Designing a Cockpit Functionalities Architecture for Trajectory Based Operations

A methodological agent-oriented analysis and modelling

José Miguel Canino, Luis Gómez
Signal and Communications/Electronic Engineering Dept.
University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
jcanino@dsc.ulpgc.es, lgomez@diea.ulpgc.es

Jesús García
Computer Science Department
Carlos III-University of Madrid
Madrid, Spain
jgherrer@inf.uc3m.es

Juan Besada, José Ramón Casar
Signal, System and Radiocommunication Dept.
Polytechnic University of Madrid
Madrid, Spain
besada@grpss.ssr.upm.es, jramon@grpss.ssr.upm.es

***Abstract*— Trajectory Based Operations (TBO) will require new procedures and systems to achieve a suitable automation of air-traffic operations. Procedures and systems for automated operations are closely related and therefore frequently they need to be modeled in a combined way. Our group is currently employing recent agent-oriented methodological approaches to obtain conceptual models about TBO scenarios. Conceptual models define roles of air traffic entities as well as their interactions together with a detailed description of the entities' architecture and dynamic behaviour. In this paper we present a cockpit functionality architecture built upon a methodological analysis and design of a TBO scenario as a multi-agent system. The proposed design has the advantage of mapping to an executable model for analytical simulation of TBO concepts and its modular architecture allows for a progressive integration of additional underlying models with specifIc functionalities.**

Keywords- Trajectory Based Operations; automated flight procedures; cockpü system; analysis and modeüing; muüi-agent system

I. INTRODUCTION

The Trajectory-based Operations (TBO) concept [1] has been proposed moving the current clearance-based and centralized air traffic control concept towards a trajectory-based and decentralized one. However, real implementation of TBO concepts will require several efforts aimed at:

- Defining the detailed roles of air-crew and air traffic controllers in order to provide efficient air traffic flows which take into account user preference trajectories (UPT). It requires developing new air and ground procedures to lead trajectories planning, coordination and execution tasks. Then, procedures should guide ad hoc protocols and processes for: *(i)* air-ground and air-air trajectories negotiation, *(i)* monitoring aircraft states and intentions, *(iii)* solving unexpected events during the procedure execution, etc.
- Developing systems and new human-machine interfaces to execute the above procedures: e.g. cockpit system for Airborne Assurance System (ASAS) and Cockpit Display Traffic Information (CDTI) [2-3] that includes suitable user interfaces for performing air-ground and air-air negotiation processes; FMS with four-dimensional trajectory guidance capabilities (4D-FMS) [4]; ground systems for sequencing, de-conflicting and monitoring of arrival traffic, etc. In addition, a new high level natural language is necessary to achieve a precise intercommunication between aircraft systems and ground systems [5]. Moreover, this language should enable human-readable compression of communication processes.
- Deploying new underlying mathematical models and algorithms to support the mentioned functionalities: i.e. trajectory synthesis models, conflict detection and resolution algorithms, four-dimensional (4D) trajectory guidance models, etc.

The high interdependence of above requirements often makes impossible to outline a preliminary design of procedures, support systems, underlying models and communication ¿mguages in an independen! way.

Henee, it is necessary to develop conceptual models of air traffic scenarios to provide a highly detailed description of these interdependencies. The architecture of these conceptual models should be robust enough to: *(i)* obtain a basic executable model to analytical simulation (discrete events and/or dynamic simulation), *(ii)* add new specific underlying models and functionalities as they are designed.

Modelling complex and distributed air traffic scenarios can be considered as a software problem for which the agent-oriented programming provides a natural response.

Current agent-based approaches to modelling and simulating have focused on models that represent several functionalities of physical entities in air traffic scenarios: aircraft, air-traffic services providers, airlines, etc. [6-7]. CNS (Communication, Navigation and Surveillance) aspects (delays and information uncertainty) have also been modeled as agents in the simulation environment [8]. Other approaches investigated the application of multi-agent coordination techniques using some generic practical coordination models [9-10]. In addition, decentralized Air Traffic Management approach was proposed [11], which focused on the design of an automated arrival/departure system for non-controlled airports. Moreover, studios of modelling and simulation of air transport systems suggest there is a need for new contributions of more integrated and flexible models [12-14]. In particular, procedures and the corresponding air and ground systems functionalities for automated TBO are not often modeled in a combined manner [14].

Our group has been developing during the recent past years, a conceptual agent-based model of air traffic scenarios under a TBO perspective. The mentioned interdependences between procedures, systems and underlying mathematical models have been taken into account within the model. The scenario under study focused on arrival and approach air traffic. This scenario represents a significant variability of the operational conditions and a higher workload for crews and air traffic controllers. Therefore it is easily extendable to gate-to-gate scenarios.

