

AN AGENT ORIENTED ANALYSIS AND MODELING OF AIRBORNE CAPABILITIES FOR TRAJECTORY BASED OPERATIONS

José Miguel Canino, Luis Gómez, Signal and Communications Department and Electronic Engineering Department, University of Las Palmas de Gran Canaria, Las Palmas de G. C., Spain
Jesús García, Computer Science Department, Carlos III-University of Madrid, Madrid, Spain
Juan Besada, José Ramón Casar, Signal, System and Radiocommunication Department, Polytechnic University of Madrid, Madrid, Spain

Abstract

Current and future air traffic is requiring new procedures and systems to achieve a greater automation of air-traffic operations. Particular difficulty presents the automation of arrival air-traffic operations in Terminal Areas due to aircraft speeds and environment variability into a delimited airspace where multiple aircraft converge. Several projects have proposed guidelines to implement new operational concepts as well as airborne and ground systems to carry out corresponding procedures. Developing procedures and systems are closely related. Therefore, usually it requires to analyze and to design them in a combined manner.

In this paper we present an agent-oriented analysis and modeling of airborne systems capabilities to perform automated arrival and approach procedures based on user preference trajectories. A detailed architecture model of airborne capabilities is achieved through a methodological analysis of an arrival traffic scenario within the trajectory based operations paradigm.

Introduction

New CNS-ATM technologies will improve the current air traffic operations. Moreover, current operation concepts and procedures will be replaced or enhanced including a more automated operation from gate-to-gate. Particular difficulty presents the automation of arrival air-traffic operations in Terminal Areas due to aircraft speed and environment variability into a delimited airspace where multiple aircraft converge. So that outcomes on arrival and approach operations can be extended to gate-to-gate operations.

Several proposals have been developed in order to provide automatic tools for air traffic control. Some of them have been focused on enhanced

current air traffic procedures and systems [1-3]. In addition numerous projects have proposed new operational concepts and guidelines to implement self-spacing flight operation for merging an in-trail separation in Terminal Areas (e.g. [4-5]) and for air-ground negotiation of free of conflict user-preferred four-dimensional (4D) trajectories (e.g. [6])

As result of above works Trajectory-based Operations (TBO) concept [7] has been proposed moving the current clearance-based and centralized air traffic control concept to a trajectory-based and decentralized one. However implementation of TBO concepts will require several efforts for:

- Defining detailed roles of air-crew and air traffic controllers in order to provide efficient air traffic flow taking into account user preference trajectory (UPT). It requires developing new navigation and air traffic control procedures. Procedures should contain ad hoc protocols for: (i) air-ground and air-air trajectories negotiation, (ii) monitoring aircraft states and intentions, (iii) solving unexpected events during the procedure execution, etc.
- Developing new human-machine interfaces to execute above procedures: e.g. Airborne Assurance System and Cockpit Display Traffic Information (ASAS-CDTI) [8-9], FMS with four-dimensional trajectory guidance capabilities (4D-FMS) [10], ground systems for sequencing, de-conflicting, and monitoring arrival traffic, etc. In addition a new height level natural language is necessary for obtaining a precise intercommunication between aircraft systems and ground systems [11]. Moreover, this language should enable

human-readable compression of communication processes.

- Deploying new mathematical models and algorithms to support the above mentioned functionalities: i.e. trajectory synthesis models, conflict detection and resolution algorithms, trajectory guidance models, etc.

The high interdependence of above requirements often makes impossible conducting a preliminary design of procedures, support systems and underlying models and communication languages in an independent way.

Hence it is necessary to create conceptual models of air traffic scenarios that provide high levels of detail of these interdependencies. Architecture of these conceptual models should be robust enough for: (i) obtaining a basic executable model to analytical simulation (discrete events and/or dynamic simulation), (ii) adding new specific functionalities as they are designed.

Modeling complex and distributed air traffic scenarios is a software problem for which the agent-oriented programming provides a natural response.

Current agent-based approaches to modeling and simulating have focused on models that represent several functionalities of physical entities in air traffic scenarios: aircraft, air-traffic services providers, airlines, etc. [12-14]. Also CNS aspects (delays and information uncertainty) have been modeled either as agents in the simulation environment or directly at the infrastructure level by developing delay, media-error, congestion, etc [16]. Other approaches investigated the application of multi-agent coordination techniques using some generic practical coordination models [16-17]. In addition [18] proposed a decentralized Air Traffic Management approach, centered on the design of an automated arrival/departure system for non-controlled airports.

Therefore, procedure and corresponding air and ground capabilities for automated TBO are no often modeled in a combined manner.

Our group is developing a conceptual agent-based model to analyze and represent air traffic scenarios under a TBO perspective. Scenario under studio was focused on arrival air traffic. This scenario represents a significant variability of the operational

conditions and therefore it is easily extending to gate-to-gate scenarios.

The model describes air traffic operations by means of interactions between agents and internal agent's behaviors (multi-agent system model). Interactions are described in terms of coordination of autonomous agents that cooperate for: (i) increasing their environment knowledge to take decisions, (ii) negotiating features of trajectories.

Modeling referred scenarios as a multi-agent system was carried out through a structured and consistent methodological process. Current agent's technology provides practical and formal methodologies to analyze and to design in a structured and consistent manner the following aspects: (i) functionalities of autonomous entities (agents) that take part in an operational scenario, (ii) interactions between agents (iii) inner architecture and behavior of agents.

Prometheus agent-oriented well-established methodology has been selected to provide guidelines to develop the mentioned multi-agent system [19-20]. Several multi-agent methodologies approaches have been proposed in recent years and comparison between them are beginning to appear [21]. However, we argue that Prometheus suits well to solve our problem due to: (i) the high level of detail of guidelines for the initial system specification, (ii) the modularity of the agent's internal architecture around the concept of capability (providing a direct correspondence between capabilities and specific functionalities of airborne and ground systems).

In this paper we present part of above work consisting of an agent-oriented analysis and modeling of airborne systems capabilities for TBO. Detailed analysis and modeling of capabilities are achieved through a detailed design of the implied agents in the arrival procedure (aircraft agent in this case).

According to Prometheus methodology, the detailed design of agents is a third phase that requires two previous phases: a *system specification* phase and an *architectural design* phase. The system specification focuses on identifying goals and basic functionalities of the system. The architectural design uses the mentioned analysis to identify agents and interactions between them and with the environment. These activities have been performed by our group in previous work consisting, mainly, of defining an

architectural system and agent descriptors as well as proposing air-ground and air-air negotiation protocols [22-23].

