TOWARDS A FORMAL LANGUAGE FOR THE COMMON DESCRIPTION OF AIRCRAFT INTENT

Miguel A. Vilaplana, Eduardo Gallo, Francisco A. Navarro
Boeing Research and Technology Europe, Madrid, Spain
Sip Swierstra, Eurocontrol Headquarters, Brussels, Belgium

Abstract

The Air Traffic Management (ATM) system is continuously evolving to enhance system capacity whilst maintaining safety. To this effect, sophisticated Decision Support Tools (DSTs) are being developed and implemented in airborne and ground-based automation systems.

Interoperability among these DSTs is key for the healthy evolution of the ATM system. Unfortunately, most DSTs incorporate their own trajectory predictor (TP) tailored to their specific needs. As a consequence, predicted trajectories lack consistency, DSTs have limited interoperability and evolution is difficult if not impossible. This may lead to inefficient flight profiles, loss of ATM capacity and potentially introduces a safety issue.

To achieve consistency it is essential that, besides using similar estimates of meteorological conditions and aircraft performance, the various TPs share a common view of the way in which the aircraft will be operated over the extent of the prediction period, i.e. of the *aircraft intent*. The prediction period may cover the entire flight, from gate to gate.

This paper summarizes the key characteristics of several different methods applied to encode aircraft intent in TP implementations. The differences appear to be significant. We conclude that a standardized formal method to express aircraft intent information (understood as input to the trajectory computation process) is a key enabler to achieve the required consistency.

This paper identifies the need for a *formal language* for the common description of aircraft intent in the context of trajectory computation. Such a language, denoted as *Aircraft Intent Description Language (AIDL)*, would allow expressing the aircraft intent with different levels of detail within a single standard unifying framework. By using the AIDL as a common standard for intent sharing, different TP clients could interoperate on the basis of a consistent input to the TP, regardless of their individual requirements. The proposed approach is considered as a possible solution to ensure interoperability among future ATM automation components.

Introduction

This paper is part of a set of papers produced by the Core Team of Action Plan 16 of the FAA-Eurocontrol Coordination Committee. Other papers in this set address the proposed structure of a common Trajectory Predictor [1], the Validation methodology [2], the validation metrics to apply [3] and an initial comparison of aircraft performance models [4].

The current and envisioned Air Traffic Management (ATM) systems contain a variety of applications relying on trajectory prediction to provide the expected functionality of Flight Planning / Re-planning , Traffic Flow Management, Flight Data Processing, conflict probing, sequencing & metering and last but not least, Flight Management in aircraft. The ground portion of the ATM system aims at providing optimum services to airspace users considering safety, capacity and cost whilst minimizing the impact on the environment. To achieve this in an efficient way, Air Service Providers target a single operational concept, but with multiple modes of operation adapted to the instantaneous traffic demand and specific local conditions. To achieve this, additional automation to what is already available today in the air and ground systems will need to be introduced to accommodate the projected increase in traffic demand. Trajectory prediction is the cornerstone of this automation.

In 2000 a Technical Interchange Meeting (TIM) on “Shared Flight Intent Information and Aircraft Intent Data” was held at the FAA William

0-7803-9307-4/05/$20.00 ©2005 IEEE.

J Hughes Technical Centre at Atlantic City NJ/USA [5]. The meeting was a combined session of five Action Plans working in the framework of the FAA/Eurocontrol R&D Co-operation agreement. The meeting agreed on the definition of "Flight Intent" being "*the future aircraft trajectory expressed as a 4-D profile until destination. This trajectory will be constrained by aircraft performance, weather, terrain, Air Traffic Control, etc. It will be calculated and "owned" by the aircraft FMS and communicated to ground and air systems".* This definition implies an operational concept that is very different from today's modes of operation and assumes the availability of an infrastructure in air and ground systems that does not exist today. From the lessons learned earlier, it is concluded that this vision is rather optimistic when considering application in the short to medium term future. For example, it is not expected that RNP-RNAV, where specifications were completed in 1996, will be fully implemented before 2015. Therefore it is very probable that even if work on defining functional and operational requirements were complete now, airborne derived 4D control is unlikely to be achievable cost effectively before 2020-2025. Consequently we believe that it is more appropriate to consider ways in which early implementation could be achieved using 3D RNAV supported by appropriately configured ground based prediction and management tools. As soon as full 4-D profile control will become available, this infrastructure can provide the necessary practical to support the transition and to provide the necessary contingency for airborne equipment and communication failures.