Modelling referred scenarios as a multi-agent system was carried out through a methodological process. Current agent technology provides practical and formal methodologies to analyze and design, in a structured and consistent manner, the following issues: *(i)* roles and functionalities of autonomous entities (agents) that take part in an operational scenario, *(ii)* interactions between agents (or agent protocols) and (iii) inner architecture and dynamic behaviour (processes) of agents.

Several multi-agent methodologies approaches have been proposed in recent years and comparative analysis between them are beginning to appear [15]. Prometheus agent-oriented well-established methodology has been selected to provide guidelines to develop the mentioned multi-agent system [16]. We argue that Prometheus suits well for solving our problem due to: *(i)* the highly detailed 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 functionalities of airborne and ground systems), *(iii)* the easy translation from the conceptual model into an executable model by means of current agent platforms such as JADE [17], JADEX [18], JACK [19], etc.

In this paper we present a review of the mentioned related work. We specifically focus the analysis to modelling cockpit systems capabilities for TBO.

The main result from this work is the design of an architecture for cockpit capabilities for TBO operations. Three central airborne capabilities were identified: Trajectory Guidance, Navigation Procedure Management and Contingence Management capabilities. These cockpit capabilities extend functionalities of current Flight Management Systems (FMS) and autopilot/autothrottle (AP/AT) systems. In addition, other airborne capabilities were considered for managing aircraft environmental information and system alarms and for providing conflict detection and resolution. Moreover, an initial characterization of airborne events for managing procedures and contingences as well as data to represent T B O procedure states is provided.

This paper is organized as follows: first a brief overview illustrates a Prometheus-based methodological approach for analyzing and designing a conceptual model of an arrival T B O scenario as a multi-agent system. Next the detailed design of an aircraft agent results in the capabilities-based cockpit architecture1. Later on, we focus on the Navigation Procedure Management capability to illustrate: *(i)* the core of a new cockpit system for procedure managing, *(ii)* how agent (or cockpit) capabilities can integrate new capabilities, events, plans, data, etc., and *(iii)* how future underlying mathematical models for cockpit system can be implemented within the capability. Then, from above design, a procedure state data structure for automated flight procedure management is identified. In addition, the aircraft architecture is mapped to cockpit-system architecture for TBOs. Finally, conclusions are presented.

II. PREVIOUS WORK: DEVELOPING A TBO CONCEPTUAL MODEL UNDER A AGENT-BASED METHODOLIGICAL APPROACH

Prometheus methodology proposes developing multi-agent system models through an iterative process performed on three phases (see Figure 1): specification system, architecture design and detailed design. Each of these phases provides artefact design (either as final artefact or either as intermediate artefacts). Final artefacts produce structured elements for modelling at multiple levels of abstraction [16]. Intermediate artefacts capture details for final artefacts. The structured nature of design artefacts facilitates crosschecking for completeness and consistency of the model in each phase.

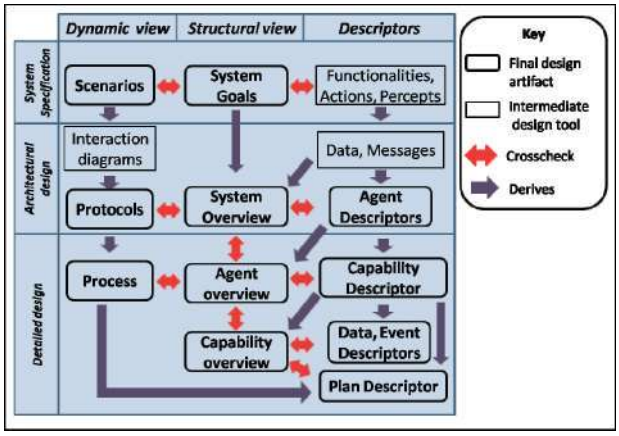

Figure 1. Prometheus methodology

[1] Note that in this paper the word *capability* can have two similar but distinct meanings: a general meaning synonymous of ability (referred to functionalities abilities) and a Prometheus meaning (that defines it as agent modules used to separate processes into individual components). In this case it refers to the last one.

A. System specification

The system specification defines the *goals* of our model. Goals can be captured by developing *scenarios* that illustrate essential aspects of system operations. Scenarios and goals help to analyze the main system functionalities and 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.

To define use cases scenarios, an automated air traffic general scenario was considered as a distributed process where several autonomous and proactive entities (agents) plan and execute a set of coordinated tasks to provide an arrival and approach free of conflict 4D trajectory. Moreover, guidelines from scenario proposed in DAG-TM (CE-11) project have been taken into account [20]. 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 (or restrictions) has been assigned.

