51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0443

Challenges in Achieving Trajectory-Based Operations

Karen T Cate[1]
NASA Ames Research Center, Moffett Field,CA, 94035

In the past few years much of the global ATM research community has proposed advanced systems based on Trajectory-Based Operations (TBO). The concept of TBO uses four-dimensional aircraft trajectories as the base information for managing safety and capacity. Both the US and European advanced ATM programs call for the sharing of trajectory data across different decision support tools for successful operations. However, the actual integration of TBO systems presents many challenges. Trajectory predictors are built to meet the specific needs of a particular system and are not always compatible with others. Two case studies are presented which examine the challenges of introducing a new concept into two legacy systems in regards to their trajectory prediction software. The first case describes the issues with integrating a new decision support tool with a legacy operational system which overlap in domain space. These tools perform similar functions but are driven by different requirements. The difference in the resulting trajectories can lead to conflicting advisories. The second case looks at integrating this same new tool with a legacy system originally developed as an integrated system, but diverged many years ago. Both cases illustrate how the lack of common architecture concepts for the trajectory predictors added cost and complexity to the integration efforts.

Nomenclature

CAS	=	calibrated airspeed
EDA	=	Efficient Descent Advisor
ERAM	=	En Route Automation Modernization
ERAMr1	=	ERAM Release 1
ETA	=	estimated time of arrival
TAS	=	true airspeed
TBO	=	Trajectory-Based Operations
TMA	=	Traffic Management Advisor
TP	=	Trajectory Predictor

I. Introduction

ONE of the largest challenges faced by the Air Traffic Management (ATM) community is the integration of new tools and concepts into the existing airspace system. These new concepts often need to be integrated with legacy systems. In the past few years much of the global ATM research community has proposed advanced systems based on Trajectory-Based Operations (TBO)[1]. The concept of TBO uses four-dimensional aircraft trajectories as the base information for managing safety and capacity. Both the US and European advanced ATM programs call for the sharing of the trajectory data between decision support tools for successful operations. However, the actual implementation of sharing trajectory information presents many challenges. Many advanced tools and concepts define functional and accuracy requirements for the trajectory predictor to meet their specific needs. These requirements can often be inconsistent or even conflicting across different systems. Two integration case studies, involving three systems, are discussed. These cases will focus specifically on the trajectory prediction functionality, a common feature for all three systems but very different in requirements and implementation. The first case study examines the issues with integration of the Efficient Descent Advisor (EDA), a tool which provides advisory conflict-free trajectories to meet a scheduled time, with the conflict detection/resolution functions of ERAM. These systems were developed completely independently. The integration of these tools is complicated by

---

[1] Aerospace Engineer, AIAA Senior Member.

American Institute of Aeronautics and Astronautics

the fact that they perform similar functions but are driven by different requirements. The difference in the resulting trajectories can lead to conflicting advisories. The second case describes the issues with integrating the existing operational scheduler, the Traffic Management Advisor (TMA) with the EDA, which generates speed and altitude advisories to meet the TMA scheduled times. Both tools were originally developed by NASA from the same code baseline. It was anticipated that the integration of the tools would be simplified by their common source. However, recent efforts at NASA to integrate new concepts into the Operational TMA baseline, which has diverged significantly from the NASA research baseline, have shown that this is not the case. Difficulties have arisen due to the divergence of code and requirements development of the two tools.

II. History

There have been many studies about the synchronization of trajectory predictions between airborne avionics systems and ground-based systems.[2,3] These studies show that performance of the ground-based trajectory predictors can be improved by the reception of aircraft state and intent data. Of less consideration has been the integration of different ground-based decision support tools that perform overlapping but not identical functions. This paper will examine issues with the operational integration of three such tools, ERAM, Efficient Descent Advisor (EDA) and Traffic Management Advisor.

ERAM is the system that replaces the En-Route HOST computer. In this paper, the primary ERAM function of concern is the detection of potential conflicts and maintenance separation between aircraft in the en-route airspace[4].

