IADIS International Journal on Computer Science and Information Systems
Vol. 3, No. 2, pp. 36-50
ISSN: 1646-3692

# EATS: AN AGENT-BASED AIR TRAFIC SIMULATOR

José Miguel Canino Rodríguez*. Universidad de Las Palmas de Gran Canaria. Las Palmas de G.C., Spain. jcanino@dsc.ulpgc.es*

Jesús García Herrero*. Departamento de Informática Universidad Carlos III. Madrid, Spain jgherrer@inf.uc3m.es*

Juan Besada Portas. *Dpto. de Señales, Sistemas y Radiocomunicación. Universidad Politécnica de Madrid, Spain.besada@ grpss.ssr.upm.es*

Luis Gómez Déniz
*Dpto. Ingeniería Electrónica y Automática, Universidad de Las Palmas de Gran Canaria. Las Palmas de G.C., Spain, lgomez@diea.ulpgc.es*

**ABSTRACT**

We present an Experimental Air Traffic Simulator (EATS).It is conceived as a tool for preliminary evaluation of flight procedures, algorithms and human-machine interfaces to be used in future Navigation and Air Traffic surrounding the new Communication, Navigation and Surveillance System to Air Traffic Management (CNS/ATM). The proposed EATS simulator version provides realistic data for the aircraft dynamic and includes the exchange of information among the aircraft from the point of view of the Air Traffic Control (ATC). It also takes into account the meteorological conditions and terrain constraints. This system has been designed as a Multi-Agent System and implemented on a JADE framework. Its architecture facilitates its later extension to incorporate and to evaluate new communication protocols and negotiation between agents operating in a specific air space.

**KEYWORDS**

CNS/ATM system, Free Flight, Air Traffic Simulation, Multi-agent system.

## 1. INTRODUCTION

The increase of Air Traffic during last decades as well as its growth forecast in the future and the use of new technologies in civil aviation like the CNS/ATM system *(ICAO 2007)* will suppose, in next years, a substantial transformation of the current procedures for air navigation and air traffic control *(Hoekstra et al. 2002)*

CNS/ATM System will allow that all agents implied in air traffic (Flight Crews, Air Traffic Controllers, Airlines, etc.) to share real time information about on board navigation, air traffic and other environmental data. This easiness to share information combined with flexibility of satellite-based air routes suggests new air traffic scenarios where aircraft can choose their own flight trajectories in an efficient and secure way, avoiding the non-optimal current system based on rigid flight routes.

In this context, the Radio Technical Commission for Aeronautics (RTCA) has proposed the User Preferred Trajectory or Free Flight operational concept defined as *(RTCA 1998)*: "A safe and efficient flight operating capability under instrument flight rules (IFR) in which the operators have the freedom to select their path and speed in real time. Air traffic restrictions are only imposed to ensure separation, to preclude exceeding airport capacity, to prevent unauthorized flight through special use airspace, and to ensure safety of flight." An example of "mature" free flight is the Distributed Air-Ground Traffic Management (DAG-TM) concept developed as part of Advanced Air Transportation Technologies (AATT) NASA project *(AATT 1999).* DAG-TM concept proposes several guidelines to develop navigation and air traffic procedures to make feasible free flight operations. In contrast with the current form of ATC organization, which is a full centralized system on the ATC, the DAG-TM is an advanced concept in which Flight Crew, Air Traffic Service Provider, and Airline Operations Center use a distributed decision making scheme *(Ballin et al. 2002).*

To carry out the design and preliminary evaluation of these new procedures and their associated subsystems, it is necessary to make use of virtual platforms that simulate the Air Traffic in a free flight environment. Air traffic simulation is becoming an important tool to provide synthetic data to validate operational future concepts *(Valenti et al 2002; Prevôt et al. 2003).* Besides, the air traffic scenario described above needs from complex decision making models which can be naturally modelled as a Multi Agent System (MAS). Therefore, agent oriented programming and technologies provide a suitable research tool for analyzing, designing and simulation of new navigation and air traffic procedures as well as their required CNS-ATM based subsystems.

Several application of MAS paradigm has been focused on air navigation and air traffic researches. First works proposed overall modelling of free flight scenarios. For example, *(Wangermann & Stengel 1998)* suggested a cooperative structure for interactions of agents (aircraft, airlines, and ground control centers). The cooperative model was defined as an Intelligent Aircraft/Airspace Systems (IAAS) where the agents cooperate on the basis of Principled Negotiation. Principled Negotiation applied to air traffic operations allows much greater freedom for optimization by system users while maintaining safety *(Wangermann & Stengel 1999).* An agent-based modeling and a basic analysis tool for evaluation ATC performance in a free flight environment is proposed by *(Harper et al. 1999).*

Other works have centered their effort to model particular aspects and systems. For instance *(Shandy & Valasek 2001; Rong et al. 2002; Wollkind 2004)* propose agent architectures and negotiation protocols in order to solve conflicts between autonomous aircraft; *(Painter 2002)* presents an agent characterization of an on board system for autonomous operations and *(Ding et al. 2003)* proposes a preliminary simulation methodology based on agents to design an automated arrival/departure system for no-controlled airports.

Some approaches to implementation aspects have been raised by *(Sweet et al. 2002),* who describes a prototype of Airspace Concept Evaluation System (ACES). A later extension of this work presets an approach to incorporate CNS functionalities into the mentioned gent-based modelling and simulation system. *(Satapathy & Manikonda 2004)*. *(Pechoucek et al. 2006)* use the A-Globe Multi-Agent Platform to model deconflicted free-flight of collective of fully autonomous aircraft and wherein the ATC has not been primarily designed a flight planner. Also, *(Callantine et al. 2006)* described an ATC agent model and architecture and illustrated how it can be applied to analysis of future air traffic operational concepts.

