. Cost *D* depends on the distance between waypoints where the instructions take place and on the vertical distance between the RFL and the flown FL. E.g. figure 1 where the blue line indicates the trajectory requested in the FFP and the red line indicates the trajectory change that is ordered by the CD&R algorithm and that transforms the FFP in a different, Current Flight Plan (CFP). The grey area corresponds to the deviation integral – cost *D* is the sum of all deviation integrals in a given solution – that is essentially the multiplication of the distance between waypoints *43N020W* and *43N030W* by the difference between the RFL and the flown FL (1000 ft).

Finally, the chosen cost function was:

$$f = \lambda_N N + \lambda_D D \tag{1}$$

where λ_N and λ_D are weight coefficients.

So as to reduce the computational time while still finding an *optimal solution*, branch-and-bound algorithm with a depth-first configuration was used. After the first solution plan is found, upper bounds restricting criteria *N* and *D* – N_{max} and D_{max} respectively – are available, and may be used to prune branches, diminishing the computational time even more. At this phase, with two criteria to try and lower, the concept of *optimal solution*

[1]Efficiency is a measure of how closely an aircraft flies its designated flight plan, as defined by Krozel et. al in [16].

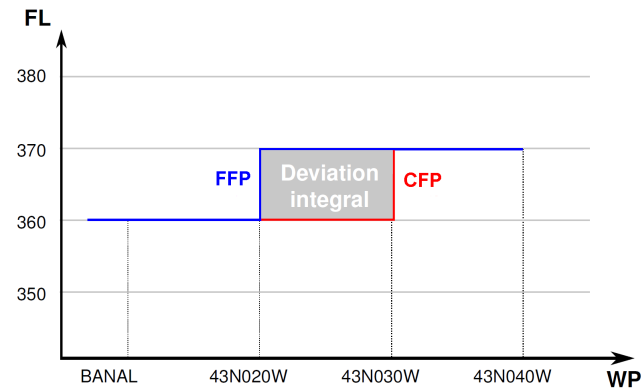


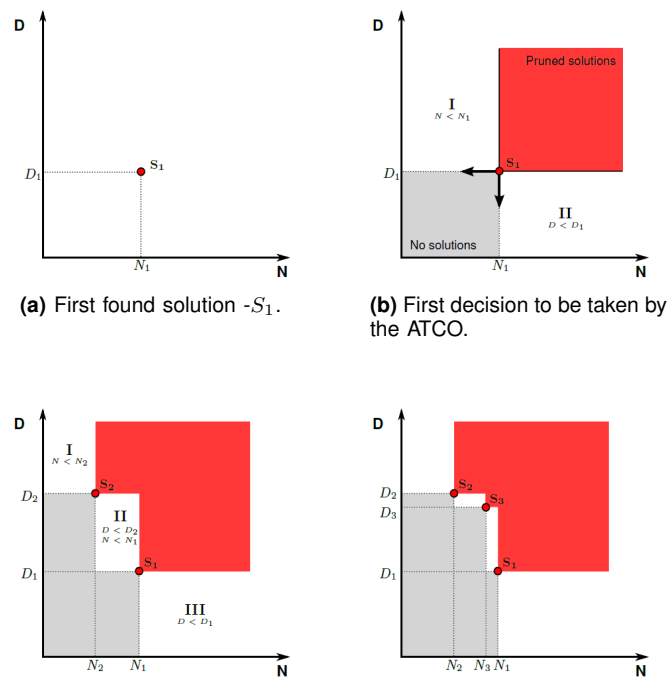
Figure 1: Deviation integral illustration.

for a given problem vanishes since a solution might minimize one criteria but not the other. Instead, *Pareto optimal solutions* (also known as *Pareto front*) came into place and a strategy regarding how these solutions were calculated had to be chosen. Between the following three categories of strategies, that differ on how they calculate the *Pareto front* and on how much the user has to be involved: *a priori* methods, *a posteriori* methods and interactive methods – the last one was chosen.

In the interactive approach, the algorithm alternates between phases of calculation and phases of interaction with the user, allowing him to participate in the optimization process by answering some algorithm's queries. The main advantages of this approach are that, unlike the *a posteriori* one, it avoids the calculation of the whole *Pareto front*, thus saving both time and memory, and that it allows the user to set his preferences based on some knowledge about the *Pareto front*, unlike the *a priori* approaches. On the other hand, the main flaw of the interactive methods is that when the user states his preferences, his knowing about the *Pareto front* might not be sufficiently broad, which can ultimately lead to settling for solutions that are local bests and far from global bests.

