

A FLIGHT DECK DECISION SUPPORT TOOL FOR AUTONOMOUS AIRBORNE OPERATIONS

Mark G. Ballin*, Vivek Sharma†, Robert A. Vivona‡,
Edward J. Johnson§, and Ermin Ramiscal¶

NASA is developing a flight deck decision support tool to support research into autonomous operations in a future distributed air/ground traffic management environment. This interactive real-time decision aid, referred to as the Autonomous Operations Planner (AOP), will enable the flight crew to plan autonomously in the presence of dense traffic and complex flight management constraints. In assisting the flight crew, the AOP accounts for traffic flow management and airspace constraints, schedule requirements, weather hazards, aircraft operational limits, and crew or airline flight-planning goals. This paper describes the AOP and presents an overview of functional and implementation design considerations required for its development. Required AOP functionality is described, its application in autonomous operations research is discussed, and a prototype software architecture for the AOP is presented.

Introduction

IN 1995, the RTCA and others released concepts for the “free flight” operational paradigm, which reduces reliance on centralized air traffic management. Free flight is defined as a safe and efficient flight operating capability under instrument flight rules in which operators have the freedom to select their path and speed in real time.¹ Proponents of free flight believe the concept will enable future growth of the National Airspace System (NAS) in a manner that scales directly with increases in air traffic, mitigates the need for costly expansions of ground-based infrastructure, and provides operational benefits to users of the system.

While the general concept of autonomous aircraft operations, under instrument flight rules, has been studied since 1965,² no single concept of operations has emerged as the best solution. Some concepts set aside specific airspace in which aircraft would be permitted to manage their operations autonomously,^{3–5} while other concepts propose a mixed environment of autonomous and managed aircraft.⁶ Further, much research and development remains to be performed to establish proof of feasibility and economic viability for any of the proposed concepts. The complex issues of the transfer of authority and responsibility between centralized ground controllers and airborne participants, the exchange of data between participants, and system stability and safety must be resolved. Activities to date have focused primarily on the chal-

lenges of airborne separation assurance and, hence, airborne conflict management (ACM). Most ACM concepts identify the need to provide a cockpit display of traffic information (CDTI), and a system of traffic conflict detection and resolution (CD&R) to participating flight crews.^{7,8} Therefore, much effort has been expended on the development of CD&R algorithms and on CDTI approaches.^{9–12} While this research is useful, it alone cannot address concept feasibility, and it does not address the need for improved traffic management, which must increase NAS throughput and accommodate airspace user preferences in addition to assuring separation.

In collaboration with industry and the international R&D community, the National Aeronautics and Space Administration (NASA) is maturing the Distributed Air/Ground Traffic Management (DAG-TM) concept as part of its Advanced Air Transportation Technologies (AATT) Project. To support DAG-TM research, NASA is developing an interactive flight deck decision support tool, referred to as the Autonomous Operations Planner (AOP). This paper describes the need for the AOP, and presents an overview of functional and implementation design considerations required for its development. AOP functionality is outlined, its application to autonomous operations research is discussed, its software design is described, and a research and development plan is presented for the AOP.

Need for an Airborne Decision Support Tool

Although CDTI and ACM concepts may alone provide benefits, an increase in flight deck planning capabilities may be necessary for concepts that significantly increase the airborne role in traffic management decision-making. While research suggests that a tactical system meets the needs of safe separation assurance,¹³ other research results indicate these airborne capabilities should have a strategic planning compo-

*Associate Fellow, AIAA, NASA Langley Research Center, Hampton, VA

†Senior Member, AIAA, Titan Systems Corporation, Hampton, VA

‡Senior Member, AIAA, Titan Systems Corporation, Billerica, MA

§Senior Member, AIAA, NASA Langley Research Center, Hampton, VA

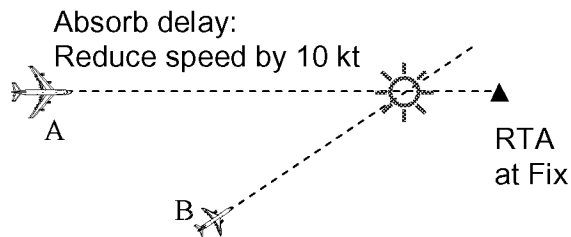
¶Senior Member, AIAA, NASA Langley Research Center, Hampton, VA

nent.^{14–16} For concepts that involve airborne integration with ground-based air traffic service providers (ATSP) who manage traffic flow, look-ahead horizons of more than 15 minutes may be beneficial, whereas tactical concepts may only involve horizons of five minutes or less. A flight-deck-based crew decision support research system is needed that supports the investigation of these and other issues in order to develop feasible and economically viable concepts for distributed air/ground traffic management.¹⁷

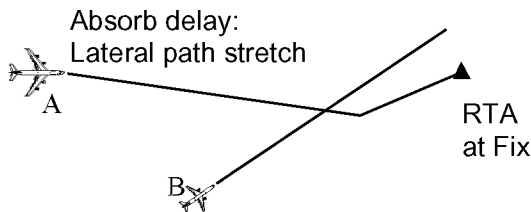
The following examples illustrate some potential benefits of using a flight crew decision aid that provides increased sophistication in its planning assistance.

Figure 1 illustrates the potential advantages of a planner that accounts for a global traffic situation. In Figure 1(a), there is no consideration of the traffic situation, either because of an insufficient look-ahead time or because of an unsophisticated ACM system. Aircraft A must absorb delay to meet a required time of arrival (RTA) at the fix, and chooses to reduce speed. By doing this, Aircraft A inadvertently causes a conflict with Aircraft B. This creates additional workload and a potential for controller intervention. In Figure 1(b), Aircraft A chooses a lateral path stretch as the best solution, thereby achieving efficient compliance with the ground-supplied restriction while minimally impacting other aircraft and controllers.

