Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

APPLYING A DISPARATE NETWORK OF MODELS FOR COMPLEX AIRSPACE PROBLEMS

Frederick Wieland
Rohit Sharma
Ankit Tyagi
Michel Santos
Jyotirmaya Nanda
Yingchuan Zhang

Intelligent Automation, Incorporated
Air Traffic Management Research and Development Division
15400 Calhoun Drive
Rockville, MD 20866 USA

ABSTRACT

Modeling and simulation in the aviation community is characterized by specialized models built to solve specific problems. Some models are statistically-based, relying on averages and distribution functions using Monte-Carlo techniques to answer policy questions. Others are physics-based, relying on differential equations describing such phenomena as the physics of flight, communication errors and frequency congestion, noise production, atmospheric wake generation, and other phenomena to provide detailed insight into study questions. Several years ago, researchers at Intelligent Automation, Incorporated (IAI) recognized that many of the physics-based aviation models, while conceptually similar, were difficult to interoperate because of varying assumptions regarding particular aspects of flight dynamics. Despite this difficulty, the aviation community routinely use these diverse physics-based models for a single coherent study. IAI researchers have since constructed an automated method for interoperating these models in a manner that produces consistent, coherent, and comparable results even with computations that otherwise use different assumptions.

1 INTRODUCTION

The aviation community, similar to many other communities, analyzes and models existing conditions, prepares future projections, and provides guidance on policy for a variety of stakeholders. With regard to the aviation community specifically, the stakeholders are as diverse as businesses (airlines), infrastructure owners (airports), government organizations (such as the Federal Aviation Administration (FAA) or any Airspace Service Provider (ASP)), and the traveling public. Major changes to the Air Traffic Control (ATC) System require a sequence of activities, including preliminary analysis, procedure development, virtual and live testing, safety assurance, public comment, legal review, rule changes, pilot and controller training, and implementation. In the early stages of this process, statistically-based fast-time models are often used. As the process matures, detailed physics-based analysis is required to understand the full impact of the proposed change on jet routes and airport procedures, aircraft spatial density, workload of individual controllers, aircraft fuel burn, community noise impact, and environmental emissions. If the proposed change gains traction, then human-in-the-loop studies and live flight testing is required. This paper will concentrate on the physics-based analysis step. Ove the years, the aviation community has developed (and continues to develop) different tools for different aspects of the problem: air traffic control performance, controller workload, environmental compliance, airspace impact analysis, and others. In this paper, we

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

present a networked system of these disparate tools, where the tools have been seamlessly and coherently joined together to produce an end-to-end analysis environment for comprehensive studies.

Examples of end-to-end aviation analyses are common. For example, the FAA, through its Joint Planning and Development Agency, recently completed an end-to-end study of the proposed NextGen system (Gawdiak 2009; Eckhause et al. 2013). There have been comprehensive studies of future aircraft types on the aviation system (Blake et al. 2009), environmental modeling of advanced vehicle concepts (Rachami et al. 2009; Pfaender and Mavris 2010), safety analysis of hazard scenarios (Fraccone et al. 2011), and many others. Many of these studies focus on either one specific area of interest—such as safety analysis or performance analysis. Other studies consider two or more areas at once, in which disparate models are run end-to-end, the output of one being extensively analyzed (and perhaps modified) before providing it as input to the next. Consistent end-to-end analysis of concepts is elusive, in part because the various economic, safety, environmental, and performance models have been separately produced for different reasons using different programming styles by different groups of people. In an attempt to rectify this potential issue, the current end-to-end integrated analysis environment has been created.

2 RE-THINKING AVIATION ANALYSIS TOOLS

In creating an end-to-end analysis system, some thought must be given to its purpose and use upfront. We know from past experience that the world changes rapidly, that new technologies are continually introduced, and therefore the requirements for such a platform will change rapidly. While integrating existing tools is a starting point, a necessary requirement for such a system is that new tools be rapidly integrated into the environment in a seamless manner. We also know from past experience that re-analyzing past work is sometimes required, especially when a question about its credibility or validity arises. These requirements motivate the need for a common abstraction in which the tools can operate.