TMA is a legacy system deployed and maintained by the Federal Aviation Administration (FAA)[5,6]. The purpose of TMA is to reduce congestion in the terminal airspace during high traffic periods. Predictions are made of when the aircraft will transition from the en-route to the terminal airspace. These predictions are used to create a sequence for the aircraft to enter the terminal area using a first-come, first-serve algorithm. Controllers use time of crossing into the terminal airspace to ensure appropriate spacing between arrivals. If necessary, aircraft are assigned some amount of delay in order to cross at their assigned time, which the controller may absorb by issuing speed changes, temporary altitudes, or flight path modifications.

EDA was designed by NASA as an enhancement to the TMA system[7,8]. EDA provides the controller with advisories on how to meet the crossing time assigned by TMA. These advisories specify trajectories that meet the assigned time while minimizing fuel usage.

The integration of the TMA and EDA systems would seem to be fairly straightforward as both systems originated from NASA and were built on a common software baseline. The TMA software was delivered to the FAA over 15 years ago. Initially, an effort was undertaken to synchronize the two code baselines so enhancements and code corrections could be shared. This became burdensome as some of the fixes and requirements made to support the operational system were incompatible with changes or functionalities for the research baseline and vice versa. The decision was made to abandon joint development resulting in the divergence of the operational and research baselines. During the period of divergence many of the prototypical algorithms were removed from the code baseline in order to reduce code complexity and minimize risk of unintended behavior. The research prototype meanwhile underwent several efforts of refactoring in order to better support EDA and other topics under investigation. At this point, these must really be considered to be two separate systems. Although the two code bases started with the same aircraft models and equations of motion, the outcome of these years of divergence is that the two code bases have very significant differences in the parsing of constraints and intent. Consequently, the trajectory requests still share a common data structure, but the resulting trajectories may look significantly different. This becomes problematic for the integration of EDA and TMA if the TMA system provides a scheduled time that is unattainable by EDA's methods of calculation.

Lockheed Martin is the contractor tasked by the FAA to enable EDA in the ERAM architecture. As the current custodians of the software, they have been enhancing the functionality of the ERAM predictor to improve performance in the transition airspace and to introduce the EDA concepts into their legacy system. However, there are significant differences in aircraft modeling and the equations of motion used, as the two systems were developed independently. The complexity of integration of these two systems in not unexpected.

III. Issues

The problems faced in integrating EDA into the ERAM/TMA environment are fairly common in the development of large software systems. Many of the issues that arise when running these systems in an integrated environment is that while the tools have overlapping functions, several underlying requirements are incompatible. ERAM detects conflicts based on its best estimation of the trajectory the airplane will fly, using all available intent

information and historical profiles modified by observed flight performance in the adapted airspace. In contrast, EDA and TMA base their advisories on fuel-optimal profiles that the airline would prefer to fly and more detailed models of aircraft performance. These tools use airline or aircraft manufacturer data to select speeds to descend at the latest possible point and still make the crossing restriction at the transition point into the terminal airspace. This can lead to very different shapes and flight times between ERAM and EDA/TMA trajectories.

Another significant issue in the maintenance of large-scale legacy software systems is ensuring that the validation process adequately tests that the performance meets the specified requirements for a project, particularly when those requirements may evolve over time. The primary metric for validation of TMA was a time accuracy metric. However, analysis of software change requests shows correction of errors for large jumps in the estimation of the time of arrival as a flight progresses toward the meter fix as opposed to the accuracy of the original ETA. These jumps would cause issues in the scheduling algorithm and also reduce controller confidence in the predicted times. The testing procedures were not updated to reflect the addition of this "stability requirement." This stability requirement could complicate integration of new EDA functionality as some of these modifications modify or suppress changes to the trajectory shape or default speeds, which are degrees of freedom used by EDA to build advisory trajectories.

