

The Distributed Space Exploration Simulation (DSES)

Edwin Z. Crues
Simulation and Graphics Branch
Automation, Robotics and Simulation Division
NASA Johnson Space Center
2101 NASA Parkway
Houston, TX 77058
281-483-2902
edwin.z.crues@nasa.gov

Victoria I. Chung
Flight Simulation and Software Branch
Flight Research Services Directorate
NASA Langley Research Center
24 West Taylor Street
Hampton, VA 23681
757-864-6406
Victoria.I.Chung@nasa.gov

Mike G. Blum
Aviation Systems Division,
Aerospace Simulation Operations Branch,
NASA Ames Research Center
Moffett Field, CA, 94035
650-604-4591
mblum@mail.arc.nasa.gov

James D. (Dan) Bowman
Missile Systems Group
Teledyne Brown Engineering
300 Sparkman Drive
Huntsville, AL 35805
256-726-1712
dan.bowman@tbe.com

Keywords:

Distributed Simulation, Space, Exploration, NASA

ABSTRACT: *The paper describes the Distributed Space Exploration Simulation (DSES) Project, a research and development collaboration between NASA centers which focuses on the investigation and development of technologies, processes and integrated simulations related to the collaborative distributed simulation of complex space systems in support of NASA's Exploration Initiative. This paper describes the three major components of DSES: network infrastructure, software infrastructure and simulation development. In the network work area, DSES is developing a Distributed Simulation Network that will provide agency wide support for distributed simulation between all NASA centers. In the software work area, DSES is developing a collection of software models, tool and procedures that ease the burden of developing distributed simulations and provides a consistent interoperability infrastructure for agency wide participation in integrated simulation. Finally, for simulation development, DSES is developing an integrated end-to-end simulation capability to support NASA development of new exploration spacecraft and missions. This paper will present current status and plans for each of these work areas with specific examples of simulations that support NASA's exploration initiatives.*

1. Introduction

The Distributed Space Exploration Simulation (DSES) is a NASA research and development project focusing on the technologies and processes related to the collaborative simulation of complex space systems involved in the exploration of the Earth's solar system.

The traditional approach to the simulation of space vehicles has been through disjoint collections of simulations. Each individual simulation usually focuses on a specific domain aspect of a space vehicle and is developed, maintained and executed within the confines of the facility having the particular domain expertise. While this provides high fidelity modeling

in a particular area, other domain areas are usually modeled to a lower fidelity. When a simulation requiring higher fidelity models across the simulation is required, a larger and more complex simulation is built; these simulations often re-implement the existing domain specific models in a new simulation architecture.

However, a new approach to building large scale simulation is emerging. This is the area of distributed simulation. In this case, the simulation is actually a collaborative association of a number of simulations working in coordination to simulate the system. This approach has a number of benefits. For instance, organizations with specialized domain expertise can

contribute to the aggregate simulation thus combining their individual expertise into a larger body. Another benefit is the ability to restrict access to or hide the details of proprietary technologies or algorithms yet still making the service available to the simulation. Yet another benefit is the ability to distribute the cost of development across multiple organizations and utilize the shared computer resources of those organizations.

One of the leading distributed simulation technologies in use today is the High Level Architecture (HLA) which was originally developed and deployed by the U.S. Department of Defense (DoD) Defense Modeling and Simulation Office (DMSO). This is an evolution of previous DoD distributed simulation technologies and has now been adopted in a modified form as an international standard by the Institute of Electrical and Electronics Engineers (IEEE) as the IEEE 1516 HLA standard. [1][2][3][4]

2. Project Overview

The principal objective of the DSES project is to investigate the use of distributed simulation in general (initially the use of HLA) and build large scale integrated space vehicle simulations in support of the Constellation Program (CxP). The Constellation Program is the organization responsible for implementing the vision of the Exploration Systems Mission Directorate (ESMD), which leads NASA's new exploration initiatives. This will involve the technical expertise and active participation of scientists and engineers from all 10 of NASA's research and flight facilities.

Like most projects, the DSES project did not emerge fully formed in either scope or participation. A number of NASA centers have been investigating distributed simulation technologies. One precursor to the DSES project was the DIstributed Simulation (DIS) project. This project's purpose is to build a distributed simulation for use in flight procedures development and training of the HII-A Transfer Vehicle (HTV) and its operations in proximity of the International Space Station (ISS). This simulation is referred to as the HTV Flight Controller Trainer (FCT) simulation. This is a collaborative effort between the Japanese Aerospace eXploration Agency (JAXA) and the NASA Johnson Space Center (JSC) Mission Operations Directorate [5][6]. In this simulation, the HTV components run at the JAXA facilities in Tsukuba, Japan and the ISS components run at JSC in Houston, Texas. A graphic scene generated from this simulation is shown in Figure 1.