Creating a common ontology in which aviation models can interoperate is a rather straightforward task. Aviation models usually contain objects that are associated with aircraft, objects that describe particular flights, objects that describe the airports, airways, enroute airspace architecture, passenger demand, capacity limitations, and other ancillary concepts. The models can typically be categorized as to whether or not they are statistically-based (queueing representations being the most common) or physics-based (using actual weather, aircraft dynamics, and control parameters to simulate flights), although there are hybrid models that use each abstraction. Because aviation models, both statistical and physics-based, have similar if not identical ontologies, the need for a hierarchical-based composability paradigm for databases and ontologies is unnecessary (for details of such an interoperability representation in other domains, refer to reference (Tolk, Diallo, and Turnitsa 2007)). The main incompatibility problem between aviation models lies in their representation of performance of aircraft (their climb, cruise, and descent characteristics by weight and altitude). These problems are minimal because almost all aviation models use the Base of Aircraft Data (BADA)-formatted aircraft performance tables derived from work by Eurocontrol (Nuic et al. 2005).

The process of analysis always begins with design. In the physics-based aviation world, the design consists of airports, their taxiways and runways, the surrounding airspace, the enroute airspace, and rules that govern the use of these components. Examples of rules include limits on the speed of taxiing aircraft, altitude and speed restrictions on airways, required navigation performance standards for keeping aircraft within a lateral and vertical boundary while flying a procedure, and the required equipment aboard each aircraft to safely execute a procedure. The designs and rules can be encoded in a number of graphical interface tools. The Terminal Area Route Generation and Traffic Simulation (TARGETS) is an FAA-owned product that allows analysts to import existing airspace structure (waypoints, routes, and restrictions) as well as generate new routes or add new restrictions to existing routes (MITRE CAASD 2007). The Terminal Area and Airport Surface Editor (TAASE) allows analysts to import existing gates, taxiways, and runways, while also allowing them to add new runways and taxiways (for airport expansion studies) or change existing taxiways or gates. The Airspace Visualization Tool (ASVT) allows animation of existing airspace structures with current or future projected traffic. Both TAASE and ASVT are tools developed by Intelligent Automation, Incorporated (IAI).

After design, the process moves to analysis. There are two key components to most aviation studies, the first being performance analysis of the system, the second being environmental compliance. Performance analysis includes computing aircraft throughput, passenger delay, controller workload, and aircraft performance. There are many extant models that operate in this area, running the gamut from statistical abstractions to detailed physics-based computations. For the physics-based infrastructure, two different models are used. The first is called Metrosim, a detailed model of an individual metroplex[1] complete with all interactions between the airports as well as surface traffic movement and gate assignments. The second model is the Airspace Concepts Evaluation System (ACES) tool was selected. Both models compute the state of each aircraft (its altitude, latitude, longitude, airspeed, heading, bank angle, turn rate, and so forth) periodically in simulation time, and imposes the airspace structure on the flight (all the routes, altitude and speed restrictions). ACES simulates flights from wheels-up to wheels-on, and includes sector boundary handoffs, and center crossings. Metrosim only simulates flights within a control region of a metroplex, typically set at 250 nautical miles (nm). ACES models surface traffic using a plug-in to the ACES program called the Surface Traffic Limitation Enhancement (STLE), which computes taxi paths to and from the gate and holds aircraft for departure in a queue next to the departure runways. Metrosim models surface traffic using a set of mixed integer-linear programs. Details of ACES can be found in (George et al. 2011), while details of Metrosim can be found in (Wieland et al. 2016).

The second key component to most aviation studies is environmental analysis, in which the fuel burn, noise, and emissions are computed. The environmental analysis must be consistent with the performance analysis for the overall study to be valid—that is, the same fleet of aircraft, engine types, delay maneuvers, trajectories, and waypoint crossing times must be used in both systems for a consistent study. The Aviation Environmental Design Toolkit (AEDT), a Department of Transportation (DoT)-owned model, is used in the infrastructure for this task. AEDT is itself a federation of previously disparate models, where the federation ensures that each model uses a common database for engine types and aircraft configuration. The previously existing tools that were integrated into the common AEDT infrastructure includes the Integrated Noise Model (INM), the Emissions Dispersion Modeling System (EDMS), the System for Assessing Aviation's Global Emissions (SAGE), and the Model for Assessing Global Exposure to the Noise of Transport Aircraft (MAGENTA) (Pfaender and Mavris 2010). All these tools are part of AEDT and therefore become part of the infrastructure discussed herein.