The software architecture can be a hindrance to integrating new concepts into legacy systems. For ERAM, altitude restrictions are critical intent information that must be met as part of the generated profiles. Altitude restrictions are locations where aircraft must comply with a specific altitude for procedural separation and workload balance. These altitude restrictions have been implemented in the software in such a way that they cannot be ignored during a trajectory prediction. Conversely, EDA seeks to avoid the fuel penalty of the aircraft leveling off at the altitude restriction by providing advisories for an uninterrupted descent profile that meets the scheduled time. The trajectories generated do not account for the altitude restrictions in any condition. The EDA-generated trajectory may cause ERAM to detect a conflict, as the aircraft is no longer flying to the intent that ERAM expects. EDA advisories are vetted for conflicts but due to the different models used by the two systems, may not find the same results. At best, these inconsistencies could lead to unnecessary iteration and less than optimal descent profiles. These differences in software architecture will complicate merging the two algorithms.

Ideally, the capabilities should all be functions of compatible requirements if not the same trajectory predictor, but how to integrate and validate the different algorithms so that the functionality and accuracy are sufficient for all applications is difficult. One of the difficulties in integration is that the boundary between the functionality of the trajectory predictor and that of the "client" application is drawn in different places.[10] The Action Plan 16 group, founded by an initiative of the joint FAA/Eurocontrol Cooperative R&D Committee, developed a Common Trajectory Predictor (TP) structure to capture the essential components of a trajectory predictor [9]. A major point of contention in the development of this structure was the determination of which components were intrinsic to the trajectory predictor and which were functions of the decision support tool. Often, for convenience or improved performance, decision support tool functions are embedded in the trajectory prediction software. This can make integration into a single baseline difficult. The TP functions may expect specific data to be processed prior to being called, or the decision support tool or system may rely on the TP to perform certain functions specific and unique to its own need.

One example of this would be in the parsing of the flight plan route provided by the aircraft. This route is comprised of a series of navigation fixes and airways, which must be decomposed into physical locations. For ERAM, this function is grouped into the trajectory prediction capability. For TMA and EDA, the decision support tool parses the route, as it may modify the fixes as part of its process in analyzing or advising the controller on the best ATC instructions to issue. EDA uses the lateral path as a degree of freedom for meeting a TMA scheduled arrival time, while TMA in some areas may modify the proscribed jet route in order to support requirements for getting traffic to the meter fix earlier (essentially to mimimize back-up during congestion).

Vivona et al.[10] proposes separating these functionalities using the TP boundary rule in development of the software architecture. The TP Boundary Rule is "...if the capability directly supports a key function of the client application, then the capability is considered a client application capability and outside the scope of the TP." However, refactoring two large legacy systems, while necessary, would be time consuming and expensive.

IV.   Case Studies

A. Integration of ERAM and EDA

American Institute of Aeronautics and Astronautics

In 2010, a comparison of the functional requirements that drove the development of the trajectory predictors for ERAM Release 1 and EDA/TMA was conducted by NASA and Lockheed Martin[11]. This study compared the tools for the aircraft behavior modeled and the mathematical assumptions that were used to calculate the trajectory based on those behaviors. It was critical to break down the comparison into these two factors as the former determines which profiles could be processed and the latter accounts for the differences in profiles. As illustrated in the paper and Table 1, both EDA and ERAM handle similar types of altitude constraints.

Table 1: Vertical Constraints Handled

TP	Cruise Altitude	Interim Altitude	Departure/ Arrival Speed Limit Altitude	Transition Alt (altimeter setting)
EDA	•	•	•	•
ERAM r1	•	•	•	

It is in the handling of the constraints where the difference emerges (Table 2). ERAMr1 uses an empirically based model of aircraft speed and vertical rate as the best approximation of speed in an environment where the speeds are unknown. The empirical model is based on analysis of thousands of trajectory histories at a particular site, where speed and altitude profiles are averaged over many operational conditions. EDA, conversely, uses a speed schedule of mach and CAS as these are the values that would be modified as advisories to meet a time, along with a high fidelity model of the aircraft performance for that specific type of aircraft and a nominal model of expected pilot behavior, assuming no intermediate restrictions from controllers prior to the TMA meter fix.

