CS 582 Modeling and Simulation II
Spring 2013

Interoperability / SimSmackdown

Final Report

May 1, 2013

Authors	
John Bland	Bill Tucker (team leader)
Justin Brown	Bingyang Wei
Jonathan Patrick	Tymaine Whitaker
Melissa Redmond	

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THIS DOCUMENT WAS PREPARED BY THE STUDENTS AS A PART OF A UAHUNTSVILLE CLASS PROJECT. THE REPORT IS AN ACCOUNT. OF WORK PERFORMED DURING THE SIMSAMCKDOWN PROJECT. THE CONTENT IS THE SOLE RESPONSILITY OF THE STUDENTS AND NO EXPLICIT OR IMPLIED WARRANTY OF THE ACCURACY OR FITNESS FOR ANY USE IS PROVIDED

NEITHER UAHUNTSVILLE NOR ANY OF THE COMMERCIAL OR PRIVATE SPONSORS:

(A) MAKE ANY WARRANTY OR REPRESENTATION WHATSOEVER, EXPRESS OR IMPLIED, (I) WITH RESPECT TO THE USE OF ANY INFORMATION, APPARATUS, METHOD, PROCESS, OR SIMILAR ITEM DISCLOSED IN THIS DOCUMENT, INCLUDING MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR (II) THAT SUCH USE DOES NOT INFRINGE ON OR INTERFERE WITH PRIVATELY OWNED RIGHTS, INCLUDING ANY PARTY'S INTELLECTUAL PROPERTY, OR (III) THAT THIS DOCUMENT IS SUITABLE TO ANY PARTICULAR USER'S CIRCUMSTANCE; OR

(B) ASSUME RESPONSIBILITY FOR ANY DAMAGES OR OTHER LIABILITY WHATSOEVER (INCLUDING ANY CONSEQUENTIAL DAMAGES RESULTING FROM ANY USERS SELECTION OR USE OF THIS DOCUMENT OR ANY INFORMATION, APPARATUS, METHOD, PROCESS, OR SIMILAR ITEM DISCLOSED IN THIS DOCUMENT.

(C) NECESSARILY ENDORSE ANY SPECIFIC COMMERCIAL PRODUCT, PROCESS, OR SERVICE BY ITS TRADE NAME, TRADEMARK, MANUFACTURER, OR OTHERWISE, REFERENCE HEREIN. NOR DOES SUCH MENTION TO NECESSARILY CONSTITUTE OR IMPLY ITS ENDORSEMENT, RECOMMENDATION, OR FAVORING

ACKNOWLEDGMENTS

The organization listed below in alphabetical order provided significant support to the UAHuntsville SimSmackdown effort and their contributions are gratefully acknowledged. In particular the tech support provided by all formed a significant part of the learning experience. Excellent raining materiel and help files were provided by all of the vendors, but Pitch is recognized for providing an introductory short course that got us all started.

Organization	Key Contributions
AEgis Technologies Group	Travel grant, ACSLx licenses, tech support. Graphics support
Analytic Graphics Inc	System Tool Kit licenses, technical support
ForwardSim	HLA toolkit licenses for use during competition and as prizes. Tech support
Johnson Space Flight Center, NASA	Distributed simulation network and a great deal of technical support
Kennedy Space Flight Center, NASA	The Unity Tool (used for visualization) and tech support. 3D visual models. Overall project leadership.
Marshall Space Flight Center, NASA	Technical Team leadership
MäK	Licenses for Real Time Interface and associated software. Tech Support
Pitch	Real Time Interface licenses and Tech Support. HLA startup kit and training materials
Simulation Interoperability Standards Organization (SISO)	Venue, A/V support, network access, and participant registration
Society for Modeling and Simulation, International (SCS)	Venue, A/V support, network access, and participant registration

ABSTRACT

The report describes work performed during the spring 2013 semester by the members of Dr. Mikel Petty's MS 582 – Modeling and Simulation II – class, working in close conjunction with other universities and with great support from NASA and the other event co-sponsors to plan and execute the 3013 SimSmackdown event. The actual event was held in San Francisco on 9 April, hosted by the joint spring conference of the Simulation Interoperability Standards Organization (SISO) and the Society for Modeling and Simulation, International (SCS).

SimSmackdown provided a unique and challenging, opportunity for the students at the various schools from universities around the world to jointly develop a complex distributed simulation of future missions to the moon, Lagrange point II, and Earth   The achievement required a great deal of commitment from the participating students and significant support from the co-sponsors but resulted in a highly rewarding educational experience.

Achieving the goal required the students to learn about distributed simulation technology, complex project planning, how to use a variety of sophisticated commercial and free software tools, all in order to develop and integrate a set of simulations.

This report describes how the UAHuntsville team achieved that goal. Because the event was strongly cooperative, the scope of the report necessarily includes some aspects of the work of the other participants and the co-sponsors.

Things that went especially well, and some things that did not go so well are discussed, and and a set of recommendations are provided about how the event can be improved in the future. They based on data collected through a series of non-attribution interviews conducted with 15 participants, leaders and supporters.

Appendices contain the team's project plan and group charters.

UAH Huntsville Participants	
Dr. Mikel Petty	CS 585 Professor and faculty sponsor
Dan Oneil	team technical advisor to the team and Smackdown technical team leader
CS 585 Students	
John Bland	Bill Tucker (team leader)
Justin Brown	Bingyang Wei
Jonathan Patrick	Tymaine Whitaker
Melissa Redmond	

Contents

LIST OF TABLES AND FIGURES

1 APPROACH

The project began with the development of the learning objectives for the course as listed in Figure 1-1. The team believes that we have achieved these goals as evidenced by this report

Goal: As demonstrated by successful execution of a distributed simulation and delivery of a statement of work, project plan, and final report show that we have learned about

1. The space domain including:
1.1. Space vehicle dynamics
1.2. Orbital dynamics
1.3. Space exploration concepts
1.4. Interplanetary logistics
2. Distributed simulation protocol and implementation
3. Working with distributed teams on integrating complex systems
4. Providing an interesting, convincing, and perhaps entertaining visualization of the complex simulation.
5. Planning and leadership of complex projects

Figure 1-1 Course Objectives

These objectives along with other planning details were incorporated into a project management plan. The students volunteered to participate in separate teams and each team developed a team charter to describe their work. To accomplish the first object, Dr Petty conducted a series of lectures to improve the student's understanding of distributed simulation in general, and HLA in particular. Also, invited lecturers came to speak on key technical topics. The project development environment, including computers, licensed commercial software products, a configuration management system, and a web site. Program planning documents including a program management plan and team charters were developed during this timeframe. The federates, i.e. the individual simulations to be integrated in the HLA simulation were developed in an iterative fashion, followed by integrated tested, and then by the conduct of the actual event. After the event participants, sponsors, and supporting staff were interviewed in a non-attribution process to collect lessons learned. Recommendations were developed on how the enterprise could improve future events, and this report was prepared. Figure 1-2 and the report outline roughly follow the life cycle of the project.