Where aviation analysis differs from one project to another typically involves the metrics computed from the simulated data. Metrics for a given analysis tend to be very specific. In one analysis, flight time and distance above and below 18,000 feet may be a critical metric. In another analysis, the number of aircraft occupying gates may be critical. Yet other critical metrics may be the fuel burn within 100 nm of an airport, the number of level-off segments arriving aircraft must perform, departure queue length, sector occupancy by time of day, the noise profile in the neighborhoods around an airport, and many other similar quantities. To handle this variability, and to allow for the creation of new metrics, all the integrated models are configured to write their detailed output to a database (either by modifying the model's source code itself or producing an external script that loads the generated output to a database). The database is then mined by writing scripts to compute specific metrics from the amassed data. New scripts can be produced to compute new metrics, and existing scripts can be repurposed, as necessary, to fulfill a typical analysis project.

3 INTEGRATING THE TOOLS

In this section we describe the process of integrating the tools briefly. For detailed information on the inputs, processes, and output of each tool, the reader is encouraged to consult the references cited in the previous section. As each model is itself a complicated program (the ACES model, for example, contains over 200,000 lines of Java code), describing each model in detail is beyond the scope of this paper.

---

[1] A metroplex is an aviation term that describes large airports in close proximity that share low altitude airspace and/or other resources.

Integrating these tools was straightforward but replete with problems. The first integration, between TARGETS, ASVT, TASSE and ACES, was straightforward. The integration involved moving the airspace design from TARGETS, the surface design from TAASE, and the aircraft performance used by ASVT to their appropriate input files for ACES. TARGETS already contains a plug-in interface that allowed users to provide their own reformatting of its output data for their own needs. Using that interface, IAI engineers produced a plug-in that converted the TARGETS airspace design into the XML format required by ACES. Integrating TAASE was straightforward, as IAI engineers had developed TAASE and could modify its source code. The TAASE surface design software was modified so that it produced output in a format readable by the STLE plug-in for ACES. ASVT and ACES both use the Eurocontrol Base of Aircraft Data (BADA) as a standard aircraft performance description, and therefore they used the same ontology for aircraft performance, simplifying the process of integration.

Integrating ACES and AEDT was more complicated. AEDT requires five sources of data: airport data, flight data, track data, a receptor grid (for noise calculations), and a scenario setup. A conversion tool scans the output of ACES and, using an airport database, provides the information necessary for AEDT. Most ACES runs use between 6,000 and 50,000 flights, depending upon whether it is configured regionally or for a system-wide study. For 1,000 flights, which is considered a "test case" for ACES, the integration resulted in an AEDT input file that is 65 megabytes in size and contains over 1.5 million lines of text.

One of the most pressing issues in integrating ACES and AEDT is to control the trajectory that AEDT uses for each flight. The trajectory control must extend to the altitude and speed of the aircraft in addition to the normal latitude/longitude ground tracks. Fortunately, AEDT provides a mechanism that allows the user to input a four-dimensional (4DT) trajectory, consisting of latitude/longitude/altitude and time at each waypoint. Using this feature, the trajectories in ACES are identical to the trajectories used in AEDT, ensuring that the performance metrics from ACES line up exactly with the noise, emissions, and fuel burn metrics from AEDT.

Another area that is critical to integrating the two models is to ensure that the aircraft mapping from ACES to AEDT is as close as possible. The ACES to AEDT conversion tool accounts for the aircraft manufacturer, aircraft engine type, number of engines, and number of passengers, all of which are important to AEDT. However, there are some considerations missing such as wing area, aspect ratio, maximum thrust, and thrust to climb, which are difficult to determine from the ACES output. The integrated toolset is shown in Figure 1 below.

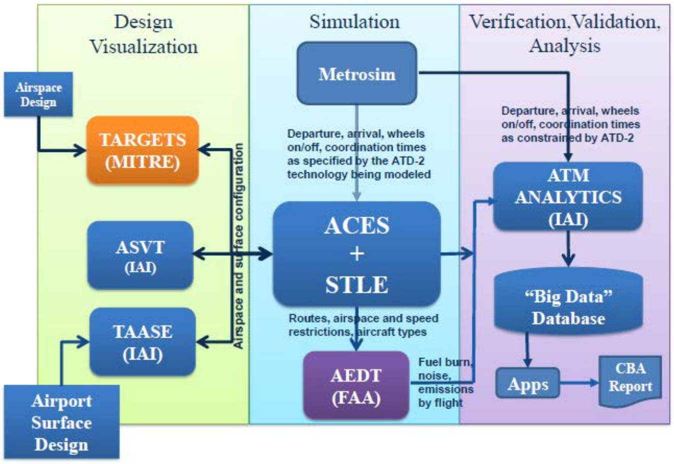

Figure 1: IAI's Integrated Model-Based Toolkit.