After a review of previous outcomes, in this paper we focused on a detailed design an aircraft agent and how it will achieve its tasks within the system (dynamic behavior). By means of an iterative top-down modeling process, we define capabilities, internal events, plans and detailed data structures for the aircraft agent. Capabilities are modules within agent used to separate processes into individual components. Rest of elements will be explained later.

As main outcome of above design two central airborne capabilities for TBO operations were identified: *Trajectory Guidance* and *Navigation Procedure Management* capabilities. In addition, other airborne capabilities were considered for managing environmental information, conflicts, system alarms and other contingences.

Trajectory Guidance capability extends navigation guidance functionalities of current FMS to provide flight guidance along negotiated 4D trajectories. *Navigation Procedure Management* capability carries out tasks of calculating and negotiating user preferred trajectories as well as generates and manages events to implement referred trajectories. These cockpit capabilities are put on top of the current ones as it will be explain late.

This paper is organized as follows: first an overview of Prometheus methodology is presented and a briefly description about how this methodology has been applied to model arrival air traffic on a TBO scenario. Next, we focused on a description of inner airborne capabilities. Later, a more detailed explanation of Navigation Procedure Management capability is offered. After that, designed agent architecture is mapped as future cockpit architecture for TBOs. Then implementation guidelines are presented. Finally conclusions of this work are presented.

Applying Prometheus Methodology for Designing an Arrival TBO Conceptual Model

Prometheus methodology carries out an iterative process on three phases previously mentioned (see Figure 1): specification system, architecture design

and detailed design. Each of these phases provides artifact design (either as final artifact or either as intermediate design tool that derives on final artifacts). Final artifacts produce hierarchical structuring mechanisms which make possible designing models at multiple levels of abstraction [19]. Structured nature of design artifacts facilitates crosschecking for completeness and consistency of the model in each phase.

System Specification

The system specification defines *goals* of our model. Goals can be captured developing *scenarios* that illustrate essential aspects of system operation. Scenarios and goals help to identify initial system functionalities as well as to analyze the system-environment interface in terms of inputs (*percepts*) and outputs (*actions*). Scenarios are use cases that contain a sequence of steps each of them relating to a goal, an action, a percept or another scenario.

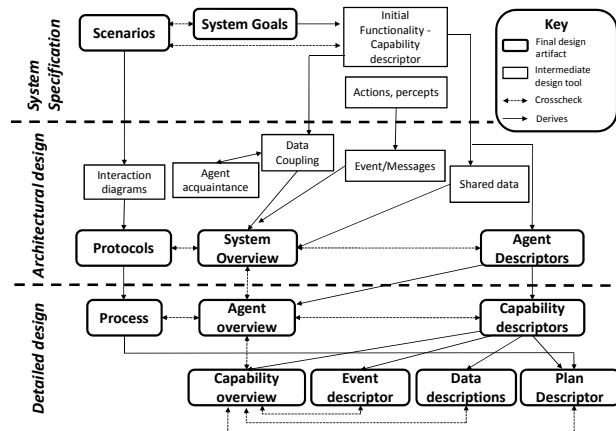


Figure 1. Prometheus Methodology Phases [20]

To define uses case scenarios, an automated air traffic general scenario was considered as a distributed processes where several autonomous and proactive entities (agents) plans and executes a set of coordinated tasks to provide arrival and approach free of conflict 4D trajectory. Moreover, guidelines from scenario proposed in DAG-TM (CE-11) project have been taken into account [5]. According to referred guidelines the flight crew: (i) could negotiate arrival preferred trajectories with ATC; (ii) is responsible for maintaining longitudinal spacing between consecutive aircraft once a trajectory (o constraints) has been assigned.

In above operational scenario, following agents¹ have been identified: Aircraft, Air Traffic Control (ATC), Meteorological Service Provider (MSP), Airspace Resources Provider (ASP) and Airline Operational Control (AOC). In addition several ATC agents could be defined in order to coordinate arrival ATC activities with en-route or departure ATC. However it is not essential when the study is focused on an airborne viewpoint. MSP, ASP and AOC agents' functionalities have been used to identify information requirements of ATC and Aircraft as well as their associated protocols to obtain referred information. In addition, human user and systems have been jointly modeled as an autonomous agent whose inner automatic processes are transparent to the human operator (crew or air-traffic controller).

Selection and organization of use case scenarios have been performed taking into account each agent perspective. Then, five root scenarios have been defined: (i) Manage Aircraft, (ii) Manage ATC, (iii) Manage Airline Operational Control, (iv) Provide Airspace Resources (v) Provide Weather Information.

Due to space limitations we focus this Prometheus methodology overview on the Aircraft Management scenario.

Manage Aircraft Scenario

Aircraft goals for TBO can be captured from this scenario which in turn contains the following four scenarios:

- *Update environmental information scenario* that covers associated processes.
- *Manage on board surveillance scenario* for monitoring flight obstacles, conflict detection and providing events containing proposed solutions.
- *Manage contingency scenario* to analyze arisen contingences and to provide advisable interventions.
- *Track trajectory scenario* to implement 4D trajectories and monitoring own trajectory tracking process.
- *Manage navigation procedures scenario* to manage trajectory planning and

negotiation processes and to generate events to trigger trajectory guidance.

Focusing on *manage navigation procedure* scenario, new scenarios were deployed. To do it, a gate-to-gate airborne trajectory based operation was considered as a sequential execution of operational procedures (taxi, departure, en-route, arrival-approach, landing, etc.). Moreover, others alternative procedures can be defined for each flight phase in order to manage abnormal and emergencies situations. Procedures contain specific attributes such as: associated 4D trajectory (expressed as a sequence of 4D points space-time constraints, vector instructions or a combination of them)², operational (or reference) points and areas to calculate and to negotiate trajectories, etc.