To better understand how the *Pareto front* equipped with the interactive approach works, let us observe figure 2. So, at the beginning, N_{max} and D_{max} are initialized as infinite since no restrictions exist yet. The branch-and-bound algorithm starts working and assuming that the given scenario is solvable, a particular solution S_1 with N_1 instructions and D_1 deviation cost is found – figure 2a. At this point the user is confronted with the first decision: either accept the solution S_1 or request the calculation of more inside region *I* (where $N < N_1$) or inside region *II* (where $D < D_1$), both depicted in figure 2b. Note that in this figure (as well as in figures 2c and 2d) red areas correspond to the *Pruned solutions*, where solutions have more *N* and more *D* than the previously found solutions, thus deterring them from belonging to the *Pareto front*. The grey regions represent areas where no solutions to the scenario can be found because they would have less *N* and less *D* than some solution in the *Pareto front*, condition that would guarantee that they would have been

returned by the algorithm instead. E.g. if there was any solution in the grey area of figure 2b it would have $N < N_1$ and $D < D_1$ which would cause the return of that same solution instead of S_1.



(a) First found solution -S_1.

(b) First decision to be taken by the ATCO.

(c) Second found solution - S_2.

(d) Third found solution - S_3.

Figure 2: Pareto optimal set calculation.

Supposing now that the user rejects the given solution S_1 and requests another with *N* smaller than N_1 – region *I* of figure 2b. A second *Pareto front* solution S_2 is then found – figure 2c – and presented to the user. Now, there are three possible regions to look out for more *Pareto front* solutions: region *I* (where $N < N_2$), region *II* (where $D < D_2$ and $N < N_1$) and region *III* (where $D < D_1$). Presuming that the user is not yet happy with the solution and wants to explore region *II* now, takes us to a solution S_3 – figure 2d. At this point, the user is happy with the resulting solution and accepts it, which causes the interactive algorithm to exit.

The study of the results coming from the simulation experiments showed that the running time t_{run} evolution with the number of flights in simulation was exponential what might constitute a problem in the scope of this paper, while memory usage was not an obstacle since it presented linear behavior.

A high variety of simulations with different traffic density and complexity were conducted and the DST proved its robustness. The results were considered very good and the DST seemed ready for the next step, i.e. test it with ATCOs.

B. Improvements

In order to better interact with the plug-in and so that the experience for the ATCOs was the most alike real ATC as possible, some improvements were made to the DST's algorithms:

- Before, the simulation time was set to be at 0800Z (Zulu time). When integrated with a simulation ATC system, that parameter need to be variable and so, the DST was changed to receive every possible start time from the plug-in.
- The Airbus A320 aircraft model was the only one considered in the simulation. As the replication as much as possible of real ATC system is the objective, having only one aircraft model detaches some credibility of the scenario files and thus efforts were made in order to have a wider variety of functional aircraft models. Common models for aircraft flying oceanic airspace were considered and 10 more of these models were added – by devising the appropriate Flight Management System (FMS) gain parameters file for each one. A total of 11 aircraft models (including the A320) were used in the simulations which led to more diversified and realistic scenario files.
- During the testing phase of the DST, t_{run} when searching for an optimal solution was found to be quite big in some cases. E.g. for a scenario consisting of 40 aircraft the mean value for t_{run} was approximately 630 seconds. Test the DST with ATCOs and allow for that kind of wait for a solution is inconceivable since a real controlled environment can change considerably during that waiting period. The solution consisted on applying an anytime algorithm to the DST. Each loop of the DST's solution finding would be tested to check if a pre-defined t_{run} was already reached. If so, and if there was one or more found solutions (non-optimal ones), the algorithm forced the return of the last found (and better for that matter) solution. Else, if t_{run} was passed but there were not solutions yet, the DST's solution finding kept running until a solution was found, moment where that same solution was returned.
- Many other minor alterations were done in order to correct some things or to adapt the DST to the plug-in, e.g. the border points connecting Piarco FIR and LPPO did not exist so they were added in the appropriate routine; the visual part of every DST's algorithm was removed since it would be of no use to the C++ user, also reducing the running time in a small, but important, amount.

C. Limitation

Some limitations were encountered in the DST. For some, workarounds were able to be found but for a specific one, hindering the use of the interactive part of the DST, no solution was established.