In the above operational scenario, the following agents[2] have been identified: Aircraft, Air Traffic Control (ATC), Meteorological Service Provider (MPS), 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, this is not essential when the study is focused on the 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 modelled as an autonomous agent whose inner automatic processes are transparent to the human operator (crew or air-traffic controller).

Selection and organization of use cases scenarios have been performed taking into account each agent's 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.

Each of the previous scenarios was deployed into several sub-scenarios (some of them are common to mentioned ones). Due to a lack of space we focus on the Aircraft Management scenario which is composed of the following sub-scenarios: *Update environmental information*, *Manage on-board surveillance*, *Manage contingencies*, *Track trajectory* and *Manage navigation procedure* scenarios. In turn, the *Manage navigation procedure* scenario consists of new scenarios that were captured considering on-board tasks for arrival and approach flight operation. To obtain more details of the mentioned scenario, a flight procedure for TBO was considered as a set of tasks in order to plan, execute or modify the trajectory for a specific flight phase. Planning a procedure consists of calculating and negotiating its associated trajectory. Executing a procedure consists of executing the referred trajectory. Besides, when a procedure is executing and contingences arise, a trajectory modification could be required. Then, a gate-to-gate airborne trajectory based operation consists of a set of flight procedures (taxi, departure, en-route, arrival-approach, landing, etc.) that must be planned and executed (and sometimes modified) in a sequential way. Moreover, other alternative procedures can be defined for each flight phase in order to manage abnormal and emergencies situations.

Apart from tasks, flight procedures also contain several specific attributes such as: an associated 4D trajectory (expressed as a sequence of 4D points space-time restrictions, vector instructions or a combination of them)[3], operational (or reference) points and areas to calculate and to negotiate trajectories, etc.

Therefore, the *Manage navigation procedure* scenario consists of the following scenarios: *Planning flight-plan*, *Execute procedure*, *Plan next procedure* and *Re-plan current procedure*.

Planning flight-plan covers calculation and communication processes for planning the flight trajectory from the current position to the destination airport. It provides initial procedure attributes for each flight phase (trajectories and flight segments to negotiate updated trajectories for each flight phase). The *Execute procedure* scenario analyzes the current procedure and generates events to implement trajectories (or partial trajectory modifications) and to trigger the next procedure planning. The *Plan next procedure* scenario carries out the trajectory planning process to update the trajectory and other attributes for a next flight phase. Finally, the *Re-plan current procedure* scenario plans partial modifications for the current executing trajectory when airborne contingences arise.

In order to define the trajectory negotiation tasks required for planning the next procedure or for modifying an executing procedure, new scenarios where identified within the above ones. Therefore, the *Plan next procedure* scenario contains *air-ground trajectory negotiation* scenarios. In the same way, the *Re-plan current procedure* scenario contains sub-scenarios to illustrate air-air negotiation and air-ground negotiation required for modifying a trajectory under execution. Note that *air-ground negotiation scenarios also belong to the Manage ATC scenarios tree*.

From the above scenarios, a list of initial aircraft goals was derived. Then, by means of an iterative process, sub-goals for each main goal were obtained. Sub-goals indicate how the corresponding parent goal can be achieved. Thus a goals tree enabled identifying the main aircraft functionalities[4]. Also, actions and percepts were identified. For the aircraft agent, actions consist of the aircraft movement and outputs for pilot

[2] Usually the identification of agents as autonomous entities is impossible before the architectural design phase. However, in air traffic scenarios it is immediate to identify autonomous entities during the system specification phase.

[3] Proposals 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 [5].

[4] For more details of *Manage Aircraft* scenario and corresponding aircraft goals see [20].

graphical interfaces. Percepts come from aircraft sensors and from the pilot-interface (options menus, controls, etc).

B. Architecture design

The architecture design phase captures: *(i)* the overall system structure (static) by means of a system overview diagram that ties agents, showing interaction protocol names, and the data used by each agent, as well as their percepts and actions, *(ii)* the system dynamic behaviour by means of a detailed design of interaction protocols or individual pathways of communication between agents. Protocols capture the timing of communication of related messages between agents. They can be depicted using agent UML (AUML) notation [21].

Apart from the system overview diagram, two interaction protocols were designed. The first one is described in [22] It consists of a basic air-ground negotiation protocol to negotiate arrival trajectories. 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, which is described in [24], involves an air-air negotiation process that illustrates re-planning modifications of an implemented arrival procedure. Mentioned protocols are included within the agent plan library as explained in *section IV*. Moreover, communication processes are carried out through an interchange of messages following the standards of agent communication language [25].