Therefore managing navigation procedure scenario consists of the following scenarios:

- *Planning flight-plan scenario* that covers calculations and communication processes for planning flight trajectory from current position to the destination airport. It provides procedures for each flight phase containing initial attributes (trajectories and flight segments to negotiate updated trajectory for each flight phase).
- *Execute procedure scenario*. It generates events to implement trajectories (or partial trajectory modifications) and to plan updated trajectories for next flight phase.
- *Plan next procedure scenario* to plan updated trajectories for a next procedure. Within this scenario, new *trajectory negotiation scenarios* can be added to illustrate air-ground negotiation processes. Last scenarios also belong to the Manage ATC scenarios tree.
- *Re-plan current procedure scenario* that plans partial modifications for the current executing trajectory when airborne contingences arises. Therefore, new sub-scenarios can be included to illustrate air-air negotiation and air-ground negotiation processes. Note that air-ground negotiation

¹ Usually identification of agents as autonomous entities is impossible until to achieve the architectural design phase. However in air traffic scenarios it is immediate to recognize autonomous entities in the system specification phase.

² Proposal about the best possible way for describing and exchanging aircraft trajectory intents in a common format for air and ground systems and for a readable human compression are currently under studio [11].

scenarios for modifying current procedure can be integrated within Managing ATC sub-scenarios.

From preceding scenarios, a list of initial aircraft goals was arranged (see Figure 2). In an iterative

process sub-goals for each main goal were obtained. Sub-goals indicate how the corresponding parent goal can be achieved. Then a goals tree enabled identifying main aircraft functionalities.

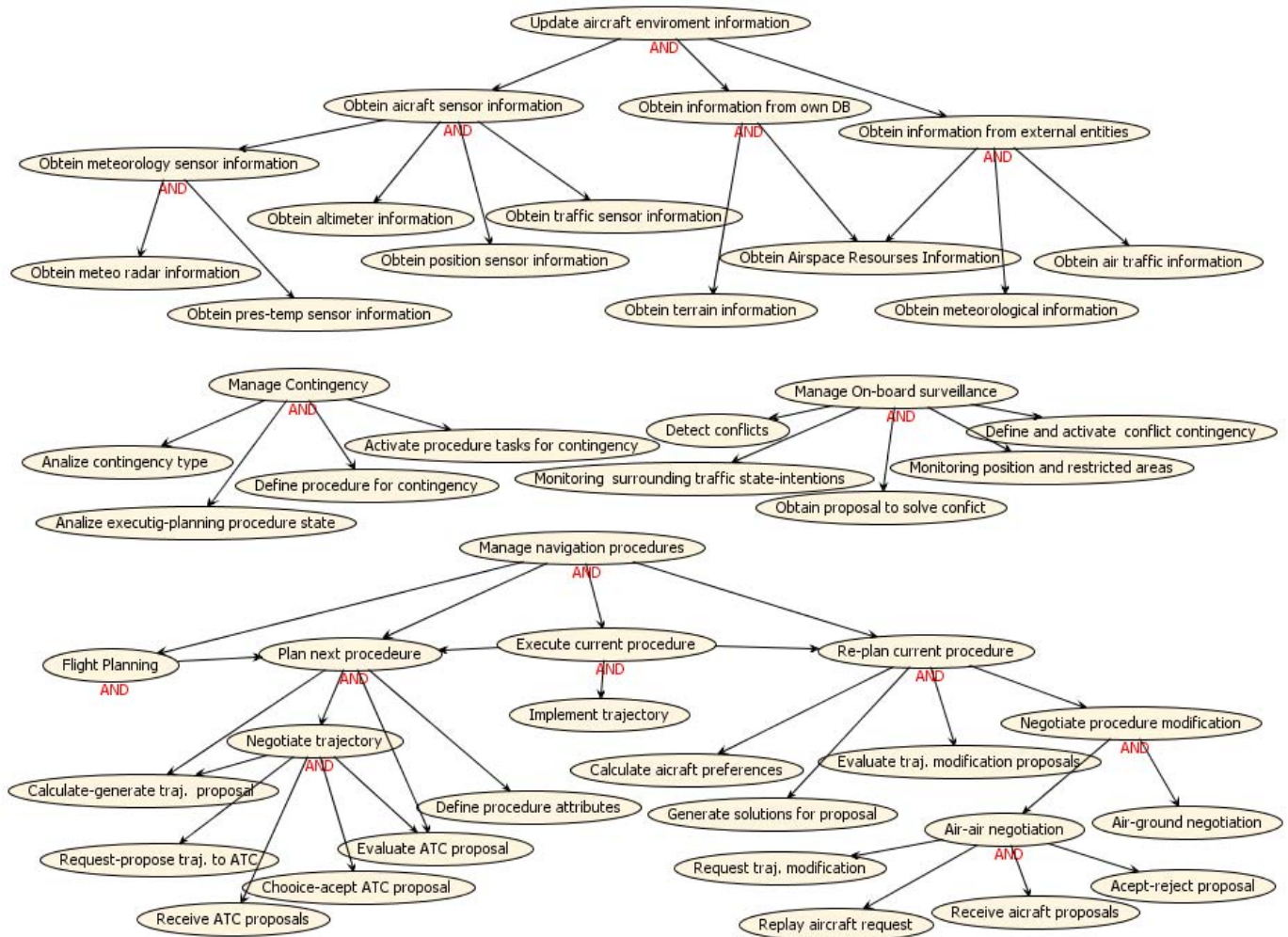


Figure 2. Aircraft Agent Goals

Architecture Design

Three aspects are essential (final artifacts in figure 1) in this phase:

- To decide about which *agents should exist*. In our case, agent identification is obvious from specification system phase. Therefore the design process continues refining agent functionalities from previous goals. Functionalities are small chunks of system behavior that allows identifying inputs,

outputs as well as used and produced data. Functionalities are the base to define *agent capabilities* described in next sessions.

- Capturing overall system structure (static) by means of a *system overview diagram*. This diagram ties agents, showing data used by each agent as well as their percepts and actions. Furthermore pathways of communication between agents are considered.