The limitation was as follows: when tests to evaluate the DST-plug-in connection began, it was noticed that every time a restriction in criteria *N* or *D* was introduced, the algorithm exited abruptly referring to a coding error in the DST's interactive algorithm. Attempts to solve this

bug were extensively made by me and by the own creator of the DST, all having the same result: insuccess. Thus, the bug stood as a somewhat big limitation to the paper since it ruled out completely the use of the interactive solution that would be one of the most interesting aspects to evaluate the human and DST interaction.

The initial objectives for the plug-in were redefined at this phase and from this point on the work made as part of those initial objectives are discarded and not showed in this paper. Instead, only the plug-in made within the redefined objectives is going to be presented and discussed.

D. Plug-in and DST communication

Due to the previously discussed interactive limitation, the connection between the plug-in and the DST had to be simplified to the one presented in figure 3 where we can see that the outputs of the plug-in are:

1) **Scenario:** it provides the DST with the information about the current time, the total number of aircraft present in the actual scenario as well as the flight plans for every aircraft.
2) $\frac{\lambda_N}{\lambda_D}$**:** the criteria weight ratio. The two weight parameters may be adjusted to benefit one of the criteria over the other: specifying $\lambda_N \ll \lambda_D$ causes the CD&R algorithm to instruct aircraft to fly as closely to their FFP as possible, no matter how many instructions it takes; setting $\lambda_N \gg \lambda_D$ causes the algorithm to choose the solutions with as few instructions as possible, neglecting vertical deviation from the RFLs. The criteria remains constant throughout a session.

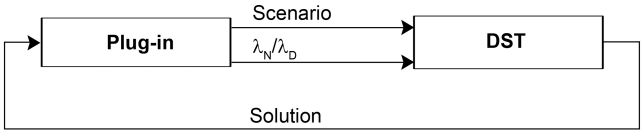


Figure 3: Scheme of the communication between the plug-in and the DST.

III. Simulation Environment and Plug-in Description

A. Simulation environment

There are many ATM simulation tools currently available. One that is considered excellent and largely used by virtual ATC communities is Euroscope. It is also the one (version 3.1d) that has been chosen in this paper since it is very much alike the one ATC uses nowadays. A print screen showing the main window of Euroscope with a simulation of the LPPO behind and with a couple of child windows is depicted in figure 4.

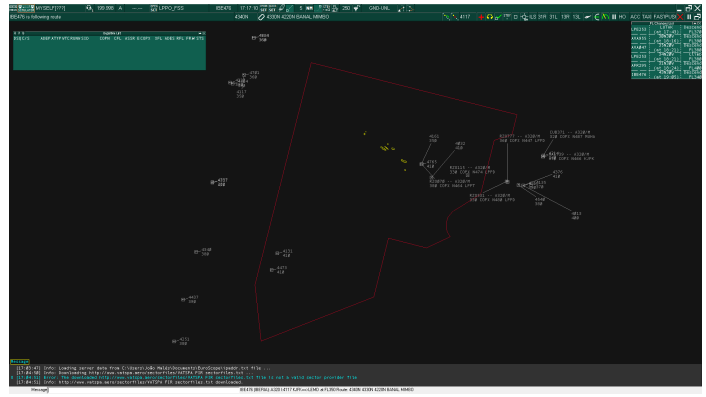


Figure 4: Print screen of the simulation environment Euroscope.

B. Plug-in's computational languages

Euroscope is programmed in C++ using Microsoft foundation classes. That is why, in order to develop a .dll plug-in that could be loaded into any computer running Euroscope, Visual C++ in Microsoft Visual Studio 2012 was chosen. Python 2.7.4 was also extensively used to make the discussed alterations in the DST – section II – and to make the bridge routine that called the multiple DST's algorithms and properly handled the responses from them.

So as to connect the two programming languages the alternative of embedding Python in C++ using the Python/C Application Programming Interface (API) was used. Some of its advantages over other reasonable options are:

1) Fairly intuitive;
2) It involved an API made by Python specially for that matter, which is always good since Python reference manuals are generally very thorough;
3) Python scripts do not need to be opened unlike for the sockets strategies;
4) There is no need for Python to be installed in the used computer – portability;
5) The connection between the two languages is achieved really fast, almost instantly (measured at 0.0093 milliseconds).

C. Plug-in's interface

At the top-right corner of figure 4 with title *FL Changes List* there is the plug-in's window made as part of this paper and one can see that it has the same style as the other child window present in Euroscope at that time, *Departure List*. This window is movable and in figure 5 we can observe one of these plug-in windows in more detail.