Figure 1: The HTV FCT Simulation

Much of the knowledge and technologies used in the HTV FCT have been directly applied to the DSES project.

Initially, the DSES project was referred to as a “coalition of the willing” in the sense that small groups of NASA scientists and engineers with similar interests and very little funding met on a regular basis to exchange information and ideas on new approaches to simulation. These groups located at JSC, NASA Ames Research Center (ARC) and NASA Langley Research Center (LaRC) configured a small informal network of computers and began testing out some simple distributed simulations. As the DSES work progressed, interest began to spread to other centers. Another interested group at NASA Marshall Space Flight Center (MSFC) soon joined the growing DSES team. Currently, a group at NASA Kennedy Space Center (KSC) is joining in on the DSES work. This will bring the count to five NASA centers with plans to extend to all NASA centers in the coming years.

Early DSES work was focused in three principal areas: Infrastructure, Expertise and Product. The first area developed was Infrastructure and required the build up of the computer, network and software systems required to support a working distributed simulation capability. The second area was Expertise and required the build up the knowledge base of the scientists and engineers who develop and use these distributed simulations. The third, and final, area was the Product and required the process of building up distributed tools and simulations used for the analysis, planning, construction, development and operation of future space exploration systems.

To build up these areas, the DSES project adopted an open-ended development plan based on iterative

evolutionary phases of development. The product of each iteration was a working distributed simulation of a space vehicle. Every iteration built up one or more of the three areas, improving NASA's overall capabilities to design, develop and deploy complex distributed simulations of space exploration systems.

In September of 2006, the DSES project successfully demonstrated a distributed version of the Orion/Ares launch vehicle (see Figure 2).

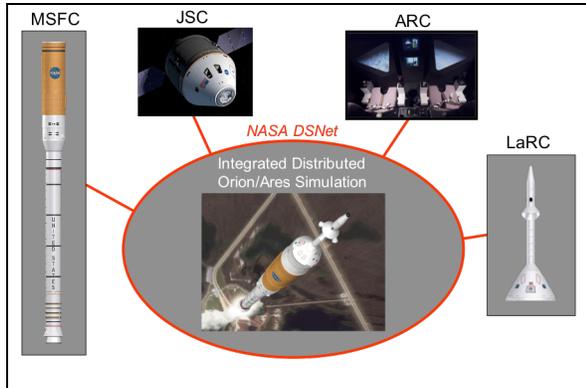


Figure 2: Orion/Ares Launch Simulation

This simulation is composed of four federates each providing functionality for important subsystems in the vehicle stack. The Ares launch vehicle or Crew Launch Vehicle (CLV) federate is an MSFC simulation. The Orion spacecraft or Crew Exploration Vehicle (CEV) federate is a JSC simulation. The Orion Launch Abort Systems (LAS) federate is a LaRC simulation. The final federate is the crew interaction simulation which runs at ARC. All these federates are run in a federation of simulations to demonstrate an integrated and distributed simulation of the Orion/Ares vehicle from launch, thru LAS jettison, to stage 2 separation and on to orbit.

As the DSES project now moves out of its early formative and development phases, the project will transition from a mostly research and development project into a primarily product development project. In that regard, the DSES project will again focus on three areas; however, this time they are: Network Infrastructure, Software Infrastructure and Simulation Development.

3. Network Infrastructure

While most simulations require an infrastructure of computers and software, distributed simulations also rely heavily on a network infrastructure. As a result, the DSES project will focus on building up NASA's

network infrastructure needed to support distributed simulation.

3.1 NASA Distributed Simulation Network (DSNet)

The DSES network infrastructure is currently based on an interconnection of facility local area networks through the NASA Integrated Services Network (NISN). This with a combination of externally accessible IP addresses and access allocations through the facility network firewalls forms the NASA Distributed Simulation Network (DSNet). While the current network has limited levels of service guarantees, the determination of the form and necessity of network guarantees is an area of DSES investigation.

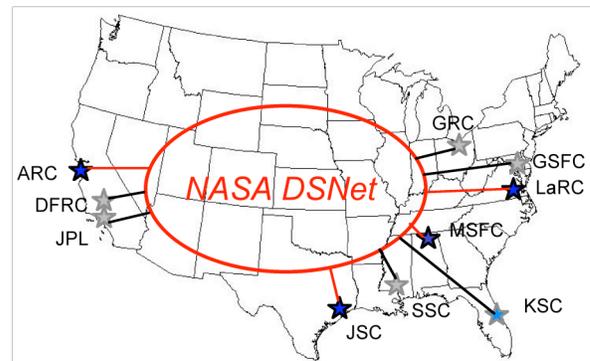


Figure 3: Distributed Simulation Network

Currently, there are 5 NASA centers connected through the DSNet: ARC, JSC, KSC, LaRC and MSFC. There are plans to connect the remaining 5 NASA centers as project participation expands.

