

## FAST-TIME SIMULATION SYSTEM FOR ANALYSIS OF ADVANCED AIR TRANSPORTATION CONCEPTS

Douglas N. Sweet\*

Seagull Technology, Inc., Campbell, California 95008

Vikram Manikonda†

Intelligent Automation, Inc., Rockville, Maryland 20855

Jesse S. Aronson‡

Science Applications International Corporation, Arlington, Virginia 22201

Karlin Roth§

NASA Ames Research Center, Moffett Field, California 94035

Matthew Blake\*\*

### Abstract

This paper describes the approach for creating a prototype modeling and simulation system that captures the interactions between key participants within the National Airspace System. The current operational paradigm for the National Airspace System has nearly reached its limits and cannot accommodate the projected increases in air travel demand. As a result, systems-engineering practices are being used to develop and evaluate candidate operational concepts for air traffic control. As these new concepts are considered, it is imperative that adequate and credible models are available to: 1) perform conceptual trade evaluations covering many issues and metrics; 2) provide detailed evaluations from many viewpoints of changes to the system prior to their implementation; and 3) conduct real-time and non-real-time analyses of system-wide performance. It is planned that an effective modeling and simulation capability, known as the Airspace Concept Evaluation System, will be achieved by improvements to existing models, as well as development of extensive new modeling capabilities. The initial effort concentrates on the development and validation of a toolbox of compatible models that can be configured to address many different concepts and evaluation criteria. This

modeling strategy is supported by the emergence of distributed simulation capabilities together with the availability of a suite of models that represent key components of the air traffic system that can be integrated into a gate-to-gate modeling tool. In particular, the Airspace Concept Evaluation System prototype utilizes the High-Level Architecture together with agent-based software to create the large-scale, distributed simulation framework necessary to support system-wide evaluations. The complete simulation system will contain the run-time simulation engine together with scenario generation utilities, databases and post-processing tools. The prototype that is presented in this paper focuses on development of the run-time simulation engine.

### 1 INTRODUCTION

Year after year, the demand for air travel has continued to increase. Prior to September 11, 2001, the demand for air travel was growing at an average rate of approximately 4% per year. Even with the dramatic events of September 11th and resulting lower traffic levels, the Federal Aviation Administration (FAA) is predicting that air travel will reach pre-September 11th levels by 2003 and that growth rates will return to pre-September 11th rates by 2004<sup>1</sup>.

---

\* Systems Engineer, Senior Member AIAA

† Director, Distributed Intelligent Systems

‡ Systems Engineer

§ Chief, Aerospace Operations Modeling Office, Associate Fellow AIAA

\*\* Chief Operating Officer, Senior Member AIAA

National Airspace System (NAS) performance measures have supported the consensus opinion that the existing operational paradigm cannot provide the capacity to accommodate the forecasted demand. Flight delays have continued to increase year after year as air traffic demand increases year after year. The inability of the NAS to keep up with increased demand cannot be traced to a single cause, but to many factors, including:

- Hub-and-spoke flight scheduling
- Air traffic controller workload
- Insufficient capacity at major airports
- Limitations in the existing surveillance
- Limitations with the VOR-based navigation systems
- Non-optimal decision making from a lack of adequate predictive data

The FAA, the National Aeronautics and Space Administration (NASA), and the aviation industry recognize the limitations of the current NAS operational paradigm and are looking into a wide variety of possible technologies and procedural improvements to support the expanding demands on the air transportation system. The FAA's Free Flight Program and Safeflight 21 Program are two such programs looking at integrating new technologies and procedures into the NAS<sup>2,3</sup>. The FAA's Operational Evolution Plan<sup>4</sup> has created an integrated plan to incorporate these technologies into the existing system.

In addition to these focused programs, there is a need to look beyond the evolution of the existing NAS and to investigate revolutionary new Air Traffic Management (ATM) concepts that hold the promise of dramatically increasing the capacity and safety of the NAS. NASA, as a leader in ATM research, has made the improvement of the NAS one of its highest priorities. In fiscal year 2002, NASA initiated the Virtual Airspace Modeling and Simulation (VAMS) Project, a five year research and development effort, in response to the projected growth of the demand for air travel and lack of sufficient future capacity to meet this demand. The need to evaluate the costs and benefits of new operational paradigms early in the development process is of paramount importance in identifying the most promising concepts. In recognition of this critical evaluation function, one objective of the NASA VAMS Project is to develop a national simulation and modeling capability for design and tradeoff studies of system-level concepts within the national air transportation system. This element, which is designated as Virtual Airspace Simulation Technology

(VAST), must address the technical challenges of: a) understanding the limitations of current modeling and simulation methods; b) leveraging existing capabilities and facilities; c) developing models and methods for real-time and non-real-time analyses of the system; and d) achieving consensual data definition and extraction.