C. Detailed design

In the detailed design phase, the internal agent architecture (agent overview) and internal agent processes were developed. The agent architecture is defined by several capabilities that exchange data by means of inner messages. Processes are derived from interaction protocols. They are implemented within capabilities. Some capabilities are described using new capabilities at a lower level. At the 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 behaviour. 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 chosen implementation platform. Events consist of the arrival of a percept, arrival of a message from another agent or an internal message (event) within the agent. Details of the aircraft agent design are provided next. Following the Prometheus methodology [16], the notation used to capture the agent's design is depicted in Fig. 2.

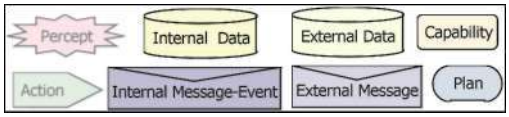

Figure 2. Notation used in agent and capability overview diagrams

III. AICRAFT AGENT DESIGN: COCKPIT CAPABILITIES

Taking into account artefacts mentioned in previous design phases, the aircraft agent architecture was obtained. The agent overview diagram depicted in Fig. 3 shows the aircraft architecture described in this section. In the referred figure, actions, percepts, messages (inner o external) and share data are linked to the following six cockpit capabilities: *(i)* Aircraft Environment Information Management, *(ii)* Alarms Management of Aircraft Systems, *(iii)* Conflict Detection-Resolution, *(iv)* Airborne Contingency Management, *(v)* Trajectory Guidance and *(vi)* Navigation Procedures Management.

Taking into account an on-board automated procedure management, *Navigation Procedure Management* capability is the core of the aircraft capability-based architecture. The rest of them provide environmental and airborne data to manage the procedures. Also, as explained later, this capability consists of several sub-capabilities.

A. Aircraft Environment Information Management

The main goal of this capability consists of maintaining an updated onboard environmental knowledge. Information is obtained from *percepts* of the aircraft sensor systems and from incoming agent messages. Plans of this capability update and store *data* information from: sensor data, weather forecast, restricted areas, air space recourses (e.g. available arrival routes and gateways), surrounding air traffic, etc. Besides, this capability generates specific *contingency events* (represented as inner messages in Fig. 3) when significant environmental changes are detected.

B. Alarms Management of Aircraft Systems

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

C. 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. Also, this capability is used for testing conflicts and provides solutions during planning trajectory processes. Plans that implement several models to detect conflicts are included within a *Conflict Detection Sub-capability*. In the same way, plans to provide initial conflict solutions are included within an *Initial Conflict Solution Sub-capability*.

D. Airborne Contingency Management

This capability deals with deciding procedural tasks according to received *contingency input events* from other capabilities or external agents. The following contingencies inputs have been identified: *(i)* Contingency of environmental