While the DSNet was originally intended for basic connectivity to DSES distributed simulations, its scope has expanded to include additional Constellation projects like the Command, Control, Communication and Information (C3I) Constellation of Labs (CofL) and the Distributed Systems Integration Labs (DSILs).

There are currently plans for the DSNet to provide two categories of service: Facility and Commodity. The Facility category of service will provide Constellation facilities at various NASA centers a lab facility to facility high bandwidth, high reliability connectivity with quality of service guarantees. The Commodity category of service will provide general distributed simulation connectivity between any computer at any facility at any time.

3.2 Network Status and Performance Tools

Unfortunately, it is not enough to just establish connectivity between computers at the geographically dispersed NASA centers. It is also important to have insight into both the current status and performance of the network connections. This is being accomplished through the creation of a collection of DSN status and performance tools.

HLA messaging and Trick simulations were established between JSC and AMES by early 2006. Subsequently, a backup network path through the open Internet was established. The freely available network test software package known as iperf was pressed into regular service to measure network TCP/IP and UDP packet statistics, and showed an initial directional packet loss of 2% on one of the connection links, but more than 30% going the other direction! The DSES team worked with NISN, feeding them performance measurements while NISN tried different solutions. The performance did improve, with packet loss dropping to less than 2% across all links.

At each participating center, the typical template for network architecture resulted in computers behind firewalls whose rules required specific modifications and additions. These actions were necessary to allow the DSES computers they protected to establish sessions and to allow 2-way packet exchanges with all other external DSES nodes.

This proved to be the most time consuming and error-prone step. As LaRC and MSFC joined, it was clear that there was a definite risk that there might be errors introduced by someone at one of the new or established DSES centers. These errors typically result in packets blocked at some firewall, in effect breaking an existing previously proven connection to other nodes on the network. This did happen, resulting in the DSES practice of banning the attempted implementation of firewall rule changes in the weeks leading up to important simulations. Fortunately, backup hosts were able to perform when primary hosts suffered broken connections. AMES work in 2007 involves setup of a more forward end node with fewer firewall rules to contend with, operating initially in parallel with the default original architecture, in order to test proxy server concepts with JSC, latency, and reliability issues.

By this time it was clear that network connectivity and performance tests were routinely required to verify the state of all connections. The testing was automated in order to eliminate the otherwise roughly 2 hours per week of coordinated manual participation involved. By the time of the main DSES demo in September 2006, software had been written by AMES and MSFC to

automate the collection and presentation of jitter and packet loss information on all 6 connection links between the 4 NASA centers. Anyone with a browser and proper permission could access an AMES server and retrieve current network status and performance charts and plots.

There is an ongoing program of testing tool improvement in concert with the exploration of alternative network architectures and simulation executives. The work focuses on more accurate testing, measurement of more parameters, and more intelligent automation. In 2006, the network testing across all links was hourly and used only one procedure each time. In 2007, the network health tools are becoming sensitive to bandwidth issues, and becoming more scalable by running mostly isolated link tests, and by using larger numbers of packets only during certain times; for example, the hours immediately before an integrated simulation.

4. Software Infrastructure

The DSES software infrastructure consists of identifiable collections of software assets located at each of the participating NASA facilities. These software assets include a common distributed simulation software framework. For the initial phases of the DSES project, this framework will be the IEEE 1516 High Level Architecture. However, other distributed simulation software infrastructures are being investigated. Each NASA facility provides simulation components for the distributed simulation using in-house simulation software and systems but rely on the common distribution framework for external interfaces, timing and execution control.

4.1 DSES Federation Object Model (FOM)

Since the DSES project simulations are currently HLA based, they depend on a common description of interoperable data types that are defined in the DSES Federation Object Model (FOM). The DSES FOM is one of the principal development items. It currently consists of a simple set of objects that are used to exchange space vehicle state information like position, velocity and acceleration. The DSES FOM also defines a collection of Interactions that are used to exchange vehicle command and status information.

As the project advances, it is expected that both the state objects and system interactions will expand. The focus will be on identifying generalized abstractions for these with specificity through attribute and parameter values. Ultimately, the DSES FOM will

have to be compatible with and captured in the coordinated and standardized Constellation Project knowledge base called the NASA Exploration Information Ontology Model (NExIOM).

4.2 NASA Simulation Software

Obviously, NASA has been using simulation for many years and simulation is an important tool in almost every aspect of the agencies daily activities. As a result many different simulation tools and systems have been developed. One of the challenges faced by the DSES team was the integration of HLA capabilities into the various tools used by the teams at the various NASA centers.