- Describing system dynamic behavior by means of interaction *protocols*. Protocols capture timing of communication of related messages between agents. They can be depicted using agent UML (AUML) notation [24].
- In a previous work two interactions protocols have been designed. The first ones, consists of a basic air-ground negotiation protocol to negotiate arrival trajectories [22]. This protocol represents the core of an arrival-approach procedure planning process. Arrival-approach procedure planning process (and therefore the mentioned protocol) is activated by a specific event produced while previous en-route navigation procedures is executing. A second protocol involves an air-air negotiation process that illustrates re-planning modifications of an implemented arrival procedure [23]. Mentioned protocols are included within the agent plan library. Moreover, communication and negotiation processes are carried out through an interchange of messages following the standards of agent communication language [25]

Detailed Design

In detailed design phase, the internal agent architecture (*agent overview*) and its dynamic behavior are developed. As it was explained before, agent activities are grouped within *capabilities*. Then, activities carried out by capabilities and agent inner messages between capabilities are specified. Some capabilities are described using new capabilities at a lower level. At bottom level, capabilities are defined in terms of *plans*, *events* and *data*. Plans define different ways of responding to an event and, therefore, they describe the agent dynamic behavior. Each plan is divided into a number of sub-tasks which are triggered by a specific event. Tasks are implemented using conventional programming structures according to the platform for implementations. Events consist of the arrival of a percept, arrival of a message from another agent or an internal message or subtask within the agent.

Refining interaction protocols, scenarios and goals results into *process specifications*. *Process*

diagrams depict inner agent processes that are triggered by an incoming message or inner event.

Prometheus methodology as described in [20] includes notations for capturing designs. Notation used in next aircraft agent and capabilities diagrams is depicted in Figure 3.

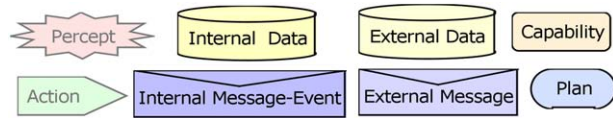


Figure 3. Notation Used in Agent and Capability Overview Diagrams

Airborne Capabilities

Taking into account goals and functionalities, six airborne capabilities have been considered (see Figure 4).

- Aircraft Environment Information Management
- Alarms Management of Aircraft Systems
- Conflict Detection-Resolution
- Airborne Contingency Management
- Trajectory Guidance
- Navigation Procedures Management

Aircraft Environment Information Management

Main goal of this capability is maintaining an updated onboard environmental knowledge. Information is obtained from aircraft sensor systems and from incoming agent messages.

Capability plans provide information about: sensor data, weather forecast, restricted areas, air space recourses (e.g. available arrival routes and gateways), surrounding air traffic and contingency events concerning to significant environmental changes.

Alarms Management of Aircraft Systems

Alarm system outputs are managed by this capability to provide contingency events related to alarm characteristics.

Conflict Detection-Resolution

As its name suggests, it is responsible for detecting conflicts with other aircraft or obstacles (terrain, adverse weather areas, etc.). It also provides a set of ranked proposals for conflict resolutions. Furthermore, proposals are negotiated and/or implemented by means of other capabilities.

To achieve above goals capability is composed of two sub-capabilities: (i) *Conflict Detection Capability* and (ii) *Initial Conflict Solution Capability*.

Conflict Detection Capability detects conflicts and provides detailed data about it. Also it is used to test trajectories synthesized during trajectory planning processes. This capability is constituted by several plans each of them contains a specific model to detect short, medium and long term conflicts. Plan

inputs are predicted trajectory, restricted areas as well as surrounding traffic state and intentions.

Conflict data calculated by previous plans are integrated by a new plan to obtain a detailed conflict description and to generate conflict events.

Initial Conflict Solution Capability uses several inner plans to supply solutions according conflict data input. Results of referred plans are applied to other one that provides conflict contingency events (including conflict information and a set of ranked conflict solutions)

Airborne Contingency Management

This capability deal with deciding procedural tasks according to received contingency input events. Procedural tasks are defined as contingency output events to be considered within the current or the next procedure.

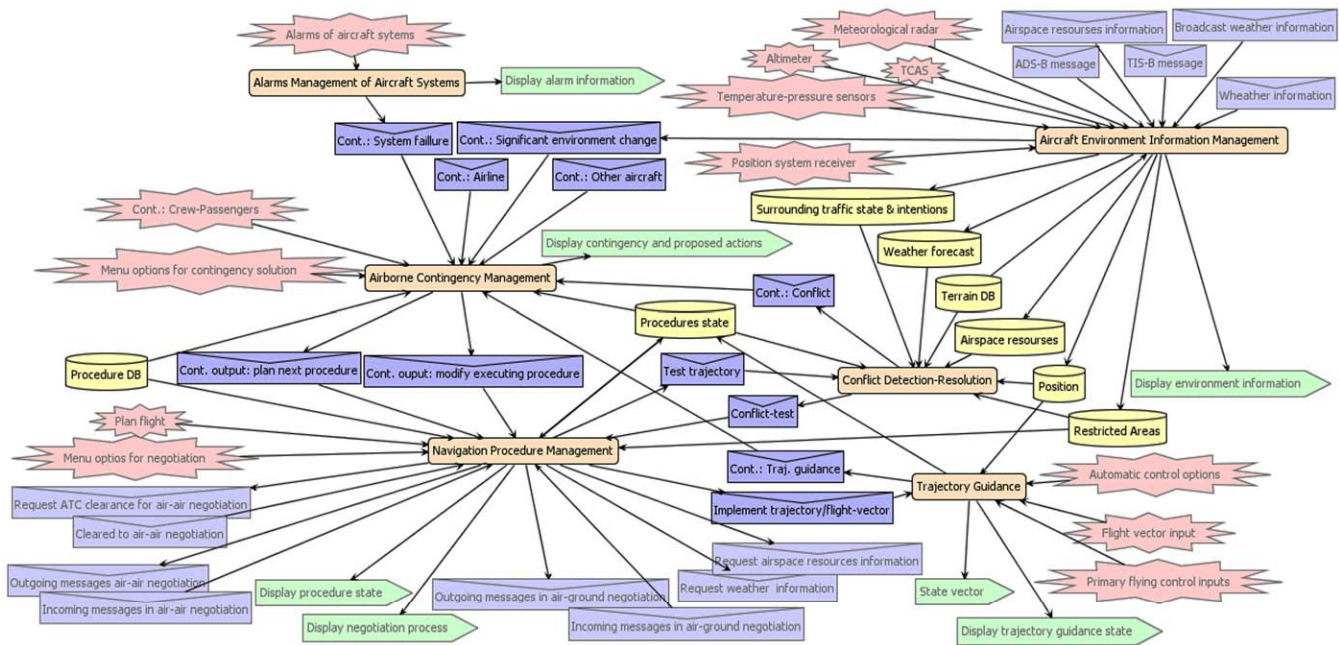


Figure 4. Aircraft Agent Architecture

The following contingencies inputs have been identified:

- Contingency of environmental significant changes.
- Failure system contingency: indicates failure details as well as proposed procedure, maneuver or actions according

to normal, abnormal or emergency procedures.