In this paper we present and describe an Experimental Air Traffic Simulator (EATS) developed to model situations with multiple aircraft executing their own flight plans and following instructions from a traffic controller. The architecture of this simulator extends the conventional Air traffic simulators used in Air traffic controller training, by using two types of consoles. In one of them an Air Traffic Controller (ATC) monitors on the radar screen the current position of different aircraft. In the other one a user pilot (also named pseudopilot user) receives orders from the ATC (via voice) and carries out the necessary actions to fly the aircraft according to these orders. In EATS, the communication between the air controller and the pilot can be done through voice or data. The proposed architecture we present in this paper makes easier later extensions in order to add new algorithms (i.e. conflict detection and resolution algorithms, arrival sequence algorithms, etc), air-air and air-ground negotiations protocols, human-machine interfaces and decision-making support systems, etc.

EATS has been designed within a Multi-Agent architecture, implemented on a JADE platform *(Bellifemine et al. 2007).* The advantages of this advanced design can be summarized as:

a) *Transparency in communications*: communication protocols are automatically managed by the JADE library resources. Also, JADE uses an Agent Communication Language specified by FIPA (FIPA-ACL) *(FIPA 2008).* FIPA-ACL standard derives from the linguistic theory of speech acts and provides a set of 22 communicative acts which are very appropriate for coordination situations such as those appearing in Air-Ground communication procedures *(ICAO 2005)*.
b) *Decentralized System*: each agent is executed in an independent way and the interaction with other agents takes place through messages. Furthermore, a multi-agent system facilitates the distributed simulation of different agents that contains complex hybrid control system (aircraft) and utilizes multiple computing resources.
c) *Modularity:* It is easy to add new elements in next versions, or to modify them without affecting the rest ones. For instance, it provides a platform to add algorithms and decentralized procedures to Air Traffic Control or to add new sub-agents to help for or to perform specifics pilot and air controller tasks.
d) *Scalability:* The behaviour of multiple aircraft operating with new procedures can be simulated in a natural form simply adding more agents in the system with the predefined behaviours and coordination mechanism. Besides, JADE allows the use of multiple platforms for those situations demanding a high CPU load (tens or hundreds of aircraft).
e) *Behaviour-based agent architecture*: although JADE doesn't define specific agent architecture, it provides a set of functionalities to achieve efficient architectures of autonomous agent. Moreover, the extended JADE version (JADEX) will allow building BDI *Agent (Pokahr et al 2003)*. BDI Agent could be used to model specific human behavior and the reasoning process applied in the navigation and air traffic control tasks *(Urlings et al. 2006).*

The paper is organized as follows. In Section 2, we describe the general architecture of the simulator explaining the main characteristics of each agent. In Section 3, we analyze the communication between the aircraft agent and the ATC agent. In Section 4, to illustrate the most relevant functionalities of this simulator a complete realistic simulation example is under consideration. Finally, in Section 5, conclusions and scope for future work are presented.

## 2. GENERAL ARCHITECTURE OF THE SIMULATOR

The overall simulator architecture is shown in Fig. 1. The system is composed of six agent classes in JADE which, when instantiated, are registered within Agent Management System (AMS) and subscribed to Directory Facilitator (DF), before starting their specific tasks and services. From this figure we can see that the application contains the following agents that represent real world physical entities.

- *The Aircraft Agent*. This agent simulates the behaviour of real aircraft following a flight plan which consists of a set of three dimensional points (way points or WP). Each aircraft can modify its own flight plan when receiving an ATC instruction message (or orders), to change the altitude, the heading and/or the speed. These orders are shown on the pilot display, named in this work, Pseudopilot Radar View. A pilot user can execute them in a manual mode using the aircraft control window or in a automatic mode which allows introducing data instruction directly in the aircraft flight management system. Besides, to illustrate adding future aircraft navigation and air-ground communication functionalities, in this work we have implemented an algorithm to compute the *Top Of Descent* (TOD) to carry out continuous descent arrivals *(Clarke et al. 2006)* from cruise level to a final approach fix to landing. Then, when aircraft detects an arrival route, it computes the TOD and modifies the vertical profile of this route. This modification is also notified to ATC.
- *The ATC Agent*. It receives messages from the aircraft agents about their position and flight plans. This information is shown in the ATC Radar View display. The ATC agent is provided with a control window interface to send instruction messages to a particular aircraft. Also, two algorithms have been included in the description of the ATC agent to extend its possibilities. One of them, proposed by *(Isaacson, & Erzberger 1997)*, detects possible conflicts between two aircraft. The other one is a simple algorithm to solve the conflict. When a conflict is detected, the ATC agent sends a first message to the involved aircraft notifying it. Latter, the ATC sends a second message with specific instructions to avoid the collision.
- *The Meteorology Agent*. This agent provides information about the atmospheric and wind conditions in the flight region under consideration. In our work, this information contains updated meteorological data required by the aircraft to compute predicted trajectories for continuous descent approach and its associated aerodynamic parameters (ground speed, true air speed, aerodynamic coefficients, etc.). Capabilities of this agent can be extended in the future to include autonomous management of current and forecast meteorological data obtained from several sources.
- *The Terrain Agent*. It integrates information about the set of possible flight routes as well as the operational altitudes *(EUROCONTROL 2004)* in the reference airport. The Terrain agent's data will also be used by the ATC agent to represent these routes in the ATC radar view display. Latter extensions of this agent could consider autonomous capabilities to

manage available air routes taking into account ground restrictions such as minimum altitudes, restricted airspace, etc.

Note that ATC, Meteorology and Terrain Agents represent functionalities that must be supplied by the Air Traffic Service Provider.