Table 2: Vertical Speed Models

	Descending Segments			Level Segments	
	Capture Speed	Constant Speed		Level Segments	
TP	Descent	Idle Thrust	Flight Path Angle	Capture Speed	Constant Speed
CTAS ER	CAS, Mach	CAS, Mach	CAS, Mach	CAS, Mach	CAS, Mach
ERAM r1	TAS			TAS	TAS

ERAM handles additional functionality of waypoint-defined "AT or BELOW" constraints, where the constraint is a limit rather than an exact target state. These could be of great use to EDA but would require additional integration efforts to minimize possible negative effects such as ETA jumping when iterating to meet a TMA advised time.

B. Integration of EDA and TMA

The change in the operational TMA system that would have a great impact on the EDA system is the modification of nominal speed profile used for the calculation of descents. The nominal speed value was originally selected based on typical airline and aircraft manufacturers' preferences. However, in response to an FAA requirement to increase the number of aircraft at the beginning of each arrival rush (known as "front loading" to increase pressure on the runways), the operational TMA system was modified to increase the indicated nominal descent speed.

Doing so produced the desired earlier scheduled times of arrival, but had several potential unintended consequences. First, these speed changes can degrade the accuracy of the trajectory prediction made by TMA. For example, the nominal speed for A320 aircraft flying to LAX used by the TMA system is given as 310 kts. Fig. 1 shows estimated calibrated airspeeds flown by aircraft in descent into LAX. TMA and EDA model the descent of a jet such as the A320 using a speed profile of a constant mach speed segment (used for acceleratingto a descent speed higher than the cruise speed, if necessary) followed by a constant CAS segment to the arrival fix. This constant CAS segment can be viewed in Fig. 1 as the approximately vertical values from 25,000 to 10,000 ft. As can be seen, for this day the tendency for aircraft of this type was to fly in the constant CAS portion in the range of 250-300 kts. Using the higher values as a nominal descent speed can also increase the risk that the software will be unable to
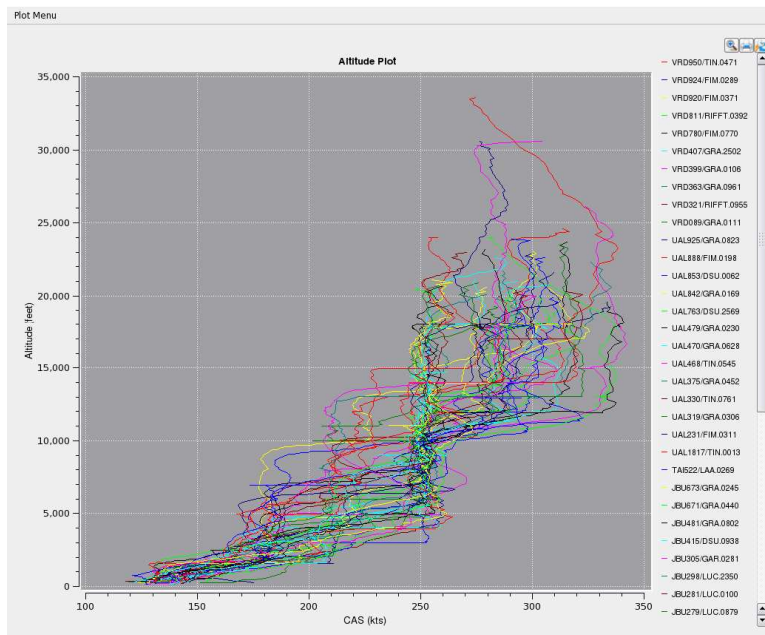
Figure 1. Calibrated airspeeds for Airbus 320 aircraft flying into LAX over a 24-hour period, 2011.

calculate the trajectory that can meet all the aircraft constraints. In both these cases, the FAA has validated that these risks do not have a significant effect on the performance of TMA.