This paper provides a summary of the approach taken in developing a non-real-time, NAS-wide simulation capability to meet NASA research objectives. The paper also provides current status of the development effort. The overarching objective of the system is to develop a flexible NAS simulation and modeling environment that can assess the impact of new NAS tools, concepts, and architectures, including those that represent a significant departure from the existing NAS operational paradigm. First, the simulation requirements are presented from the viewpoints of assessment, architecture and modeling. Next, the development approach for the prototype Airspace Concept Evaluation System (ACES) is described in detail. These details include a demonstration of the basic architectural principles of utilizing the Department of Defense's High-Level Architecture (HLA)<sup>5</sup> in a fast-time simulation of the NAS environment and the adaptability of agent-based, object-oriented software in conforming to the HLA architecture. Lastly, the functionality of the initial prototype is analyzed. The paper focuses on the run-time aspects of the system, as these represent the current design foci of ACES development.

## **2 SIMULATION REQUIREMENTS**

Years ago, before deregulation and the adoption of the hub and spoke paradigm, the NAS could be viewed as a relatively loosely coupled system. Any disruption of service was generally localized. A flight going from New York to Denver was affected by conditions at New York and Denver but not by conditions elsewhere in the country. Today's NAS, however, is a complex, tightly coupled, highly interactive system. In today's NAS, a flight from New York to Denver is affected not only by conditions at New York and Denver, but by conditions in other key locations around the country that, at first glance, may appear to be completely unrelated. To fully understand the operation of today's NAS, the interactions among the thousands of NAS participants must be considered. As air traffic levels continue to grow, the interdependencies among NAS participants will also increase. A NAS-wide simulation capability is required to fully capture current and future NAS performance and to evaluate the costs and benefits of new operational paradigms.

From an assessment point of view, to evaluate the costs and benefits of a NAS concept, the ACES simulation capability needs to support evaluations of:

- NAS operational performance including system-level, multi-objective assessments (e.g., capacity, financial, and safety)
- National economic impact of new technologies or products in terms of costs and benefits to all participants
- Dynamic effects of human agent interactions with the system (e.g. pilots, controllers, and airline operation centers)
- Affects of uncertainty within the system and within the models used to simulate the system
- Affects of infrastructure constraints, such as Communications, Navigation and Surveillance (CNS) and information sharing, on the NAS
- Impact of transitory constraints on the system (e.g. weather and special use airspace)

To create a viable simulation system that supports the assessment requirements, the simulation architecture needs to provide:

- A flexible simulation environment ("plug and play") that can be tailored to the individual researcher's needs and for efficient use of computational and network resources
- An extensible simulation framework to provide for varying levels of complexity
- Standardized modeling interfaces to allow for easy integration of new models and legacy simulations
- A centralized simulation control and visualization capability to control and monitor a distributed simulation during runtime execution
- Data collection that is easily adaptable to the researcher's specific needs

From a modeling perspective, the simulation system requires a "modeling toolbox" with:

- Models that provide the data necessary to support the desired assessments
- Models that support the "plug and play" paradigm by conforming to the standardized architectural interfaces
- Models of varying levels of fidelity to create a tailored simulation of an appropriate level of complexity to meet specific assessment needs

- Models that "mirror" the various NAS components (current and future) to enable assessments of the dynamic interactions among the NAS entities

While pre-simulation scenario development and post-simulation analysis are critical components of the overall ACES Simulation System, we will concentrate on a discussion of the run-time functionality.

### **3 OVERVIEW OF THE ACES APPROACH**

Current NAS simulation architectures<sup>6</sup> do not provide the fidelity, flexibility, and expandability to support the simulation requirements described in Section 2. High fidelity models and simulation tools have typically been restricted to simulation of specific activities or domains within the NAS. Extending or integrating these models for NAS-wide assessments is difficult for several reasons. First, legacy systems were designed for stand-alone operation and were not designed to be interoperable. Different modeling approaches and information exchange techniques make it difficult to interconnect these systems and have them operate as a single simulation. Second, the model approach is often not amenable to modeling individual NAS participant's actions and interactions with other NAS entities. The required data to assess the system is therefore not produced by these simulations. Third, the resources and time required to maintain and enhance these systems is often extensive, bringing to question the practicality of this approach.