At ARC, a real time Operating System (OS) known as microTau was developed in house for high fidelity Vertical Motion Simulator activities which currently include Shuttle, and Lunar Lander simulations. The environment allows relatively fast development of, or hook up to, new software models, additional computing hardware and force feedback inceptors. Simulation parameters can be changed on the fly. The overall system involves local Ethernet, SCRAMNET, and XIO memory sharing tools, and interfaces to other protocols and media.

ARC also has an extensive HLA environment bridging 3 facilities, adding different motion simulators, and a 360 degree panoramic control-tower-style visualization and control center to any distributed simulation. Ongoing work includes the addition, to the DSES network, of the Integrated Vehicle Health System (IVHS) and Advanced Diagnostics and Prognostics Testbed (ADAPT) Laboratories.

Over the years, JSC has developed and used a number of simulation tools. Recently, divisions in the Engineering Directorate at JSC have been adopting a common simulation development tool: "The Trick Simulation Development Environment" [7][8][9]. Trick automates many of the more tedious processes in constructing simulations and provides a number of generalized simulation capabilities. This helps to standardize simulation data definition, execution, and post run analysis. In addition to its adoption of Trick as a simulation construction tool, JSC has also been assembling a collection of common models. These models, while hosted in the Trick environment, are not necessarily Trick dependent. This goal is to develop a suite of space systems models that can be shared across projects and facilities. The JSC DSES simulations are built using Trick and models from the common model library.

One of these Trick based simulations is the Advanced NASA Technology Architecture for Exploration Studies (ANTARES). ANTARES is a multi-purpose dynamics analysis simulation supporting the Crew Exploration Vehicle (CEV) Guidance Navigation and Control (GN&C) team. ANTARES supports a wide variety of analyses, including requirements assessments, design trades, GN&C flight software evaluation and test, and pilot in the loop interactions. ANTARES supports computational and parametric studies as well as hardware and crew in the loop facilities. ANTARES is based on a common model library architecture, leveraging off simulation development across several organizations at JSC.

At LaRC, the Flight Simulation and Software Branch has been providing flight simulation products and services since the 1950's including simulation for the Mercury, Gemini, and Apollo missions. Simulation software for modeling, analyses, and system interfaces has evolved throughout the years. Due to the needs to provide customers from the government, industry, and universities with cost-effective and high quality simulation products across multiple simulator facilities among a large pool of developers, the Langley Standard Real-Time Simulation in C++ (LaSRS++) framework was established in the Mid 1990s. This framework is an object-oriented application framework for constructing continuous cyclic simulations. The framework was developed using modern object-oriented programming and design techniques.

LaSRS++ consists of a suite of software libraries that can be shared among developers, projects, and facilities. The framework is intended to provide all of the critical services required to perform simulations and to allow developers to focus only on vehicle model development. The benefits of using the common framework for simulation development include the increase of productivity and ease of maintenance due to high magnitude of framework software reuse, cohesive software practices, and common coding standards for project software development.

The framework supports multiple, heterogeneous models in a simulation. While originally developed to support aircraft simulation, models are not restricted to aircraft and can represent any interactive item. The framework also contains multi-processor support which allows the framework to run N models on M processors. Alternatively a model can be divided across processors. Vehicle models are portable across simulators with little to no modifications to its "hardware interface" code. The simulation software is also portable across IRIX, Linux, Solaris, and Windows operating systems. See references [10]-[19]

for detailed descriptions of the LaSRS++ framework and application software.

The LaSRS++ framework includes various world models such as Earth, Lunar, and Mars atmospheres for simulation. The framework supports real-time human-in-the-loop, hardware-in-the-loop, and parametric simulation and analyses. The framework supports multiple simulation aspects such as Guidance, Navigation, and Controls, vehicle dynamics and behavior, trajectory analysis, vehicle performance assessment, crew displays, and flight deck avionics and layout for part task or end-to-end full mission simulation. The LaRC DSES simulation is a project-specific software derived from the generic spacecraft model of the LaSRS++ framework.

Over the years MSFC has developed the Marshall Aerospace Vehicle Representation in C II (MAVERIC-II) program [20]. This program is designed to more rapidly create flight simulations for spacecraft and launch vehicles. It can be used for all steps of vehicle development — from concept design to actual flight of a finished vehicle. MAVERIC-II simulations can provide detailed predictions of how vehicle designs will actually perform before the craft is built and flown.