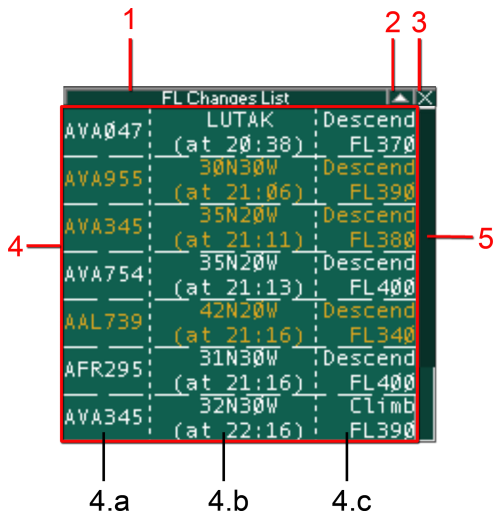


Figure 5: Interactive plug-in's interface

Looking at that detailed image of the plug-in, multiple features can be observed. They are hereafter explained:

- **1:** title bar with a brief description of the plug-in's intention. It is also the only grabbing area for when the user wants to move the window;
- **2:** minimize/maximize button;
- **3:** close button;
- **4:** display area of the plug-in where one of the following options is in use: the start button, the set of instructions or a message indicating that either the scenario has no solution, no conflicts were detected or that any other error was encountered. In case the message indicating that there is no solution is showed, that message is accompanied by the pair of aircraft precluding the resolution of the scenario with the current set of conditions. This way, the ATCO can still solve a conflict via giving instructions in non-standard positions i.e. not only after waypoints reports like the DST was intended to.
  As for when the set of instructions is displayed, those instructions are ordered by time of occurrence and the ones shown in yellow, instead of the regular white, indicate that the user has already carried them out. In the specific case of figure 5 the set of instructions are in display and each instruction is separated by the following fields:
  - **4a:** callsign of the aircraft in question;
  - **4b:** waypoint where the instruction shall take place and the predicted hour of arrival at that waypoint;
  - **4c:** instruction, on whether the aircraft is supposed to descend or climb, followed by the resulting FL of that instruction;
- **5:** scroll bar, that only appears when the number of instructions is superior to a pre-defined value and that has variable size depending on the total number of instructions.

Note in figure 5 that there are two instructions for aircraft AVA345: first descend to FL380 at waypoint *35N20W* and then climb back to FL390 at waypoint *32N30W*. However, the controller will not have the opportunity to deliver those two instructions to the aircraft in the duration of a cycle of instructions since, as mentioned before, in the aircraft's next waypoint another call to the DST is made and another set of instructions is presented – if the conditions of the simulation change only accordingly to the predicted instructions changes, the second instruction (climb to FL390 at *32N30W*) is then going to be given as the first instruction to aircraft AVA345.

Rather, future instructions are also showed in order to give controllers a correct idea about the size of the current set of instructions. Also, it allows ATCOs to conceive a better mental image about the future work that they will have if they adopt this or that DST's advice.

D. Plug-in and DST connection periodicity

We have talked about the first time a call to the DST is made. Now, let us talk about the other hundred times per session that this communication is needed.

Ideally, a new call to the DST was to be made when:

1) A new airplane entered the simulation;
2) An aircraft crossed some of its route's waypoint.

and the information coming from those calls was to be displayed for a reasonable amount of time.

However, and even with the implementation of the *anytime algorithm* – subsection II-B – the DST running time was often impracticable. E.g. in a scenario with 30 aircraft where the average time window having any aircraft crossing a waypoint was 2 minutes and the DST's average running time was 3 minutes, it was impracticable to call the DST in every situation of the two mentioned above because it would seldom present the solutions in the plug-in's window and rather the message "Finding solution". In other words, it would spend more time in discovering the solutions than in presenting them which is not of course the intention for a tool that aids decision.

Between the multiple strategies that were thought of to solve this problem, the implemented one was the following:

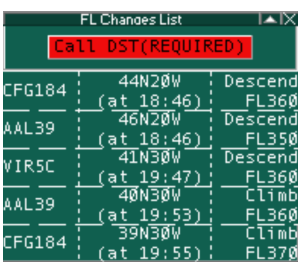
- Let the ATCO decide when to call the DST, using a button created for that effect. While a new solution is being sought the last valid solution is showed in the plug-in's display area and the created button is not available for the controller to press.
  On the upside, this approach: is less chaotic than the other considered strategies since the DST does not need to be constantly called in crowded scenarios; it allows for a better approximation of the real conventional ATC, since in it controllers have to receive the aircraft positions, input it in the system and only after shall they give the instruction to the airplane. With this strategy, the controller has to manually call the DST after issuing an instruction to an aircraft, better approaching the amount of workload needed at each waypoint in real oceanic ATC.