These effects would have more impact on the EDA systsm. Both the trajectory accuracy and failure issues could cause the iteration algorithm used by EDA to perform suboptimally. Most significantly, the increase of speeds removes a degree of freedom from EDA as there would be less ability to use faster speed to meet scheduled times.

V. Conclusion

Integration of new concepts into legacy systems will always be challenging. Even systems that were created from a common software baseline may not be trivial to recombine after a significant period of isolated development. The trajectory prediction functions in EDA, ERAM and TMA systems have all been developed separately to meet the needs of their specific system. The trajectory predictors are composed of many optimizations to meet their project's requirements that should not be lost in the process of integration.

The issues with integration of the trajectory predictors could be addressed by longer-term consideration of future requirements and how they would be implemented. These do add complexity to the code baseline. Uncertainty in the acceptance of the new capability also discourages retention of the additional functionality. Since many of the tools are deployed and maintained by different organizations and contractors during different timeframes, it is challenging to find and maintain commonality between the tool's trajectory predictors. Still, many of these advanced capabilities can be maintained with low risk to the performance of the initially implemented systems and save considerable money and development time in the future.

Both case studies illustrate the need for clear, consistant and cross-comparible TP requirements, developed at as early a stage as possible. It would be difficult to build a single trajectory predictor which could meet the requirements of all systems. If the requirements were written in a consistant format, this could drive the code implementation to find common structures and minimize architectural differences. Similarly, consistant application of the TP boundary rule would enable more common TP structure between systems, allowing the TPs to function interchangeably.

Acknowledgments

References

[1]Joint Planning and Development Office: Concept of Operations for the Next Generation Air Transportation System, Version 2.0, June 2007 *Available for public download from http://www.jpdo.gov/library/NextGen_v2.0.pdf*.

[2]Chan, D.S.K., Brooksby, G.W., Hochwarth, J., Klooster, J., and Torres, S., "Air-Ground Trajectory Synchronization -- Case Studies and Metrics", 30th Digital Avionics Systems Conference, Seattle, Washington, 16-20 October 2011

[3]Torres, S., Klooster K. J., Ren, L., and Castillo-Effen, M., "Trajectory Synchronization between Air and Ground Trajectory Predictors", 30th Digital Avionics Systems Conference, Seattle, Washington, 16-20 October

[4]Crowell, A., Fabian, A., Young, C., Musialek, B. and Paglione, M., "Evaluation of Prototype Enhancements to the En Route Automation Modernization's Conflict Probe," DOT/FAA TC-TN12/3, 2011

[5]Green, S. M., and Vivona, R. A., "En route Descent Advisor Concept for Arrival Metering," AIAA-2001-4114, AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, Aug. 2001.

[6]Coppenbarger, R. A., Lanier, R., Sweet, D., and Dorsky, S., "Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering," AIAA-2004-4875, AIAA Guidance, Navigation, and Control Conference, Providence, RI, 16-19 Aug. 2004.

[7]Nedell, W., and Erzberger, H., "The Traffic Management Advisor," 1990 American Control Conference, San Diego, CA, May 1990.

[8]Swenson, H. N., Hoang, T., Engelland, S., Vincent, D., Sanders, T., Sanford, B., and Heere, K., "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France, June 1997.

[9]FAA/Eurocontrol Action Plan 16 Common Trajectory Prediction Capability: Generic Trajectory Predictor Structure *Available for public download from http://acy.tc.faa.gov/cpat/tjm//.*

[10]Vivona, R. A., Cate, K. T., and Green, S. M., "Abstraction Techniques for Capturing and Comparing Trajectory Predictor Capabilities and Requirements," AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI, 18-21 Aug. 2008.

[11]Vivona, R. A., Cate, K., and Green, S., "Comparison of Aircraft Trajectory Predictor Capabilities and Their Impacts on Air Traffic Management Automation Interoperability," AIAA-2011-6856, 11th AIAA Aviation Technology, Integration, and Operations Conference, Virginia Beach, VA, 20-22 Sep. 2011.