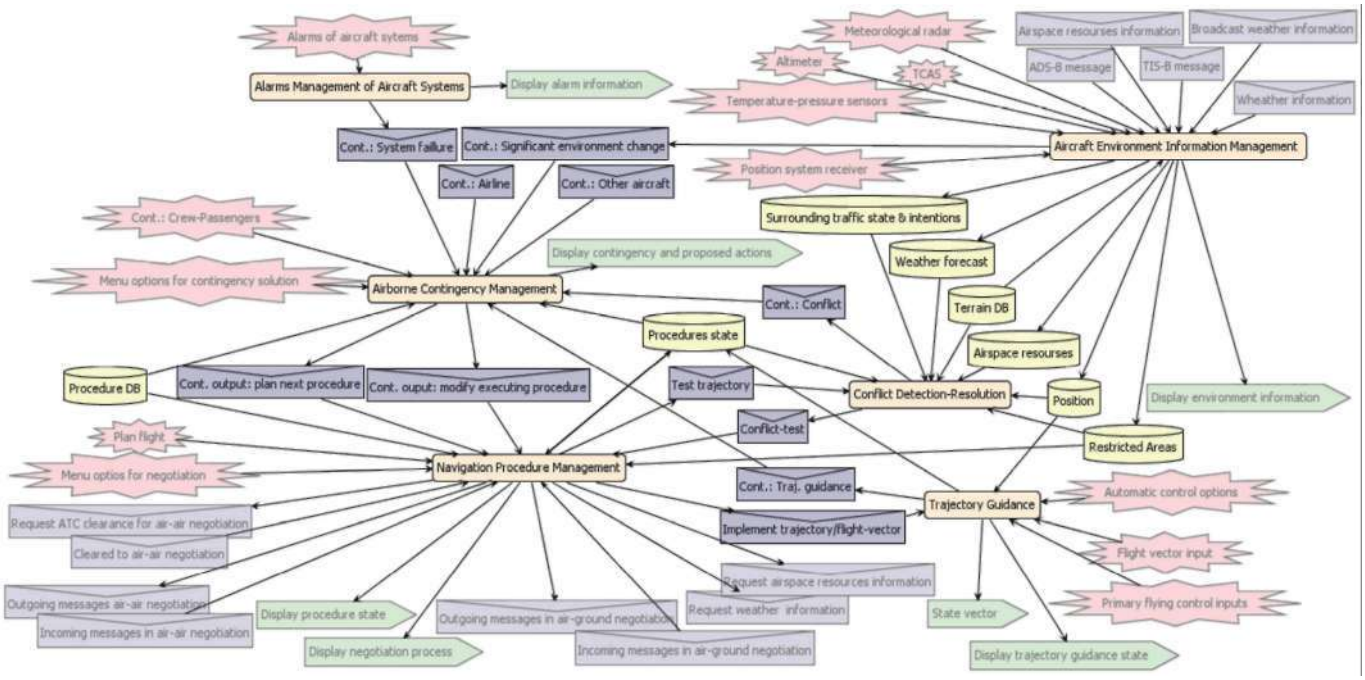

Figure 3. Aircraft agent architecture

significant changes, *(ii)* System failure contingency (that indicates failure details as well as proposed procedure, manoeuvre or actions according to normal, abnormal or emergency procedures), *(iii)* Conflict contingency (that includes information about solutions proposed by the conflict detection-resolution capability), *(iv)* Contingency from other aircraft (i.e. requirements from other aircraft requesting to solve conflicts, to modify arrival sequence, etc.), *(v)* Airline contingency, requesting to modify intended flight plan, *(vi)* Contingency of ATC (e.g. changes regarding previous agreement, and *(vii)* Contingency of crew or passengers defined through an on board options menu.

Advisable procedural tasks are suggested by means of *contingency output events* in order to be considered within the current procedure (e.g. *modify executing procedure* event) or within the next one (e.g. *plan next procedure* event). The contingence solving process can be treated in an automatic manner and therefore this capability will require future efforts to design new 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.

E. Trajectory Guidance

This capability represents a flight control system 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 the current *FMS* and *Flight Director* to perform 4D-trajectory guidance. Also it provides information about the real-time guidance process which is required for other capabilities or agents. Trajectory guidance state is part of the real-time procedure state data that will be described in the next section. Apart from the mentioned information, this capability provides contingence events when trajectory tracking difficulties arise. Besides, actions are represented by the state-vector from aircraft dynamic. Flight

Information about the trajectory to be flown by the aircraft is provided by the implement *trajectory/flight-vector* event that comes from the *Navigation Procedures Management Capability*. Other inputs come from the user interface (e.g. automatic control options which define automation levels to execute trajectories, flight vector input or flight control inputs).

F. Navigation Procedures Management

According to correspondent scenarios and goals, this capability is responsible for managing procedures for planning, implementing or modifying trajectories of each flight phase. The mentioned capability together with the Trajectory Guidance one represents the core of a next-generation of *Navigation Management System* for TBO. Details of this capability are described next.

IV. NAVIGATION PROCEDURES MANAGEMENT CAPABILITY

This capability overview diagrams is depicted in Fig. 4. It consists of four sub-capabilities (*Flight Planning*, *Procedure Executing*, *Next Procedure Planning* and *Current Procedure Re-planning*) as well as a plan to read and display information about the current planned and executing procedure state.

Despite our conceptual model being focused on arrival air traffic operations, the *Flight Planning Capability* has been considered in order to define trigger-events for planning the arrival-approach procedure while the preceding en-route procedure is being executed. Besides, it enables a framework for including new navigation procedures.

A basic Flight Planning capability has been modelled by means of two plans. An initial plan contains tasks leading to an

automatic air-ground negotiation process for a gate-to-gate trajectory. A second plan fills attributes of the initial gate-to-gate procedures list.

The *Procedure Executing Capability* contains two main plans. One of them executes tasks of the current procedure such as: generating events and data to implement trajectories or to trigger the next procedure planning process, storing updated procedure states, etc. The second one updates procedure attributes once a trajectory modification has been negotiated and the corresponding event has been received. A procedure state data has been defined to capture the dynamic behaviour of the procedure management process. This discrete variable, also, facilitates a human-readable automated procedure management. In addition, this information can be exchanged with ground and other aircraft systems in order to improve surveillance and coordination mechanisms in TBO scenarios.