The MAVERIC software was developed in-house at Marshall Space Flight Center (MSFC) in Huntsville, Alabama. The vehicle models are totally constructed via user defined inputs for flight modes, vehicle configurations (single stack to shuttle), and user algorithm “plug-ins”. The “plug-ins” range from generic forces to detailed engine and guidance/control algorithms. This makes MAVERIC adaptable to any flight configuration.

MAVERIC simulations have been developed for a Saturn V vehicle, a Shuttle-derived cargo vehicle, a Lockheed-Martin 2-stage-to-orbit launch vehicle concept studied under the SLI program, Orbital Space Plane (OSP) concepts studied by Lockheed-Martin and Boeing, and CEV Launch Vehicle concepts. MAVERIC is also being used for a 6-DOF simulation of the X-37 vehicle trajectory. Some of these vehicle simulations are considered generic (i.e., non-proprietary). The input data files for these generic simulations may be used as templates for building simulations of other vehicle types and configurations.

The current MAVERIC simulation program contains generic models for the (1) propulsion system, (2) the reaction control system (RCS), (3) the vehicle aerodynamic characteristics, (4) the atmosphere (GRAM, US62, US76), (5) mass properties, (6) winds

and (7) algorithms that emulate flight software for guidance and navigation systems. GN&C flight software algorithms have been provided (or are being developed) to accommodate various flight phases such as (1) Ascent (closed-loop and open-loop), (2) Entry, (3) TAEM, (4) Approach, (5) Landing and (6) On-orbit. Additional flight software algorithms to be added later include models for Mission Manager, Propellant Utilization System, sensor models and Performance Monitor. Engine failure abort scenarios can be simulated with MAVERIC. A Mission Manager model will allow the simulation of such scenarios as ground controller commands which can redirect the vehicle to an alternate landing destination or enable/disable certain flight management options.

MAVERIC can be executed in both 3-DOF and 6-DOF simulation modes. In 3-DOF mode the flight control system algorithm is by-passed and the vehicle attitude is set to the attitude command issued by the Guidance System (i.e., a “perfect” control system is assumed). This mode is typically used, for example, when design changes in the guidance algorithms - or new guidance algorithms - are being evaluated. In 3-DOF mode, an engine gimbal moment balance function can be selected to compute engine gimbal angles that will cause the resulting thrust forces to balance the aerodynamic moments. In the DSES federation configuration, MAVERIC is executed in full 6-DOF mode.

4.3 Trick HLA Classes

In addition to modifying existing simulation tools, the team at JSC is creating a generalized set of HLA classes that can be incorporated into a Trick based simulation. These Trick HLA classes take advantage of some of Trick’s generalized IO and variable access capabilities to provide a model that can be included into almost any Trick based simulation to allow it to join into almost any DSES Federation. The mapping of FOM objects, attributes, interactions and parameters are defined in data at run time. In this way, many of the details and intricacies of HLA development will be hidden from the simulation developers.

These classes will be made available to all Constellation simulation developers who are developing Trick based simulations. This will make it much easier for Constellation simulations to provide services in a larger coordinated distributed simulation environment.

5. Simulation Development

While the DSES project is extremely interested in the technological aspects of distributed simulation (network and software infrastructure), the main objective is to develop products (models, simulations and analysis) with meaningful scientific and engineering application to NASA's space exploration initiatives. To accomplish this goal, the DSES project is using an evolutionary development process. In this process the various models and systems required for a working distributed simulation of a space exploration vehicle are developed in phases. The early phase concentrated on technology demonstrations and simulation architecture development. The next phase will demonstrate modeling capabilities, data exchange, model exchange and coordinated execution. The final phase will demonstrate integrated system execution and analysis products.

5.1 DSES Initialization and Startup Process

One of the challenges faced by the DSES team was how best to coordinate the initialization and startup of the collection of federates in any given DSES Federation. It was determined that there was a need for a generic initialization sequence that can be used to create a federate that can:

1. Properly initialize all HLA objects, object instances, interactions, and time management
2. Check for the presence of all federates
3. Coordinate startup with other federates
4. Robustly initialize and share initial object instance data with other federates.

To address these requirements, the following initialization process is used in all DSES federates:

1. Create the federation
2. Publish and subscribe
3. Create object instances
4. Confirm all federates are joined
5. Achieve "initialize" Synchronization Point
6. Update object instance(s) with initial data
7. Wait for object instance reflections
8. Set up time management
9. Achieve "startup" Synchronization Point
10. Start execution

The details of the DSES initialization and startup process are beyond the scope of this paper and are covered in a companion paper [21].

5.2 Dynamics Comparison

One significant requirement for a distributed or collaborative simulation is that the fidelity of the

component simulations be compatible. This requires coordination between the DSES participants to assess and compare selected aspects of the fidelities of their simulation models and environment(s). More specifically, this requires a comparison between DSES participant space environments and space vehicle dynamics.