Although the Trajectory Predictor is the key, common component in the air and ground automation systems, it has evolved into a collection of disparate TPs with differences in approach, data requirements, performance, capabilities and design. Action Plan 16 of the FAA/Eurocontrol R&D Committee on Common Trajectory Prediction Capability" has worked towards a generic TP structure (Figure 1). It was developed by comparing several existing trajectory predictors, identifying aspects found to be common and combining them into a single yet flexible logical structure consistent with all existing TPs. Once the structure was developed, it was presented in 2004 at the 2nd US-Europe Technical Interchange Meeting on Common Trajectory Prediction [6]. Consensus was achieved on the part of participating European and US experts in ground and airborne automation that the structure was consistent with existing TPs, including airborne applications. Although past attempts have found it a challenge to relate one TP to another, this structure lends itself as a generic link between any two TPs. Based on this consensus, the definition of aircraft intent was redefined in line with the proposed TP structure. Aircraft intent comprises the information that is input into the TP process, rather than the output of this process, viz. the computed trajectory. In practice, the interoperability of the various automation systems in air and ground applications can be ensured by synchronizing the intent information at the input level of the TP, viz. the Flight Script.

The Trajectory Prediction process in Figure 1 consists of a *data container*, the Flight Script and a *processor*, the Trajectory Engine (TΣ). The flight script contains all the flight-specific data required by the Trajectory Engine to compute a trajectory. The flight script is formulated to remove all intent ambiguity. The flight script includes "instruction" type data such as how to compute a turn, or how to execute a climb (i.e., the operational constraints on the trajectory elements to be modeled as well as the criteria for transitioning between trajectory elements). Consistency must be ensured between the flight script and the engine so that the engine supports all instructions commanded by the script. Following the TP structure, the Flight Script is built by the Preparation process. Input data needs range from flight plan, aircraft operating procedures, ATC constraints, aircraft performance and procedures to now/forecast atmospheric conditions. As currently envisaged, these data may be transported via System Wide Information Management (SWIM).

This paper describes a few different methodologies to describe aircraft intent in the Flight Script of TPs currently fielded. It concludes by proposing the development of a formal description of aircraft intent on the basis of the experiences gained and by borrowing concepts from the theory of formal languages and the field of compiler design. We believe that this approach facilitates the definition of a formal structure on the basis of which a formal language can be defined:

the Aircraft Intent Definition Language, AIDL. When adopted, this will be an efficient vehicle for distributing consistent aircraft intent information to the various TPs in the air and automation components of the ATM system.