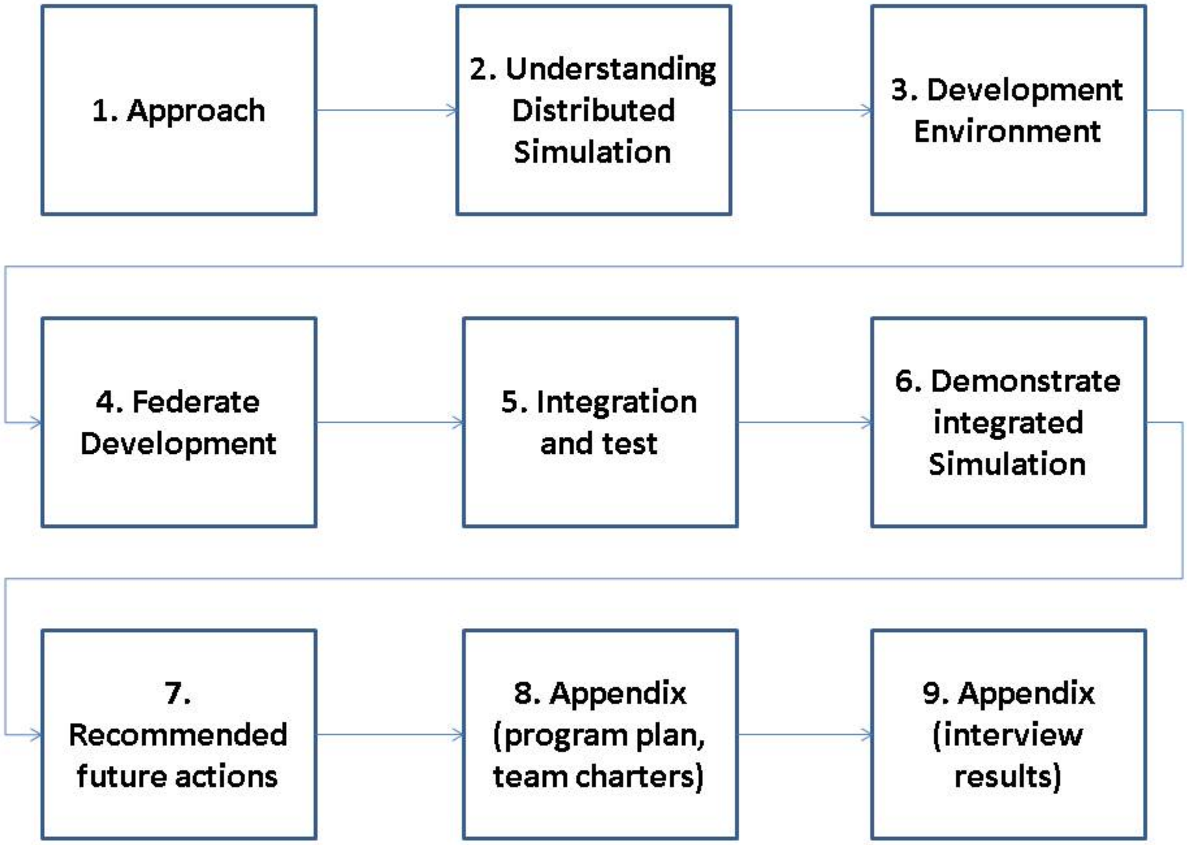

1. Approach
2. Understanding Distributed Simulation
3. Development Environment
4. Federate Development
5. Integration and test
6. Demonstrate integrated Simulation
7. Recommended future actions
8. Appendix (program plan, team charters)
9. Appendix (interview results)

Figure 1-2 Report Organization

2 IMPROVE UNDERSTANDING OF DISTRIBUTED SIMULATION

.The Students enrolled in the class came from widely differing backgrounds. Some had experience related to distributed simulation, but none done actual hands on development of distributed simulations, and none had experience working with a geographically distributed set of fellow students and collogues. To prepare the team for tis experience, a variety of training opportunities were provided, and the students were tested on the knowledge they gained from the lectures.

Event	Type	Provider
HLA Overview	**Lecture (multiple)**	**Dr. Petty**
Time Management	**Lecture**	**Dr. Petty**
Environment Federate	**Lecture**	**Dr. Cruz**
Coordinate Transformations	**Lecture**	**Andy…**
Introduction to HLA	**Lab Exercise**	**Pitch**
Network Operations & Interface descriptions	**Reference Documents**	**NASA**
On Line Training	**Videos, Briefings**	**AGI, ForwardSim, Matlab, Pitch, MäK**
On Call Technical Support (product)	**Email and phone**	**AGI, ForwardSim, Pitch, MäK**
On Call Technical Support (space domain and network)	**Email and phone**	**NASA KFC**

Table 2-1 Training Opportunities

The other forms of available training were less formal, and knowledge of their availability was limited. Compared to other universities, UAHuntsville had a significant advantage in that, since the project was organized as a class, relevant training part of the plan. We also had the opportunity to invite special lectures on topics we found especially difficult.

Among other things, the available training rightly led the team to focus on the ability of the models to interact in the HLA environment. Model complexity was limited to that necessary for the mission. Other teams focused their initial efforts on modeling, leaving them not ready for the early and frequent testing that is so critical to success in a project of this type.

3 ESTABLISH DEVELOPMENT ENVIRONMENT AND PLAN

The first requirement for setting up a local test environment was to establish a working lab with the hardware and software needed for each federate along with this there was a challenge acquiring the needed licenses required for the software. Once this was established we were then tasked with setting up the virtual private network (VPN) connection to the NASA provided test environment which included both MAK and Pitch RTIs. Once all of this was complete the teams were able to test their federates using both RTIs. Figure 3.1 describes the initial test and integration network. The network used in the actual event was significantly
different:

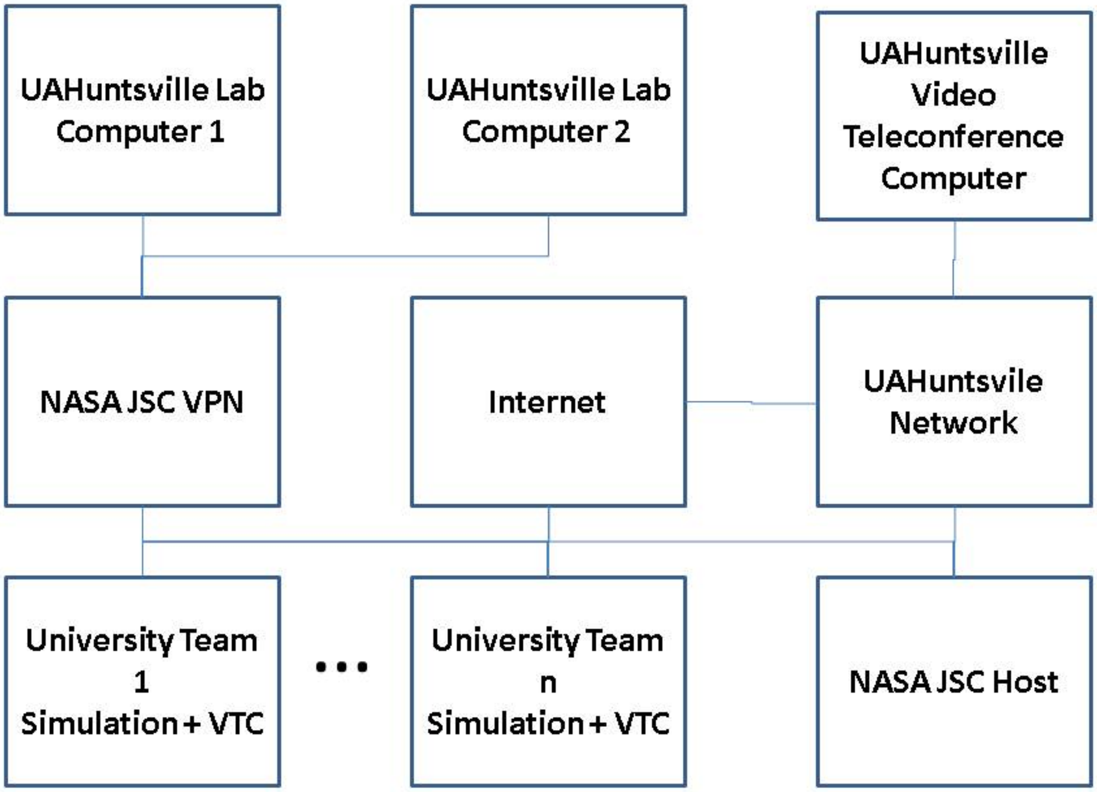

UAHuntsville Lab Computer 1
UAHuntsville Lab Computer 2
UAHuntsville Video Teleconference Computer
NASA JSC VPN
Internet
UAHuntsvile Network
University Team 1 Simulation + VTC
• • •
University Team n Simulation + VTC
NASA JSC Host

Figure 3-1 Integration & Test network

. Note that the lab computers when connected to the VPN had no access to other university computers or to the public internet, and that a separate computer and network connection was required for the computer used to communicate with other participants. The VPN software did not work well when computers were connected to it via any wireless network since any dropout and reconnection used up another license from their pool, eventually making it impossible for anyone to connect to the network. It was necessary to use a VPN to control access to the NASA computer resources and to protect the network especially since all connected computers were required to run with their firewalls turned off. Only two wired network connections were availble to the UAHuntsvile Smackdown lab. Table 3-1 describes some consequnces of this architecture

Feature	Challenges presented
Use of VPN for testing	Very difficult to update software and data during testing
Limited number of network drops	Constant network configuration was required to conduct required test-fix-test operations as 7 students contended for two drops University wireless security rules frequently denied access to students, making it an unreliable backup connection – possibly due to the special security settings required by the VPN Network access challenges led to very frequent computing and network changes, making software version control very difficult
Requirement for firewalls to be off during VPN testing	Disconnecting from the VPN and connecting to the public internet was risky because of the probability that people would not turn their firewalls back on. Computing and network challenges, especially combined with the difficulty of shipping desktop computers led to students using their personal laptops without firewall protection

Table 3-1 Laboratory Test and Integration Environment Challenges

For the Pitch RTI, we only needed the trial version of the Pitch software distributed from http://www.pitch.se for testing purposes. This was because Johnson Space Center (JSC) provided the licensed Pitch RTI over a Virtual Private Network (VPN) which did not have a bound on the number of federates it could run, and thus we were able to connect to their RTI and not worry about the bounds on our trial version RTIs.
For the MAK RTI, each computer had to have a license that was tied to the machine's MAC address. Again, JSC was running a MAK RTI on a VPN, but for the MAK software a license was required to connect to JSC's RTI.