To complete the software description of the EATS simulator, two other agents need to be issued: the *Configuration Agent* and the *Pseudopilot Agent.* The Configuration Agent is required to define the set of initial simulation parameters, which is done by means of an ASCII input file. The Pseudopilot Agent has been designed with a twofold purpose. First, it is a desk control that allows to an unique pilot-user to have control over several aircraft. Second, it represents a graphical display, providing significant information about the state and intentions of surrounding traffic for each selected aircraft. This interface has been implemented as a separated agent (and not like an aircraft agent component), to centralize in an unique interface the access to each aircraft. Besides, it plays an important role (especially in the near future scenarios) to design and evaluate specific on board man-machine interfaces like Cockpit Display Traffic Information (CDTI) *(Bone 2005).* The CDTI cockpit display allows seeing the surrounding traffic and, what is more relevant, the intentions of the surrounding aircraft. To access to a particular aircraft, a mouse click over the icon symbol is required. Once the aircraft is selected, it is placed at the central position of the pseudopilot radar window, and the movement and the position of other aircraft are represented in relation to it. At the same time, the control window of the selected aircraft will be opened.

To carry out the communications between agents, a Communication class with specific methods has been designed. In particular, the air-ground communication between the aircraft agents and the ATC agent is carried out with the following messages:

a) *Messages sent by the aircraft to the ATC*: message to inform about the state vector; message to inform about the possible modification of the altitudes of the flight plan to initiate a continuous descent to the airport.
b) *Messages received in the aircraft from the ATC*: instruction messages (changing altitude, heading, speed, a flight plan waypoint, etc.) and messages of conflict detection with other aircraft.

   Starting from this nucleus of air-ground communications, new types of messages can be implemented in future EATS extensions with the purpose of establishing more complex negotiations between aircraft and ATC.

Apart from the previous communications, there are other communications involving the agent meteorology (to obtain atmospheric information), the terrain agent (to request the available routes). Moreover, the communication between the aircraft agent and the pseudo pilot agent represents the communication between a physicals agent (the aircraft) and a man-machine interface like the CDTI. Finally a communication air-air provides information to each aircraft about its surrounding air traffic.

As it is known, all agents in JADE implement predefined tasks (behaviours) associated to events, which are executed keeping its state of activation *(Bellifemine et al. 2007).* Complex agent behaviours can be defined stating from simple behaviours. In EATS application the following JADE simple behaviours were implemented: (i) *on-shot behaviour*, an atomic task to be carried out once, used here for initialization tasks; (ii) *cyclic behaviour*, which is iterated while exists, such as messages listening and processing; (iii) *waker behaviour*, or a one-shot behaviour invoked after a certain time; and (iv) *ticker behaviour*, or a cyclic behaviour which performs a series of instructions executed keeping a certain fixed time, used in the platform for simulation numeric computation and graphical output. Also, a composite behaviour that executes a set of sub-behaviours concurrently *-parallel behaviour-* has been implemented. In this way, agents are able to concurrently to carry out different tasks and to keep simultaneous conversations. Messages follow FIPA-ACL format, and they are shared by all behaviours.

The architecture described in this section allows future extensions of the capacities of EATS incorporating new agents (for instance Airline Operational Agents) as well as more complex behaviours to each agent. These behaviours will also be able to include specific algorithms for the aircraft navigation (for example trajectories 4D tracking or for new airborne separation assistance systems) and for Air Traffic Control (e.g. to solve the aircraft sequencing problem).