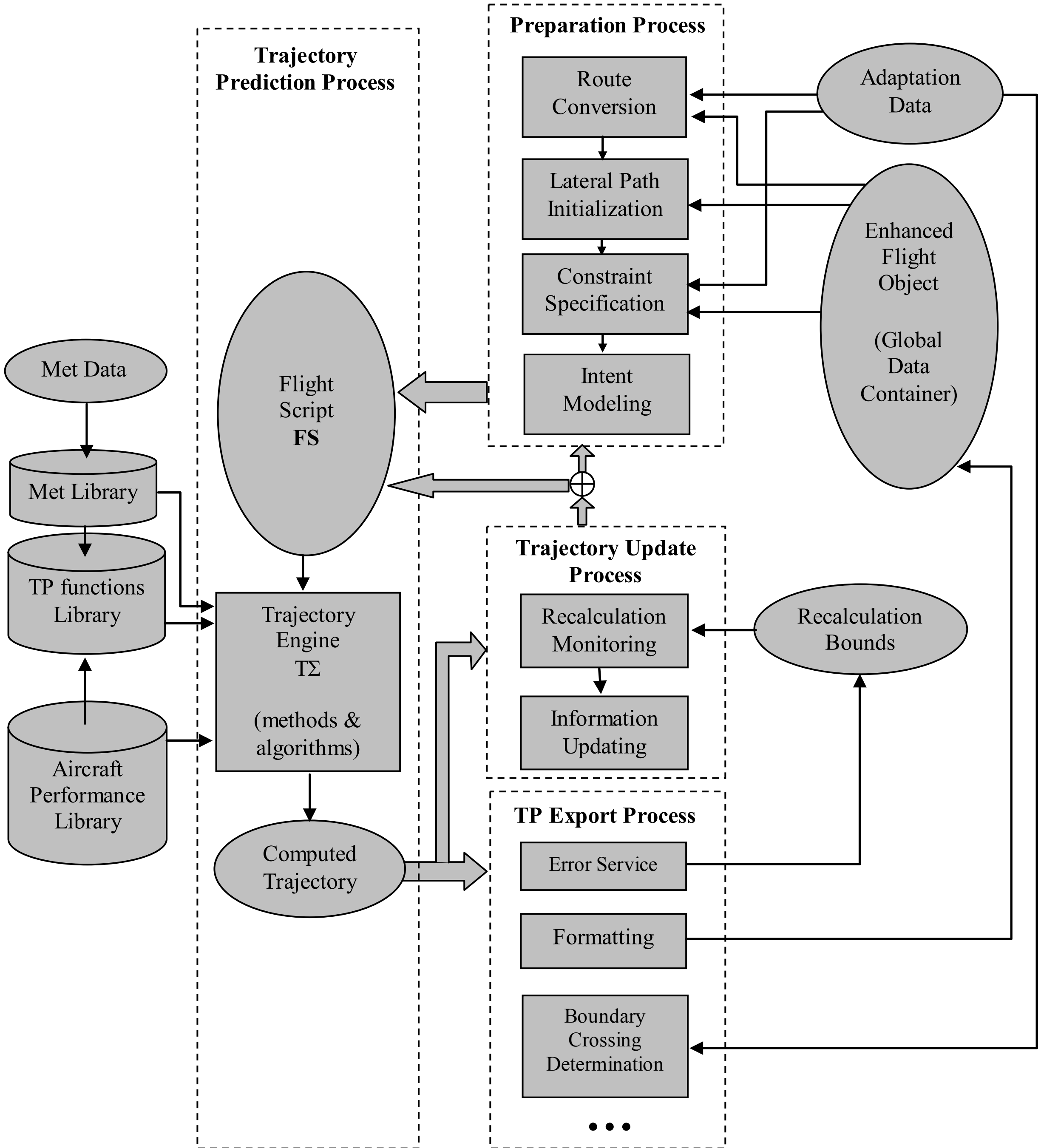

Figure 1. TP Structure

Background

In this section, we review some of the currently used and proposed methods for describing aircraft intent. The methods reviewed are designed to express aircraft intent in a manner suitable for trajectory prediction.

On board the aircraft, the Flight Management System (FMS) predicts the aircraft's future trajectory based on the active route. The description of a flight plan commonly includes a sequence of waypoints together with a set of operational constraints to be complied with at or between waypoints (speeds, altitudes, etc). The lateral path between waypoints is normally described using the standard ARINC 424 leg types [7]. In addition to constraints (established by procedures or introduced by the pilot), the operational flight plan contains parameters encoding airline preferences, such as the cost index. FMS trajectory predictions are based on the decoupling of the lateral and vertical profiles. The intent primitives for the vertical profile emulate the different vertical guidance modes. For long-term predictions, simplifying assumptions are commonly made to define the lateral path. In order to accurately predict the lateral path, it would be necessary to use intent primitives that model how the guidance function steers the aircraft along the ARINC 424 legs.

The Trajectory Synthesizer, TS, used in the CTAS toolset developed by NASA Ames [8] is also based on the decoupled description of horizontal and vertical trajectory elements. The underlying approach to describe aircraft intent for trajectory synthesis in CTAS is outlined in [9]. The aircraft intent for the horizontal path is modeled through a procedural approach. In the vertical plane, multiple altitude and speed restrictions can be specified with realistic modeling of aircraft behavior. The TS Trajectory Engine first computes the horizontal path based on an approximation of the vertical path, then the vertical profile and, subsequently, builds the output 4D-trajectory by mapping the vertical profile onto the horizontal path.

One of the challenges when applying this method is to estimate the aircraft speed and the wind conditions during turn maneuvers. In enroute sections of the flight, the horizontal flight path does not normally include significant changes in track heading. In this part of the flight this issue is of less concern. However flight operations in Terminal airspace may include significant changes in track heading, e.g., when progressing from down-wind to the localizer when at the same time the aircraft may change altitude, reduce speed, adapt aircraft configuration by extending flaps and slats and can be subject to significant changes in wind vector [10].