The challenges in creating the ACES fast-time simulation capability are considerable. To address these challenges, a unique approach has been taken, utilizing a number of existing technologies and methodologies to provide a flexible, extensible simulation framework. At the core of the simulation's architecture is a distributed simulation approach called the High Level Architecture (HLA)<sup>5</sup>. HLA is a set of processes, tools and middleware software, developed by the Department of Defense, to support "plug-and-play" assembly of independently developed models and databases. HLA, which has been in existence since 1995, is the result of significant government investment in simulation technology and has formed the basis for large-scale simulation systems including STOW, JSIMS and numerous others<sup>7,8</sup>.

HLA provides a blueprint for the integration of existing simulations and provides guidelines for how to create future simulations with interoperability in mind. HLA specifies a set of rules, an interface specification to a communications infrastructure, called the Run-Time Infrastructure (RTI), and a template to use for defining

message types for information interchange, called the Object Model Template.

Individual simulations, "federates" in HLA terminology, are integrated using the HLA RTI. The RTI provides the common set of services required by a distributed simulation system. The integrated set of simulations is called a "federation." The object model that provides the data definition for the federation is known as the "federation object model" (FOM). The FOM defines the communication protocol between the various federates (individual simulations) that make up the federation (the distributed simulation system).

Individual simulations are made up of combinations of models, where each model represents a single aspect of a simulation. Weather, human behavior, aircraft dynamics, and flight planning are all examples of models. Models cannot usually stand on their own, and to operate properly are combined with other models to create simulations using a particular simulation engine. Models of varying degrees of fidelity, representing a broad range of NAS components and systems, are available within the ACES "Model Toolbox." The Model Toolbox provides the ability to populate the simulation with the models necessary to emulate the desired operational environment and provide the researcher with the desired operational data.

For ease in integration and efficient runtime execution

of the simulation, the ACES simulation framework includes an agent-based modeling and simulation engine, which provides a software layer between the models and the HLA RTI. From the models' perspective, the underlying Agent Infrastructure supporting the simulation engine, provides a well defined model interface, independent of the specific HLA implementation. Another important feature of the Agent Infrastructure is that it provides a filtering mechanism to minimize HLA network traffic and improve overall system performance.

Together, the ACES simulation architecture combines the HLA, the Model Toolbox, and the Agent Infrastructure (see Figure 1). Each of these key elements is discussed further in the sections that follow.

### 3.1 ARCHITECTURE COMPONENTS

The top-level ACES architecture is shown in Figure 2. At its lowest level the architecture contains a number of communications protocols, including the aforementioned HLA RTI for run-time simulation communications, SQL/XML for data initialization and storage, remote method invocation (such as SOAP or CORBA) and standard web HTTP protocols, used outside of run-time for access to model and data repositories.