On the other side, the disadvantage is that the presented solution could be outdated and worthless, causing the ATCO to give instructions that should not be given.

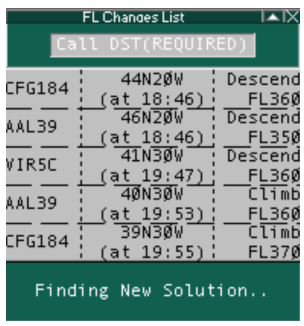
Some alterations were made to the plug-in's interface in order to accommodate the new requirements. Those alterations and the resulting interface are then discussed.

First, a button on top of the display area titled "Call DST" was created. It allows for new calls to the DST and whenever there is new information – i.e. when conditions 1 or 2 are met – the button turns red and the text is modified to "Call DST (REQUIRED)", so as to notify the responsible controller that a new call to the DST is advised – situation depicted in figure 6a.

The other modification is that while finding a new solution the last valid one is showed in different color and background so as to allow the ATCO to recognize it as being old. Also, the "Call DST" button becomes grayed out i.e. not available to use until the current finding returns a solution – figure 6b. While the button is grayed out it is still updated and thus it can be grayed out and indicating that a new call to the DST is necessary after the conclusion of the current search.



(a) Detail of button "Call DST (REQUIRED)"



(b) Detail of showing the last valid solution while making a new call to the DST

Figure 6: Details of the final plug-in's interface version

The windows depicted in figure 6 constitute the final version of the plug-in and hence the one presented to the ATCOs.

E. Additional window for Euroscope

Besides the previous plug-in interface, designed to be used only when ATCOs want DST's help, an additional window was made for controllers to use whether they are controlling a simulation with or without the help of the DST.

Like discussed before, in oceanic control, pilots are responsible for reporting each aircraft's crossed waypoint to the ATCO in charge via voice communication. In Euroscope however, that would be possible only by assigning each aircraft to a trained person entrusting them of being the pilot and making the voice communications – such is done in virtual air traffic simulation. That would of course require a big amount of human resources in relevant tragic scenarios for our purposes i.e. where there can be 40, 50 or 60 aircraft at once in the scenario.

So as to replace the voice communications for something with the same valuable information and thus help controllers by not compelling them to constantly check for each flight's position in search of waypoint reports, a window that constantly shows and updates the last reported waypoints was created.

An example of one of these referred windows is depicted in figure 7. Analysing that figure, from left to right, we observe that each report contains the following information: callsign of the aircraft in question in the left, name of the reported waypoint followed by the hour of that report in the middle and ground speed followed by the flight level at that reported waypoint in the right.

Each report stays available for 70 seconds, moment when it disappears from the list. The value of 70 was devised to allow controllers enough time to notice the report and act on it if necessary.

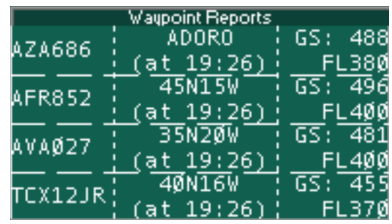


Figure 7: Waypoint reports window

IV. Simulation and Results

A. Simulation framework

To completely define the simulation framework used to test the DST integrated in Euroscope via plug-in, the same assumptions made for the DST in subsection II-A were used, with two exceptions:

- **Aircraft models**: As mentioned before, an effort to have a wider variety of functional aircraft models, instead of just the A320 used in the DST's simulations, was made. Thus, the available models were: A319, A320, A321, A332, A333, A343, A346, B77W, B744, B752, B763.[2]
- **Simulation duration**: A time window of one and a half hours for the DST was used.

Also, the value of 10 was used for the criteria weight ratio $\frac{\lambda_N}{\lambda_D}$. Therefore, more emphasis on having fewer instructions was given at the expense of having a somewhat bigger vertical deviation from the RFLs.

As for the generation of the traffic scenarios that are loaded by Euroscope, two approaches were followed: one that generates a random scenario based on a nominal flight plan pool and another that replicates a real traffic situation that occurred in a certain oceanic airspace at

[2]The models with first letter A like the A319 are Airbus while the ones with first letter B like B77W are Boeing.

a given time. In order to generate traffic scenarios as real as possible, both use real flight plans. The first by downloading the flight plan pool from the Data Repository Demand 2 (DDR2) of EUROCONTROL's extranet and the latter by using EUROCONTROL's modeling tool Network Strategic Tool (NEST) to download and export to readable content the numerous Aeronautical Information Regulation and Control (AIRAC) available cycles[3].