In the second example, illustrated in Figure 2, Aircraft A through E have estimated times of arrival (ETA) at a fix that are incompatible with flow manage-

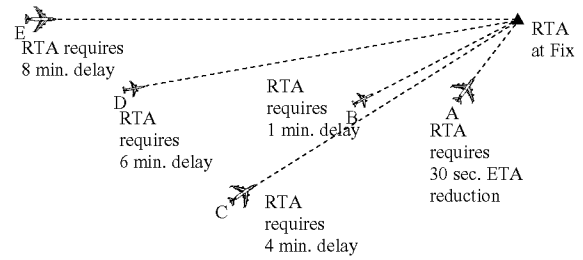


a) Constraints management without consideration of traffic.

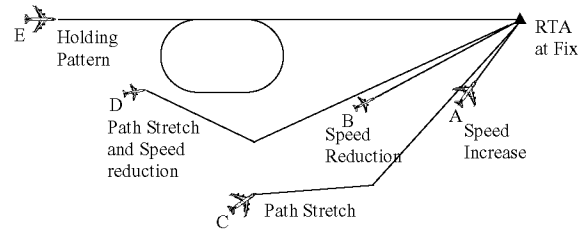


b) Constraints management with consideration of traffic.

Fig. 1 Potential benefits of including situational planning in conflict and constraints management.



a) ATSP-provided RTAs require ETA adjustments.



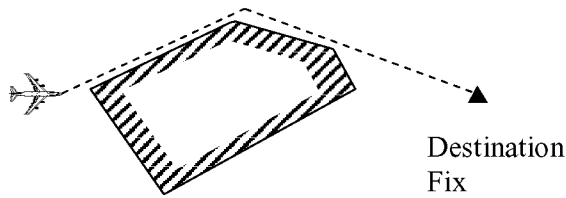
b) Airborne decision aid determines appropriate strategy for each aircraft.

Fig. 2 Airborne decision aid must choose appropriate strategy for meeting constraints.

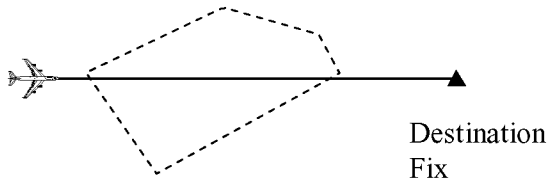
ment planning. A ground-based scheduler has resolved arrival capacity issues by assigning each aircraft an RTA for crossing the fix (Figure 2(a)). Aircraft A must increase speed so its ETA matches its assigned RTA, while Aircraft B through E must absorb increasing amounts of delay. For each aircraft, a flight deck planning system identifies a conflict with the assigned constraint, and determines the most appropriate strategy (Figure 2(b)).

Figure 3 illustrates the possible benefits of a flight deck system that combines flight management planning with goals or preferences of the flight crew. The crew needs the earliest possible arrival over the destination fix. In Figure 3(a), the flight plan accounts for dynamic special use airspace (SUA), a region of airspace that is temporarily inaccessible. In Figure 3(b), the airborne planning aid has detected the removal of the airspace constraint, and either through direct crew input of its goals or through inference, has determined that an early arrival is the crew's goal. It provides the new trajectory as an advisory to the crew, who accept it.

These examples suggest that flight deck automation to support crew planning may be useful. As currently defined by the International Civil Aviation Organization (ICAO), a conflict is “a predicted converging of aircraft in space and time, which constitutes a violation of a given set of separation minima”.⁵ A crew decision support system as described above expands upon this definition: rather than being limited to traffic, the system identifies and resolves conflicts with



a) Crew needs earliest arrival possible; plan accounts for dynamic SUA.



b) Airborne decision aid detects removal of SUA and re-plans for earliest arrival.

Fig. 3 Airborne decision aid may require knowledge of crew or company planning goals.

all hazards, aircraft performance limits, externally-imposed constraints, and crew or company preferences and goals. Such a system is envisioned to contain several characteristics:

1. Because DAG-TM and most other distributed traffic management concepts are human-centered, the system must provide advisory information to the flight crew. It must be designed to enable the crew to execute a recommended action in a way that minimally impacts crew workload. Because the crew will have final authority in all flight-management decisions, the system may need to provide interactive tools to assist the crew in generating alternative solutions. It may also need input or derived information about mission goals to incorporate operator preferences in its advisories.
2. Distributed traffic management concepts may only be viable if they are capable of providing benefits in dense and highly constrained traffic environments.¹⁷ Such environments are common today and are expected to remain common in the future. Therefore, the cockpit decision aid must be able to account for all known constraints while determining the appropriate trajectory for the aircraft. Constraints include nearby traffic, special use airspace, weather and other environmental hazards, and arrival time, speed, or altitude constraints imposed by the ATSP for safety or to expedite traffic flow.

3. Tactical conflict resolution has been defined as a maneuver that resolves a conflict, but does not account for the own aircrafts flight plan in doing so. Strategic resolutions have been defined as those that, if executed, include recovery of the intended flight plan.¹⁸ Both types of resolution have advantages and disadvantages. To continue research on the characteristics and desirability of strategic and tactical ACM and the relationships between them, a research decision support system must provide independent strategic and tactical ACM functions. It must also provide a flexible and re-configurable alerting and advisory system that integrates the two approaches in a way that conveys the appropriate information to the crew at the appropriate time. It may also be required to provide advisories that include trajectory optimization or desired operational procedures, such as flight-idle descents, since future commercial systems, must ultimately be economically viable for the airspace user.

4. To support investigations of the safety aspects of system functions, operating independently or in combination, the system must be designed to maintain independence between these functions, and it may need to provide redundancy.
5. The system must also support research that defines airborne and ground-based communication, navigation, and surveillance (CNS) infrastructure requirements. The system must therefore be designed to deal with a variety of data types, formats, and data link characteristics. The system must be robust to missing data and account for redundant data provided from several sources. It must be flexible enough to allow for exploration of varying data exchange content, update rates, and availability requirements. Research may indicate a need to reconstruct missing information from a data linked source or infer the intent of proximate traffic. Monitoring of an intruders conformance with its provided intent may also be needed.
6. Although it is designed for research, the system must consider integration with existing flight deck systems and flight deck procedures to the extent possible. It should consider current guidelines and standards that the aviation industry is developing for the future, such as ARINC 660A.¹⁹
7. The research system must be highly flexible so that currently proposed and future functional components may be incorporated, as they become available. It must have a well-designed functional architecture that utilizes stable and documented interfaces. This facilitates the investigation of differing approaches to a given function without requiring extensive system modification. It must

be capable of operation in several research environments, from air traffic simulation to flight. Finally, it must be capable of supporting several research approaches, from human-in-the-loop investigations of system utility and usability to batch-mode verification and validation analysis.