Licenses were also required for the ForwardSim HLA Toolkit for those developers that used it – primarily the team working on the visualization federate.

The lunar communication federate used the free version of Analytic Graphics System Tool Kit (STK), so no additional license was needed. The team developed a connection between the STK application programmers interface (API) and the HLA standard interface using JAVA developed in the Eclipse environment.

A full license for STK was obtained after some delay. STK was interfaced with Matlab to evaluate the degree of coverage provided by various lunar orbits to help pick the right orbits for the satellites in the communication federate

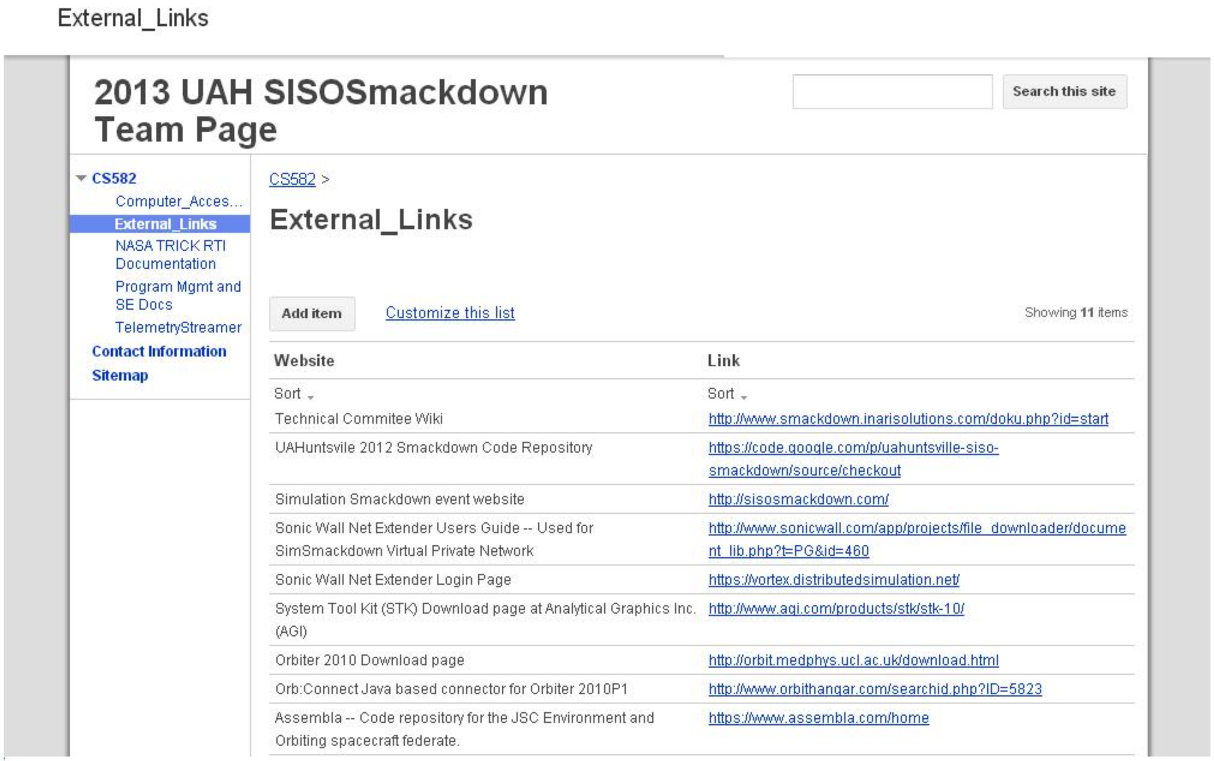
.

For applications written in C++, we used Microsoft Visual Studio 2010 to ensure compatibility with the libraries for Pitch and MäK which were compiled with 2010

Since the federate used the Orbiter software (orbit.medphys.ucl.ac.uk/) we also used Orbiter 2010 for development, along with the LMD scenario.

The NASAS provided Unity 3D game engine based executable used for the visualization federate, was named The Unity Tool (TUT). It was developed by Kennedy Space Center. The 2012 visualization federate was written in Matlab, primarily developed by ForwardSim. It used Virtual Reality Modeling Language (VRML) to create the images. This approach required access to Matlab toolkits that were not available to the 2013 team, so a different approach was used in 2013 and the VRML code was removed. The 2013 team developed a C++ program to stream data into the unity tool. The interface between the streaming program and the HLA interface was written in Matlab, using the ForwardSim HLA toolbox.

The the team utilized Google Sites for sharing general information as well as Google Project Hosting as a code repository. The external links page form the site is shown in figure 3-2.



Website	Link
Sort	Sort
Technical Commitee Wiki	http://www.smackdown.inarisolutions.com/doku.php?id=start
UAHuntsvile 2012 Smackdown Code Repository	https://code.google.com/p/uahuntsville-siso-smackdown/source/checkout
Simulation Smackdown event website	http://sisosmackdown.com/
Sonic Wall Net Extender Users Guide -- Used for SimSmackdown Virtual Private Network	http://www.sonicwall.com/app/projects/file_downloader/document_lib.php?t=PG&id=460
Sonic Wall Net Extender Login Page	https://vortex.distributedsimulation.net/
System Tool Kit (STK) Download page at Analytical Graphics Inc. (AGI)	http://www.agi.com/products/stk/stk-10/
Orbiter 2010 Download page	http://orbit.medphys.ucl.ac.uk/download.html
Orb:Connect Java based connector for Orbiter 2010P1	http://www.orbithangar.com/searchid.php?ID=5823
Assembla -- Code repository for the JSC Environment and Orbiting spacecraft federate.	https://www.assembla.com/home

Figure 3-2 Google based team website

4 DEVELOP FEDERATES

Figure 4-1 provides an overview of the scenario, describing how the simulations created by the University teams fit together into an integrated simulation of a future lunar mission.

To support this overall scenario, the 2013 UAHuntsville team developed four federates:

1. Lunar Communication System and Satellites Constellation federate (LCANSat2013);
2. Lunar Mass Driver federate (LMD);
3. 3D lunar surface visualization federate (fsi3DViewer-UAH);
4. Test driver federate.