The *Next Procedure Planning Capability* is responsible for selecting the next procedure from the procedure list and planning its linked trajectory. Also it provides data about the real-time planning process. These data are jointly stored with the current procedure state. Figure 4 shows the main inputs and outputs of this capability. In Fig. 5 the referred inputs and outputs are associated to their corresponding plans. As it was explained, plans of this figure represent the bottom level of the mentioned sub-capability design. According to Fig. 5 selecting a next procedure is performed by a plan (named *Select Next Procedure and Start Planning*) procedure. This plan is triggered by events coming from the *Procedure Executing Capability* or from the *Contingency Management Capability*. Besides, it generates events that trigger new specific plans for planning or updating attributes of different procedures. Also, in Fig. 5 data, events and messages used/produced by the *Arrival-Approach Planning* plan are depicted. This plan implements the air-ground trajectory negotiation protocol described in [22]. Therefore incoming and outgoing communication messages refer to the mentioned negotiation process.

The *Current Procedure Re-planning Capability* (see Fig. 4) contains a plans library to modify in several ways the current trajectory attributes. Although inner design of this capability is not depicted, it presents a similar design scheme in respect to the previous one. When the mentioned capability is being executed, a plan is selected according to the information provided by the event that comes from the contingence management capability. We have focused our implementation work 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 the current arrival aircraft sequence has been implemented in previous work [23].

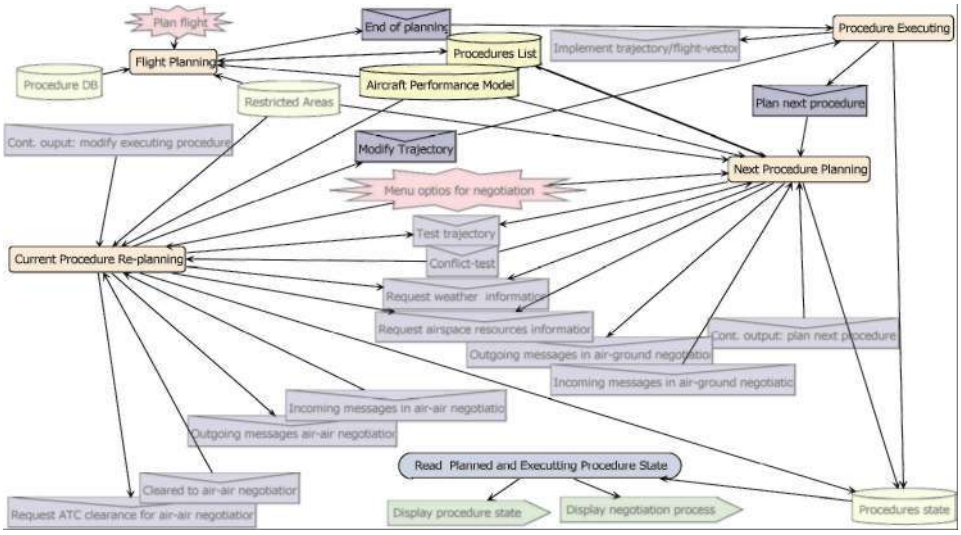

Figure 4. Navigation Procedures Management Capability

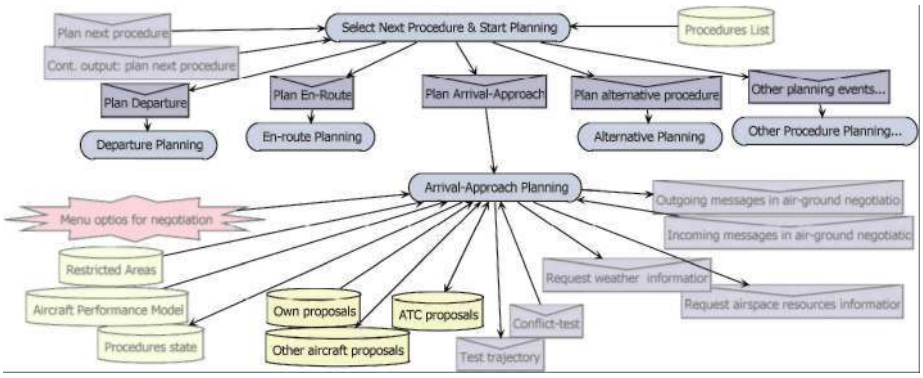

Figure 5. Next Procedure Planning Capability

V. FROM AGENT CAPABILITIES TO COCKPIT SYSTEMS

Aircraft agent capabilities enable a direct correspondence with cockpit functionalities for future TBOs. Figure 7 shows the cockpit system architecture obtained from the previous capability-based architecture.

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

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

CMS maps the *Contingence Management* capability and therefore uses the mentioned events and other ones (e.g. incoming messages, percepts from pilot interface) to provide decisions events. Consequently, CMS represents a first layer decision making 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, the 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 the current FMS.