The Autonomous Operations Planner

The AOP is a flight deck-based decision support research system that is currently under development to meet the above needs. It assists a flight crew in mission planning and execution, as needed for future civil operations under the DAG-TM paradigm. The crew interacts with the AOP to perform trajectory planning that accounts for conflicts with (1) traffic hazards; (2) ownship aircraft performance limitations; (3) system flow constraints that are imposed by the ATSP; (4) airspace constraints, such as severe weather and dynamic SUA; and (5) operator flight goals, such as efficiency and schedule.

The AOP assists crew decision-making through (1) presentation of the most relevant information; (2) the accommodation of crew planning preferences; (3) alerting advisories; and (4) automated negotiations with crews of other aircraft (if necessary) and ground-based air traffic controllers. It analyzes surveillance and constraint data for potential airspace or traffic conflicts; it calculates conflict resolution options that optimize parameters specified by the flight-crew; it provides tactical information for conflict-free maneuvering; and it analyzes over-constrained problems for viable solutions. It will manage information received by the flight deck from several sources that are expected to exist in future operational environments. These sources include the direct broadcast of position and intent information from other aircraft in proximity, ground-based traffic information services, and ground-based flight information services. The AOP will fuse these data and manage any incomplete, redundant, or ambiguous information. Figure 4 is a block diagram of AOP functionality expected at its completion.

Research Applications

To facilitate autonomous flight management research, the AOP is designed for integration into three specific research environments:

1. A concept-level distributed traffic simulation environment will be used for operational feasibility assessments, system-level requirements definition, airborne and CNS technology requirements determination, and human-centered design and assessment.²⁰ A workstation-based aircraft simulation, referred to as the Aircraft Simulation for Traffic Operations Research (ASTOR), is designed to host the AOP, an enhanced-capability software FMS, and airborne CNS systems.

2. High-fidelity flight deck simulators that make up the Langley Research Center Cockpit Motion Facility (CMF) will be used for investigations that require a full crew simulation in a high-fidelity cockpit environment. These simulators utilize a real-time simulation environment, so the AOP is designed to accommodate normal simulation operating modes, including initialize, trim, hold, operate, and reset.
3. The NASA Airborne Research Integrated Experiments System (ARIES), a specially instrumented Boeing 757, will be used to support in-flight validation activities.

The AOP is also designed to support both human-in-the-loop (HITL) and batch modes of operation. HITL is defined as a mode in which subject pilots, subject controllers, and/or subject dispatchers interact with systems or simulators that will allow them to make flight- or traffic-management decisions, as they would in a deployed system. HITL will be used for human factors research and AOP crew interface development. Batch is a mode in which all system operations are simulated with full repeatability using automated human operator models to represent humans in the simulated system. Batch mode will enable simulations with statistically significant sample sizes.

Functional Design

The AOP is designed around a set of core functions that are central to ACM applications: mission planning, data management, interfacing with other on-board systems, and interacting with the flight crew.

The core AOP responsibilities are: (1) generating and managing ownship and traffic trajectories; (2) collecting and maintaining data on area hazards that are comprised of hazardous weather, SUA, or other unusable air space; (3) probing the ownship trajectories for conflicts with the nearby traffic trajectories, area hazards or constraints; (4) detecting and resolving conflicts with hazards and constraints as well as the strategic planning goals and preferences; (5) outputting data to be displayed to the crew, in the form of alerts, advisories, and flight management options; and (6) responding to crew inputs. In order to achieve this functionality, the AOP functional design must consider the ownship aircraft, traffic and area-hazards in the surrounding airspace, conflict detection and resolution mechanisms, output integration, and external interfaces.

Ownship

The AOP uses the ownship state and a representation of the ownship intent in order to create a set of trajectories. The AOP receives the ownship state information from the FMS, and the intent information from the flight plan buffers in the FMS and the current mode control panel (MCP) settings. Once the

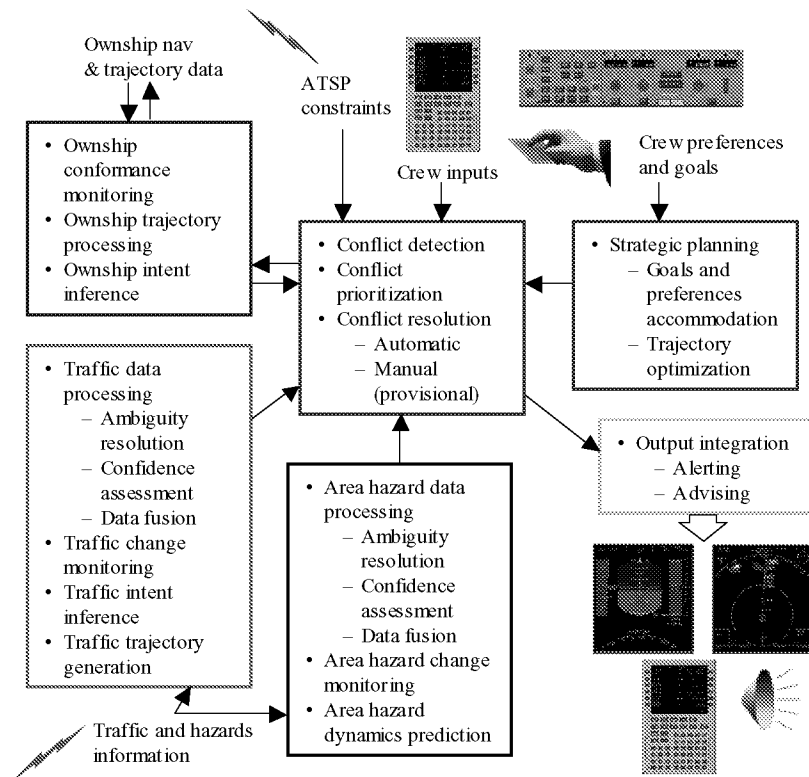


Fig. 4 Autonomous Operations Planner functions upon completion.

ownship state and intent information is available, the AOP checks whether the ownship state is in conformance with its intent.