4 DEFINING AN ANALYSIS ENVIRONMENT

As difficult as it may sound, the integration of these existing tools is simple compared to the analysis task. A typical run of the software infrastructure produces about 2 gigabytes of data from all the systems combined. To analyze this volume of data and produce meaningful results is a challenging task in itself. To make the process easier, the infrastructure is configured to capture all the output data and direct the results to a database, currently an open-source MySQL database. Once in the database, the results can be queried and cross-compared to produce metrics.The metric production requires small scripts, or "apps," to be built, each one producing a different metric. The set of metrics produced by the infrastructure is shown in Table 1. For a given project, only a subset of these metrics may be used. If a new metric is needed, only the "app" needs to be built, computing the metric from the stored information.

Some of the metrics may be confusing to non-aviation analysts. For example, the "Average time (min) within 250 nm" seems to conflate the time and spatial component. This metric, and others like it, represent the average flight time, measured in minutes, for the portion of the aircraft's flight that lies within 250 nm of the study airport. To compute such a metric for arriving flights, a virtual clock starts when the aircraft pierces a 250 nm boundary as it approaches the airport, and the virtual clock ends upon landing. For departing aircraft, the virtual clock begins on takeoff and ends when the aircraft pierces a circle whose radius is 250 nm from the airport. The time on this virtual clock, in minutes, is the reported metric. A higher average and higher variance of this metric represents poorer performance of the system. Higher variance, for example, suggests that aircraft followed different flight paths as they approached (or departed) the airport. The different paths might arise from holding patterns flown to absorb delay absorption, vectoring off-route for spacing, and crossing to a different arrival or departure route, all of which generally indicate less efficient operations.

Table 1: Metrics computed by "apps" connected to the output database.

Category	**Metric**
Flight efficiency	Average Distance (in nm) Within 250 nm
	Average Time (min) Within 250 nm
	Average Distance from Top of Descent to Threshold (nm)
	Average Time from Top of Descent (min) to Runway Threshold
	Average Distance to Top of Climb to Threshold (nm)
	Average Time to Top of Climb (min)
	Average Time Below 18000 ft (min)
	Average Distance Below 18000 ft (nm)
	Average Time Below 10000 ft (min)
	Average Distance Below 10000 ft (nm)
	Average Number of Level-offs per Flight
	Average Time in Level Flight (min)
	Average Distance in Level Flight (nm)
	Average Percent Time in Level Flight
	Average Time Weighted Altitude (ft)
Throughput and workload	Peak Hourly Throughput
	Average Number of Speed Change Commands
Environmental	Average Aircraft CO2 emmisions (kg)
	Average Fuel Burn (kg)
	Total Arrival and Departure Noise Receptors above 65 db DNL
Surface	Average Taxi Time (min/flight)
	Average Taxi in/out Delay (min/flight)
	Average Runway Occupancy Time (sec/flight)
	Average Taxi Speeds (knots)
	Average Taxi Distance (nm)

	Average Dep Queue Length
	Average Taxi Out Time to Queue (min)
	Average Taxiing Aircraft Count (Arrival and Departures)

5 EXAMPLE ANALYSES

The model-based infrastructure described in this paper has been used in numerous studies. Examples include the effect of imperfect snowstorm forecasts on airline and passenger planning (Wieland, Sharma, and Zettlemoyer 2013), combined airport arrival and departure scheduling (Tyagi and Wieland 2012), analysis of near-term NextGen operational improvements at Atlanta (Federal Aviation Administration 2013), and analysis of direct Q-Routes (Tyagi and Wieland 2012), among others. In this section we will focus on a previously unpublished result regarding Optimal Profile Descents (OPDs) at Atlanta Hartsfield-Jackson Airport (ATL).