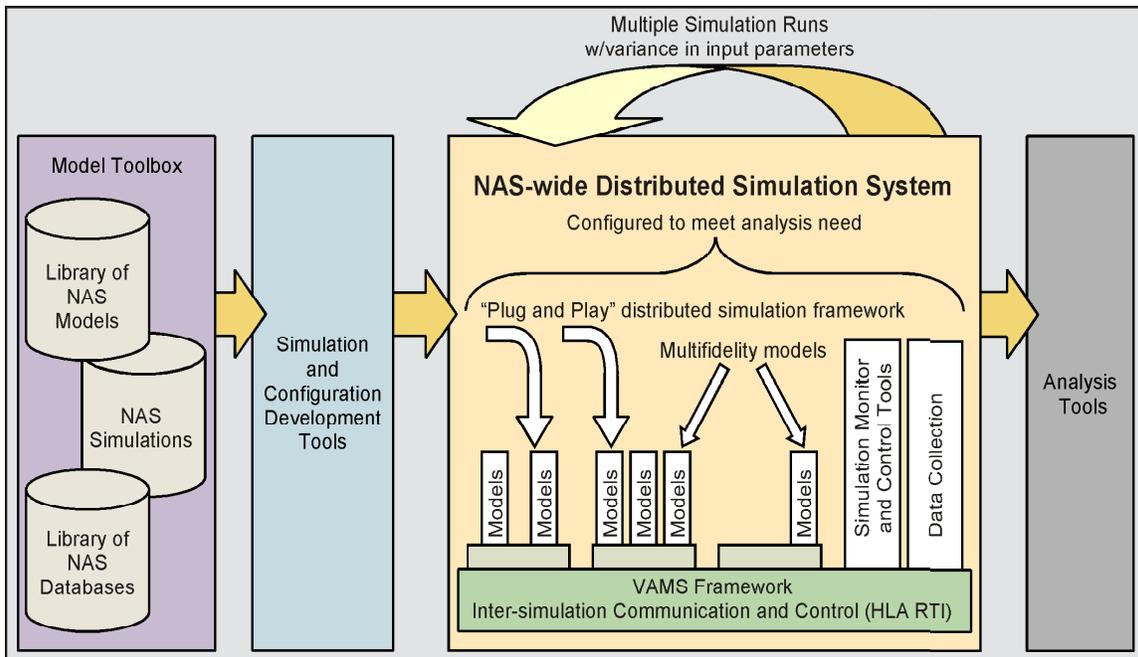


Figure 1: VAMS Simulation System Overview

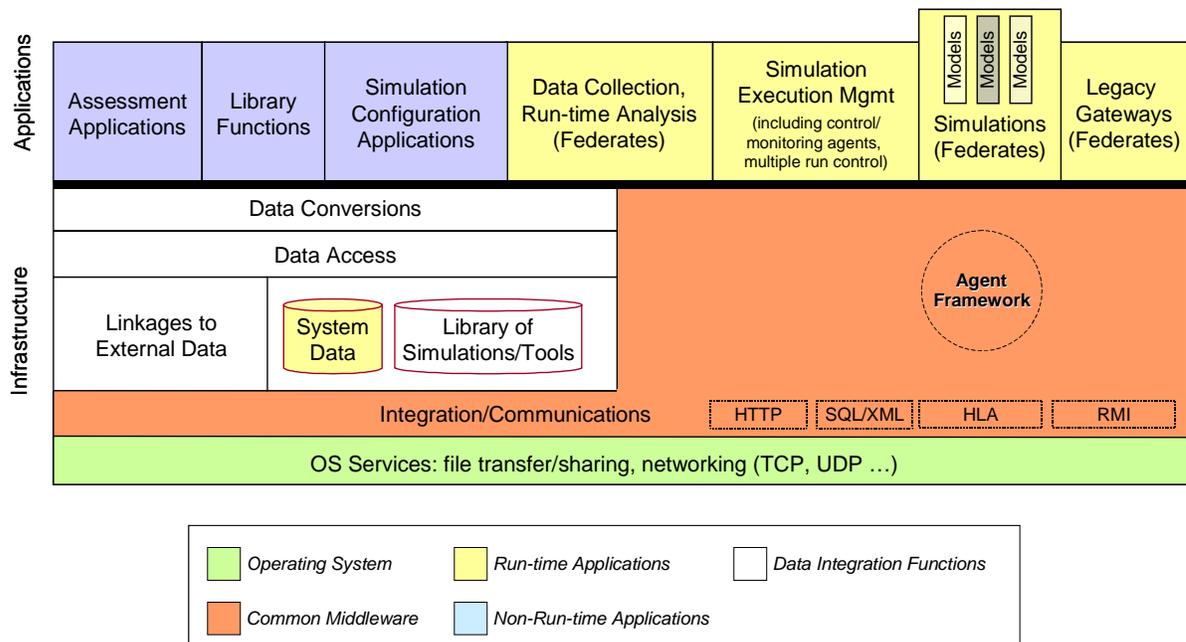


Figure 2: ACES Simulation System Architectural Approach

Above this layer, the architecture contains an optional simulation framework. HLA supports heterogeneous federates and so the specific design and implementation of the simulation engine can vary from simulation to simulation. In practice, however, simulations can be developed most efficiently and models reused most effectively through the employment of a standard simulation engine. Most of the ACES federates are built using an agent-based modeling and simulation engine, described further in a subsequent section.

The Applications layer of the architecture consists of applications built on this common core infrastructure. This include simulation federates as well as utilities such as simulation control, visualization, data collection, and analysis tools. One of the advantages of using an HLA approach is that a large number of HLA-compliant Commercial off the Shelf (COTS) and Government off the Shelf (GOTS) tools are available for integration with federations.

The system also contains a number of non-runtime components, shown on the left-hand side of Figure 2. These components focus on scenario generation, configuration of simulation applications from ACES toolbox components, data management and assessment tools for analyzing simulation results. The non-runtime aspects of the system are not covered in the current paper.

### 3.2 AGENT-BASED MODELING AND SIMULATION FRAMEWORK

To support the structured development of a scalable and reconfigurable, plug and play modeling capability, ACES adopts an agent-based<sup>††</sup> approach to model and simulate the various entities of the NAS. The agent-based modeling approach promotes an Activity Centric Programming paradigm. The paradigm encourages the decomposition of agents (e.g., FlightAgent) into activities based on logical decomposition of roles (e.g., pilot and physics) and interactions (e.g., handoffs). The interactions between agents are implemented using a message-based publish-subscribe communication paradigm.