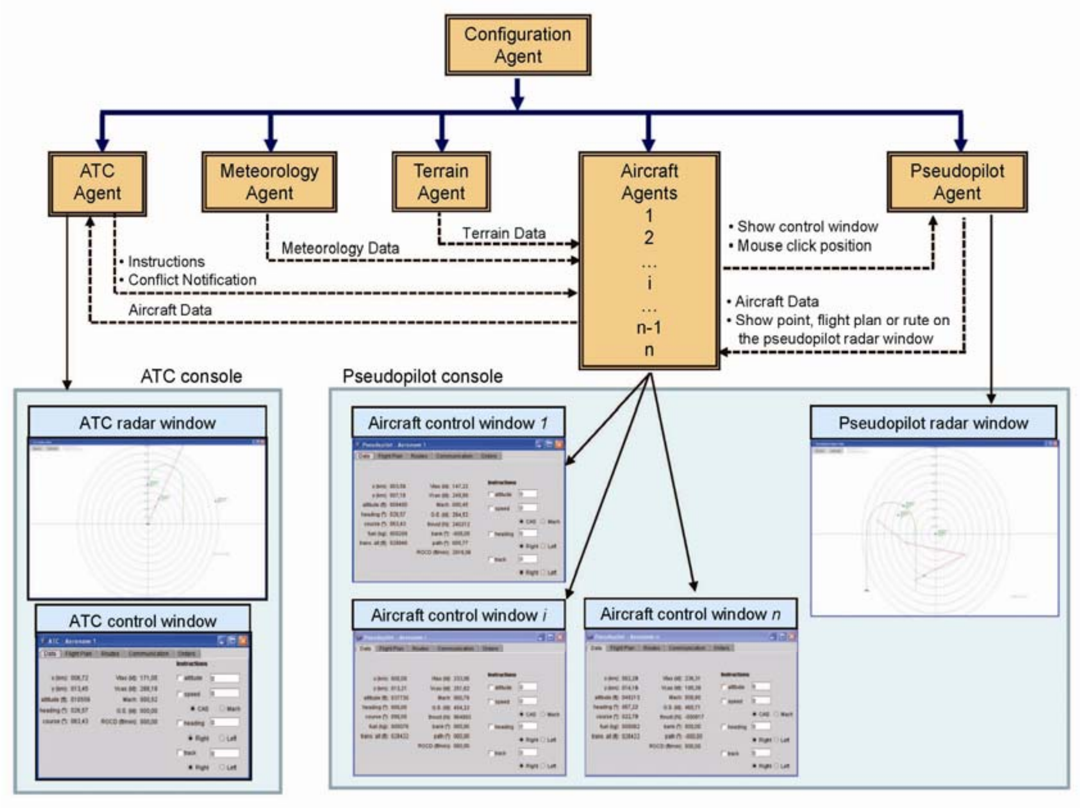


Figure 1. Overall architecture of the simulator

# 3. DESCRIPTION OF THE MAIN AGENTS: AIRCRAFT AND ATC AGENTS

## 3.1 The Aircraft Agent

Navigation and communication capabilities have been implemented for the aircraft agent. Navigation capabilities consist of aircraft aerodynamic behaviours under pilot control and estimations about trajectory predictions. Communication functionality mainly manages message exchanges between aircraft and ATC.

In our application, the flight is carried out according flight plan routes or ATC orders. Among other tasks, the main functions of the aircraft agent consist of calculating its state vector and sending this information to the ATC and to the Pseudopilot agents via ACL messages. Also, as it was explained before, it must compute the TOD for arrival routes, modifying the descent profile and communicating this event to the ATC.

### 3.1.1 Model to the Aircraft Aerodynamic Behaviour

The aerodynamic behaviour of aircraft can be modelled by means of a hybrid dynamic system (Fig. 2), constituted by two main blocks *(Glover & Lygeros 2004).* The first one models the aircraft dynamic. The second one is the Flight Management System (FMS) or a control system that computes the input control to the aircraft dynamic model as function of required flight conditions.

The aircraft dynamic is properly formulated by means a Mass Point Model (MPM) that consists of the following motion equations:

$$\begin{aligned}
\dot{x} &= V\cdot\cos(\varphi)\cdot\cos(\gamma)+w_x \\
\dot{y} &= V\cdot\sin(\varphi)\cdot\cos(\gamma)+w_y \\
\dot{z} &= V\cdot\sin(\gamma)+w_z \\
\dot{V} &= \frac{1}{m}\left(T-\frac{C_D S\rho}{2}V^2\right)\left(1-f\{M\}\right) \\
\dot{\varphi} &= \frac{C_L S\rho}{2}\frac{V}{m}\sin(\phi) \\
\dot{m} &= -\eta T
\end{aligned} \tag{1}$$

In this system, x, y, z represents the aircraft spatial coordinates, V denotes the aircraft speed respect to the surrounding air mass (also named True Air Speed or TAS), $\varphi$ is the aircraft heading, $\gamma$ is the path angle, and $\phi$ is the bank angle. Then, the above output system provides the aircraft vector state $\chi$=(x, y, z, V,$\varphi$, m).

Also *T (throttle), $\phi$ (bank angle) and,* $\gamma$ (path angle) are the components of the input control vector to the aircraft dynamic model.

The atmospheric conditions are given by the wind vector *($\omega_x$, $\omega_y$, $\omega_z$)* and the atmospheric density, $\rho$. Aircraft parameters are the surface area (*S*), mass (*m*), dry coefficient ($C_D$), lift coefficient ($C_L$), thrust specific fuel consumption ($\eta$), throttle power ($T$) and $f\{M\}=\left(1+\frac{v}{g}\frac{dV}{dz}\right)^{-1}$ is the energy share factor.

Values of above aircraft parameters are obtained from the Aircraft Database of EUROCONTROL (BADA) *(EUROCONTROL 2004).* BADA is constituted by a group of ASCII files that provides information about the aerodynamic parameters and standard operative speeds of 295 civil aircraft.

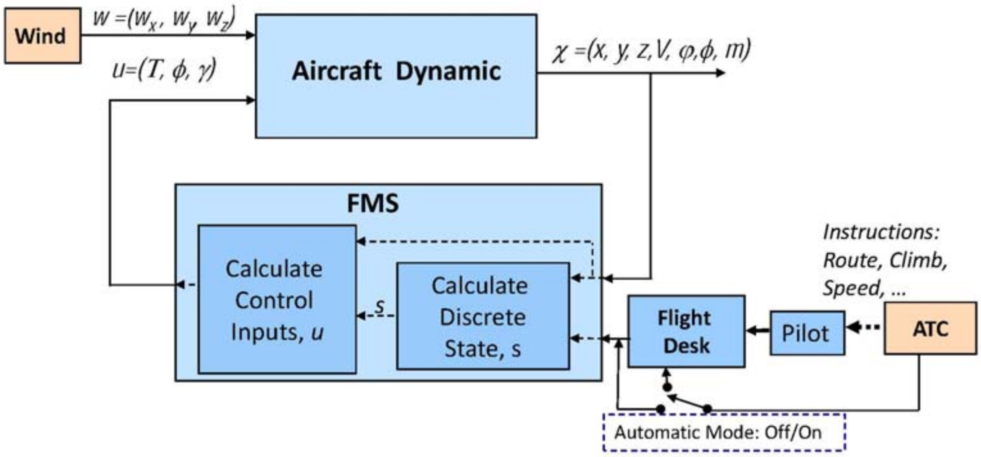


Figure 2. Aircraft model components

The FMS module computes the control inputs T, $\phi$ and $\gamma$ that the aircraft needs to reach the state vector value according to the conditions imposed by the flight plan or by the ATC instructions. This computation is carried out in two phases. First, the FMS compare current vector state and mentioned requirements to identify several discrete states, such as Climb, Descent, Accelerate, Turn, etc. Then, discrete state vector and state vector information are used to compute the control inputs.

Differential equation system (1) is also used to compute the TOD by means of a backward Runge-Kutta integration scheme from a final approach fix to landing to aircraft cruise level.