CITRAC (Common Interface for Trajectory Computation) [11] is an aircraft intent description method aiming to serve as a standard interface between an application and different trajectory predictors. Several ATM system manufacturers in Europe apply an Application Program Interface (API) similar to CITRAC. In essence, CITRAC divides the aircraft's intended trajectory in segments and defines a set of parameterized primitives that can be combined to define the horizontal and vertical profiles for each segment. The standard ARINC 424 legs can be modeled using CITRAC's primitives. In the CITRAC approach, the vertical profile is defined by defining constraints for speed and altitude targets to be achieved at the end of the segment and of a *pilot profile*, which defines how the segments are to be flown. In effect, the aircraft intent in the vertical plane is mapped onto the individual segments that describe the horizontal path. CITRAC provides the means to describe with a high degree of accuracy the geometric aspects of an intended trajectory. Certainly in Terminal area operations, it provides a more accurate means to synchronize the horizontal and vertical profile definition.

The concept of *path objects* [12] represents an attempt to develop a common aircraft intent description language that can be used by both FMSs and ground-based DSTs. The central premise of this concept is to facilitate the sharing of aircraft intent information between the air and the ground (intent synchronization) through the use of a common set of primitives. In essence, path objects are parameterized basic geometric shapes (in line with CITRAC's primitives) that can be concatenated to construct intended aircraft paths. In principle, most common paths flown by aircraft in the ATM context can be modeled using a relatively small number of path objects, which in turn are defined by a small number of parameters. However, the fact that path objects only encode geometric aspects of the intended trajectory and the lack of a formal treatment of the concept hamper their applicability.

The above methods are efficient in their respective areas of application if there is a well-defined relation between the aircraft intent in the vertical and horizontal plane. This is often the case for trajectory calculations that are aircraft-centric, i.e. to meet the optimum trajectory objectives from the perspective of the individual aircraft whilst respecting standard speed and altitude constraints imposed by ATC. However, in high-density traffic conditions it is key to be able to operate the ATM system at maximum capacity. In these conditions air traffic controllers will need to adapt standard routes through radar vectors in order to meter the aircraft to achieve minimum separation distances. It is expected that Decision Support Tools that are able to suggest radar vectors for merging traffic onto the localizer will be needed to efficiently introduce 3D-RNAV procedures in the major part of the flight. The requirement to accommodate Continuous Descent profiles, preferably from cruise level, complicates the requirements for the flexibility of the flight script. In the further future airborne automation systems may assist through ASAS (Airborne Separation Assurance System) supported sequencing and merging functions.

The CINTIA Trajectory Predictor [13] uses a flight script structure in which the horizontal path, the speed profile and the vertical profile are defined as independent entities. In each of the "profiles" the trajectory is described from the specific perspective of the aircraft speed profile, the progress in the vertical plane and the sequence of horizontal maneuvers. For example, a change in a "target" value in the speed profile is identified by a "profile break point" that can be defined by a time, altitude value, geographical position or path distance to a target point. This latter functionality facilitates the simple definition of Continuous Descent profiles whilst providing the same accuracy in the definition of the geometric details of the trajectory as the CITRAC method.

We observe that today, trajectory predictors can be very disparate. The internal data structures and methods are often tuned to meet the requirements of the specific client application. Consequently, the differences in Flight Script methodologies appear to be significant. Hence a novel approach is required to facilitate efficient intent synchronization that will constitute the basis for the essential interoperability among the various automation components.

Aircraft Intent in the ATM Context

We define aircraft intent as the set of *instructions* that have to be provided to the *trajectory engine* in order to compute a trajectory that realizes a given intended trajectory, i.e. that complies with all the constraints and requirements defined by the client trajectory-based application. We consider instructions as atomic and indivisible inputs to the trajectory engine. When combined in the appropriate manner, a set of instructions constitutes an instance of valid aircraft intent, providing all the necessary information required to integrate the equations of motion unambiguously.

In this section we present a tentative set of allowable instructions considered sufficient to elicit any aircraft behavior in the ATM context. The list attempts to be exhaustive regarding the different types of instructions available while remaining open to the addition of specific instructions within the types presented. The methodology followed consisted of identifying and abstracting the key inputs available to the pilot/FMS for commanding the aircraft motion within the ATM environment.

We distinguish the following four broad types of instructions: *constraints*, *configuration*, *control* and *objectives*. This classification groups together those instructions whose contribution to an operation have the same mathematical effect in the equations of motion.