Thus, the agent-based approach enables the development of modular code with one-to-one correspondence between (i) data structures and NAS static components (such as sectors and centers) (ii) Logical processes (program control threads or “agents”) and NAS agents (such as pilots, controllers, centers, etc.) and (iii) Inter-process messages and NAS communications (such as clearances). In addition, message based communication enables users to isolate, modify or replace a particular agent.

<sup>††</sup> From a software perspective, an agent is an event driven, persistent software entity that encapsulates the behavior of a user/entity and interacts with other agents using a message-based communication paradigm.

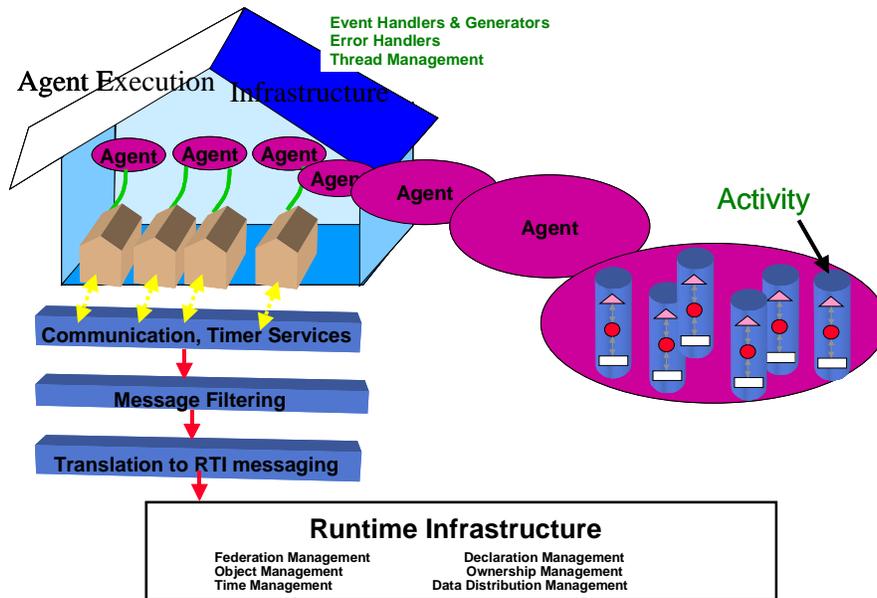


Figure 3: Integration of the Agent Simulation Engine with the RTI

For simulating these agents and their interactions, at the individual federate level, the ACES agent simulation engine provides the runtime environment for the control and execution of the agents. The simulation engine defines a standardized set of interfaces to code the agents, activities and the interactions. The engine is built on an agent infrastructure that provides a service-layered architecture with plug-and-play services such as communication, timer, error handling, thread-management and event management. Integration of services, such as communication and time-management, of the agent simulation engine, with the RTI, and compliance to the HLA interface specifications enables the ability to easily federate a group of these simulations. While communication control and execution within the federate is handled by the simulation engine, the communication and execution infrastructure between federates is provided by the RTI. For efficient execution, the simulation engine also attempts to optimize the network traffic between federates by providing appropriate filtering and data-translation capabilities. The use of a HLA compliant standardized simulation engine enables the user to mix and match a set of agents into a federate depending on the applications and available hardware. Figure 3 shows the overall

approach of integrating the agent-simulation engine with the RTI.

The agent-based modeling and simulation paradigm provides an approach for modeling (enforcing inter-agent interactions via messages, and discrete event-based simulation of agents) while providing the flexibility necessary in integrating, modifying, and configuring the models in the ACES simulation architecture.

### 3.3 THE MODEL TOOLBOX

The NAS is a collection of physical systems, human operators, and rules and procedures by which they operate. To simulate the NAS and evaluate a new tool or concept, a full range of domain and human behavior models is required. In addition, multiple levels of fidelity are needed to focus on the specific components of the NAS while maintaining the integrity of the overall simulation. For a specific concept, domain and human behavior models must be created or enhanced to address the substantially different rules, procedures, technologies, and the basic interactions between the entities in the future concept. The modular HLA based ACES framework provides a flexible architecture to assemble models in the most functional way to meet the specific operational concept requirements.