Based on the available information, i.e., ownship state, intent and conformance, the AOP creates following trajectories.

1. The AOP creates a **state-projection trajectory** by projecting the aircrafts current state parameters forward in time, along the linear velocity vector, ignoring turn rates. This trajectory is used for tactical conflict detection (e.g. blunder protection and tactical maneuvering). There is always one, and only one, state-projection trajectory for the ownship aircraft.
2. The AOP creates a **commanded trajectory** by evaluating all available command information from the active FMS route, the MCP, and autopilot / autothrottle modes. This trajectory represents what the aircraft will do if the crew makes no changes to the automation settings in the cockpit. There is always a commanded trajectory for the ownship aircraft when an autoflight system is engaged.
3. The AOP creates a **planning trajectory** by evaluating all available “known intent”²¹ information from the AOP, the MCP, and the FMS.
4. The AOP may create a **provisional trajectory** from either a non-active route or by modifying some element of the active route. Each provisional trajectory represents a potential change to the aircrafts current known intent that needs to be evaluated either by the crew or by the internal AOP automation. Provisional trajectories are used in conflict resolution, provisional planning, self-spacing operations, and internal AOP functions. Several provisional trajectories may exist.
5. The AOP has a provision to create an **inferred-intent trajectory**, if research determines that intent-inferencing is necessary. This trajectory uses known intent information and additional forms of potential intent information (e.g. a return to the planned route after traversing a

weather-impacted region). This trajectory represents the AOPs “best guess” at ownship intent when the aircraft is not maneuvering consistently with a “known intent”. There is an inferred intent trajectory for the ownship aircraft only if the aircraft is not conforming to the known intent, and if an intent inferencing function is available.

Traffic Hazards

The AOP uses the traffic state and, if available, traffic intent information in order to create four-dimensional trajectories for the nearby aircraft. In the current implementation, the AOP may receive this information via either the Automatic Dependent Surveillance-Broadcast (ADS-B) or the Traffic Information Service (TIS) interface. The AOP fuses these data and generates appropriate four-dimensional trajectories depending on the information broadcast by the traffic aircraft. When only state information is available for a traffic aircraft, the AOP creates a state-projection trajectory. When intent information, e.g., a set of trajectory change points (TCPs) and target state information, is received, the AOP creates additional intent-based traffic trajectories, namely the TCP-based trajectory and/or the target-state trajectory. For every traffic aircraft, the AOP checks whether the traffic state is in conformance with its broadcast intent. If the state for a traffic aircraft is not in conformance with its broadcast intent, the AOP creates an inferred-intent trajectory, representing the inferred path of the traffic aircraft over a defined “look-ahead” period.

Area Hazards

The AOP needs accurate data on the boundaries of area hazards in space and time in order to determine if any of the ownship trajectories conflict with the hazard within a defined “look-ahead” period. The AOP can detect conflicts with different types of area hazard models. Expected area hazards to be detected include adverse weather, special use airspace, turbulence, volcanic ash, and terrain. The AOP design allows for several levels of intensity and dynamics for area hazards as well as several methods of modeling, including N-sided polygon, grid-based, and centroid-vector approaches. The three core types of models in the AOP are static, time-variant static, and dynamic.

A **static area hazard** is an area hazard whose properties (such as position, geometry, and altitude bounds) are constant with respect to time. Static area hazards include terrain, low altitude man-made obstacles, and restricted airspace. A model for this type of area hazard includes the number of vertices, the latitude and longitude of each vertex, and the minimum and maximum altitudes of the area hazard.

A **time-variant static area hazard** is a static area hazard whose requirement for avoidance varies with time. Two examples of such an area hazard are a dy-

namically activated special use airspace (SUA), and a sector impacted by congestion. A model for this type of area hazard includes, in addition to the static model, the activation schedule.

A **dynamic area hazard** is an area hazard whose position (and potentially the shape) is variable with respect to time. Examples include turbulence, thunderstorms, wind shear, heavy precipitation, hail, icing, volcanic ash, airborne hazardous materials, and fronts. A model for this type of area hazard could include, in addition to the static model, a velocity vector associated with each vertex.

Hazards Avoidance

The AOP independently uses both the state-based and the intent-based algorithms for conflict detection and resolution with respect to the traffic and area hazards. In the event that the constraints need to be relaxed to achieve a resolution, the AOP utilizes a hazards prioritization function.

State-based Conflict Detection and Resolution

The AOP probes the ownship state-projection trajectory for conflicts with the traffic state trajectories and the area hazard boundaries to determine state-based conflicts. Currently, the AOP uses the National Aerospace Laboratory of the Netherlands (NLR) strategy to determine state-based conflicts, and to generate conflict prevention bands that appear on the crews cockpit displays.²² Conflict prevention bands indicate no fly zones in terms of the aircraft heading, speed, and vertical rate.