OPDs are procedures that have recently been introduced into the National Airspace System (NAS) that allow aircraft to fly continuously, uninterrupted by “level offs,” from their cruise altitude to the initial approach fix for landing at an airport. A “level off” occurs when either a controller requests one or the published altitude restriction at a particular waypoint requires one, and involves the aircraft interrupting its descent and establishing level flight. To do so, the pilot must increase the throttle to add energy to the aircraft in order to interrupt the descent. The procedure, therefore, requires more fuel to be consumed than if the aircraft continuously descended to its approach to the runway.

The analysis day used in the study was September 5, 2011, a day when ATL was operating under instrument flight rules, the operations were eastbound, and there were 1,238 arriving flights with 1,226 departing flights. The study focused on those flights from the Northeast, a heavily traveled corridor, and ignored departures (because OPDs are inapplicable to departures).

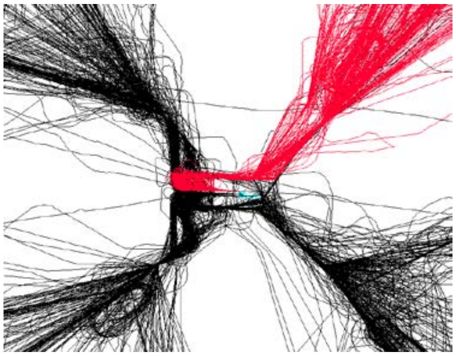

Figure 2: Flight tracks at ATL for September 5, 2011, arrivals in red and departures in black.

The model-based infrastructure was configured as follows. The ACES tool with the STLE plug-in was configured with a detailed runway model of ATL, along with all standard approach routes and final approach routes from aviation charts published by the FAA and current in 2011. ACES was configured with the Kinematic Trajectory Generator (KTG) (Zhang et al. 2010) for computing the aircraft state every half a simulated second. Twenty-two different types of aircraft were modeled by ACES, KTG, and AEDT, spanning the gamut from a small Canadair Regional Jet to a large MD90. Three standard arrival routes, identified by FAA databases using identifiers FLON7, PECHY7, and WHINZ1, were configured for the Northeast corridor. In addition, ACES was programmed to use “extended downwind” procedures for delay absorption, whereby flight paths are extended well beyond the runway to space them properly with aircraft ahead that have not yet landed. The tool that converts the ACES output to the AEDT input was used, and in this example the control of AEDT trajectories, especially with the baseline and treatment cases having very different trajectories, was critical to the success of the analysis. We assumed that all aircraft had the proper equipage to safely execute the OPD approach.

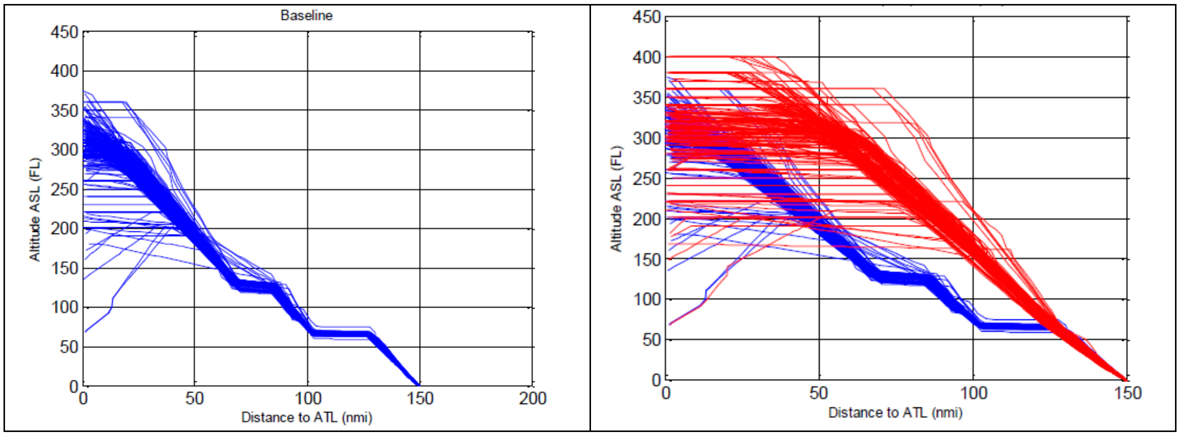

Figure 3: Original flight tracks in blue on the left chart, and those same flight tracks overlayed with the OPD flight tracks (in red) on the right chart.