B. Ideal trials with ATCOs

The objective of the trials with ATCOs was to ascertain which would be the potential benefits and drawbacks of using the constructed plug-in in a simulated ATC environment. It would serve as an intermediate step while evolving the DST into one integrated with the real ATC world and used on the day-to-day of controllers.

Ideally the trials would be taken by several sets of four ATCOs, each set having the configuration depicted on table I. The conditions to take into account when attributing each experience framework were:

- **Scenario files**: *I* and *II*;
- **Conditions while controlling:** with the help of DST (*DST_ON*) and without the help of DST (*DST_OFF*).

The duration of each trial would be of four hours, which is half of the typical duration of a controllers' shift.

		ATCO			
		A	**B**	**C**	**D**
Trial	**1st**	*I* *DST_ON*	*II* *DST_ON*	*I* *DST_OFF*	*II* *DST_OFF*
	2nd	*II* *DST_OFF*	*I* *DST_OFF*	*II* *DST_ON*	*I* *DST_ON*

Table I: Scheme plan of the experiments with four controllers

In order to have interesting and reliable data to analyze corresponding to two scenario files, 12 or 16 controllers (three or four sets of trials respectively) would be enough. The reliable factor is related to the elimination as well as possible of the ambiguity aspect that is inherent to all experiments. In this case, the ambiguities' elimination can be pinpointed by understanding that the scheme's distribution of table I is not random. Instead, three conditions rule the distribution:

- Each scenario file is attributed only once to each controller, assuring that ATCOs do not supervise the same scenario file more than once, precluding a preconditioning in the results – coming from previous insight – on the second time the scenario would have been controlled.
- The order in which the DST is used is inverse for the first and latter two ATCOs. This guarantees that a better result in the second trial cannot be attributed solely to the experience that the controllers have gained from the first trial. I.e. if the first trials were all made with *DST_OFF/DST_ON*, a possible better result in the second round of trials could prone one to attribute that success to the bigger ATCO's experience on handling Euroscope, thus neglecting the fact that the DST had worked well or not – for having initial *DST_OFF* and *DST_ON* respectively.
- Each scenario file is going to be controlled in every possible situation, i.e. in both the first and second trials with both *DST_ON* and *DST_OFF.* That is why the defined number of ATCOs for a set was four and not five, three or two. Thus, final results' conclusions cannot be attributed to scenario *I* being much easier or much harder than scenario *II.*

In the other hand, to turn the experiment into one with enough dimension to withdraw statistical significance, the more human resources the better of course, but something along the lines of testing 4 pairs of scenario files and hence 48 to 64 ATCOs would be sufficient.

C. Pilot trial

Since the necessary human resources to make this experiment into one with valid statistical results were fairly big, and therefore outside the range of our capabilities to get, a pilot trial serving as a proof of concept was conducted.

The trials were planned for two real ATCOs who made themselves available in their free time for participating in the experiences but incompatibility in schedules coupled with the approach of the thesis delivery time lead us to consider other options.

As an alternative to real ATCOs, virtual ATCOs were chosen. It is important to clarify at this point that even though the virtual ATC world tries to replicate the real one as much as possible, they are still two completely separate contexts and so there are still differences between them. With that said, virtual ATCOs despite doing this as an hobby, can still be considered excellent elements for the experiences and they have one advantage since they work on a daily basis with Euroscope and hence are best fitted to assess the plug-in's interface integration with the air traffic simulation environment in question.

The two chosen virtual ATCOs are very active members in the Portuguese VATSIM community, where ATCO A, between many functions, is supervisor and instructor to new members in various Portuguese FIRs, being also certified to operate in LPPO and having 5 and a half years of experience in the virtual ATC world whereas ATCO B is the director of LPPO, has 7 years of virtual controlling experience.

Thus, this trial was arranged with two virtual ATCOs during the 11th of October 2014. Both trials were done in the same day and only separated by time of day because it was a good idea to maintain myself (the plug-in developer) available for any possible doubt or error

[3]The AIRAC cycles have duration of 28 days and are used to disseminate information concerning significant changes.

during the simulations and that was achievable only by having the tests happening at different times.

The duration of each trial was modified from four hours to one hour so as to accommodate the fact that the ATCOs, doing the trials without being paid and on their free time, could not spend too much time in the trials. That did not constitute a problem because a one hour trial is a very good sample of what a four hour trial would be like and is enough to study the ATCOs response to the developed plug-in.