Constraint Instructions

Constraints are instructions that impose a restriction on the ensuing aircraft motion. Such restriction, which generally fixes a degree of freedom, is characterized by a *constraint equation* involving one or more aspects of the aircraft motion. This equation must be satisfied during a certain time interval (the execution interval of the instruction). The execution interval may be defined directly (stating the length of the interval) or indirectly (through a termination condition). Thus, a certain type of constraint (e.g. hold altitude) may have different variations depending on the termination condition used to define the execution interval (e.g. direct statement of the execution length, terminate when a certain speed is reached, terminate when a certain along track distance has been flown, etc).

Constraints are viewed as instantaneous instructions, i.e. they are seen as being issued at a certain instant in time and declaring (directly or indirectly) how long the restriction must be active for. The following types of constraints have been identified:

Basic Constraints

In these instructions, a certain geometric or kinematic feature of the aircraft motion is to remain constant during the execution interval. The constraint equation is fairly simple and involves just one aspect of the aircraft motion. The following sub-types are distinguished:

Include the reference information in the following order:

- *Basic geometric constraints*, such as *hold bearing* (magnetic or true), *hold ground track, hold path angle*, *hold altitude* (pressure or geometric) and *hold roll angle*.
- *Basic kinematic constraints*, such as *hold speed*, which can refer to one of the following: indicated airspeed (IAS), true airspeed (TAS), calibrated airspeed (CAS), Mach number, ground speed, rate of climb (ROC) or rate of descent (ROD).

Advanced Constraints

In these instructions, the constraint equation can be fairly complex and may involve more than one aspect of the aircraft motion. The following sub-types are considered:

Include the reference information in the following order:

- *Advanced geometric constraints*, which typically require a certain geometric attribute of the aircraft motion to be an extremum (maximum or minimum) at every time instant during the execution interval[1]. Examples of advanced geometric constraint instructions are *maximum path angle* (climb), *minimum path angle* (descent) *and maximum bank angle* (turn).
- *Advanced kinematic constraints*, which are analogous to advanced geometric constraint equations but applied to kinematic attributes of the aircraft motion. Examples of instructions of this type are *maximum ROC* and *minimum ROD*.
- *Other advanced constraints*. We will consider as belonging to this category complex constraint instructions currently in use as well as other constraint instructions that can be of interest in the future. As examples, we consider *thrust corresponding to engine rating*, *track geometric path* (2D or 3D), and *follow speed law* (*speed vs altitude, speed vs time,* etc).
- The trajectory calculation is a dynamic process. Conflicts among constraints cannot always be avoided at flight script compilation time. The trajectory engine needs to have access to strategy information on how to allocate priorities among conflicting constraints and/or how to relax constraints.

Configuration Instructions

Configuration instructions capture direct pilot inputs to cockpit controls such as flap deflection lever or landing gear switch, and are inherently open-loop, i.e. the time response of the parameter affected by the instruction is not affected by the ensuing aircraft dynamics.

These instructions are considered instantaneous, i.e. they are issued at a specific instant in time. However, the response of the aircraft to the instruction is not necessarily instantaneous: a transient may be exhibited before reaching the target specified in the instruction (e.g. set flaps to 15, where 15 is the target). In order to compute the aircraft motion resulting from this type of instructions, the Aircraft Performance Model (APM) in place must include a characterization of such transient responses. The length of the transient response determines the execution interval of the instruction. The following types of configuration instructions are considered:

[1] The constraint equation for such instructions would be obtained by setting a certain derivative equal zero, so as to force the attribute in question to be an extremum.

- *Flaps extension/retraction (high-lift devices).*
- *Landing gear up/down.*

It is observed that in many ATM applications configuration instructions will not be defined explicitly in the flight script, but will be defined implicitly through the speed and/or altitude profile.

Control Instructions

Control instructions model the inputs available to maneuver the aircraft in flight (control of aerodynamic surfaces), to control the propulsion (throttle and, if applicable, mixture strength and propeller control) and, in principle for the future, to steer and brake while on the ground (landing gear steering and wheel braking). In general, the pilot and/or FMS have direct access to such inputs and can dynamically manipulate them. These instructions are useful to compute aircraft trajectories considering short-term dynamics, such as the response to FMS guidance (e.g. roll in/roll out when turning).