As seen in the flight tracks on the left figure, there are two prominent places where flights level off, or stop their descent, on the way to ATL. On the right chart, with the OPD routes overlayed in red color, it can easily be seen that the OPD routes stay higher longer and then descend without any altitude level-offs. We call the extra time (and distance) that the OPD flights spend in cruise mode the *extended cruise segment.* The baseline case in this analysis are the flights shown in blue, the current procedure with level-offs, while the treatment case, shown in red, are the OPD flights without any level-off. The hypothesis is that the overall fuel burn for the OPD flights is less than for the flights using the current procedure.

A first test of this hypothesis revealed a problem with the AEDT model, which was subsequently overcome through several additional experiments: adding a different descent angle to the OPD route, limiting the comparison to only those flights that were common between the ACES and AEDT scenario, and correcting for anomalous fuel burn "spikes" by smoothing the dataset. The figure below shows the three scenarios that were ultimately compared (the baseline flight tracks that were unmodified, and two treatment cases each with a different descent angle).

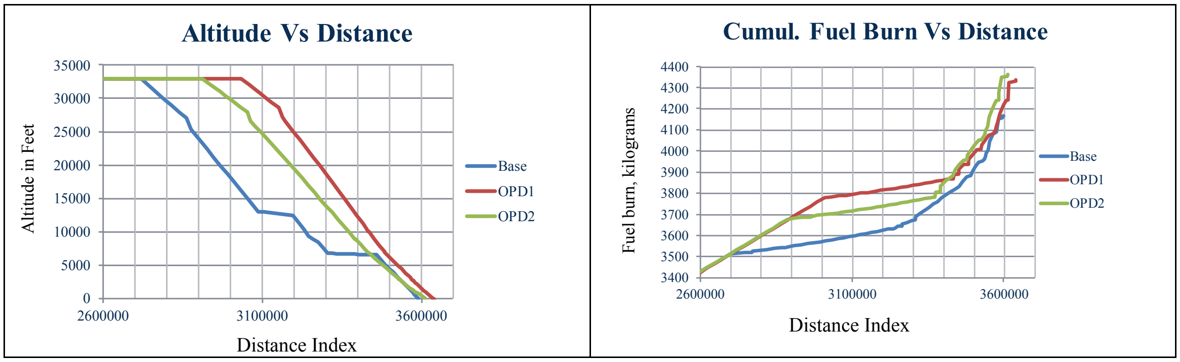

Figure 4: Original flight track (blue) and two different descent angles (OPD1 and OPD2) with the altitude and cumulative fuel burn plotted.

The results differ from the original hypothesis. Because the OPD flights remain in cruise airspace longer, their fuel burn while in cruise is higher than the comparable fuel burn while in cruise for the flights using the current procedure. Also, the length (in time and distance) of the continuous descent is larger when the flight starts descending sooner. As a result of remaining in cruise mode longer, each flight in the OPD procedure burns, on average, 106 kg *more* fuel than the comparable flight using the current procedure (which exits cruise and starts descending sooner). However, as expected, there is less fuel burn by OPD flights during descent, because flights using the current procedure must burn more fuel to level-off than the OPD flights, which have no level-off segments. Each flight in the OPD procedure during descent burns, on average, 103 kg *less* fuel than the flight using the current procedure burns during descent. The overall effect, therefore, is that the OPD procedure saves about 3 kg of fuel for each flight. In other words, the two procedures, in this experiment, are virtually the same in terms of overall fuel burn.

Other studies have shown fuel burn savings of about 100 kg for the OPD flights (see, for example, figure 5 from reference (Shresta, Neskovic, and Williams 2009)), which is much larger than the benefit computed in this study. The difference is due partly to the fact that the fuel burn model used is different between the two studies (BADA fuel burn rates are used in the referenced study as opposed to AEDT fuel burn computations used in this study), and partly because the extended cruise segments for OPD flights in the referenced study is 25% shorter than the extended OPD cruise segments in this study. Because the referenced study had a shorter extended cruise segment, and therefore a longer continuous descent segment, it computed a lower fuel burn for OPD flights with concomitantly higher benefits than the current study.