### 3.1.2 The Aircraft Behaviour

To organize and coordinate all the previous processes, three JADE simple behaviours in the aircraft agent have been defined. Also some of these behaviours have been grouped into a Parallel Behaviour.

The simple behaviours implemented are the following: First an one-shot behaviour is used to obtain and compute data about the initial conditions of the simulation and to store the results in different recipient objects. A second behaviour is a ticker behaviour which constitutes the core of the simulator (see Fig. 3).This behaviour computes the aerodynamic state vector and sends messages with this data to the ATC and to the pseudopilot agent. The first three steps of the diagram represent the calculations carried out by a hybrid dynamic system like the one showed in Fig. 2. Later, these data are sent to the mentioned agents. Next, when aircraft agent detects that current route is an arrival route, its computes the TOD to carry out a continuous descent arrival and approach and therefore a new flight is sent to the ATC agent. Then, the time period to execute the ticker behaviour is provided by the configuration agent. This ticker behaviour is executed if the aircraft does not reach the last point of its flight plan. When the aircraft arrives to the last point, the following tasks are performed: (i) the corresponding data of the whole flight trajectory is stored in an output file for a later analysis, (ii) the aircraft agent sends an arrival message to ATC agent and configuration agent and (iii) the configuration agent will destroy the aircraft agent.

The last simple behaviour is a cyclic behaviour which has been implemented to receive messages from other agents as it is shown in Fig. 4.

Finally the above TickerBehaviour and CyclicBehaviour are included as sub-behaviours of a ParallelBehaviour in order to their respective tasks can be executed in a concurrent way.