Intent-based Conflict Detection and Resolution

The AOP probes all available forms of the ownship intent-trajectory for conflicts with the available traffic intent information and the area hazard boundaries to determine intent-based conflicts. Currently, the AOP uses a protected zone conflict detection strategy and a genetic algorithm approach for the conflict resolution strategy.^{23,24}

Output Integration

The goals of the output integration function of the AOP are: (1) to integrate and format the ownship, hazards, and conflict data; (2) to determine the information to be sent to the crew-interface; (3) to associate an alerting level to each resolution advisory for the detected conflicts; and (4) to determine appropriate time to display and purge an advisory. The data available to the output integration function includes ownship trajectories, traffic trajectories, area hazard boundaries, number and type of conflicts, conflict prevention bands, conflict resolution advisories, and other mission planning information. The current AOP design provides up to five internal alert-levels. A Level-0 alert means that the conflict will not occur if the two aircraft continue to follow their strategic plans, but the aircraft

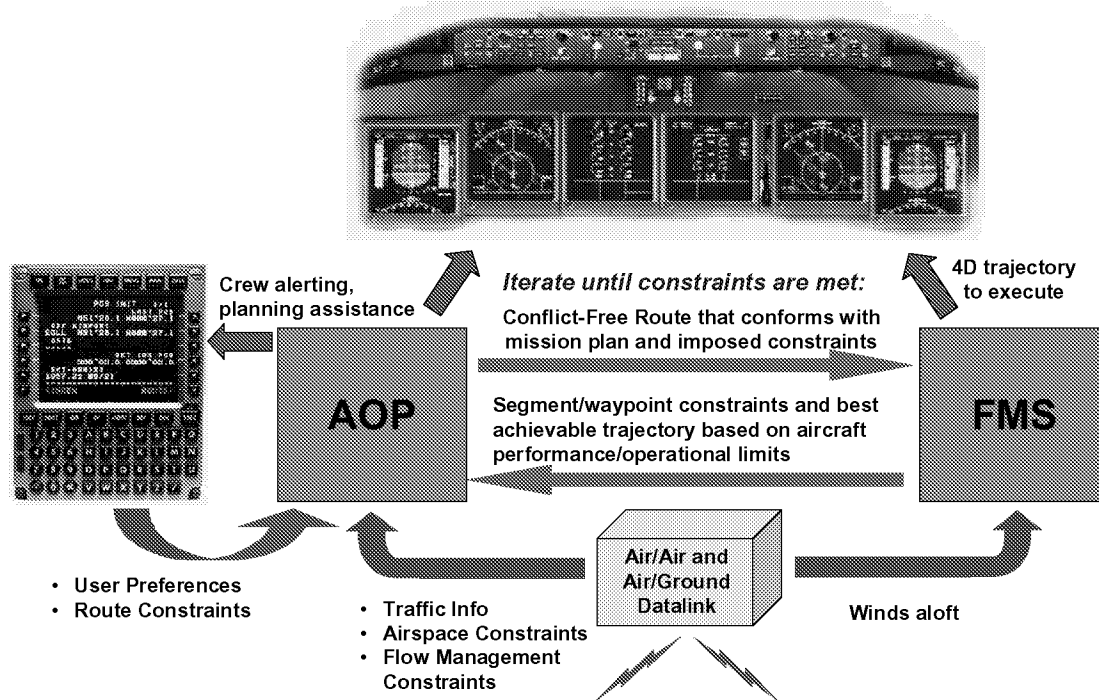


Fig. 5 Integration of the AOP with the FMS and flight deck systems.

are sufficiently close so that a tactical maneuver by the ownship or the traffic aircraft could cause a conflict. A Level-1 alert means that a conflict will occur if neither aircraft maneuver or change their strategic plans, but the traffic aircraft is required to resolve that conflict. The Level-0 and the Level-1 alerts are considered “point out” alerts because they do not require any action of the ownship crew. A Level-2 alert differs from a Level-1 alert in that the ownship needs to resolve the conflict. A Level-2 alert becomes a Level-3 alert when the time to loss of separation has fallen below a specified number of minutes. When separation is lost, a Level-4 alert appears. Researchers can combine AOP alerts and/or specify the format and actions associated with alerts.

External Interfaces

The AOP interfaces with a number of aircraft systems, or simulated aircraft systems, to receive input data from and to send output results to. All communications between the AOP and the aircraft data buses are based on modifications to existing aircraft standards.

The AOP obtains the UTC time from the global positioning system (GPS) to synchronize its internal clock with the aircraft clock. It uses the ADS-B traffic data, and may use TIS data to determine state and intent information about the other aircraft. The AOP uses enhanced Flight Information Services (FIS)-like data to determine the position, shape and dynamics of the area hazards. It sends its results to be displayed

on the navigation display (ND), and the primary flight display (PFD). The AOP currently receives the crew inputs from a simulated multi-purpose control and display unit (MCDU). The AOP uses the flight plan information in the FMS, and the settings on the MCP, to build ownship intent.

The AOP interfaces with the aircraft FMS to obtain information specific to the equipped aircraft. Specifically, the AOP utilizes the trajectory integration capabilities of the FMS to generate ownship trajectories that are consistent with the FMS’s aircraft performance model, navigation database, crew preferences, cost function data, and constraint data. These trajectories are probed against hazards, external constraints, and crew goals and preferences, to detect potential conflicts. If the AOP detects conflicts, it uses the FMS again to generate four-dimensional trajectories during the iterative conflict resolution process until all conflicts are resolved. After finding a solution, the AOP uploads the conflict-free trajectory to one of the FMS flight plan buffers for the crews evaluation. By taking advantage of the FMS’s capability to predict accurate trajectories, based on ownship performance and company- or crew-specified constraints, the AOP generates trajectory change advisories that the aircraft is capable of flying. This approach also enables the AOP to be designed as a generic system with applications to a wide range of aircraft types.

Figure 5 provides an overview of AOP integration with other flight deck systems, the FMS, and onboard data links. In this figure, an advanced FMS is assumed

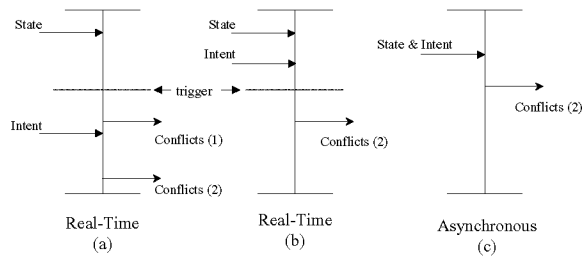


Fig. 6 Comparison of real-time and asynchronous processing of ownship state and intent data.

that can support an iteration loop with the AOP to generate potential ownship trajectories, in the background, while still performing normal FMS functions.