- Conflict contingency: provides information about solutions proposed by the conflict detection-resolution capability.
- Other aircraft contingency: consists of a requirement from other aircraft requesting

to solve conflicts, to modify arrival sequence, etc.

- Airline contingency: requesting to modify intended flight plan.
- Contingency of ATC: e.g. changes regarding previous agreement.
- Contingency of crew or passengers defined through an on board options menu.

The contingency solving process can be treated in an automatic manner and therefore this capability requires future efforts to design suitable decision-making schemes. In addition, a feasible air crew intervention within the decision process can be performed by means of an *ad hoc* options menu.

Trajectory Guidance

This capability (Figure 5) represents flight control systems for trajectory guidance at several automation levels (3D/3.5D/4D) or for flight-vector guidance (Autopilot and/or Flight Director). Therefore it requires extending functionalities of current *FMS* and *Flight Director* to perform 4D trajectory guidance.

Main inbound data to this capability consists of an event (*implement trajectory-vector event*). Mentioned event is produced by the *Navigation Procedures Management Capability* that it will be described later. Moreover it triggers a plan that updates and defines trajectory format to be used by a 4D FMS or an Auto-pilot/Auto-throttle system.

Other inputs coming from the user interface are: automatic control options, flight vector input and primary flight control input. Automatic control options define automation levels to execute trajectories.

Information about real-time guidance process (trajectory state) is stored and contingency events are generated to identify guidance difficulties.

Finally, aircraft actions are represented by the state-vector from aircraft dynamic.

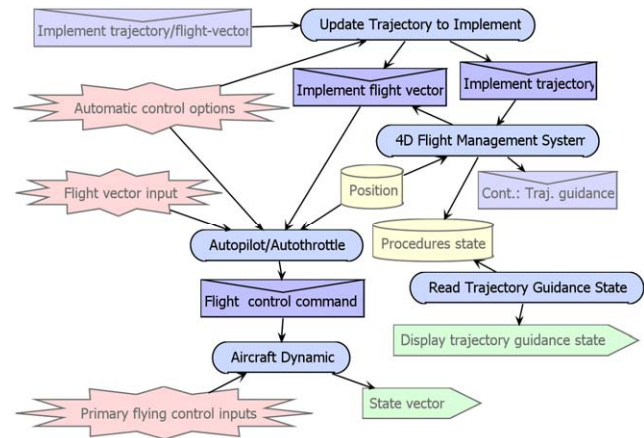


Figure 5. Trajectory Guidance Capability

Navigation Procedures Management

According to correspondent goals, this capability is responsible for managing procedures for planning, implementing or modifying trajectories of each flight phase. Mentioned capability, together with the Trajectory Guidance one, could represent the core of a next-generation of *Navigation Management Systems*. Details of this capability are described next.

Navigation Procedures Management Capability

This capability encloses four sub-capabilities (Figure 6):

- Flight Planning
- Procedure Executing
- Current Procedure Re-planning
- Next Procedure Planning

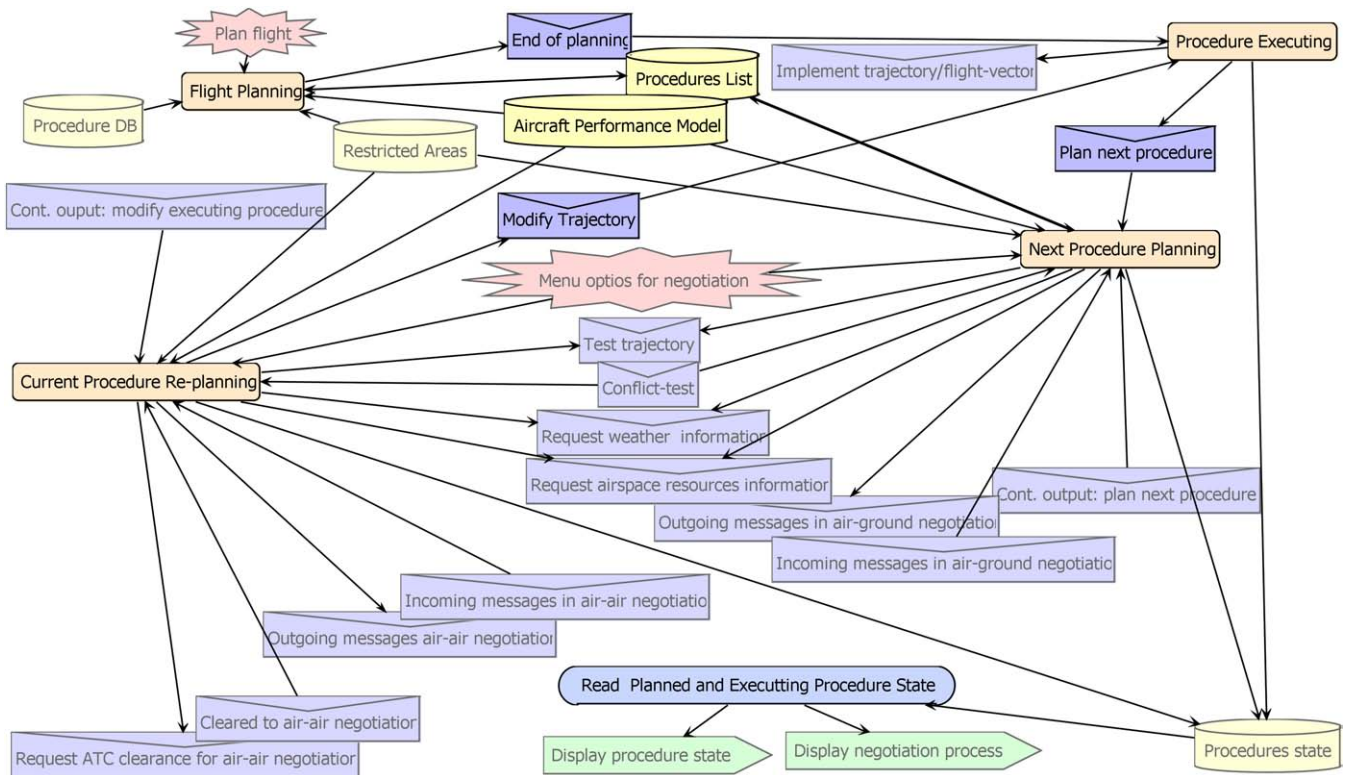


Figure 6. Navigation Procedures Management Capability

Despite our conceptual model is focused on arrival air traffic operations, *Flight Planning Capability* has been considered in order to define previous trigger events for planning the arrival-approach procedure during the preceding en-route procedure execution. Besides, an overall overview of navigation procedures is obtained for future extension of this work.