Except for the instruction commanding the throttle, which, as we shall see below, is considered instantaneous, control instructions are considered continuous. The following types of control instructions are considered:

- *Lateral-directional command (roll in/roll out).* Since symmetric flight is assumed, ailerons and rudder are coordinated and only one type of instruction suffices to capture lateral-directional steering commands.
- *Longitudinal command (pull up/push down).* This instruction models steering commands for pitch maneuvers.
- *Speed brakes command.* This instruction models pilot's inputs to the speed brakes.
- *Throttle command (throttle up/throttle down/reverse).* A throttle command (considered an instantaneous instruction) consists of setting a target value for the throttle (thrust). As it was the case for configuration instructions, the thrust transient response to an instantaneous throttle command is assumed to be provided by the APM.
- *Wheel brakes hold/release.* This instruction represents commanding the wheel brakes.
- *Turn left/right*. This instruction models the input available to control the steering of the aircraft on the ground.
- *Other control instructions*, such as air bleeds for air-conditioning and anti-icing equipment, propeller control, mixture strength control and afterburner control. In the future one may also have to consider specific control instructions for new vehicle classes to be integrated in the capacity-constrained airspace, e.g., helicopters, tilt rotors, UAVs, etc.

It is observed that in many ATM applications control instructions will not be defined explicitly in the flight script, but will be defined implicitly through the speed and/or altitude profile.

Objective Instructions

Objectives are considered instantaneous instructions. They define the reaching a state of motion characterized by a certain geometric or kinematic features. Two sub-types are distinguished:

- *Basic geometric objectives*, such as *acquire target bearing* (magnetic or true), *acquire target ground track angle, acquire target path angle, acquire target pitch angle and acquire target roll angle*.
- *Basic kinematic objectives*, such as *acquire target speed*, which can refer to one of the following: Indicated and True Air Speeds, Mach number, ground speed, Rates of climb and descent.

Intent Instructions

Instructions can be *explicit* or *implicit*. Explicit instructions define the exact initiation of the action to be performed in the horizontal or vertical plane. For example, a turn maneuver specified as a "fly over" [14], holds that the trajectory engine will start calculating the turn maneuver at the moment that the computed aircraft position is equal to that of the reference point specified for that maneuver. If the turn maneuver is defined as a “fly by”, then the start of the turn calculation depends, inter alia, on the level of detail that the Trajectory Engine applies when calculating the turn maneuver. The start point will be different if the turn calculation in the trajectory engine considers the roll-in and roll-out phases of the turn or assumes that the turn will be performed at constant maximum bank angle. Similar cases exist in the definition of the speed and altitude profile. An instruction is explicit if it defines a change of speed and/or altitude at a defined event. It is an implicit instruction if the target conditions are expected to be achieved at the specified event and, in the latter case; it is up to the trajectory engine to estimate the actual start point of the change.

Execution

The aircraft intent described in the flight script is executed by the trajectory engine. The execution intervals of different instructions may overlap, i.e. more than one instruction may be active simultaneously. As a result, the Trajectory Engine may be executing several instructions at the same time. For example, during the execution of a turn maneuver, the aircraft may intend to initiate a speed change at one moment and an altitude change at another moment. These changes do not necessarily need to be completed at the end of the turn maneuver. Instructions whose associated execution intervals can overlap are referred to as *compatible*. The compatibility of two instructions is possible only if the responses they elicit from the aircraft (dictated by the equations of motion) are simultaneously feasible, e.g. an aircraft cannot be requested to simultaneously climb and descend.

The Aircraft Intent Description Language (AIDL)

We compare the process of expressing aircraft intent in a manner suitable for trajectory computation to that of writing a computer program. Computer programs encode instructions using a standard language so that any computer can understand those instructions and execute them. In the same way, we propose to define a formal aircraft intent description language (AID language) to encode aircraft intent information in a rigorous and standardized manner understandable by any Trajectory Predictor. Ultimately, every Trajectory Predictor in the ATM system should be capable of processing the aircraft intent information, expressed in AIDL directly. Figure 2 illustrates how, during a transition period, the legacy trajectory predictors TP1, 2 and 3 will first translate the flight intent in AIDL through an interpreter process into their own flight script language. Subsequently, the individual Trajectory Engines will execute the local instruction to produce a 4D trajectory.