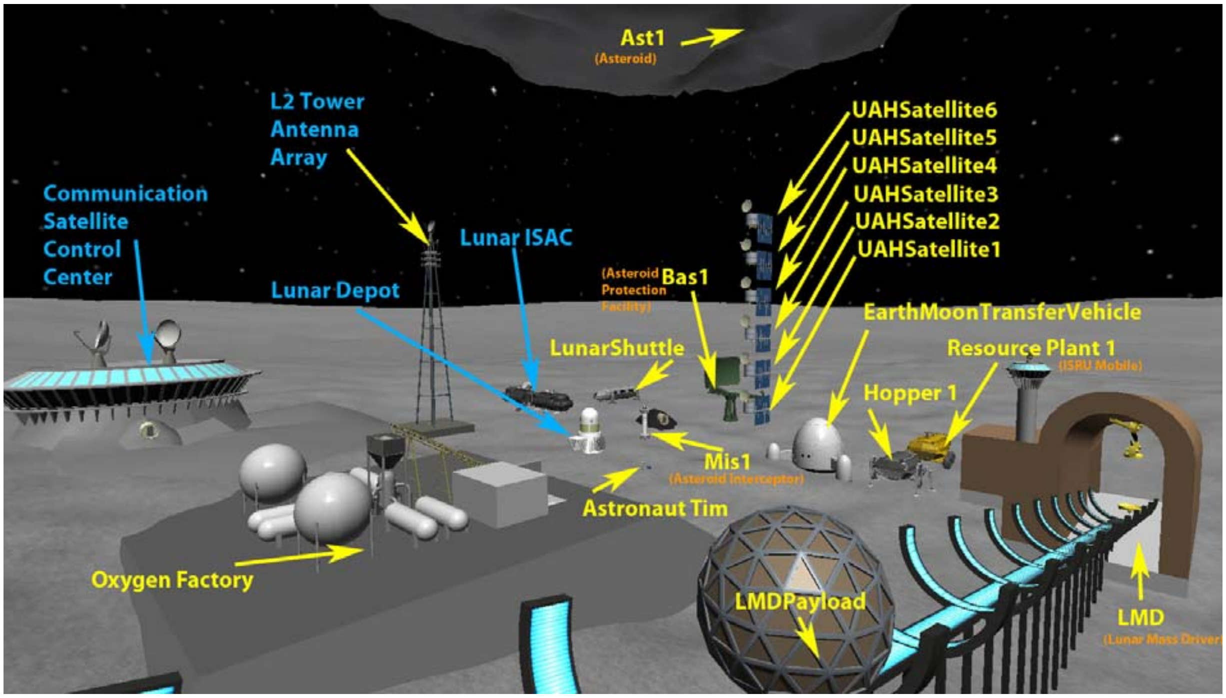


Figure 4-1 Operational context of Smackdown federates

4.1 LCANSat2013 federate

The 2013 UAHuntsville team decided to develop a more compact and comprehensive federate by integrating LCANServ, LCANSat, 3D satellite constellation visualization federate and part of the lunar surface visualization federate. The name of the new federate is Lunar Communication System and Satellites Constellation federate 2013. We use the acronym name LCANSat2013 to denote our new federate.

This federate was largely based upon the four federates developed by the 2012 UAHuntsville Smackdown team: Communication network federate (LCANServ) which handles all the interaction messages between different federates in the Smackdown

Federation, Satellite federate (LCANSat) for propagating a 4-satellite constellation, 3D satellite constellation visualization federate and 3D lunar surface visualization federate.

This new LCANSat2013 federate provides the following functions in the lunar resupply mission:

1. Propagation of constellations of satellites orbiting the moon so that enough coverage of Aitken Basin is achieved.
2. Inter-communication capabilities between physical entities in the mission.
3. Visualization of satellites constellation in 3D and all the other physical entities in 2D.

In order to fulfill the challenging requirements of our LCANSat2013 federate, our development of the federate needs to rely on robust and free software development kits. By investigating and researching different software, we found that the software applications and development kits offered by Analytical Graphics, Inc. (AGI) is the best choice for modeling, simulating and analyzing the Smackdown space mission.

4.1.1 Overview of LCANSat2013 federate:

The LCANSat2013 federate is an aerospace Java application built on top of AGI's proven Systems Tool Kit(STK) Software Development Kits: the STK Components for Java SDK. STK Components is a family of low-level class libraries that provide access to specific analytical and 3D visualization capabilities for space exploration.

At the core of STK Components is the Dynamic Geometry Library (DGL). DGL provides various specialized propagators for modeling the motion of satellites, aircraft, and surface vehicles. Additionally, algorithms like inter-visibility between physical entities given a number of complex, simultaneous constraining conditions in the space are also provided so that the line-of-sight(LOS) information critical in our lunar communication system can be obtained easily. Since DGL adopts the industry recognized time and position which is consistent with that of NASA's environment federate, it makes the positioning of other physical entities in federation accurate. The Insight3D Visualization Library which is a part of DGL allows us to build lightweight yet powerful 3D applications for performance and visualization accuracy of the lunar mission.

Other relevant capabilities provided by DGL:

1. Accurate moon modeling;
2. Precise platform positioning and orientation modeling for orbit and waypoint propagation;
3. Complex numerical and geometry algorithms;
4. Manipulate points, axes, vectors, and reference frames.

All the capabilities of STK Components make our LCANSat2013 federate possible. In the following two subsections, two main modules that constitute LCANSat2013 federate: Satellite Constellation & Lunar Visualization module and Communication Server module are illustrated in detail.

4.1.2 Satellite Constellation & Lunar Visualization module

The Satellite Constellation & Lunar Visualization module propagates and renders two different satellites constellations for the 2013 SISO Smackdown. The two satellites constellations are tested against PITCH Runtime Infrastructure (RTI) and MAK Runtime Infrastructure (RTI) respectively.

We use STK's numerical propagator to generate orbits of satellites in LCANSat2013. Our two constellations of satellites are achieved by STK's validated and verified High Precision Orbit Propagator (HPOP) capability provided by the Orbit Propagation Library (OPL). By numerically integrating the various forces affecting satellites, OPL brings high fidelity orbit propagation into our LCANSat2013.

When the whole federation is running on PITCH RTI, a 6-satellite constellation is designed which could provide a lot of hang time both over Aitken Basin and Hadley Rille to maximize the time each satellite is in view of the surface entities as shown in figure 4-2.

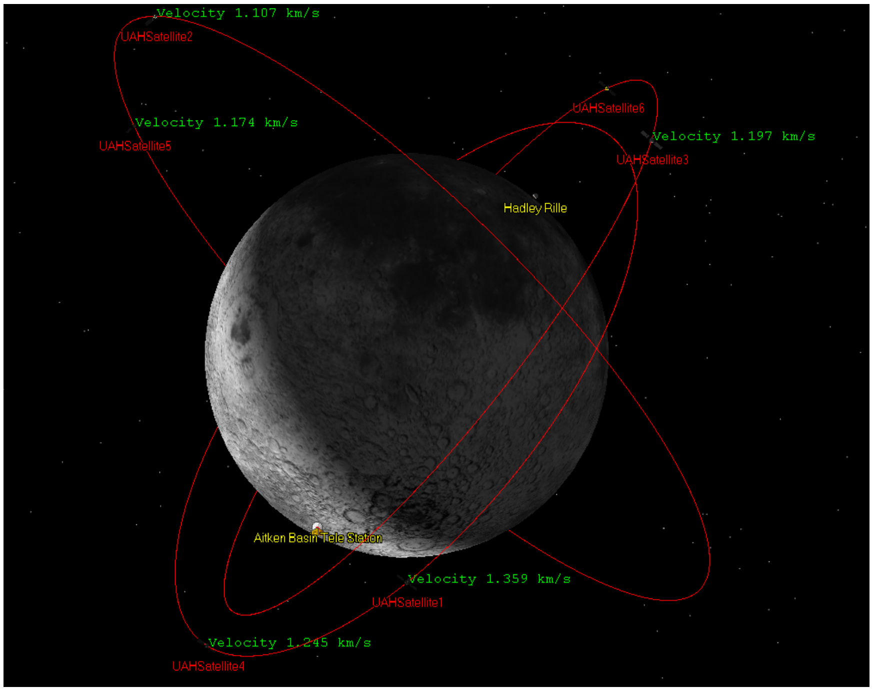


Figure 4-2 Satellite constellation 1

When the federation is working with MAK, we designed a 14-satellite Walker Constellation (Figure 4.3) that provided 100% communication coverage of both the Aitken Basin and Hadley Rille and coverage of 100% of the lunar surface at least 90% of the time. Both constellations provide a full time relay capability to allow communications between the Earth and the L2 outpost. Since the L2 point is on a line from the Earth to the moon, but on the opposite side of the moon, line of sight communication between them is not possible without some kind of assistance.