Initially, it may seem sufficient to have each DSES participant build a 6 Degree of Freedom (DOF) space based simulation, agree on initial conditions, execute the simulation and compare state and environment variable histories. However, the likelihood of getting exact or even numerically equivalent matches between these simulations rapidly approaches zero given the diversity of model implementations and simulation environments. Therefore, a multi-step testing process was developed in order to facilitate a more systematic approach. This allows for the progressive increase in modeling complexity and for the systematic identification and categorization of the sources and sizes of comparative modeling differences.

The test plan for the DSES Phase 2 test consists of following ten principal "unit" comparison test cases and a final fully integrated comparison test case:

- Test Case 1: Earth Modeling Parameters
- Test Case 2: Keplerian Propagation
- Test Case 3: Gravity Modeling
- Test Case 4: Planetary Ephemeris
- Test Case 5: Atmospheric Modeling
- Test Case 6: External Force Affects
- Test Case 7: Combined Translational Test
- Test Case 8: Torque Free Rotation
- Test Case 9: Torque Driven Rotation
- Test Case 10: Gravity Gradient Torque
- Full Test: Integrated 6 DOF Test

Each test case consists of one or more run scenarios. The test cases and their associated scenarios are designed to test specific contributions to the dynamic propagation of a 6 DOF simulated space vehicle. Each test case has essentially the same configuration differentiated by a select parameter or associated parameters.

Like the DSES initialization and startup process, the details of the dynamics comparison process are beyond the scope of this paper and will be covered in a companion paper.

5.3 End-to-End Simulation Capability

The ultimate products for the DSES are simulations and the analysis and training that the simulations can

provide. Specifically, DSES will be providing end-end-to-end simulation capability for all the mission segments in the Constellation program. This includes simulations that range from ground operations, to launch, to orbit, to interplanetary transit, to planetary operations, to Earth return and finally to entry, descent and landing.

5.3.1 Vehicle Elements

In order for DSES to provide simulation support for all Constellation mission scenarios, it will be necessary to provide simulation capabilities for all the elements of the Constellation vehicles listed below:

- International Space Station (ISS)
- Orion - Crew Module (CM)
- Orion - Service Module (SM)
- Orion - Launch Abort System (LAS)
- Ares I- First Stage
- Ares I- Upper Stage
- Ares V - Core Stage
- Ares V - Solid Rocket Boosters (SRBs)
- Earth Departure Stage (EDS)
- Lunar Surface Access Module (LSAM)

Currently, DSES has vehicle models for the ISS, CM, SM, LAS and Ares I First Stage and Ares I Upper stage. These vehicle models vary in their level of fidelity and are continually being modified to track vehicle development and enhanced with improving levels of fidelity. The LSAM, EDS and Ares V elements are currently in development.

5.3.2 Mission Scenarios

Current FY07 development plans for DSES include the following four principal mission segments:

1. Ground Operations
2. Launch and Ascent
3. Orbital Rendezvous and Proximity Operations
4. Entry Descent and Landing.

This will be expanded to include all other mission segments as the Constellation program advances.

The Ground Operations mission segment involves the ground processing of the Orion/Ares launch vehicle in preparation for and up to launch. This capability is not currently integrated into DSES but is included in the FY07 development and integration plans.

The Launch and Ascent segment involves the launch, liftoff, and ascent to orbit of the combined Orion/Ares launch system. This includes launch countdown and

interactions with the Launch Control Center (LCC) as well as the interactions with the Mission Control Center (MCC) and both nominal and contingency launch system ascents. See Figure 4 for a DSES generated view of the Orion/Ares launch vehicle lifting off.

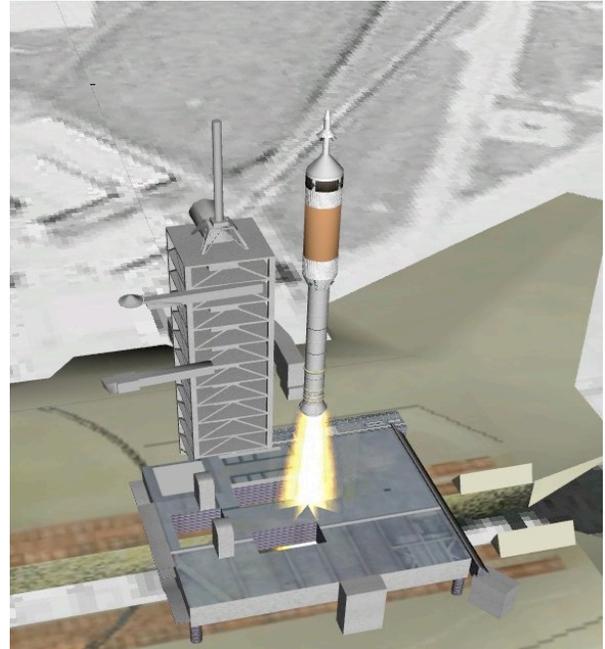


Figure 4: DSES Launch and Ascent Simulation

One of the principal early missions for the Orion spacecraft is to service the ISS. Once Orion is in orbit it will rendezvous with the ISS and then dock. This requires simulation support for rendezvous with and proximity operations around the ISS. See Figures 5 and 6 for a DSES generated view of the Orion spacecraft docking with the ISS.