Models fall into one of three basic categories: Agent, Infrastructure, or Environment. Agent models refer to the specific NAS participants, traditionally envisioned as the vertices of the air transportation triangle: the air traffic control system, the airline, and the aircraft. The airline is represented by the Airline Operations Center (AOC) and the air traffic control system is represented by the Air Traffic Control System Command Center (ATCSCC), the en route air traffic service provider (ATSP), the terminal ATSP, the airport ATSP, and the ramp manager. Infrastructure models refer to the components that operate between two NAS participants or between a NAS participant and the environment. For example, this category includes the CNS and weather forecasting systems. The Environment covers the remaining models, including the airspace definitions, airport locations and layouts, and truth state weather. Put simply, NAS Agents operate within the NAS Environment and communicate with each other and the NAS Environment through the NAS Infrastructure.

Each software agent contains a set of models that represent the three Agent, Infrastructure, and Environment modeling categories. An aircraft, for example, requires agent models of its internal components / systems (e.g., aircraft performance and dynamics, pilots, and the flight management system). Environmental models are needed by the aircraft to provide the data on the physical environment in which it will operate (e.g., the winds the aircraft is flying in, the location of the runways, the navigation aids). Infrastructure models provide the necessary CNS capabilities (e.g., radio communications, transponder) to emulate the interactions between the aircraft and the other NAS participants. Model fidelity provides another critical dimension to the modeling toolkit approach. For example, modeling of ADS-B communications can be tailored to the specific needs of a simulation. A low-fidelity model of ADS-B, modeling error free communication, will be appropriate for some basic concept studies. For other simulation studies, such as a safety study looking at using ADS-B for aircraft self-separation in congested airspace, a high-fidelity ADS-B model which explicitly models the errors and transmission limitations of ADS-B will be required.

The agent-based simulation approach described in the previous section specifies agents as the building blocks of the simulation. Agents consist of one or more activities. Each activity, in turn, utilizes a specific model or set of models to provide the desired functionality of the agent. For a given activity, different models can be incorporated to provide various levels of fidelity. The agent-based modeling

approach provides the ability to tailor a set of activities and models from the available modeling toolkit to create agents best suited for a given assessment. This approach also provides an efficient mechanism for incorporating new models into an existing structure.

## **4 ACES SIMULATION SYSTEM DEVELOPMENT**

Developing and validating ACES is a multi-phased, multi-year project. The first phase of the project was aimed at identifying the overall architectural and modeling requirements of the ACES simulation and developing a prototype system for a proof-of-concept demonstration of an agent based application transitioning to a non-realtime, distributed HLA environment. This effort was accomplished between October 2001 and February 2002. The second phase, which is currently in progress, creates the baseline ACES simulation system, with emphasis on providing a basic set of models to support NAS-wide analysis and a basic system-level functionality with regard to data collection, simulation control, and visualization. It is planned that later efforts will continue to add functionality to the Modeling Toolbox by increasing the scope and fidelity of the NAS component models. Later efforts will also concentrate on improving computational performance and user interfaces to enable effective use of the technology by airspace analysts.

### **4.1 PROTOTYPE DEMONSTRATION**

The prototype software system transitioned an agent-based legacy simulation into the HLA environment and integrated it with a centralized, HLA-based data collection and Simulation Manager. Agents En Route (AER)<sup>††</sup> is the agent-based, en route simulation that provided basic, low-fidelity models of a wide variety of NAS entities for the prototype. Within the AER legacy simulation, the ATCSCC, ARTCCs, ARTCC sector controllers, AOCs, aircraft and pilots are all represented as agents along with a set of environment and infrastructure models and datasets. NASA's FACET simulation<sup>9</sup> provided critical datasets used to

---

<sup>††</sup> Agents Enroute Software (AER), was developed by IAI under NASA SBIR Phase II contract NAS2-00012. NASA's FACET software was important in establishing a foundation for developing AER and we would like to thank the FACET team for their providing software code, datafiles, and for their support.