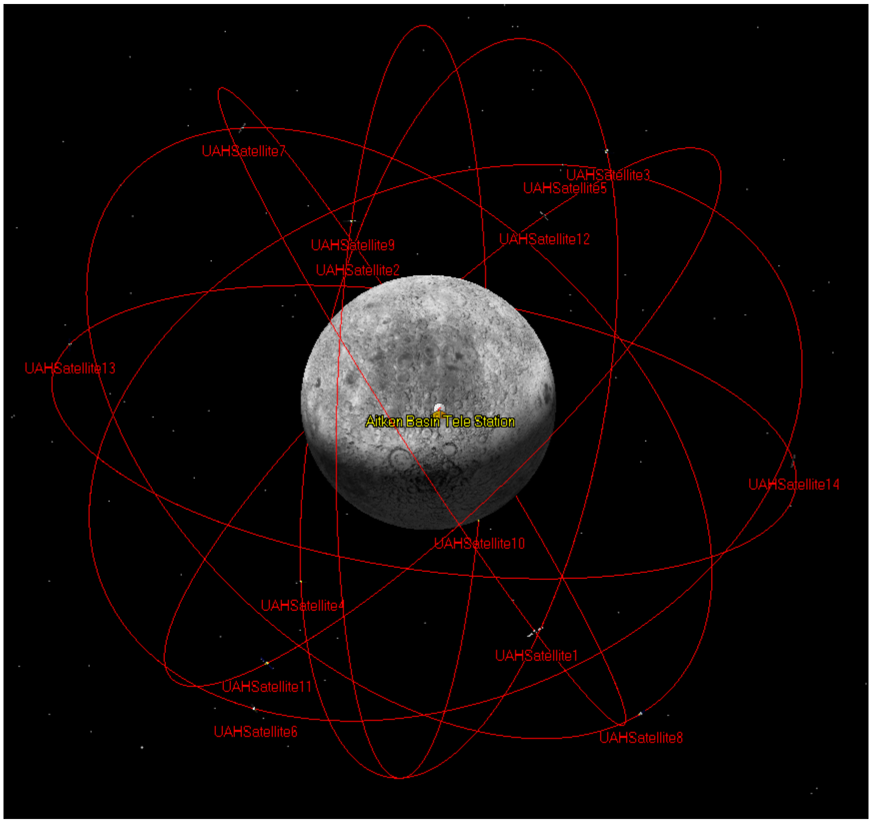


Figure 4-3 Satellite constellation 2(Walker Constellation)

The architecture of the Satellite Constellation & Lunar Visualization Module is shown in, Figure 4-4, a UML class diagram.

An instance of TheFederate class that inherits NullFederateAmbassador builds up the LCANSat2013 federate and connects it to the RTI host in Johnson Space Center (JSC) by instantiating a Connection object. Objects crucial for using HLA services like RTI ambassador, encoder factory are created and the whole federate is time constrained during the construction of federate. The TheFederate instance is then assigned to Satellite objects so that they can invoke the services provided by HLA RTI. TheFederate class also contains all the definitions of HLA callback methods: timeConstrainedEnabled, timeAdvanceGrant, removeObjectInstance, provideAttributeValueUpdate, discoverObjectInstance, reflectAttributeValues and receiveInteraction
In the Satellite module, satellites are modeled as a member of the PhysicalEntity object class defined in the SISO_Smackdown_2013_entity FOM. Once LCASat2013 federate was granted time advance by Environment federate, each satellite published its logical

time, entity name, reference frame and most up-to-date Cartesian coordinate position. Other federates which were interested in obtaining the attributes of satellites could subscribe to and receive the updates sent by the LCANSat2013 federate.

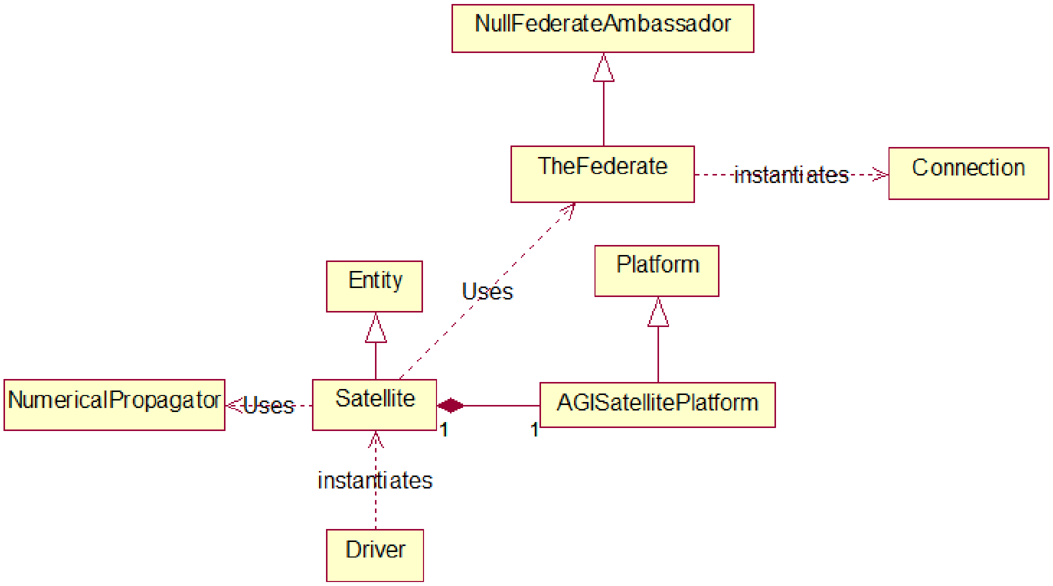


Figure 4-4 Satellite Constellation & Lunar Visualization Module Architecture

Regarding the propagation of orbits, each satellite uses a numerical propagator provided by STK to propagate its orbit. STK Components increases the fidelity and accuracy of its simulation of orbits by providing a variety of environment and force models. In our case, spherical harmonic gravity of the moon and solar radiation force are considered during the propagation of a satellite’s orbit. The gravity model of the moon is read in from *LunarGravityField_LP100K.grv* file. It is then used to construct the immutable field by selecting to the desired fidelity in the degree and order of the represented field, as well as configuring other options such as the inclusion of tidal data. This field is then used to define the force at a given position. In the end, those force models together with six orbital elements which are the parameters required to uniquely identify a specific orbit: semi major axis, eccentricity, inclination, argument of perigee, right ascension of the ascending node (RAAN) and true anomaly are used to propagate the position, orientation, and other attributes of the satellite over time using the numerical propagator which results very good and accurate orbits.
In our effort to visualize satellites and their orbits in STK’s Insight3D viewer, we adopted STK Components' Platform type which can be used to model satellites, facilities, aircraft, and other "real-world" objects. Simply put, an AGISatellitePlatform object is created for each Satellite object which stores: the name of the satellite, a time-varying position and orientation calculated by numerical propagator. By adding the AGISatellitePlatform into the Insight3D viewer, the visualization of satellites is accomplished.

The second important module in LCANSat2013 makes communication in the mission possible.

4.1.3 Communication Server module

The communication server module is used to facilitate lunar communications among different federates. Here are some potential scenarios where our communication server might be used:

The scouting hopper developed by MIT team detects some valuable regolith in one place on the moon surface and needs to send a message regarding the position of the regolith to the mobile resource utilization plant; Genoa's space guard system which tracks the approaching asteroid near the moon broadcasts the warning message to all the federates in the Smackdown federation. Penn State University team's lunar shuttle needs to send messages to the landing site on the moon surface reporting its position.

Messages in the communication system are defined as HLA interactions with transmit (TX) and receive (RX) interactions (Figure 4-5).