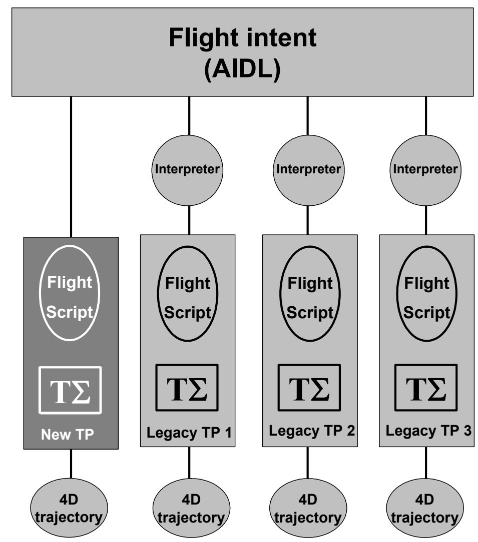

Figure 2. AIDL Implementation Scheme

This architecture will not ensure identical trajectories, because the individual trajectory predictors are optimized to meet the needs of their respective client applications. But the trajectories will be consistent, i.e. it is ensured that they are based on the same route, speed and altitude descriptions. Expressing flight intent through an AIDL common to all air and ground TP applications will facilitate the necessary consistency and pave the path towards common trajectory prediction in the future.

Conclusions

We have concluded that the definition of aircraft intent as expressed at the US/Europe Technical Interchange Meeting on “Flight Intent” in 2000 may not meet the requirements of the ATM systems that will be in operation in the short to medium term, say up to 2025.

We have summarized the main characteristics of several methods and techniques applied today for describing aircraft intent and indicated the pros and cons in relation to the target operational requirements of the trajectory predictors to which they relate. The differences in Flight Script methodology appear to be significant. Thus a novel approach is required to facilitate efficient intent synchronization to ensure that the automation systems comply with the essential interoperability requirements among automation components in the future ATM system.

Based on these lessons learned, we propose a common approach to the description of aircraft intent based on the theory of formal languages. The Aircraft Intent Description Language, AIDL, is intended to meet the requirements of air and ground automation systems. Through an appropriate "interpreter", the aircraft intent expressed in AIDL can be processed by legacy trajectory predictors that are fielded today. The proposed approach strikes a balance between supporting the legacy equipment and, at the same time, being general enough to enable the future generations of the ATM system. As a result, an AIDL built on the concepts proposed in this paper can serve as the foundation for a common intent sharing capability that can ensure the required level of system interoperability.

References

[1] Mondoloni S., S. Swierstra, 2005, Commonality in Disparate Trajectory Predictors for Air Traffic Management Applications, 24th DASC, Washington, 2005

[2] Paglione M., C. Garcia-Avello, S. Swierstra, R. Vivona, 2005, A Collaborative Approach to Trajectory Modeling Validation, 24th DASC, Washington, 2005

[3] Mondoloni S., S. Swierstra, 2005, Assessing Trajectory Prediction Performance - Metrics Definition, 24th DASC, Washington, 2005

[4] Garcia-Avello C., H Idris, R. Vivona, S. Green, 2005, Common Aircraft Performance Modeling Evaluation Tools and Experiment Results, 24th DASC, Washington, 2005

[5] FAA-Eurocontrol Cooperative R&D, 2000, Technical Interchange Meeting on Shared Flight Intent Information and Aircraft Intent Data, Atlantic City, USA.

[6] FAA/Eurocontrol Action Plan 16, 2004, Technical Interchange Meeting on Trajectory Prediction Validation, Toulouse, France

[7] ARINC, August 2004, ARINC Specification 427-17. Navigation System Database. Annapolis, Maryland, Aeronautical Radio, Inc.

[8] CTAS website. http://www.ctas.arc.nasa.gov

[9] Slattery, Rhonda, Yiyuan Zhao, April 1997, Trajectory Synthesis for Air Traffic Automation, Journal of Guidance, Control and Dynamics, Vol. 20, No. 2, AIAA.

[10] Swierstra, S., S. Green, 2003, Common Trajectory Prediction Capability for Decision Support Tools, 5th USA/Eurocontrol ATM R&D Seminar, Budapest, Hungary

[11] Leclaire, C., PQ. Nguyen, T. Mandon, 1995, CITRAC: Common Interface for Trajectory Computation, Orly (France), Centre d'Etudes de la Navigation Aerienne (CENA).

[12] Barrer, John, 2000, Integrating the Cockpit with Air Traffic Management: the Concept of Path Objects, 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli (Italy).

[13] RNAV-CDA Demonstrator – Technical description, 1997, Eurocontrol HQ/DAS/ATS.

[14] International Civil Aviation Organization, J., D., 1999, ICAO DOC 8168.

24th Digital Avionics Systems Conference
October 30, 2005