Figure 5: Orion Docking to ISS



Figure 6: Docking Boresight View

Once the Orion spacecraft has completed its mission at the ISS, it will separate from the ISS and return to Earth. This requires an entry, descent and landing capability. This capability is not currently integrated into DSES but is included in the FY07 development and integration plans.

6. Concluding Remarks

The Distributed Space Exploration Simulation (DSES) will provide an end-to-end simulation capability for the Constellation Program. For FY07, this includes all mission segments for the Orion spacecraft and its service missions to the International Space Station.

The DSES project is a collaborative project that currently includes 5 NASA centers: ARC, JSC, KSC, LaRC and MSFC. DSES utilizes technical domain expertise and resources from each of these centers to effectively support the Constellation Program with its simulation needs. In the process, DSES provides a unifying approach to simulation across NASA and makes efficient use of NASA's simulation development resources.

As the Constellation Program advances, new mission segments will be added. This includes missions to return humans to the Moon (See Figure 7 for a DSES generated view of the Orion/LSAM in orbit about the Moon). Ultimately, the Constellation Program will extend human presence beyond the local Earth/Moon system to the planetary neighbor Mars. As these and other mission segments evolve, DSES will evolve to provide the high fidelity mission simulation needs for the Constellation Program.



Figure 7: Orion/LSAM in Lunar Orbit

7. References

- [1] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) — Framework and Rules", The Institute of Electrical and Electronics Engineers, 3 Park Avenue, New York, NY 10016-5997, USA, 11 December 2000, ISBN 0-7381-2620-9 SS94882.
- [2] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Federate Interface Specification", The Institute of Electrical and Electronics Engineers, 3 Park Avenue, New York, NY 10016-5997, USA, 9 March 2001, ISBN 0-7381-2622-5 SS94883.
- [3] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Object Model Template (OMT) Specification", The Institute of Electrical and Electronics Engineers, 3 Park Avenue, New York, NY 10016-5997, USA, 9 March 2001, ISBN 0-7381-2624-1 SS94884.
- [4] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, "IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP)", The Institute of Electrical and Electronics Engineers, 3 Park Avenue, New York, NY 10016-5997, USA, 23 April 2003, ISBN 0-7381-3585-2 SS95088.
- [5] G. Lauderdale, E. Crues, D. Snyder, D. Hasan, "A Feasibility Study for ISS and HTV Distributed Simulation," AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas, 11-14 August 2003.