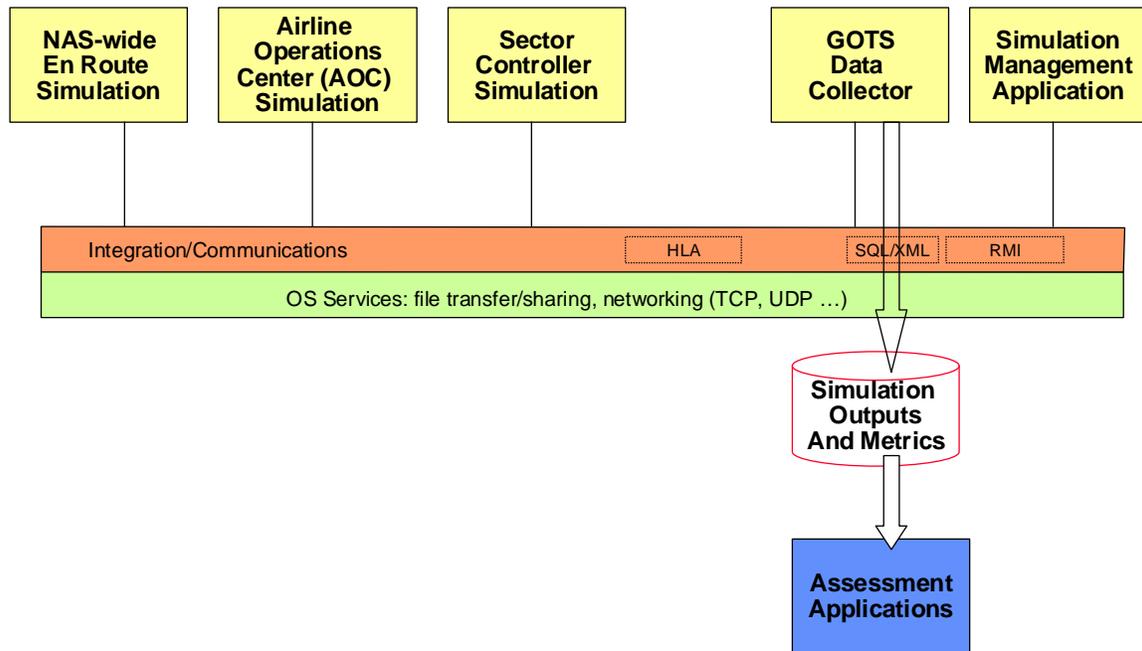


Figure 4 : Proof-of-Concept System

support the aircraft and airspace modeling within AER. For the demonstration, the AER simulation was augmented and divided into multiple federates to create a distributed simulation. Specifically, the simulation was divided into an AOC federate, an ARTCC sector controller federate, and an en route (the rest of AER) federate. In addition, a centralized data collection federate (a GOTS product provided by the Defense Modeling and Simulation Office called the HLA Data Collection Tool) and a Simulation Manager federate (an in-house design providing basic centralized simulation control functions) were integrated into the proof-of-concept demonstration system as depicted in Figure 4.

In addition to the HLA integration task, the AER simulation system models were enhanced to allow the simulation to demonstrate the benefits of the agent-based approach within a representative operational environment. The selected operational concept was a combination of autonomous aircraft<sup>§§</sup> and managed

aircraft<sup>\*\*\*</sup> flying in the same airspace, all interacting with en route controllers, AOC's and other aircraft. The scenario included a mix of different aircraft or fleets of aircraft from different airlines utilizing different operating strategies (e.g. different collision detection and resolution algorithms).

Figure 5 shows the prototype system running on three workstations with the AOC and the ARTCC on one computer, the Simulation Manager and Data Collector on another, and the En Route federate on a third. The prototype clearly demonstrated the ability to quickly integrate different NAS models and an agent-based legacy simulation, that was made HLA-compliant, together with new and GOTS support tools within the prototype simulation environment. It also demonstrated the ability to capture the interwoven behavior of NAS entities that is required for future concept assessments. Lastly, HLA successfully supported a distributed, fast-time NAS simulation.

<sup>§§</sup> Autonomous aircraft are aircraft that can detect the position of other aircraft around them and that take on responsibility for maintaining separation between aircraft. They also have the ability to change their flight plan autonomously.

<sup>\*\*\*</sup> Managed aircraft are aircraft that follow a prescribed flight plan unless directed by Air Traffic Control (ATC) to change. ATC is responsible for maintaining separation between managed aircraft



Figure 5: Prototype demonstration

## 4.2 LESSONS LEARNED

While the lessons-learned by developing a prototype were generally positive, the initial effort did demonstrate the importance of proper design of the interactions between federates. The rapid prototype adopted the existing inter-agent interaction protocol, which was designed for in-memory access speeds and funneled it through the RTI. Data was not optimized for network-speed communications and consequently, the performance was affected by the volume of data transferred. Also, the simple FOM used in the prototype allowed only minimal use of the RTI's networking optimization capabilities, further degrading system performance. Thus, in designing the baseline ACES simulation system, considerable attention is being paid to inter-federate and hence inter-model communications.