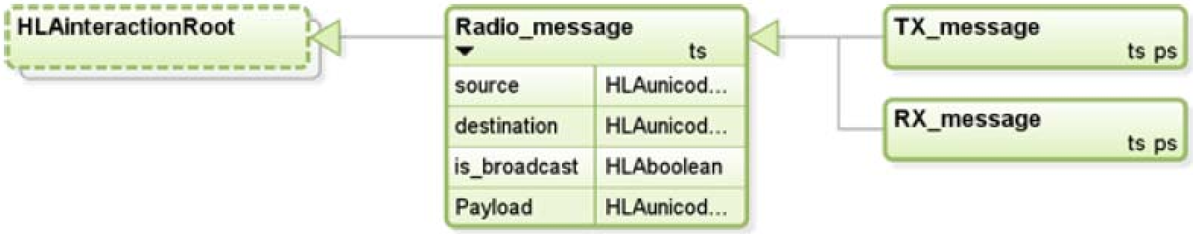
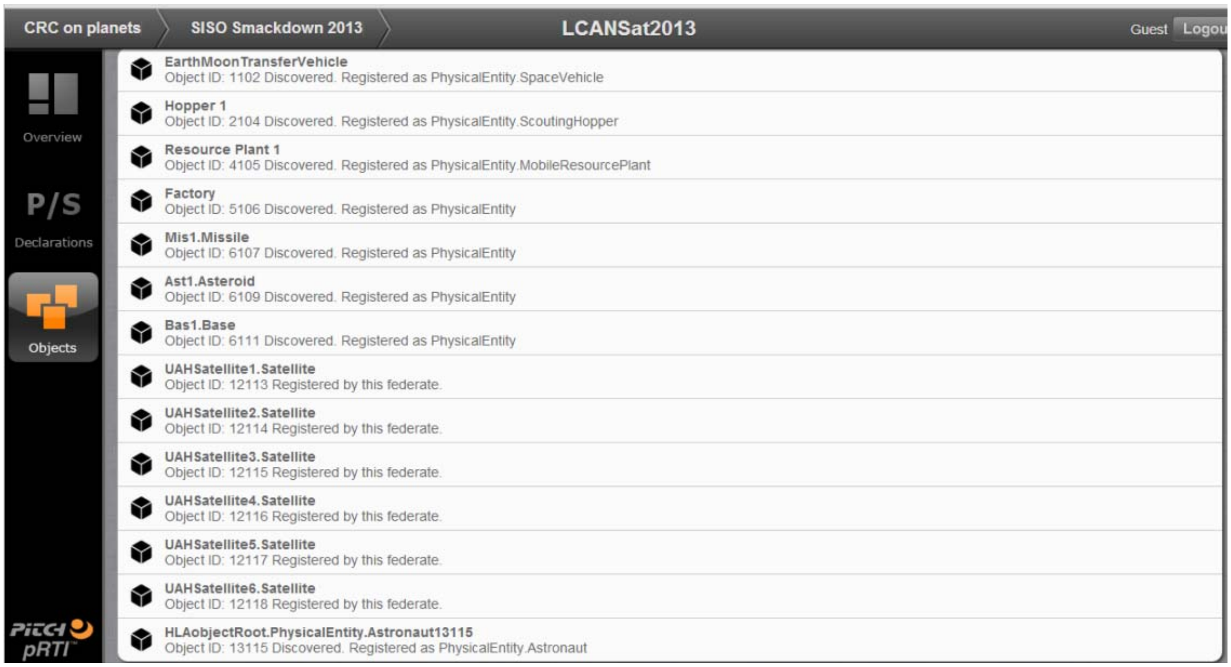


Figure 4-5 Radio message object model template

Federates using our communication service should publish the TX interaction and subscribe to RX interaction. Our LCANSat2013 communication server does the reverse order: it publishes RX interaction and subscribes to all the TX interactions.

In order to calculate line-of-sight information between two physical entities in the mission, our LCANSat2013 federate subscribes to the name, reference frame and position attributes of the PhysicalEntity object class and maintains tables of each entity's latest position in the mission during the SISO Smackdown event. Figure 4-6 is a screenshot of a display listing the discovered entities.



EarthMoonTransferVehicle
Object ID: 1102 Discovered. Registered as PhysicalEntity.SpaceVehicle

Hopper 1
Object ID: 2104 Discovered. Registered as PhysicalEntity.ScoutingHopper

Resource Plant 1
Object ID: 4105 Discovered. Registered as PhysicalEntity.MobileResourcePlant

Factory
Object ID: 5106 Discovered. Registered as PhysicalEntity

Mis1.Missile
Object ID: 6107 Discovered. Registered as PhysicalEntity

Ast1.Asteroid
Object ID: 6109 Discovered. Registered as PhysicalEntity

Bas1.Base
Object ID: 6111 Discovered. Registered as PhysicalEntity

UAHSatellite1.Satellite
Object ID: 12113 Registered by this federate.

UAHSatellite2.Satellite
Object ID: 12114 Registered by this federate.

UAHSatellite3.Satellite
Object ID: 12115 Registered by this federate.

UAHSatellite4.Satellite
Object ID: 12116 Registered by this federate.

UAHSatellite5.Satellite
Object ID: 12117 Registered by this federate.

UAHSatellite6.Satellite
Object ID: 12118 Registered by this federate.

HLAobjectRoot.PhysicalEntity.Astronaut13115
Object ID: 13115 Discovered. Registered as PhysicalEntity.Astronaut

Figure 4-6 Screenshot of Physical entities discovered by LCANSat2013 federate

The basic work flow of communication server can be explained in the sequence diagram (Figure 4-7):

- The communication server receives a TX interaction through the HLA callback method receiveInteraction().
- It extracts the source and destination information from the interaction and determines if both the sender and receiver have access to a certain satellite in the satellite constellation.
- If so, the communication server builds a RX message and uses the HLA service sendInteraction() to send the RX message, federates which subscribes to the RX message will receive the message.
- If not, that TX interaction message will be discarded.

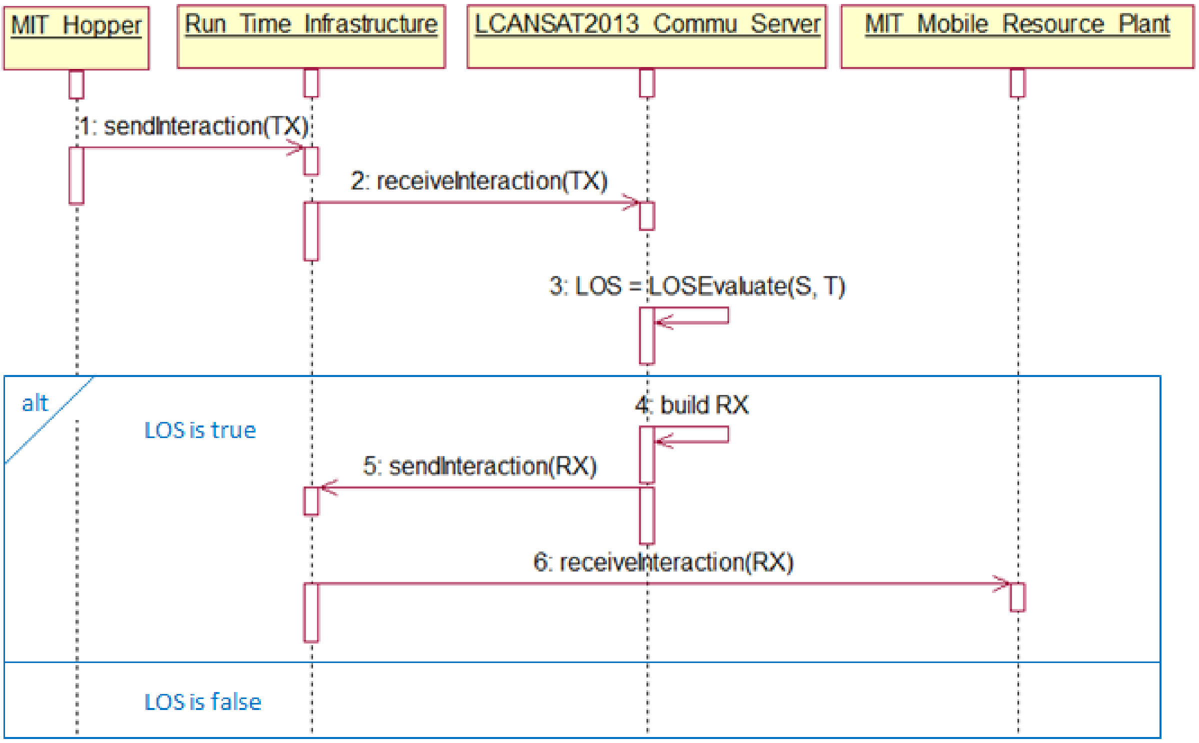


Figure 4-7 Communication Server delivering message sequence diagram

The LOSEvaluate() method invoked by the communication server in the above UML sequence diagram is used to determine the inter-visibility between entities in the lunar scenario. In STK Components, access relation issue between two entities is resolved by specifying access constraint it must satisfy. The simplest and most commonly used access constraint is the Central Body Obstruction Constraint, which requires that the view from one entity to another not be obstructed by a central body such as the moon. Further constraints can be imposed on a single access relation. In our case, we assumed communication capability between a surface entity and satellite is poor near the horizon, so we set an Elevation Angle Constraint with minimum 5 degrees angle on the access relation. This constraint requires that the angle between the link (yellow line in the Figure) and the plane tangent to the surface of the moon at the location of the entity be between 5 degrees to 175 degrees.