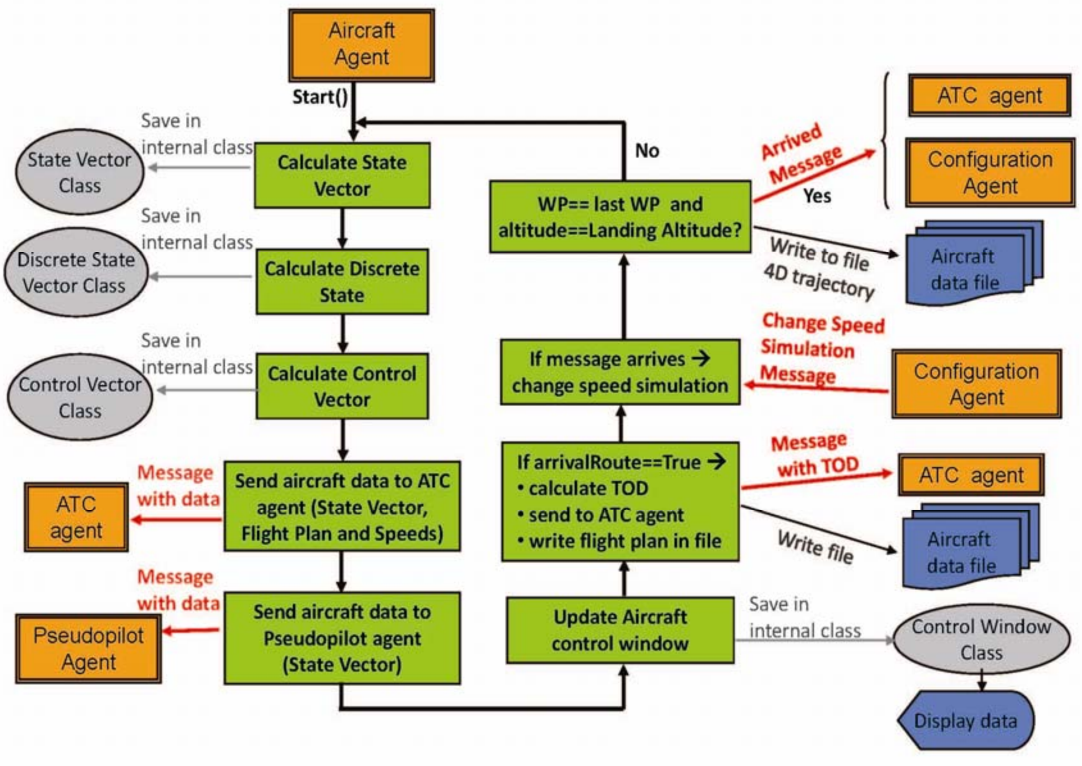


Figure 3. Aircraft ticker behaviour

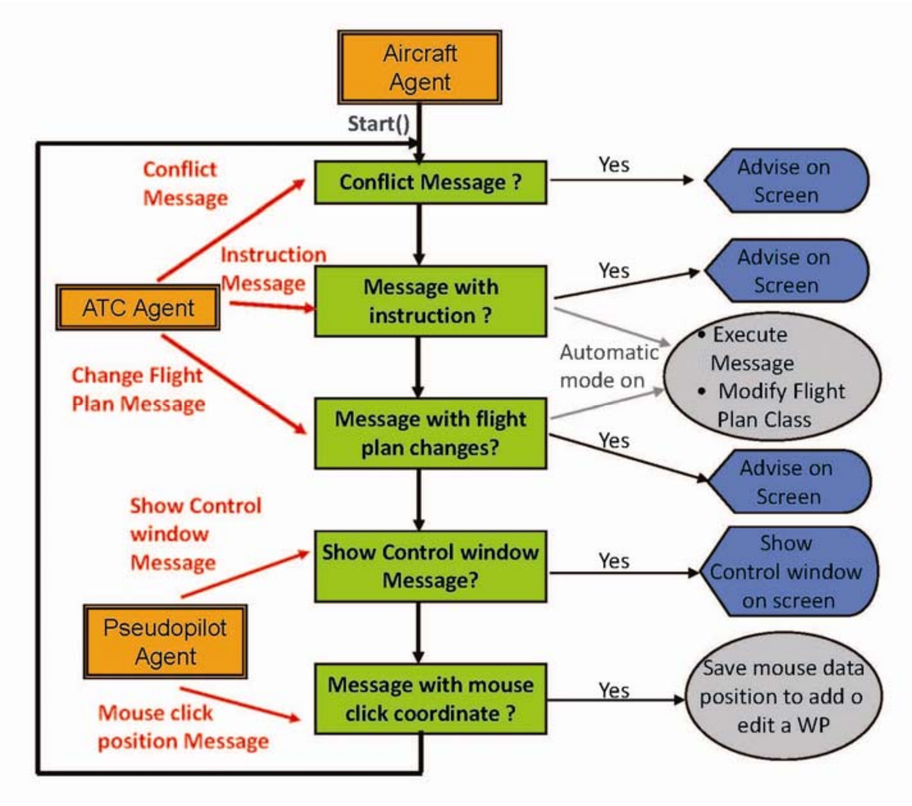


Figure 4. Aircraft cyclic behaviour

## 3.2 The Agent ATC

The ATC functionalities described in previous session has been grouped in three cyclic behaviours which are executed concurrently by a parallel behaviour. Two of them are ticker behaviours. A first ticker updates the ATC window control and the ATC radar window data. A second cyclic behaviour verifies and processes the three possible messages from the aircraft: messages asking for an identifier, messages requiring aircraft data, and messages to change the flight plan when initiating a continuous descent approach. Finally, ticker behaviour activates the collision conflict detection algorithm. When such a conflict is detected, the aircraft agent executes the resolution algorithm and sends instruction messages to the implicated aircraft to avoid the conflict.

# 4. EXAMPLE

To illustrate some of the capabilities of the EATS simulator, we consider the problem of sequencing two aircraft (Cessna type Golden Eagle and an n Airbus 320). This is typically a task of Air-Traffic Controllers in Terminal Airspace. The Cessna aircraft appears into terminal airspace in the instant of time 0 in the route number 0 with an initial altitude of 10,000 feet (see Fig. 5). Then, the Airbus 320 enters into the terminal airspace 10 seconds later flying at the same altitude and tracking the same flight plan that the Cessna aircraft.

Due to the A320 aircraft flies faster than the Cessna, the distance between them will be smaller in time. When the distance is less than 5 nautical miles, the ATC agent detects a collision conflict and notifies it to both aircraft. Next, the ATC agent will provide specific instructions to each aircraft to solve the conflict. In this case the Cessna altitude is multiplied by a 0.8 factor (descending 6124 feet). At the same time, the ATC orders to the A320 aircraft to add a new waypoint located between the current position and the next waypoint. Besides, the ATC orders to this aircraft to increment its current altitude by a factor of 1.2 or 8862 feet. This procedure of modifying the altitude of the aircraft using a numerical factor is not practical in civil aviation, where the standard procedure consists of increasing or reducing the aircraft altitude based on flight levels with a minimum separation of thousand feet between them. However, in this case the assignment of the altitude level is not relevant because the fundamental purpose is to verify the communication between agents in order to solve the conflict.

In Fig. 5, we can see the trajectory in the horizontal plan for both aircraft, as well as the original flight plans (that are the same in this case). Fig. 6 shows the vertical trajectory and the altitude variations of both aircraft as a function of time.