Hence, the trials consisted of two tests, each with duration of one hour, with the help of two ATCOs. They took place accordingly to the scheme depicted in table II in order to reduce the ambiguity of the tests. Within this table, one can notice that two of the principles for ambiguity's reduction discussed before for the trials with sets of four ATCOs are still verified:

- Each scenario file is attributed only once to each ATCO.
- The order in which the DST is used is inverse for the first and second ATCOs.

		ATCO	
		A	**B**
Trial	**1st**	*I* *DST_ON*	*I* *DST_OFF*
	2nd	*II* *DST_OFF*	*II* *DST_ON*

Table II: Scheme plan of the experiments with two controllers

However, and even though the ambiguity is almost eliminated by having used table II, there is still a third condition that cannot be taken into account with two ATCOs only: if the difficulty of one scenario is much different than the other, the validity of the results is compromised.

For that reason, the chosen scenario files were similar in both the number of aircraft and in the number of initial conflicts. Both were chosen from real traffic situations from the 17th of January 2014, during different periods of time. The first, *I*, taken from 12:00 to 13:00, had 50 aircraft in the simulation at the moment of the start and other nine planning to enter the simulated FIR at some point during the simulation and the initial number of instructions to solve potential conflicts was of seven. In the other hand, scenario *II*, taken within 14:00 and 15:00, had 56 aircraft at the beginning and the same nine planning to enter the FIR. The number of initial instructions of this second scenario file was nine.

Before the trials started, a 10 minute explanation about the DST objectives was made. Also, the simulation framework previously discussed was presented to the controllers in order for them to act according to the role intended for them. After that, and before the first trial, the ATCOs were asked to use the presented tools for a couple of minutes to become accustomed to them.

D. Results

During the trials it was observed that each ATCO had a very different style of controlling. Whereas ATCO A used all available tools in Euroscope while controlling the FIR, ATCO B had a very conventional style, measuring for example the separation distances by eye. The result of those different practices together with the fact that the number of airplanes to control was significantly large, lead to imperfect controlling in the scenarios where the DST was not used. I.e. the first ATCO failed to notice one of the conflicts and was late on noticing another, what resulted in giving an instruction out of the standard period for instruction issuing – after waypoint reports – while the second, managing the separation mentally only, discovered every conflict – even though violating for a couple of times the horizontal separation minima – but gave relatively more instructions in comparison.

In the session with the DST's assistance, it was noticed that the ACTOs did not have much work on their hands, taking advantage of their free time to check which aircraft conflict generated each DST solution's instruction. That was one of the improvement points an ATCO brought up saying that "the plug-in could have a functionality that shows which was the conflict each instruction was solving, by moving the mouse over each instruction. That way, the controller would be left with an even easier task.". This suggestion, although good and feasible has to be carefully considered because of one aspect earlier discussed: the DST has to provide the exact right level of automation, not too much so as to hinder the controller's ability to quickly reassure manual operation in case of emergency. The suggestion is then left as a medium term work, when (and if) the level of ATC automation becomes much higher.

All in all, objective results for the trials were ascertained by having the ATCOs present in the trial answer an inquiry and by asking them, and recording with their consensus, other development questions directly.

For the first case, the ATCOs' filled inquiries reveal a very positive feedback, where they both strongly agree that the DST reduced their workload by allowing the conficts' prevention to occur more easily. In the end, both wished to see the developed tool used in the virtual ATC system which constitutes a very positive reaction and is aligned with this paper's objectives.

On the other hand, for the qualitatively feedback, the most relevant answers and the respective questions asked to obtain them are presented below:

- **Were you able to trust the DST's solutions? Explain why.**
  ATCO A said "Yes, because the DST was able not only to predict a near-term conflict but also one that was only going to take place many minutes from then, something that a controller would not be capable of observing in that immediate" while controller B replied: "Yes, and the trust in the DST's solutions came from finding that the conflicts were

being solved with the DST's advisory instructions".