A basic *Flight Planning Capability* has been modeled by means of two main plans. An initial plan contains tasks leading an automatic air-ground negotiation process for a gate-to-gate trajectory. A second plan fills attributes for a gate-to-gate procedures list.

Procedure Executing Capability contains three mains plans for: (i) updating current attributes of running procedure (triggered when a modification has been planned), (ii) updating and ordering the procedures list (when current procedure ends), (iii) executing tasks of the current procedure. Last plan is responsible for generating events and data to execute trajectories and to start next procedures

planning. Updated procedure state is also stored. Procedure state variables describe planning or execution phases for each procedure. For example, a procedure can be in one of the following four states: (i) waiting for planning, (ii) planning, (iii) waiting for executing, (iv) executing or (v) executed. In turn, previous states include new sub-states and so on to represent new particular phases (e.g. phase of a negotiation process)

Current Procedure Re-planning Capability contains a plan library to modify in several ways current trajectory attributes (see capability overview in Figure 7). The selected plan depends on the information included in the corresponding contingency output event. We have focused on a plan for modifying the arrival-approach procedures when it is being flown. This plan, in turn, generates events to trigger new specific plans that perform explicit modifications. For example, a plan to trigger and to drive an air-air negotiation to modify current arrival aircraft sequence has been implemented in previous work [23].