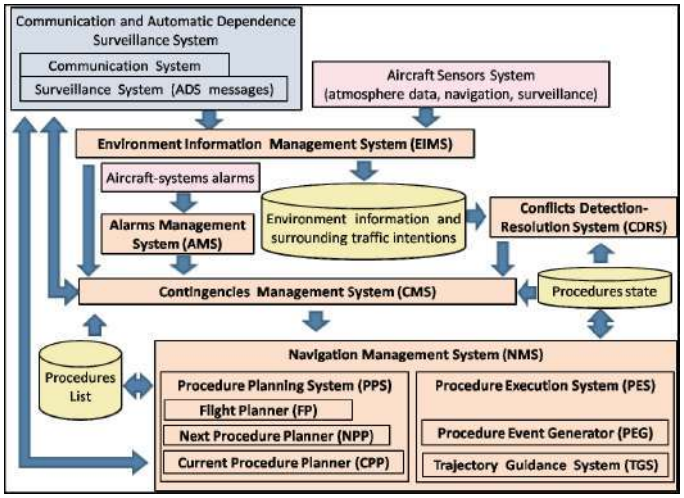

Figure 7. Cockpit functionalities

VI. IMPLEMENTATION ISSUES

Taking into account the issues pointed out in the introduction section, several aspects and directions for future works should be consider from the described conceptual model. We focus this section exclusively in the implementation topic, due to its importance into the life-cycle of the executable model and the validation process.

The descriptors of final artefacts of our conceptual model provide details enough to an easy translation into an executable model by means of the current agent development tools.

Moreover, Prometheus methodology provides a full life-cycle support tool to develop multi-agent systems: the Prometheus Development Tool (PDT). Current version of PDT provides support for: *(i)* designing most the design artefacts within the Prometheus methodology, *(ii)* cross-checking for consistency and completeness for the conceptual model, *(iii)* automatic generation of skeleton code in JACK agent-oriented programming language [19].

The proposed conceptual model is at implementation phase. Although facilities of automatic code generation of PDT, we opted for the Java Agent Development Framework (JADE) Platform [17] due to: *(i)* it is one of most extended multi-agent platforms and, *(ii)* it provides (conform to FIPA standards [25]) infrastructure for inter-agent communications and for managing software agents distributed across multiples hosts.

Architecture of JADE agent is built upon the behaviour concept rather than a plans-based architecture. Therefore, plans are included within several JADE behaviours.

On the other hand, continuous simulation requires, in nature, implementing the aircraft dynamic in order to represents model behaviour 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, the proposed conceptual model enables discrete event simulation. In this case, events can be generated by random functions implemented within capabilities plans representing underplaying models as *black boxes*. In this way, for an initial implementation phase, random functions to generate events are implemented within described agent plans. Then, the executable model will improve its performance when functions are replaced by specific underplaying models as they are developing.

VII. CONCLUSIONS

In this paper we have presented a cockpit system architecture for TBO. The proposed design illustrates a practical application of previous work developed by the authors aimed at achieving an integrated vision of automated procedures and corresponding system prototypes for TBO.

For this purpose a conceptual model that represents TBO scenarios as a multi-agent system was developed. A practical and formal methodological approach has been used to analyze and to design the mentioned scenarios in a structured and

consistent manner. By means of an iterative top-down modelling process the detailed agent architecture was designed based on capabilities, internal events, plans and data structures.

The airborne view point was described in this paper through a detailed aircraft agent design. This architecture is oriented to execute several processes in order to plan, execute or modify trajectories in a coordinated way.

The modularity of the internal architecture of the aircraft agent around the concept of capability provides a direct correspondence between them and the on-board systems to manage their respective procedures. Procedure Management Capability together with Trajectory Guidance and Contingency Capability are the core around which a future Navigation Management Systems (NMS) for TBO could be developed. Thus, NMS is described as an on-board 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.

Directions for future work include extending the described conceptual model to gate-to-gate operations, as well as obtaining an executable model for analytical simulation according to the described requirements. Furthermore, once a first version of a simulation platform has been implemented and validated, new underlying models and functionalities could be included to be analyzed as they are designed.

VIII. REFERENCES

[1] E. Courses, T. Surveys, "4-Dimensional Trajectories and Automation Connotations and Lessons learned from past research, Integrated Communications, Navigation" and Surveillance Conference, ICNS'07, pp. 1-10, 2007, Bethesda,

[2] P. Brooker, "Airborne Separation Assurance Systems: towards a work programme to prove safety", Safety Science, 2004,vol. 42, pp. 723–754.