It can be concluded that the benefits of OPD routes, at least from a fuel burn perspective, is sensitive to the top-of-descent point for the OPD flight as well as the details of the descent trajectories for both the OPD and the reference baseline case. An interesting study question, unanswered herein, is to compute the descent point at which the OPD and non-OPD procedures burn the same amount of fuel. Descents that begin beyond the computed breakeven point would have higher fuel burn for OPD than non-OPD flights, while descents that begin before the computed breakeven point would have lower fuel burn for OPD than non-OPD flights.

6 CONCLUSIONS

Creating an infrastructure where an analyst can seamlessly transition from airspace and airport design through performance analysis and then through environmental analysis, using standard government off-the-shelf tools, has proven valuable in many respects. The value of computing environmental metrics in a way that is compatible with the performance analysis results produces an analysis that is credible and contains useful information for decision makers. In the sample analysis shown herein, the trajectories designed and modeled during the performance phase were the exact trajectories and aircraft types used in the fuel burn computations, even though the performance models and environmental models were created by different groups for different purposes over different periods of time using different assumptions and programming languages. The key parameters that need to be common—aircraft types and trajectories—were successfully exported from one model to another such that the result is credible.

The result is also interesting. In this particular case, a savings of 3 kg of fuel per flight, although desirable, is a fairly small number. If the OPD procedure is easy to implement and has few problems, then such a procedure becomes attractive. If, however, the OPD procedure is complicated to manage or execute, then the benefit of 3 kg of fuel saved per flight becomes less of a factor. Experience has shown that, in the case of OPD procedures, their implementation can be problematic for a number of reasons, and hence their benefits must be much higher than projected in this study for a credible business case to emerge.

ACKNOWLEDGEMENTS

The work described herein was funded by NASA under a variety of contracts, over a period of many years with a variety of participants. The author would like to acknowledge the contribution of Jeremy Smith and Patricia Glaab of NASA Langley for their help and support over the various contracts represented herein.

REFERENCES

Blake, M., R. Young, J. A. Smith, K. Wright, R. Mideiavilla, and F. Wieland. 2009. "Advanced Vehicle Concepts in the Next Generation Air Transportation System." *AIAA Aviation Technology, Integration, and Operations Conference.* Hilton Head, NC.

Eckhause, J., D. Long, R. Hemm, M. Alcabin, J. Creedon, F. Wieland, C. Murphy, T. Thompson, N. Dikerson, and D. Ballard. 2013. "Future National Airspace System Architecture Evaluation: Methods and Intial Results." *Aviation Technology, Integration, and Operations Conference.* Los Angeles, CA: AIAA.

Federal Aviation Administration. 2013. *Integrated National Airspace System (NAS) Design and Procedure Planning (INDP) Report on Simulation Results, Revision 2.* Washington, DC: Federal Aviation Administration.

Fraccone, G. C., V. Volovoi, A. E. Colon, and M. Blake. 2011. "Novel Air Traffic Procedures: Investigation of Off-Nominal Scenarios and Potential Hazards." *Journal of AIrcraft* 48 (1).

Gawdiak, Y. 2009. "JPDO Portfolio Analysis of NextGen." *AIAA Aviation Technology, Integration, and Operations Conference.* Hilton Head, SC.

George, S., G. Satapathy, V. Manikonda, K. Palopo, L. Meyn, T. Lauderdale, M. Downs, M. Refai, and R. Dupee. 2011. "Build 8 of the Airspace Concepts Evaluation System." *AIAA Modeling and Simulation Technologies Conference.* Portland, OR.

MITRE CAASD. 2007. "Terminal Area Route Generation, Evaluation, and Traffic Simulation."

Nuic, A., C. Poinsot, M.-G. Iagaru, E. Gallo, F. A. Navarro, and C. Querejeta. 2005. "Advanced Aircraft Performance Modeling for ATM: Enhancements to the BADA Model." *24th IEEE Digital Avionics Systems Conference.* Washington DC.

Pfaender, H., and D. N. Mavris. 2010. "System-Wide Environmental Impact Analysis for Advanced Vehicle Concepts in the Next Generation Air Transportation System." *AIAA Aviation Technology, Integrations, and Operations Conference.* Fort Worth, TX.

Rachami, J., J. Page, L. Zhou, and B. Kim. 2009. "Environmental Modeling of Advanced Vehicles in NextGen." *AIAA Aviation Technology, Integration, and Operations Conference.* Hilton Head, SC.