- [6] G. Lauderdale, E. Crues, D. Snyder, D. Hasan, "Further Studies On The Feasibility Of A Distributed ISS and HTV Simulation," Proceedings of the Fall 2003 Simulation Interoperability Workshop and Conference, Orlando, Florida, 15-18 September 2003.
- [7] K. Vetter, G. Hua "The Trick User's Guide 2005.7.0 Release," NASA Technical Publication, NASA Johnson Space Center, June 2006.
- [8] K. Vetter, "Trick Simulation Environment – User Training Material 2005.7.0 Release," NASA Technical Publication, NASA Johnson Space Center, June 2006.
- [9] E. J. Paddock, A. Lin, K. Vetter, E. Z. and Crues, "Trick: A Simulation Development Toolkit", AIAA Modeling and Simulation Technologies Conference and Exhibit, AIAA 2003-5809, Austin, TX, 2003.
- [10] Madden, M. M. Architecting a Simulation Framework for Model Rehosting. AIAA Modeling and Simulation Technologies Conference, Providence, Rhode Island, August 16-19, 2004, AIAA 2004-4924.
- [11] Neuhaus, J. Modeling Mass Properties in an Object-Oriented Simulation. AIAA Modeling and Simulation Technologies Conference, Providence, Rhode Island, August 16-19, 2004, AIAA 2004-5166.
- [12] Madden, M. M., and Neuhaus, J. A Design for Composing and Extending Vehicle Models. AIAA Modeling and Simulation Technologies Conference, Austin, Texas, August 11-14, 2003, AIAA-2003-5458.
- [13] Kenney, P Sean, Simulating the ARES Aircraft in the Mars Environment. AIAA Modeling and Simulation Technologies Conference, Austin, Texas, August 11-14, 2003, AIAA-2003-6579.
- [14] Madden, M. M. Examining Reuse in LaSRS++-Based Projects. AIAA Modeling and Simulation Technologies Conference, Montreal, Canada, Aug. 6-9, 2001, AIAA-2001-4119.
- [15] Sugden, P. C.; Rau, M. A.; and Kenney, P. S. Platform-Independence and Scheduling in a Multi-Threaded Real-Time Simulation. AIAA Modeling and Simulation Technologies Conference, Montreal, Canada, Aug. 6-9, 2001, AIAA-2001-4244.
- [16] Neuhaus, J. An Object-Oriented Sensor and Sensor System Design. AIAA Modeling and Simulation Technologies Conference, Montreal, Canada, Aug. 6-9, 2001, AIAA-2001-4123.
- [17] Madden, M. M. An Object-Oriented Interface for Simulink Models. AIAA Modeling and Simulation Technologies Conference, Denver, Colorado, Aug. 11-14, 2000, AIAA-2000-4391.
- [18] Madden, M. M.; Glaab, P. C.; Cunningham, K.; Leslie, R. A.; Kenney, P. S.; and Geyer, D. W.: Constructing a Multiple-Vehicle, Multiple-CPU Simulation Using Object-Oriented. AIAA Modeling and Simulation Technologies Conference, Boston, Massachusetts, Aug. 9-11, 1998, AIAA-98-4530.
- [19] Leslie, R. A.; Geyer, D. W.; Cunningham, K.; Glaab, P. C.; Kenney, P. S.; and Madden, M. M.: LaSRS++ - An Object-Oriented Framework for Real-Time Simulation of Aircraft. AIAA Modeling and Simulation Technologies Conference, Boston, Massachusetts, Aug. 9-11, 1998, AIAA-98-4529.
- [20] A. Hill, J. Compton, J. McCarter, "MAVERIC Release 2.3 User's Guide" NASA Technical Publication, NASA Marshall Space Flight Center November 2005.
- [21] R. Phillips, D. Dexter, E. Crues, "A Coordinated Initialization Process for the Distributed Space Exploration Simulation (DSES)," Proceedings of the Spring 2007 Simulation Interoperability Workshop and Conference, Norfolk, Virginia, 25-30 March 2007.

Acknowledgments

The authors would like to thank Don Monell and Bill Othon for their interest and support in distributed simulation in general and the DSES project specifically. The DSES project would not exist without them.

The authors would also like to thank the many talented scientists and engineers who contributed to the DSES project: Cathy Alofs, Paul Bielski, Mark Coffman, Chris Daum, Dan Dexter, Jim Gibson, Joe Hawkins, Missy Hill, Sean Kenny, Jason Neuhaus, Michael Madden, Marty Pethtel, Russ Sexton, Paul Sugden, and Jesse Trammell. These are the people who really made DSES work.

Author Biographies

EDWIN Z. CRUES, PH.D. has supported the Automation, Robotics and Simulation Division at NASA Johnson Space Center for the past 14 years. He has been a member of the Simulation and Graphics Branch, since 2004, where he leads the research and development of distributed simulations technologies. In this capacity, he leads the development of the HTV Flight Controller Training (FCT) and the NASA Distributed Space Exploration Simulation (DSES). The DSES work is in support of the Modeling and Simulation and Data Architectures (MS&DA) group

for the Constellation program. Dr. Crues also supports dynamics model development for the Trick Simulation Environment and the Common Model set.

MIKE BLUM, M.A.SC, has 20 years experience in simulation hardware and software R&D, developing simulation applications, Real Time Operating Systems, compilers, and automation, vibration and vehicle control systems. He has been with NASA AMES Research Center SimLabs for the last 10 years, where he works for SAIC as the Principal Engineer of Simulation Development.

VICTORIA CHUNG has supported the flight simulation engineering and development functions at the NASA Langley Research Center for the past 17 years. She has been serving as the Software Group Lead in the Flight Simulation and Software Branch since 1999 and is currently the Assistant Branch Head of the branch. Ms. Chung is a Life-time Senior Member of the American Institute of Aeronautics and Astronautics (AIAA). She serves as a member of the AIAA Modeling and Simulation Technical Committee.

DAN BOWMAN is a Senior Systems Analyst at Teledyne Brown Engineering. He supports NASA's Modeling and Simulation / Data Architecture Office for the Constellation program in the integrated management of models and simulations. He leads the adaptation of the MAVERIC simulation to serve as the CLV federate in the HLA-based DSES configuration. He has over twenty years of experience in general aerospace modeling and simulation activities, in addition to real-time hardware-in-the-loop testing.