[3] A.L. Alexander, , C.D. Wickens, D.H. Merwin, "Perspective and coplanar cockpit displays of traffic information: Implications for maneuver choice, flight safety, and mental workload", The International Journal of Aviation Psychology, 2005,vol. 15págs. 1–21.

[4] B. Korn, A. Kuenz, "4D FMS for Increasing Efficiency of TMA Operations", IEEE/AIAA 25th Digital Avionics Systems Conference, DASC'06, 2006,pp. 1–8.

[5] J. Lopez-Leones, 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.

[6] K.M. Feigh, A.R. Pritchett, A.P. Shah, S.A. Kalaver, A. Jadhav, D.M. Holl, A.Z. Gilgur, , "Analyzing Air Traffic Management Systems Using Agent-based Modeling and Simulation", Georgia Institute of Technology, 2005, Baltimore, MD.

[7] T.J. Callantine, J. Homola, J. Mercer, T. Prevot, "Concept Investigation via Air-Ground Simulation with Embedded Agents, AIAA Modeling and Simulation Technologies Conference and Exhibit", 2006, Keystone, Colorado.

[8] G. Satapathy, V. Manikonda, "Agent Infrastructures for Modeling and Simulation of CNS in the NAS", 2004, Fairfax, VA.

[9] M. Nguyen-Duc, J.P. Briot, A. Drogoul, "An application of multi-agent coordination techniques in air traffic management", IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), 2003, pp. 622-625.

[10] Vladimir Gorodetsky, Oleg Karsaev, Vladimir Samoylov, Victor Skormin, "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.

[11] Y. Ding, J. Rong, J. Valasek, "Automation Capabilities Analysis Methodology for Non-Controlled Airports", AIAA Modeling and Simulation Technologies Conference and Exhibit, 2003, Austin, Texas.

[12] A. RTO 70, "ATM Modeling and Simulation Architecure Study", Tecnical Research in Advanced Air Tarnsportations Concepts Technologies (AATT) Project, NASA Ames Research Centre, 2001.

[13] L. Quon, "Modeling and Simulation Needs for Next Generation Air . Transportation System Research," AIAA Modeling and Simulation Technologies Conference and Exhibit, 2006, págs. 21–24.

[14] W. Anne, S. Mavo, y James P. McGee, "ATM Human Behavior Modeling Approach Study," AATT Project, 2001

[15] J. Sudeikat, L. Braubach, A. Pokahr, W. Lamersdorf, "Evaluation of agent-oriented software methodologies–examination of the gap between modeling and platform", Agent Oriented Software Engineering (AOSE), 2004, Springer.

[16] L. Padgham, M. Winikoff , "Developing Intelligent Agent Systems: A Practical Guide", John Wiley & Sons, 2004

[17] Fabio Luigi, Giovanni Caire, Dominic Greenwood "Developing Multi-Agent Systems with JADE", Wiley Series in Agent Technology, Hardcover, 2007.

[18] A. Pokahr, L. Braubach, y W. Lamersdorf, "Jadex: A BDI Reasoning Engine, Multi-agent System Artificial Societies and Simulated Organizations", vol. 15 2005.

[19] M. Winikoff, "JACK TM intelligent agents: An industrial strength platform", Bordini et al. , pp. 175–193.

[20] John A. Sorensen, "Detailed Description for CE-11 Terminal Arrival: Self Spacing for Merging and In-trail Separation", NASA Ames Research Center and NASA Langley Research Center, 2000, Moffett Field, CA and Hampton, VA.

[21] M.P. Huget, "Agent uml notation for multiagent system design", IEEE Internet Computing, 2004, vol. 8, pp. 63–71.

[22] J. Canino-Rodríguez, L. Gómez-Deniz, J. García-Herrero, J. Besada-Portas, J. Casar-Corredera, "A 4D trajectory negotiation protocol for Arrival and Approach sequencing", Integrated Communications, Navigation and Surveillance Conference, 2008. ICNS 2008, pp. 1-12.

[23] J.M Canino, L. Gómez, J. García, J. Besada, J.R. Casar, "An Agent Oriented Anaysis and Modeling of Airborne Capabilities for Trajectory Based Operations", In press Digital Avionic System Conference, DASC09, 2009, Orlando, FL.

[24] J. Canino, L. Gomez, J. Garcia, J. Besada, J. Casar, "Design of an air-air negotiation protocol to reorder aircraft arrivals sequence", Digital Avionics Systems Conference, DASC 2008. IEEE/AIAA 27th, 2008, pp. 3.C.6-1-3.C.6-11.

[25] Foundation for Intelligent Physical Agents. 24 Communicative Act Library Specification, Version J. http://www.24.org/specs/2400037/.