Batch Mode Design Considerations

To perform large numbers of non-human-in-the-loop simulation runs, AOP has an asynchronous mode that supports batch processing. The goal of asynchronous mode is to create valid, repeatable system performance that can be used to create statistically significant results. A valid system performance means that the processing would produce results that could occur during real-time operations. Repeatability means that for the same inputs, AOP will produce the exact same results each time it is executed. How AOP accomplishes validity and repeatability can be seen by a comparison of real-time and asynchronous mode processing.

In Figure 6, the inputs to and outputs from AOPs conflict detection subsystem are presented for real-time and asynchronous modes of operation. The timeline represents one cycle or frame of AOP processing that starts at the top of the figure. Nominally, the AOP uses real-time or simulation-time frame lengths of 1 second. During this frame, the AOP must complete a set of tasks that include execution of conflict detection algorithms. In the real-time mode, the AOP uses processing triggers for functions that are expected to complete within a frame. This ensures that processing begins with sufficient frame time remaining to complete the tasks during this frame. In the asynchronous mode, processing triggers are not required. The AOP will complete tasks assigned to this processing frame regardless of the actual processing time required. In this mode, the simulation time is not advanced until tasks are completed.

In Figure 6a, an ownship state is the only input data received by the conflict detection subsystem prior to a processing trigger. In this case, the state data are used to determine new conflicts. In case (a), an ownship intent update is received after the first set of conflicts is output from the conflict detection subsystem. This input event triggers a new conflict detection cycle that results in new conflicts (2) data being output during this AOP frame, thereby overwriting the old conflicts (1) data. In Figure 6b, both the state and intent up-

date are received prior to the processing trigger. In this case, only one set of conflicts is released (equivalent to the second set released in case (a)). In the asynchronous mode, Figure 6c, both of these real-time cases are modeled the same way. All inputs that can be processed during this frame are identified and processed at the beginning of the frame, regardless of the actual computational time required. This removes the variability associated with the timing of inputs during real-time modes, and provides repeatability for the asynchronous mode. The results of the asynchronous mode match the results from case (b), proving validity.

In the real-time mode, some calculations may take longer than a single time frame. This is a function of the speed of the processor on which the AOP is executing and the processor loading. Without accounting for these factors, the AOP asynchronous mode could potentially overestimate the real-time performance of the AOP. In batch mode, processing started within a frame is finished within that frame. This causes the conflict resolution (CR) results, for example, to apparently be finished in 1 second of simulation time even if it actually takes 4 seconds of wall-clock time. Batch processing mechanisms are being developed to account for this situation. To simulate a computer requiring 4 seconds to complete CR calculations, the CR results are “held” for three additional frames before being released. The magnitude of this “lag” is an adjustable parameter. This enables investigations into required processor performance, crew acceptability of system performance, and retrofit of AOP functionality into existing flight deck computers.

AOP Software Design

The AOP is implemented as a collection of subsystems that communicate with each other and with the external systems through a common framework utilizing mutually agreed protocols. Each of the independent AOP subsystems operates concurrently and asynchronously with the other subsystems. A multi-threaded architecture is used so that subsystems may be assigned to individual execution threads.

The AOP is an event-driven system, rather than a periodic, schedule-driven system. It will, however, automatically refresh trajectories if too much time has elapsed or trajectories have become invalid. These are both safety measures to ensure that the AOP maintains up-to-date trajectories and discards information that is no longer relevant.

A primary purpose of the AOP is to perform wide-ranging research investigations in a simulation environment. This requires the software to be extensible, flexible, and easily maintained. In order to provide the researcher with maximum flexibility, the software must be easily re-configurable at run-time. Since the AOP is being designed to operate in three different environments, all of which have different requirements,

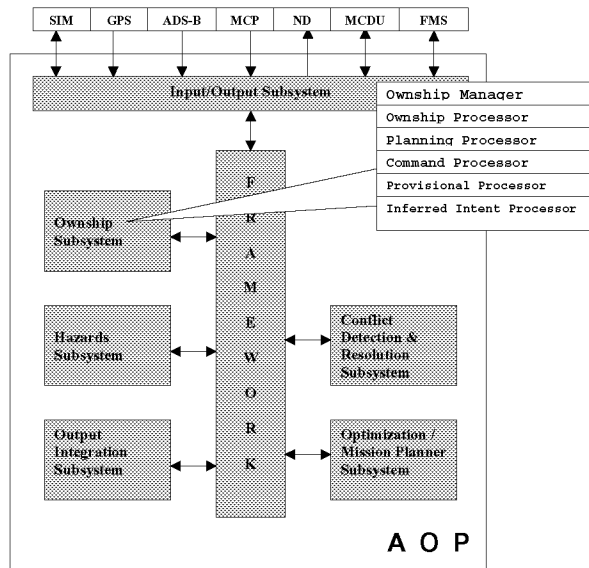


Fig. 7 Key AOP software components.

the AOP must be able to interface with a variety of external environments. In addition, the AOP must execute under Linux, Irix, and Win32. Thus, the software must be platform-independent, to the maximum extent possible.

The primary objectives of the AOP software design are (1) to instantiate the required AOP functionality described earlier, and (2) to meet the above software design constraints, in an efficient manner. This section describes the high-level software architecture of the AOP. Some of the salient software features are:

Modular Design The AOP software is written in C++, using object-oriented design patterns. The object-oriented design provides extensibility, flexibility, and re-usability of the AOP modules.

Easy to Maintain The AOP software follows the Langley Standard Real-Time Simulation in C++ (LaSRS++) guidelines developed for use in NASA Langley Research Center facilities.²⁵ These guidelines help ensure the software will interoperate with other NASA software systems, and that the AOP software will be maintainable for many years.

Platform Independence The AOP uses the Adaptive Communication Environment (ACE) as the operating system abstraction layer to retain platform independence. ACE is a high-level C++ toolkit for writing sophisticated concurrent, parallel, and distributed applications.

Flexible Interface The AOP supports data transmission using ARINC 429 characteristics over the TCP/IP in the ASTOR environment, and using

shared-memory architecture in the CMF and the ARIES environments.

Figure 7 shows the key components of the AOP software. A central component is the AOP framework which integrates the subsystems providing the AOP with the input/output (I/O), the ownership, the hazards, the conflict detection and resolution, and the output integration functions. Each subsystem has a “manager”, which is responsible for inter-subsystem communication, and one or more “processors”, which realize the functionality required of the subsystem.