- **Did you think that the DST reduced the workload in comparison to the session without the DST? Why?**
  ATCO A response was: "Absolutely! I think that it reduces approximately 70% of controller's workload and hence, airspace divided for two controllers could easily be supervised by one ATCO only if being helped by this DST ".
  ATCO B in its turn, said: "My workload was significantly diminished since I stayed completely rested, when in the session without the DST I needed to constantly find possible conflicts and solutions and then, the solutions that I have found turned out not to be the best ones".
- **Would you change anything about the DST to make it better ATC-wise?**
  Both ATCOs gave the same answer and said that it would be better if the information present on "Waypoint Reports" window stayed there indefinitely instead of just for a certain amount of time. That, coupled with the functionality of left-clicking over a report that is not needed anymore and erasing it from the list by doing so, would in their opinion establish a better way to inform the controllers about the aircraft's whereabouts.
  ATCO B also suggested that by right-clicking each report, instead of having to use the **.find** Euroscope's functionality to find each aircraft, a line showing the aircraft location should prompt.
- **Do you have any other observation?**
  ATCO A stated that "I was under the impression that the tool would be more relevant in oceanic airspace only but I was left with the certain that it would also be of great help in radar airspace areas that are sometimes full of traffic with intersecting routes and flying from different FIRs"

In summary, the overall reaction to the constructed plug-in was excellent. Saying that the workload was diminished by 70% and that you were left doing nothing in a scenario with 56 aircraft was even better than the projected ideal responses.

As for the suggestions from the question about possible DST improvements, they both seem like a good step towards evolving the plug-in and are therefore left as future work.

V. Conclusions

This paper proposes a plug-in for integration of a DST in an air traffic simulation environment – Euroscope – and tests it with virtual ATCOs to assess its validity. The data coming from the ATCOs' trials are not statistical relevant and hence they serve only as a proof-of-concept to check if the small sample of ATCOs found the plug-in and DST to help their workload diminish. Thus, their feedback was determined through an inquiry with likert scale and by oral interaction.

The obtained results illustrate the benefits of controlling with the help of the DST, at least in conventional oceanic control – which is the case where the simulations took place. The controllers' feedback was extremely positive and both said that the workload was diminished, since they found possible conflicts more easily with the aid of the DST via plug-in. They were also in favor of using the tool again in the ATC they practice nowadays.

Possible future work is related to solving the bugs found in the DST and only after that to execute more tests with more ATCOs. After all the necessary steps for validation of the tool were made, the final objective is to integrate that DST with the real ATC system, turning it into a relevant aid in the ATC automation.

References

[1] EUROCONTROL *EUROCONTROL 20-year Forecast of Annual Number of IFR Flights (2012-2035)*, June 2013.

[2] B. G. Hillburn, "Evaluating human interaction with advanced air traffic management automation," *RTO-MP-088 - The Role of Humans in Intelligent and Automated Systems*, October 2003.

[3] C. D. Wickens, *Engineering psychology and human performance* . HarperCollins Publishers, 1992.

[4] F. Freitas, "Optimal air traffic control automation: Application to oceanic control," Master's thesis, Instituto Superior Técnico, 2013.

[5] M. Brennan *et al.*, "Evaluating en route congestion management through interactive simulation,"

[6] R. Gingell, K. Corker, and E. Weber, "The feasability of measuring capacity in a real-time atm simulation independent of subjective contcontrol worlkload measurement," 2005.

[7] T. Prevot *et al.*, "Automated air traffic control operations with weather and time-constraints," in *Proceedings of the 9th FAA/Eurocontrol R&D Seminar*, 2011.

[8] P. L. de Matos and D. Marsh, "Dss on trial," *OR Insight*, vol. 13, no. 1, pp. 2–8, 2000.

[9] A. Kirlik, "Modeling strategic behavior in human-automation interaction: Why an aid can and should go unused," *Human Factors*, vol. 35, pp. 221–242, 1993.

[10] L. Major, H. Johannsson, H. Davison, E. T. Hvannberg, and R. J. Hansman, "Key human-centered transition issues for future oceanic air traffic control systems," 2004.

[11] C. D. Wickens, "Imperfect and unreliable automation and its implications for attention allocation, information access and situation awareness," 2000.

[12] U. Metzger and R. Parasuraman, "Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 47, no. 1, pp. 35–49, 2005.

[13] J. Sanchez, C. J. Krumbholz, and M. O. Silva, "Controller-automation interaction in nextgen: A new paradigm," February 2010.

[14] ICAO, "Doc. 9574 an/934," *Manual on Implementation of a 300 m (1000 ft) Vertical Separation Minimum Between FL 290 and FL410 Inclusive*, Edition 2002.

[15] ICAO, "Doc. 4444 atm/501," *Procedures for Air Navigation Services–Air Traffic Management (PANS-ATM)*, pp. 69–101, Edition 2007.

[16] J. Krozel *et al.*, "System performance characteristics of centralized and decentralized air traffic separation strategies," *4th USA/Europe ATM R&D Seminar*, 2001.