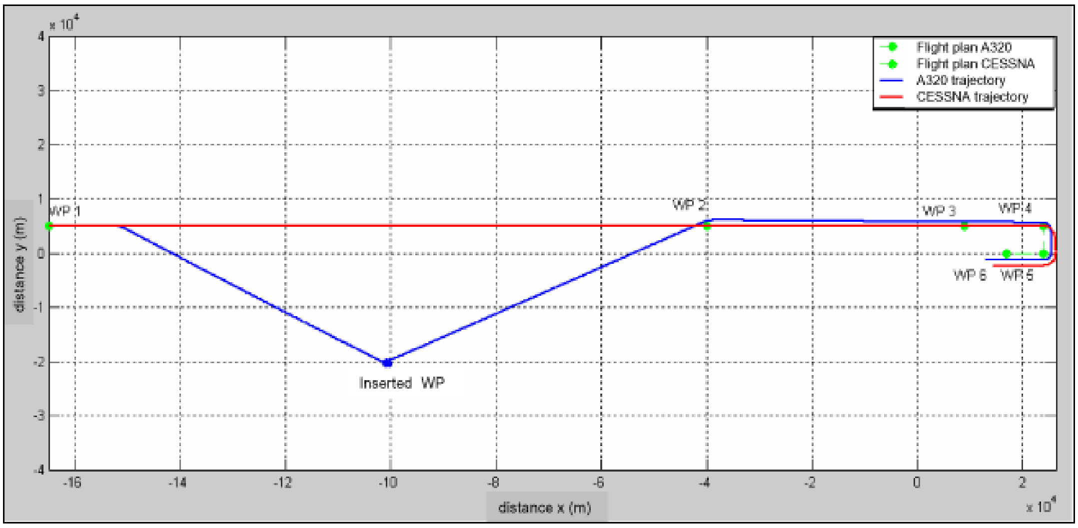


Figure 5. Horizontal flight trajectory

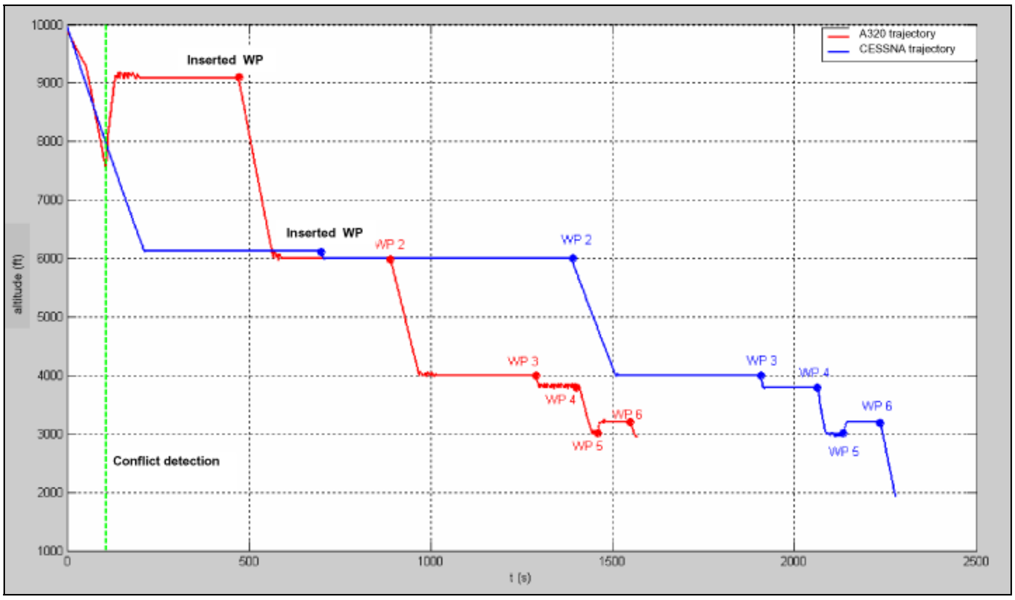


Figure 6. Vertical flight trajectory

The geometry of the new route is shown in the Fig. 7. Instructions messages are indicated in Orders tab of the ATC control window and the content is also indicated in table 1.

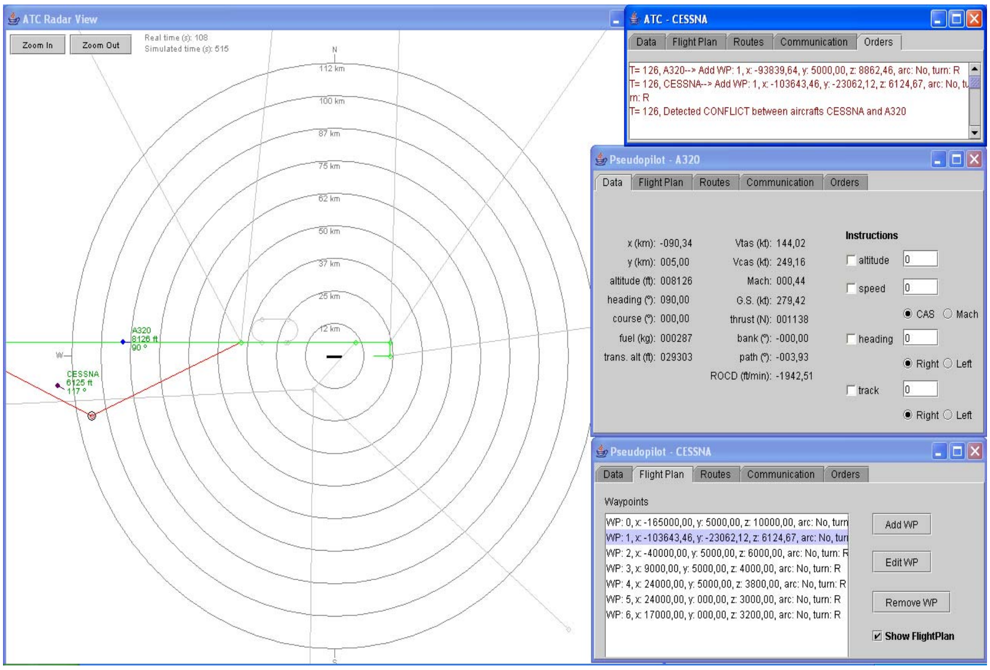


Figure 7. Application screenshot for the described scenario

Table 1. File content with the orders of ATC

| ATC Orders |
| --- |
| T= 126, Detected CONFLICT between aircraft CESSNA and A320 |
| T= 126, CESSNA--> Add WP: 1, x: -103643,46, y: -23062,12, z: 6124,67, arc: No, turn: R |
| T= 126, A320--> Add WP: 1, x: -93839,64, y: 5000,00, z: 8862,46, arc: No, turn: R |

ATC messages sent to each one of both aircraft consist of (i): A message to inform them about collision conflict detection at second 126. (ii) An instruction message ordering them adding a new way point which spatial coordinates are also specified.

Once these messages have been received, they are executed either manually by a user pseudopilot or either in automatic mode. Details of an inserted way point are also shown in Flight Plan tag of the Cessna Pseudopilot control window.

## 5. CONCLUSIONS

In this work, we have presented EATS, an Experimental Air Traffic Simulator, implemented as a Multi-Agent System. The Multi-Agent design allows the implementation of behaviors and complex decision processes resulting from coordinated tasks. The main concepts related to Air Traffic Management have been incorporated to the simulator as physical agents (realistic aircraft dynamic models, ATC, Meteorology and Terrain agent) and Configuration agents. Special attention has been paid to aircraft-ATC communications in order to deal with standard and near future flight scenarios. The fundamental component of the EATS application is the aircraft agent, which integrates the dynamic model to estimate the flight trajectory. EATS simulator provides realistic data for the validation of algorithms, man-machine interfaces and procedures for the navigation and air traffic control in a free-flight environment. Besides, main navigation and air control tasks –such as aircraft-ATC- have been included in the application. Future work will address the implementation of new aircraft-to-aircraft communication protocols for the negotiation among agents as well as a conflict detection probabilistic model to deal with uncertainty associated to aircraft position.