The AOP Framework ensures a flexible, scalable, and efficient framework for multi-threading asynchronous communication among the AOP subsystems, and reconfiguration during operation. The AOP framework and subsystems employ a number of software design patterns to achieve specific characteristics.^{26,27} Specifically, the AOP uses (1) a Component-Configurator, (2) a Subject-Observer, (3) an Acceptor-Connector, and (4) a Reactor design pattern to achieve efficiency and flexibility. The Component-Configurator design pattern allows an application to link and unlink its components at run-time without having to modify, recompile, or statically relink the main application. This pattern supports the reconfiguration of components into the AOP without having to shutdown or re-start the main application. The Subject-Observer design pattern defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically. The AOP uses a variation of this design pattern to include one or more threads per object for asynchronous execution, to push updates on message queues for protecting causal ordering, and to reference-count messages for efficient memory usage. The Acceptor-Connector and the Reactor design patterns decouple a connections “management” handling from its “service” handling. This allows a researcher to easily develop a new “service” object to either handle data differently or get new data.

Research and Development Plans

An early (pre-Build 1) prototype version of the AOP is operational in the NASA Air Traffic Operations Laboratory (ATOL).²⁸ The prototype is being used to support research studies and investigate new functionality that may be required of the AOP.²⁹ For example, RTCA concepts involving multiple protection zones surrounding the ownship will require new mechanisms both to detect conflicts and to properly inform the crew of the conflict nature.⁵ Interoperability of the AOP with airborne and ground-based systems is also expected to be an area of active research. For example, in a mixed equipage environment, some conflict resolution maneuvers may require compatibility with Traffic Collision and Avoidance System (TCAS) maneuvers, even if the ownship is not TCAS equipped.

<u>Build 1</u>	<u>Build 2</u>	<u>Build 3</u>
<ul style="list-style-type: none"> •Ownship Conformance Monitoring •Ownship Trajectory Processing •Traffic Data Processing – ADS-B •Traffic Trajectory Generation •Static Area Hazards – FIS-B •Area Hazard Modeling •Conflict Detection •Conflict Resolution – 3D and single RTA 4-D •ACM collision avoidance zones 	<ul style="list-style-type: none"> •Coordination with ground-based traffic management tools •Time Variant Static Area Hazards •Multiple RTA 4-D automatic resolution •Fusion of hazards data •Data Ambiguity resolution of hazards •Crew Alerting of unpredictable traffic •Pairwise Separation Integration - Optimized arrival spacing •Flight Crew, AOC, ATSP goals and preferences accommodation •Trajectory Optimization 	<ul style="list-style-type: none"> •Refinement of Build 2 capabilities •Dynamic Area Hazards •Traffic Intent Inferencing •Port AOP functionality to high-fidelity simulators and research aircraft

Fig. 8 AOP phased development plan.

The AOP will also need to exchange data with, and operate cooperatively with, future ground-based air traffic management systems and airborne ASAS systems using advanced data link technologies. Future CD&R algorithms, especially those that can be formally proven to result in safe resolution trajectories, will be incorporated in the AOP as they become available.³⁰

A phased development of AOP capabilities is planned, as shown in Figure 8; other capabilities may be added as research warrants. The initial build of the AOP incorporates both state-based, intent-based conflict detection and resolution and optimized trajectories with traffic and static area hazards. Traffic state-projection trajectories and traffic TCP based trajectories are generated from traffic ADS-B data. Area hazard models are generated from enhanced FIS-like data. In Build 1, to be completed in November 2002, conflict resolution takes into account priority and maneuver flight rules, user-specified resolution degrees of freedom, and optimizes a 3-D, or single RTA 4-D automatic resolution with the FMS. Conflict detection alerts, conflict resolutions and traffic are displayed to the crew through the aircrafts PFD and ND. Flight crew interaction, with the AOP, is through the MCP and FMS CDU. Support for multiple types of protection zones surrounding the ownship will be added, and a TCAS model will be incorporated in the ATOL environment.

Along with refining Build 1 capabilities, the ensuing build of the AOP will introduce automated inter-

actions with ground-based traffic management tools being developed by the NASA Ames Research Center. Additions to the airborne components of AOP will include Time Variant Static Area Hazard conflict detection and resolution, capability for multiple RTA 4-D resolutions, and conflict prioritization. Additionally, Traffic Data Management and Area Hazard Data Management will give the AOP additional capability with ambiguity resolution, confidence assessment and data fusion from multiple data sources. The AOP will be integrated with pairwise-separation to aid the crew in sequencing and self-separation during arrivals. If required, the trajectory between broadcast TCPs will be constructed for conflict detection and resolution (e.g. determining the curved traffic trajectory to Top of Climb and Bottom of Descent). Finally, optimized en route and arrival trajectories, assimilating flight crew, AOC and ATSP goals and preferences, will also be introduced.

Build 3 will primarily add the traffic intent inferencing (if found to be necessary) and dynamic area hazard prediction to the AOP. Capabilities established in the previous builds will be further refined. The AOP software will also be ported to CMF and NASA aircraft for research in high-fidelity simulation facilities and research aircraft.

Summary

In collaboration with industry and the international R&D community, NASA is developing the Autonomous Operations Planner (AOP) flight deck de-

cision support tool. The AOP is an interactive real-time decision aid that will permit flight crews to plan and safely conduct autonomous operations in a future distributed air/ground traffic management environment. It will assist in the development of plans that are consistent with constraints imposed by a ground-based ATSP and the plans of other aircraft. It has provisions to handle multiple flight management constraints, dense traffic situations, weather hazards, aircraft operational limitations, airspace constraints, and crew or airline flight planning goals. The AOP is being implemented as a generic software system that makes use of the FMS and interfaces with other aircraft systems to ensure that AOP outputs reflect parameters specific to the equipped aircraft and the manner in which it is being operated. The AOP software architecture is highly modular, flexible, extensible, and platform independent. A C++ object-oriented design, optimized for reuse, will ensure that the AOP can support wide-ranging research studies for many years.