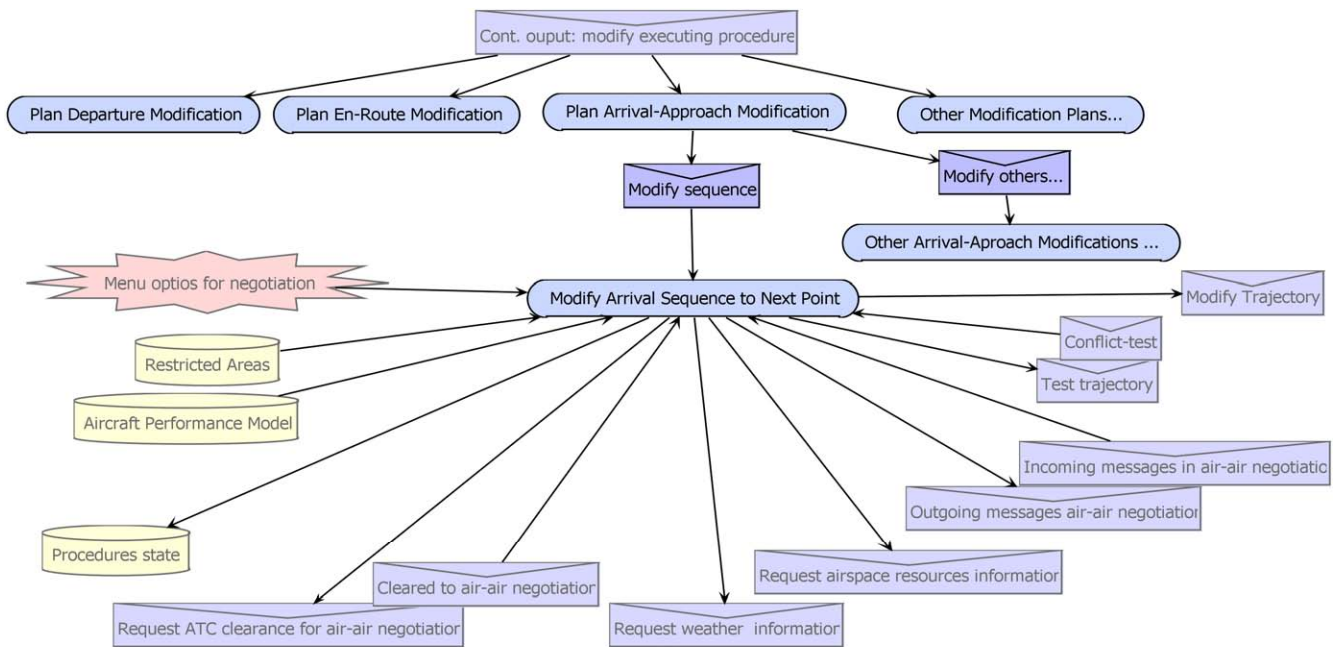


Figure 7. Current Procedure Re-Planning Capability

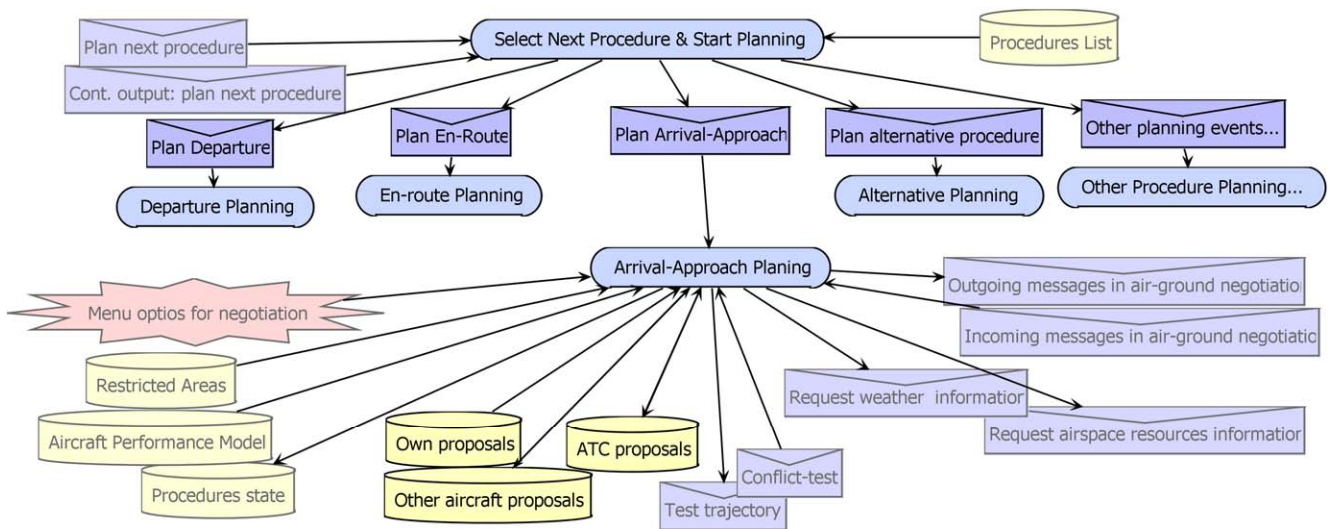


Figure 8. Next Procedure Management Capability

Next Procedure Planning Capability is responsible of selecting and planning next plan of the procedure list (see capability overview in Figure 8). Selecting a next procedure is performed by a plan (named *Select Next Procedure and Start Planning*) that generates events to trigger new specific plans for each type of procedure. Last plan is triggered by events coming from *Procedure*

Executing Capability or from *Contingency Management Capability*. Figure 8 shows data, events and messages used/produced by *Arrival-Approach Planning*. The plan manages an air-ground trajectory negotiation described in [23]. Therefore incoming and outgoing communication messages are referred to mentioned negotiation

process. The plan is triggered by the *Plan Arrival-Approach* event (inner message).

Also, data required by this capability are the following: (i) own position, (ii) aircraft performance, (iii) state and intention of rounding traffic, (iv) procedure data base. Provided data consists of information about the planning process and it is jointly stored with the current procedure state.

From Agent Capabilities to Cockpit Systems

An aircraft agent capability enables a direct correspondence with cockpit functionalities. Figure 9 shows cockpit system architecture for TBO obtained from the described agent architecture.

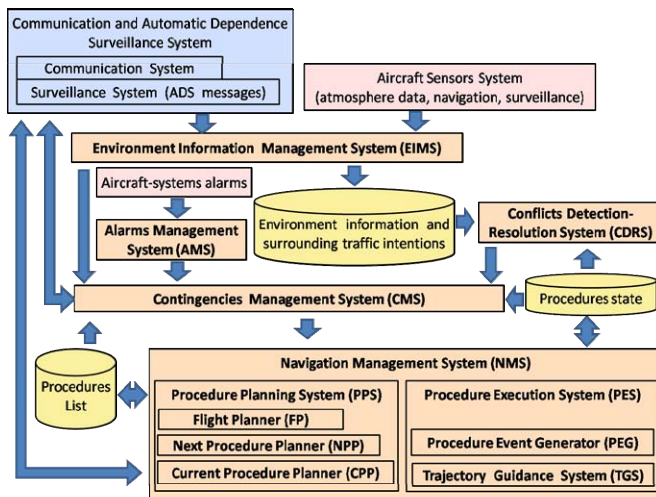


Figure 9. Cockpit Functionalities

Three main data groups have been considered: (i) Environment and surrounding traffic information, (ii) Procedures list and (iii) Procedures state and trajectory state.

Information about aircraft environment, including surrounding traffic state and intentions, are used by the Alarms System (AS) and Conflict Detection and Resolution System (CDRS) for generating specific contingency events to be treated by Contingence Management System (CMS).

CMS maps the Contingence Management capability and therefore uses mentioned events and other ones (e.g. incoming messages, percepts from

pilot interface) to provide decisions events. Consequently, CMS represents a first layer of a decision-making process, placed on top of the Navigation Management System (NMS).

NMS includes Navigation Procedure Management and Trajectory Guidance Capabilities. NMS is composed of Procedure Planning System (PPS) and Procedure Executing System (PES). Planning capabilities (i.e. Next Planning Procedure Capability and Current Procedure Re-planning Capability) are supported by the PPS system. Also, PES system consists of: (i) Procedure Event Generator (PEG) that performs Procedure Executing Capability and (ii) Trajectory Guidance System (TGS) implemented through the Trajectory Guidance Capability.

Therefore NMS extends current FMS and AP/AT functionalities in the following manner: (i) TGS extends FMS/AP/AT flight plan guidance functionality for providing 4D trajectory guidance, (ii) PEG expands flight plan managing functionalities for managing executing procedure. (iii) PPS supports full procedure planning processes (that includes air-ground and air-air negotiation) versus planning flight functionalities of current FMS.

From a Conceptual Model to an Executable Model

Conceptual model presented in this paper has been implemented in order to obtain an executable model to support analytical simulation.

Depending on functionalities implemented, the executable model can be used either in continuous simulation or discrete event simulation.

In nature, continuous simulation requires implementing the aircraft dynamic in order to represents model behavior over a continuum-time. It is essential in real-time and human-in-the-loop simulations. Also is suitable for fast analytical simulations intended for preliminary designing and evaluation of cockpit systems and underlying mathematical models and algorithms (e.g. for trajectory guidance trajectory synthesis and evaluation model, etc.). However when mentioned models are not available, proposed conceptual model enables discrete event simulation. Discrete

event simulation relies on the occurrence of specific events to advance models from a state to another over time. In this case, events can be generated by random functions implemented within capabilities planes representing underplaying models as *black boxes*.

For this purpose a Java Agent Development Framework (JADE) Platform [26] has been used for implementing an executable model for discrete event simulation. JADE is one of most extended multi-agent platforms and it provides (conform to FIPA standards [25]) infrastructure for inter-agent communications and for managing software agents. In this way, random functions to generate events are implemented using conventional java programming structures within described agent planes. In addition, planes are included within several *JADE behaviors* [26]. Then, the executable model improves its performance when functions are replaced by specific underplaying models as they are developing. More implementation details will be presented on next papers.

Conclusions

An agent oriented analysis and modeling of airborne capabilities for trajectory based operations has been presented. Despite, analysis has been focused on arrival flight procedures the obtained model can be extended to include new procedures within gate-to-gate operations.

The analysis and design of the referred model has been supported by Prometheus methodology. Airborne capabilities have been obtained as result of a detailed design of the aircraft agent architecture. Detailed design is part of previous work of our group where referred methodology was applied to model arrival TBO scenario as a multi-agent system.