Shresta, S., D. Neskovic, and S. S. Williams. 2009. "Analysis of Continuous Descent Benefits and Impacts During Daytime Operations." *Eighth USA/Europe Air Traffic Management Research and Development Seminar.* Napa, CA: Eig.

Tolk, A., S. Y. Diallo, and C. D. Turnitsa. 2007. "Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering." *IEEE Systemics, Cybernetics, and Informatics* 5 (5): 65-74.

Tyagi, A., and F. Wieland. 2012. "Combined Performance-Environmental Analysis of Q-Routes." *Institute of Electrical and Electronic Engineers (IEEE) Digital Avionics Systems Conference.* Williamsburg, VA: IEEE Digital.

Tyagi, A. 2012. "The Design and Optimization of a Combined Arrival-Departure Scheduler." *AIAA Aviation Technology, Integration, and Operations Conference.* Williamsburg, VA.

Wieland, F., R. Sharma, and M. Zettlemoyer. 2013. "Assessment of Imperfect Weather Forecasts on Airline and Passenger Planning." *IEEE Digital Avionics Systems Conference.* Syracuse, NY.

Wieland, F., R. Sharma, A. Tyagi, M. Santos, and Y. Zhang. 2016. "Initial Results from the Metrosim Project." *AIAA Aviation 2016.* Washington, DC.

Zhang, Y., G. Satapathy, V. Manikonda, and N. Nigam. 2010. "KTG: A Fast-time Kinematic Trajectory Generator for Modeling and Simulation of ATM Automation Concepts and NAS-wide System Level Analysis." *AIAA Modeling and Simulation Technologies Conference.* Toronto, Canada.

AUTHOR BIOGRAPHIES

FREDERICK WIELAND is the Director of the Air Traffic Management Research and Development Group at Intelligent Automation, Incorporated (IAI) in Rockville, MD. He has over thirty years' experience in modeling, simulation, and analysis, including ten years with defense-related analysis and twenty with aviation analysis. He is the author of over sixty papers in conferences and journals. He holds a Ph.D. in Information Technology/Applied Probability Theory from George Mason University, and a Bachelor Degree in Astronomy from Caltech. He is a Private Pilot with over 240 hours of logged flight time. His email address is fwieland@i-a-i.com.

ROHIT SHARMA is a Senior Software Engineer in the Air Traffic Management Research and Development Group at Intelligent Automation, Incorporated (IAI) in Rockville, MD. He has over 20 years of experience in developing distributed software systems for diverse domains such as aviation, pharmaceutical, supply chains and finance. He has also worked in developing embedded systems. His main interest is to work in scientific and engineering application development. He has Bachelor Degrees in Physics and Electrical Engineering. His email address is rsharma@i-a-i.com.

ANKIT TYAGI is a Senior Research Engineer at the Air Traffic Management Research and Development Group at Intelligent Automation, Incorporated (IAI) in Rockville, MD. He graduated from Purdue with Master's degree in 2010, and has been an aviation analyst and programmer since then. His interests lie in modeling and simulation and Big Data analytics. His current focus is on investigating machine learning techniques for improving the safety of operations in the National Airspace. His email address is atyagi@i-a-i.com.

MICHEL SANTOS is a Senior Research Scientist in the Air Traffic Management Research and Development Group at Intelligent Automation, Incorporated (IAI) in Rockville, MD. He has over 15 years of experience in modeling, simulation, and analysis of spacecraft and aircraft systems. He holds a Ph.D. in Aerospace Engineering from the University of Maryland College-Park, an M. Eng. in Aerospace Engineering and a B.Sc. in Mechanical Engineering from Cornell University. His email address is msantos@i-a-i.com.

JYOTIRMAYA NANDA is a Program Manager at IAI. He has developed the ASVT and TAASE tools described earlier, and has worked on a variety of Next Generation Airspace System airspace and air portal programs for NASA. He holds a Ph.D. in Industrial Engineering with a minor in High Performance Computing from the Pennsylvania State University and a Masters of Science degree in Industrial Engineering from the Pennsylvania State University in 2002. His email address is jnanda@i-a-i.com.

YINGCHUAN ZHANG has been a Senior Research Scientist at Intelligent Automation. She has experience in air traffic management related research, especially in algorithm development, modeling and simulation, and team leadership. She has helped produce the "KTG" product mentioned in this paper. She holds a Ph.D. from the Georgia Institute of Technology. Her email address is yzhang@i-a-i.com.