References

- ¹“Final Report of the RTCA Task Force 3: Free Flight Implementation,” RTCA, Inc., Washington DC, October 1995.
- ²Cotton, W. B., “New Directions In Air Traffic Control at Kennedy Airport,” Massachusetts Institute of Technology, 1965.
- ³Duong, V. N., “FREER: Free-Route Experimental Encounter Resolution—A Solution for the European Free Flight Implementation,” Air Navigation Conference, Amsterdam, 1997.
- ⁴Hoekstra, J. M., van Gent, R., , and Ruigrok, R., “Conceptual Design of Free Flight with Airborne Separation Assurance,” AIAA-98-4239, August 1998.
- ⁵“Application of Airborne Conflict Management: Detection, Prevention, & Resolution,” RTCA, Inc., Washington, DC, October 2000.
- ⁶Green, S. M., Bilimoria, K. D., and Ballin, M. G., “Distributed Air/Ground Traffic Management for En Route Operations,” *Air Traffic Control Quarterly*, Vol. 9, No. 4, 2001.
- ⁷“FAA/EUROCONTROL COOPERATIVE R&D: Principles of Operation for the Use of Airborne Separation Assurance Systems (ASAS), Version 7.1,” June 2001.
- ⁸“Airborne Conflict Management: Application Description, Version 2.5,” RTCA SC-186, March 2002.
- ⁹Eby, Martin, S., “A Self-Organizational Approach for Resolving Air Traffic Control Conflicts,” *Lincoln Laboratory Journal*, Vol. 7, No. 2, 1994.
- ¹⁰Kuchar, J. K. and Yang, L. C., “A Review of Conflict Detection and Resolution Modeling Methods,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000.
- ¹¹Johnson, W. W., Battiste, V., and Bochow, S. H., “A Cockpit Display Designed to Enable Limited Flight Deck Separation Responsibility,” World Aviation Conference, Anaheim, CA, 1999.
- ¹²Bilimoria, K. D., “A Geometric Optimization Approach to Aircraft Conflict Resolution,” AIAA-2000-4265, August 2000.
- ¹³Hoekstra, J. M., *Designing for Safety: the Free Flight Air Traffic Management Concept*, No. ISBN-90-806343-2-8, November 2001.
- ¹⁴Wing, D. J., Adams, Richard, J., Duley, J. A., Legan, B. M., Barmore, B. E., and Moses, D., “Airborne Use of Traffic Intent Information in a Distributed Air-Ground Traffic Management Concept: Experiment Design and Preliminary Results,” *4th USA/Europe Air Traffic Management R&D Seminar*, December 2001.
- ¹⁵Wing, D., Adams, R., Duley, J., Legan, B., Barmore, B., and Moses, D., “Airborne Use of Traffic Intent Information in a Distributed Air-Ground Traffic Management Concept: Experiment Design and Preliminary Results,” NASA/TM-2001-211254, November 2001.
- ¹⁶Battiste, V., Johnson, W. W., and Holland Bochow, S., “Enabling Strategic Flight Deck Route Re-Planning Within A Modified ATC Environment: The Display of 4-D Intent Information on a CSD,” SAE 2000-01-5573, 2000.
- ¹⁷Ballin, M., Wing, D., Hughes, M., and Conway, S., “Airborne Separation Assurance and Traffic Management: Research of Concepts and Technology,” AIAA-99-3989, 1999.
- ¹⁸Green, S. M., Wing, D. J., Bilimoria, K. D., and Ballin, M. G., “Personal Communication,” January 2000.
- ¹⁹“CNS/ATM Avionics, Functional Allocation and Recommended Architectures,” ARINC Inc., Report 660A, 2000.
- ²⁰Peters, M. E., Ballin, M. G., and Sakosky, J. S., “A Multi-Operator Simulation for Investigation of Distributed Air Traffic Management Concepts,” AIAA-2002-4596, August 2002.
- ²¹Vivona, Robert, A., “Applications of User Intent in Cockpit- and Ground-Based Decision Support Tools,” AIAA-2002-4556 Guidance Navigation and Control Conference, Monterey, CA, August 2002.
- ²²Hoekstra, J., Ruigrok, R., van Gent, R., Visser, J., Gijssbers, B., Valenti Clari, M., Heesbeen, W., Hilburn, B., Groeneweg, J., and Bussink, F., “Overview of the NLR Free Flight Project 1997-1999,” NLR-CR-2000-227, National Aerospace Laboratory, 2000.
- ²³McDonald, J. and Vivona, R., “Strategic Airborne Conflict Detection of Air Traffic and Area Hazards,” NASA Contract NAS2-98005, Final Report for RTO-29, November 2000.
- ²⁴Mondoloni, S. and Conway, S., “An Airborne Conflict Resolution Approach Using a Genetic Algorithm,” AIAA-2001-4054, August 2001.
- ²⁵Leslie, R., Geyer, D., Cunningham, K., Madden, M., Kenney, P., and Glaab, P., “LaSRS++: An Object-Oriented Framework for Real-Time Simulation of Aircraft,” AIAA-98-4529, Modeling and Simulation Technology Conference, Boston, MA, August 1998.
- ²⁶Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley Professional Computing Series, 1995.
- ²⁷Schmitt, D., Stal, M., Rohnert, H., and Buschmann, F., *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*, Vol. 2, 2000.
- ²⁸Mondoloni, S., Palmer, M., and Wing, D., “Development of a Prototype Airborne Conflict Detection and Resolution Simulation Capability,” AIAA-2002-4446, Guidance Navigation and Control Conference, Monterey, CA, August 2002.
- ²⁹Wing, D., Barmore, B., and Krishnamurthy, K., “Use of Traffic Intent Information by Autonomous Aircraft in Constrained Operations,” AIAA-2002-4555 Guidance Navigation and Control Conference, Monterey, CA, August 2002.
- ³⁰Muñoz, C., Butler, R., Carreo, V., and Dowek, G., “On the Formal Verification of Conflict Detection Algorithms,” NASA/TM-2001-210864, May 2001.