Experience with the prototype also enabled the identification of additional capabilities that will have to be built to provide ACES with multi-run assessment capabilities. HLA has been applied primarily to human-in-the-loop simulations and as such has no inherent capabilities for multi-run, Monte-Carlo simulations. Several programs within the Department of Defense "simulation-based acquisition" community are beginning to use HLA for analytical simulations. ACES will take advantage of these efforts to support the development of these capabilities wherever possible.

## 4.3 BASELINE SYSTEM DEFINITION

Using the lessons-learned and experience from the prototype demonstration, the baseline ACES simulation system has been refined. The system will be capable of simulating a day-in-the-NAS scenario. The current NAS operational environment is being targeted to create a validated baseline on which to grow ACES.

The overall emphasis of this development phase is to create the baseline architectural infrastructure, expanding on the minimal set of HLA services that were used by the prototype system. In addition, an

agent-based simulation infrastructure, Cybele™<sup>10</sup>, is being adapted to ease of integration of new models. A basic Simulation Control, Data Collection, and Visualization functionality will be provided with the baseline system.

The initial set of NAS models for the ACES Model Toolkit has been identified and is under development. Existing models are being utilized where possible to optimize the modeling effort. The agent-based modeling paradigm is being utilized in creating this baseline Modeling Toolkit to ensure that the system is easily adaptable to specific research needs. Traffic Flow Management (TFM) modeling and the interactions among the TFM participants will be a particular focus of the modeling effort.

## 5 SUMMARY

The ACES design provides a unique approach to the modeling and simulation of new NAS concepts. By acknowledging the need to provide a flexible and dynamic simulation capability, the ACES design utilizes the HLA / Agent Infrastructure to realize these critical requirements. The Modeling Toolkit approach, and the agent-based paradigm used to structure the models and their interactions, provides a modeling structure that correlates to the NAS operational environment. The combination of these key elements will provide a simulation tool capable of supporting complex, distributed fast-time simulation to meet NASA's research requirements.

## 6 ACKNOWLEDGEMENTS

The ACES prototype development effort is part of a three and a half year project supported by the Air Traffic Management System Development and Integration Contract team consisting of Raytheon, Seagull Technology, SAIC, Intelligent Automation, Inc. (IAI), TITAN, Metron, Booz-Allen Hamilton and others. The authors wish to acknowledge the efforts of the many individuals that have contributed to the design and development of the ACES prototype. Specifically, we would like to acknowledge Pauline Froemberg and Ed Stevens of Raytheon; Ari Stassart, Alex Huang, George Couluris, John Sorensen, Matthew Blake, George Hunter, and Jon Cate of Seagull Technology; Roger Wuerfel, Fang Yu Chi and Jim Ramsey from SAIC; Phil Goetz, Wilbur Peng, Goutam Satapathy, Anna Tietinen. Mark Becker and Renato Levy from IAI; and Paul Abramson of PDA Associates (TITAN, Inc).

## References

- [1] Federal Aviation Administration, Aviation Policy and Plans (APO), <http://api.hq.faa.gov/pubs.asp>.
- [2] Federal Aviation Administration, Free Flight, <http://ffp1.faa.gov/home/home.asp>.
- [3] Federal Aviation Administration, SafeFlight 21, <http://www.faa.gov/safeflight21/>.
- [4] Federal Aviation Administration, Operational Evolution Plan, <http://www.faa.gov/programs/oep/>.
- [5] <https://www.dmsomil/public/transition/hla/>.
- [6] Odoni, A. R., Bowman, J., Delahaye, D., Deyst, J. J., Feron, E., Hansman, R. J., Khan, K., Kuchar, J. K., Pujet, N. and Simpson, R. W., "Existing and Required Modeling Capabilities for Evaluating ATM Systems and Concepts," Final Report, International Center for Air Transportation, MIT, Mar. 1997.
- [7] Aronson, J. "The STOW System Architecture and Implementation Design." Proceedings of the 14th Distributed Interactive Simulation Conference, Mar. 1996, Orlando, FL. Paper 96-14-062.
- [8] Powell, E. T. and Pratt, D. R.. "The Joint Simulation System Architecture: A Foundation For Future Training Systems," *Proceedings of IITSEC 1997*, Orlando, FL, Nov. 1997.
- [9] Bilimoria, K. D., Sridhar, B., Chatterji, G., Sheth, K. S., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," 3<sup>rd</sup> USA/Europe Air Traffic Management Research and Development Seminar, June 2000.
- [10] OpenCybele, [www.opencybele.org](http://www.opencybele.org), Intelligent Automation Inc.