The Navigation Procedure Management Capability together with Trajectory Guidance and Airborne Contingency Capabilities are the core around future Navigation Management Systems (NMS) for TBO could be developed. Thus, NMS is described as an onboard system that includes flight planning and navigation guidance capabilities of current FMS adding other ones such as: (i) Obtaining user preferred trajectories, (ii) Leading trajectory negotiation processes, (iii) Evaluating 4D

trajectory proposals from other agents, (iv) Generating new proposals for other agents and (v) Providing flight guidance along negotiated 4D trajectories.

References

- [1] Callantine, T., T. Prevot, N. Smith, E. Palmer, 2001, Simulation of CTAS/FMS Air Traffic Management, 4th USA/Europe Air Traffic Management R&D Seminar, Santa Fe.
- [2] Bailey, E., R Verhoeven, 1999, PHARE Airborne Programme Final Report, DOC 98-70-19, EUROCONTROL, Bruxelles, Belgium.
- [3] Fairclough, Ian, 1999, PHARE Advanced Tools Arrival Manager Final Report, PHARE Document DOC, 98-70-18 (Vol 2 of 10), Bruxelles, Belgium.
- [4] Prevot, T., S. Shelden, E. Palmer, W. Johnson, V. Battiste, N. Smith, T. Callantine, P. Lee, J. Mercer, 2003, Distributed air/ground traffic management simulation: results, progress and plans, Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference.
- [5] Sorensen, John A, 2000, Detailed Description for CE-11 Terminal Arrival: Self Spacing for Merging and In-trail Separation, NASA Ames Research Center and NASA Langley Research Center, Moffett Field, CA and Hampton, VA.
- [6] Teutsch, J., E. Hoffman, 2004, Aircraft in the future ATM system-exploiting the 4D aircraft trajectory, Digital Avionics Systems Conference, DASC 04.
- [7] Courses, E., T. Surveys, 2007, 4-Dimensional Trajectories and Automation Connotations and Lessons learned from past research, Integrated Communications, Navigation and Surveillance Conference, ICNS'07, pp. 1-10.
- [8] Brooker, P., 2004, Airborne Separation Assurance Systems: towards a work programme to prove safety, Safety Science, vol. 42, pp. 723-754.
- [9] Alexander, A.L., C.D. Wickens, D.H. Merwin, 2005, Perspective and coplanar cockpit displays of traffic information: Implications for maneuver choice, flight safety, and mental workload, The International Journal of Aviation Psychology, vol. 15, pp. 1-21.

- [10] Korn B., A. Kuenz, 2006, 4D FMS for Increasing Efficiency of TMA Operations, IEEE/AIAA 25th Digital Avionics Systems Conference, DASC'06, pp. 1–8.
- [11] Lopez-Leones, J., M. Vilaplana, E. Gallo, F. Navarro, and C. Querejeta,, 2007, The Aircraft Intent Description Language: A key enabler for air-ground synchronization in Trajectory-Based Operations, Digital Avionics Systems Conference, DASC '07. IEEE/AIAA 26th, 2007, pages 1.D.4-1-1.D.4-12.
- [12] Feigh, K.M., A.R. Pritchett, A.P. Shah, S.A. Kalaver, A. Jadhav, D.M. Holl, A.Z. Gilgur, 2005, Analyzing Air Traffic Management Systems Using Agent-based Modeling and Simulation, Georgia Institute of Technology, Baltimore, MD.
- [13] Callantine, T.J., E. Palmer, J. Homola, J. Mercer, y T. Prevot, 2006, Agent-Based Assessment of Trajectory-Oriented Operations with Limited Delegation, Proc. 25th Digital Avionics Systems Conf., Portland, OR.
- [14] Callantine, T.J., J. Homola, J. Mercer, T. Prevot, 2006, Concept Investigation via Air-Ground Simulation with Embedded Agents, AIAA Modeling and Simulation Technologies Conference and Exhibit, Keystone, Colorado.
- [15] Satapathy, G., V. Manikonda, 2004, Agent Infrastructures for Modeling and Simulation of CNS in the NAS, Fairfax, VA.
- [16] Nguyen-Duc, M., J.P. Briot, A. Drogoul, 2003, An application of multi-agent coordination techniques in air traffic management, IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), pp. 622-625.
- [17] Gorodetsky, Vladimir, Oleg Karsaev, Vladimir Samoylov, Victor Skormin, 2008, Multi-Agent Technology for Air Traffic Control and Incident Management in Airport Airspace, in: Proceedings of AAMAS, International Workshop Agents in Traffic and Transportation, Lisbon, Portugal, May 12–16 2008, pp. 118–125.
- [18] Ding, Y., J. Rong, J. Valasek, 2003, Automation Capabilities Analysis Methodology for Non-Controlled Airports, AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas.
- [19] Padgham, L., M. Winikoff, 2002, Prometheus: A pragmatic methodology for engineering intelligent agents, Proceedings of the OOPSLA Workshop on Agent-Oriented Methodologies, pp. 97–108.
- [20] Padgham, L., M. Winikoff, 2004, Developing Intelligent Agent Systems: A Practical Guide, John Wiley & Sons,
- [21] Sudeikat, J., L. Braubach, A. Pokahr, W. Lamersdorf, 2004, Evaluation of agent-oriented software methodologies—examination of the gap between modeling and platform, Agent Oriented Software Engineering (AOSE), Springer.
- [22] Rodríguez, J.M.C., L.G. Déniz, J.G. Herrero, J. B. Portas, J.R.C. Corredera, A 4D trajectory negotiation protocol for Arrival and Approach sequencing, Integrated Communications, Navigation and Surveillance Conference, 2008. ICNS 2008, 2008, pp. 1-12.
- [23] Canino,J., L. Gomez, J. Garcia, J. Besada, y J. Casar, Design of an air-air negotiation protocol to reorder aircraft arrivals sequence, Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th, 2008, pp. 3.C.6-1-3.C.6-11.
- [24] Huget, M.P., Agent uml notation for multiagent system design, IEEE Internet Computing, vol. 8, 2004, pp. 63–71.
- [25] Foundation for Intelligent Physical Agents. 24 Communicative Act Library Specification, Version J. <http://www.24.org/specs/2400037/>.
- [26] Luigi, Fabio, Giovanni Caire, Dominic Greenwood, 2007, Developing Multi-Agent Systems with JADE, Wiley Series in Agent Technology, Hardcover.

*28th Digital Avionics Systems Conference
October 25-29, 2009*