## REFERENCES

AATT, 1999. *Concept Definition for Distributed Air/Ground Traffic Management (DAG-TM), Version 1.0*. NASA AATProject Office, Ames Research Center. Moffett Field, California

Ballin, M.G. et al., 2002. NASA Langley and NLR Research of Distributed Air/Ground Traffic Management. *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical*. Los Angeles, California.

Bellifemine, F.L. et al., 2007. *Developing Multi-agent Systems with JADE*, Springer.

Bone, R.S., 2005. Cockpit Display of Traffic Information (CDTI) Assisted Visual Separation (CAVS): Pilot Acceptability of a Spacing Task During a Visual Approach. *6 USA/Europe Air Traffic Management R&D Seminar*. Baltimore, MD.

Callantine, T.J. et al., 2006. Concept Investigation via Air-Ground Simulation with Embedded Agents. *AIAA*. Keystone, Colorado.

Clarke, J.P. et al., 2006. *Development, Design, and Flight Test Evaluation of a Continuous Descent Approach Procedure for Nighttime Operation at Louisville International Airport*. Partnership for Air Transportation Noise and Emission Reduction, Report No. PARTNER-COE-2006-002, Massachusetts Institute of Technology. Cambridge, MA.

Ding, Y. et al., 2003. Automation Capabilities Analysis Methodology for Non-Controlled Airports. *AIAA Modeling and Simulation Technologies Conference and Exhibit*. Austin, Texas

EUROCONTROL, 2004. *User Manual For The Base Of Aircraft Data (Bada) - REVISION 3.6*. EUROCONTROL Experimental Centre. Brussels. Available at: http://www.eurocontrol.int/eec/public/standard_page/proj_BADA.html

FIPA, 2008. FIPA Agent Communication Language Specifications. Available at: http://www.fipa.org/repository/aclspecs.html

Glover, W. & Lygeros, J., 2004. A Stochastic Hybrid Model for Air Traffic Control Simulation. *Hybrid Systems: Computation and Control*. *Lecture Notes in Computer Science. Pag(s): 372-386*. Springer Berlin/Heidelberg.

Harper, K.A. et al., 1999. Air traffic controller agent model for free flight. *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pags. pages 288–301. Portland, OR.

Hoekstra, J.M., e al., 2002 Designing for safety: the free flight air traffic management concept. *Reliability Engineering and System Safety*, Vol. 75, Issue 2, *Pag(s):* 215-232, Elsevier Science.

ICAO, 2005. *Convention on International Civil Aviation, Annex 10: Aeronautical Telecommunications, Vol II- Communications Procedures*. International Aviation Civil Aviation. Montreal.

ICAO, 2007. *Global Air Navigation Plan for CNS/ATM Systems*, International Civil Aviation Organization. Montreal.

Isaacson, D,.R. & Erzberger, H., 1997 .Design of a conflict detection algorithm for the Center/TRACON automation system. *Digital Avionics Systems Conference, 1997.* 16th DASC., AIAA/IEEE Volume 2, 26-30 Page(s):9.3 - 1-9.3-9 vol.2

Painter, J.H., 2002. Cockpit multi-agent for distributed air traffic management. En *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Monterey, CA.

Pechoucek, M. et al., 2006. Autonomous agents for air-traffic deconfliction. *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, Pag(s):* 1498-1505. Hakodate, Hokkaido, Japan.

Prevôt, T. et al., 2003. Distributed air/ground traffic management simulation: results, progress and plans. *AIAA Modeling and Simulation Technologies Conference*. Austin, TX.

Pokahr, A., Braubach, L., and Lamersdorf, W., 2003. Jadex: implementing a BDI-infrastructure for JADE agents. *EXP – In Search of Innovation (Special Issue on JADE)*, vol. 3, no. 3, pp. 76–85, http://vsis-www.informatik.unihamburg.de/projects/jadex/

Rong, J. et al., 2002. Air traffic conflict negotiation and resolution using an onboard multi-agent system. *Proceedings 21st Digital Avionics Systems Conference. Vol. 2, Pag(s): 7B2-1-7B2-12.*

RTCA, 1998. Free flight: new concepts, a new architecture, new opportunities. *Proceedings of RTCA, Inc. Annual Symposiums*. Radio Technical Commission for Aeronautics. Washington, D.C.

Satapathy, G. & Manikonda, V., 2004. Agent Infrastructures for Modeling and Simulation of CNS in the NAS. *4th Integrated Communications, Navigation, and Surveillance (ICNS) Conference*. Fairfax, VA.

Shandy, S. & Valasek, J., 2001. Intelligent Agent for Aircraft Collision Avoidance. *AIAA Guidance, Navigation, and Control Conference & Exhibit*. Montreal, Canada.

Sweet, D.N. et al., 2002. Fast-Time Simulation System for Analysis of Advanced Air Transportation Concepts. *AIAA Modeling and Simulation Technologies Conference*. Monterey, CA.

Urlings, P. et al., 2006. A future framework for interfacing BDI agents in a real-time teaming environment. *Journal of Network and Computer Applications*, Vol. 29(2-3), Pag(s):105-123.

*Valenti Clari et al., 2002*. Research flight simulation of future autonomous aircraft operations. *Proceedings of the 2002 Winter Simulation Conference*. Vol. 2. Pag(s): 1226-1234.

Wangermann, J.P. & Stengel, R.F., 1999. Optimization and Coordination of Multiagent Systems Using Principled Negotiation. *Journal of Guidance, Control and Dynamics*, Vol. 22, (1), Pag(s): 43-50.

Wangermann, J.P. & Stengel, R.F., 1998. Principled negotiation between intelligent agents: a model for air traffic management. *Artificial Intelligence in Engineering*, Vol. 12(3), Pag(s): 177-187.