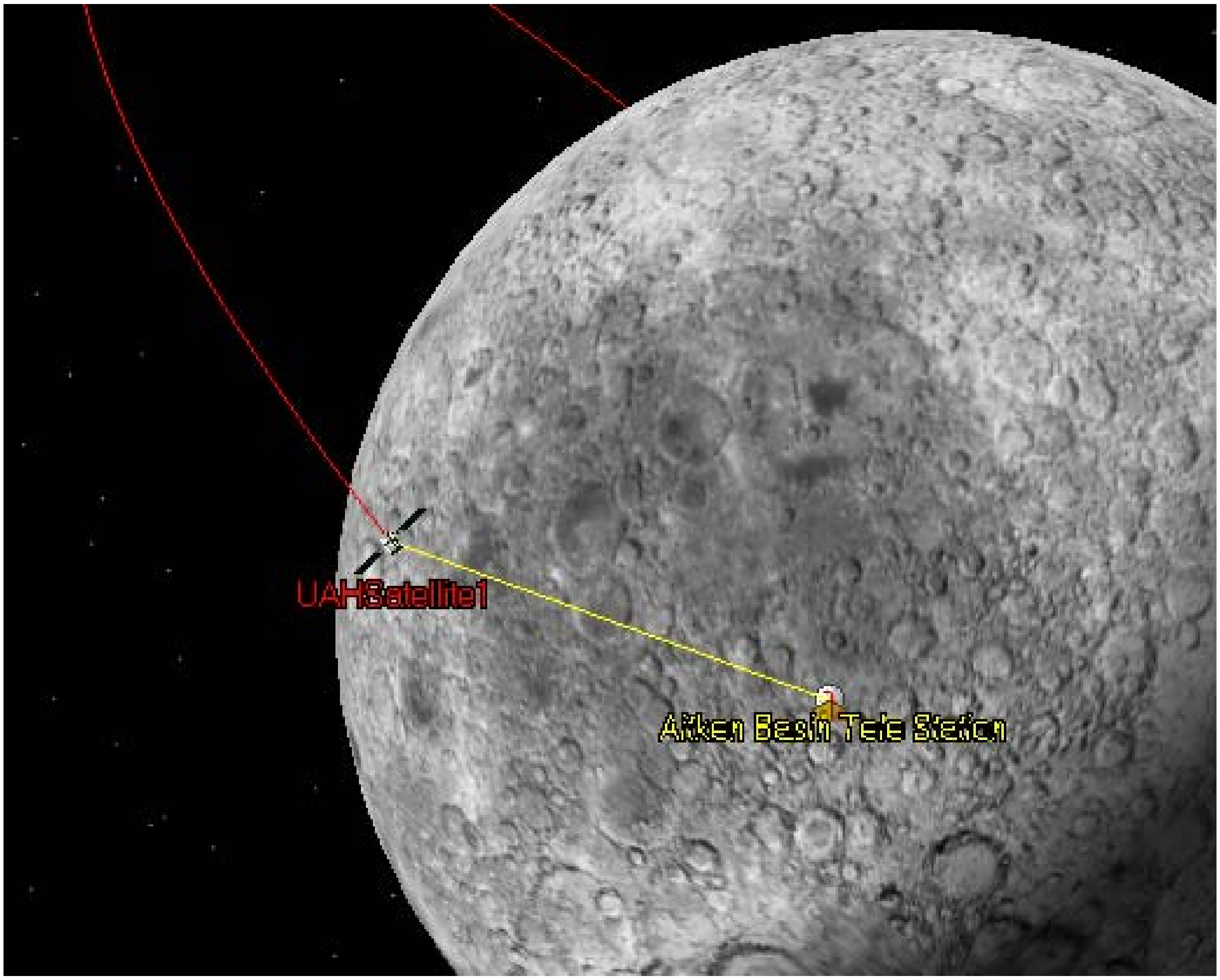


Figure 4-8 Satellite to Lunar surface object Line of Sight determination

4.2 Driver federate

The Driver federate is a testing tool that will be bequeathed to future Smackdown teams. It was built for several reasons.

First, it was built to circumvent wasting time developing testing tools. Future teams will not need to spend time developing a way to test federates against others. The Driver federate publishes attributes of all entities across the network. Therefore, testing interoperability between federates is achieved by running the Driver alone.

Second, it was built to encourage testing to begin sooner. With the Driver, testing can begin immediately. When testing is done early, problems are discovered sooner, there is more time for solutions to be implemented, and a more thought out product is produced. All of which, benefit future teams.

Third, it was built to ease the development process for future teams. Teams will have one more tool in place to utilize and one less challenge to face. The motivation for this is to continue growing upon older federates, creating newer ones, and help guide improvement.

The Driver federate was built in Matlab using ForwardSim's HLA Toolbox. The HLA Toolbox is the MATLAB interface to the High Level Architecture distributed simulation standards: HLA 1.3 and IEEE 1516. The HLA Toolbox automatically generates Matlab code based on specifications selected by the user.

4.3 Visualization Federate

The objectives established for the visualization federate team included:

1. Provide graphics and visualization to display all Federation objects, lunar surface and stellar background.
2. Develop a C++ interface between the existing Matlab federate and the Unity 3D MPC_Client.

Tools used by the team to accomplish these objectives included:

1. Matlab Rs2012
2. Microsoft Visual Studio 2010
3. FowardSim HLA Toolbox
4. The Unity Tool (TUT)

MATLAB is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, Matlab allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, the optional toolbox HLA Toolbox interfaces to the U.S. DoD High Level Architecture (HLA 1.3) simulation standard, to the IEEE 1516 standard, as well a the Distributed Interoperable Simulation (DIS) standard IEEE 1278.

The MatLab element of the visualization federate receives all current positions, translates coordinates to a common reference frame, and translates the 3D models based on those coordinates. The Matlab element consists of several files such as fsi3DViewer.m, fedAmbassador.m and telStreamFunc.m:

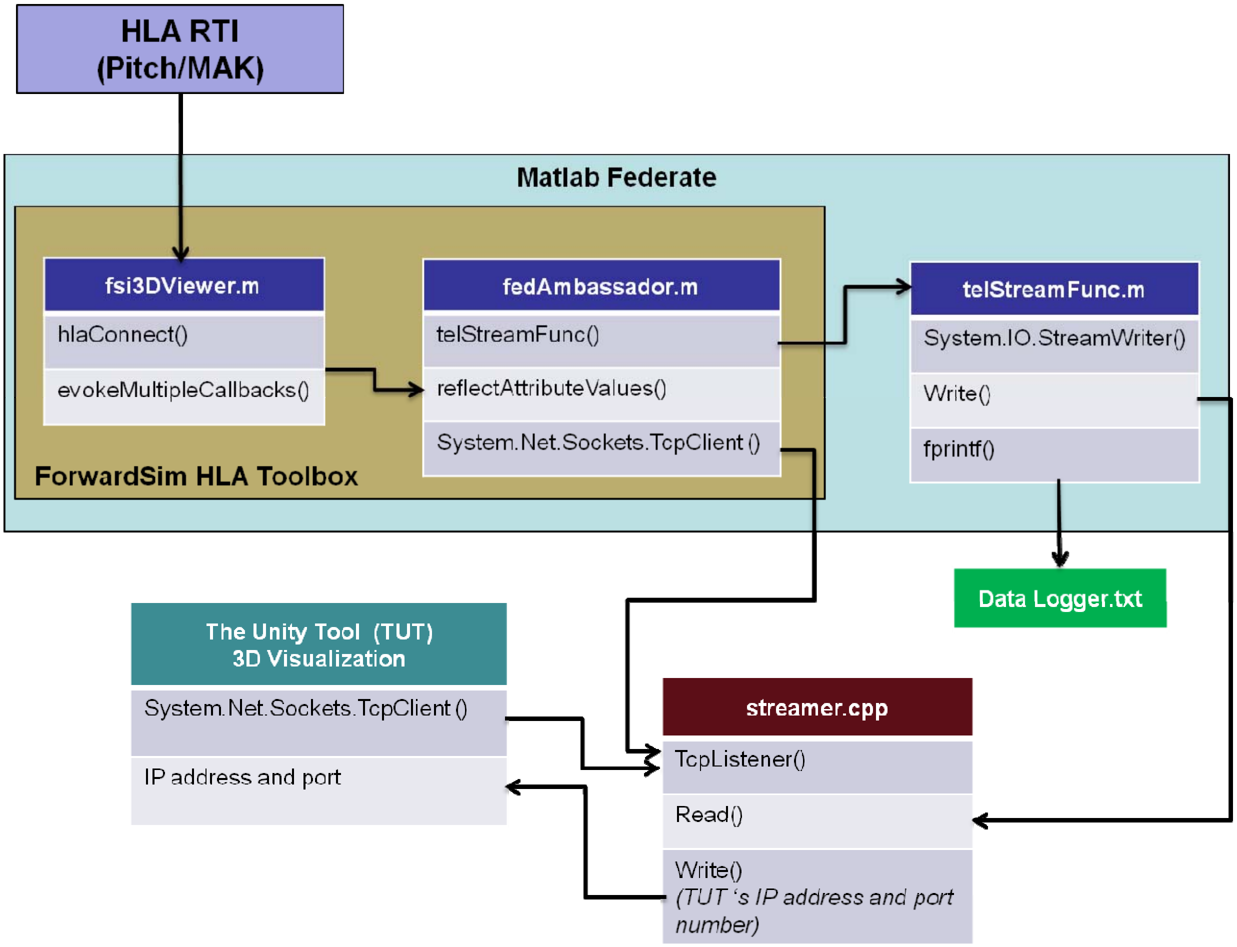


Figure 4-9 Visualization Federate Architecture

The fsi3DViewer.m file services include:

hlaConnect() - establishes a RTI connection and calls the fedAmbassador
createFederationExecution() - creates a Federation Execution
joinFederationExecution() - joins a federation execution
enableAsynchronousDelivery () - enables asynchronous delivery
hlaAutoPublishSubscribe() - subscribes to federates
evokeMultipleCallbacks() - tells the RTI that the federate is ready to process one or many callbacks
ResignandQuit() – disconnects and resigns the federate

The fedAmbassador.m file services include:

System.Net.Sockets.TcpClient ()- Provides client connections for TCP network services to the Unity 3D viewer.
reflectAttributeValues() – gets federate attributes such as position, orientation, name, etc.
telStreamFunc() – gets xml formatted string.
telStreamFunc.m

The telStreamFunc.m file reads XML data and streams it to an IP address & port.
System.IO.StreamReader() - Implements a TextWriter for writing characters to a stream
Write() - Writes a string to the stream
fprintf() – Writes string to a log file

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms or WPF applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. The visualization federate developed in Visual Studio is streamer.cpp.

The streamer.cpp creates a server that receives XML data from the Matlab federate and streams the XML data to The Unity Tool (TUT), providing the following services:
TcpListener() - Listens for connections from TCP network clients, TUT and Matlab.
Start() - Starts listening for incoming connection requests from TUT and Matlab.
AcceptTcpClient() - Accepts a pending connection request.
GetStream() - Gets a stream object for reading and writing.
Read() – Reads XML data sent by the Matlab client.
Write() – Writes XML data to the TUT client.
Close() – Shutdowns and ends connection.

The ForwardSim HLA Toolbox consists of a set of functions designed to implement HLA federates within the MATLAB environment. The function set encompasses of the entire HLA standard library and additional functionality which useful to simplify the implementation of the HLA federates. It has a streaming capability which might be able to replace streamer.cpp

The HLA Toolbox also includes a complete set of callback templates ready to be used and customized by the user. These m-files are called by the RTI in response of a Tick or EvokeCallback call as specified in the federate implementation.

ForwardSim has developed an RTI Ambassador running in the MATLAB environment and allowing the creation of m-files using the HLA functions described in the HLA standards. All the code required to create and execute an HLA federate is written in m-code (MATLAB programming language) by the user and is fully compatible with any existing MATLAB program or toolbox. Like any other MATLAB program, C/C++ code can be called from the m-files. MATLAB connects directly with the RTI without any other tool when the HLA Toolbox is installed.
The HLA toolbox functions have the same prototypes as the C++ HLA standard interface (API) in order to minimize the learning process and to be fully compliant with the HLA standards.

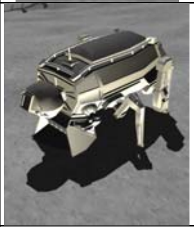
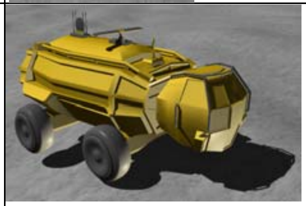
The HLA Toolbox is capable of encoding and decoding information from/to the RTI allowing information exchange with any HLA compliant federate on the same federation (independent of the OS or language used by the other federate).
It is possible to run more than one federate on one computer. Each federate requires one MATLAB instance.

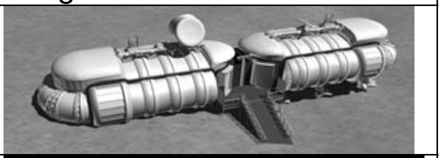
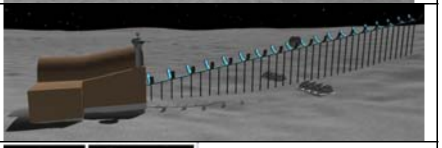
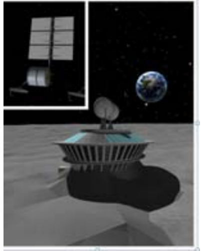
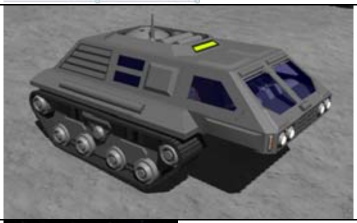
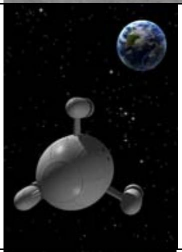
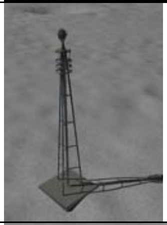
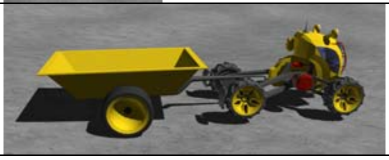
The HLA Toolbox allows the use of any other MATLAB Toolboxes to create a federate. The code can even be compiled with the MATLAB Compiler in order to generate a standalone application running on a computer without a MATLAB license.

The toolbox also provide a Graphical User Interface (GUI) used to quickly generate the m-code required to create an HLA federate simply by filling a form and selecting the information to be publish/subscribed from a tree view. This tool allows the user to create their first HLA federate in just a few minutes. The generated code will be customized by the user to define the behavior of its federate.

The Unity Tool (TUT) is a 3D visualization tool developed by NASA at Kennedy Space Center. TUT is based on Unity 3D Game Engine and it displays all federation objects, lunar surface, and stellar background and provides the following services:

Game based shared, distributed visualization
Student Simulations and Models
NASA based telemetry interface (MPC2)
Distributed and Preserved Simulation, replay without simulators

3D Model	Team Name	Image
Astronaut Tim	MIT	
Scouting Hopper	MIT	
Mobile Resource Plant	MIT	

3D Model	Team Name	Image
Lunar ISAC		
Lunar Mass Driver	UAH	
Comm Sat Control Center	UAH	
Cargo Rover	Nebraska	
Earth Moon Transfer Vehicle	NASA	
L2 Outpost	Munich	
L2 Tower Array	Munich	
Regolith Hauler		

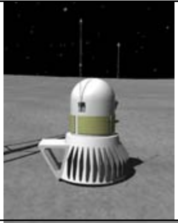
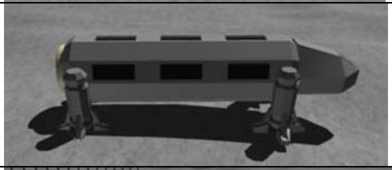
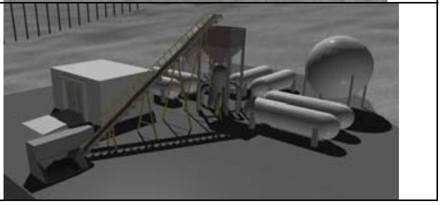
3D Model	Team Name	Image
Lunar Supply Depot		
Deep Space RADAR	Genoa	
Lunar Cargo Lander	Penn State	
Lunar Fabrication Facility (Oxygen Factory)	Brunel / Exeter	

Figure 4-10 3D models of Smackdown federates available in TUT

Other schools who were not able to get their simulations integrated in time participated as observers, including

4.4 Lunar Mass Driver federate

The LMD federate included the freely available program Orbiter 2010 which was connected to the rest of the distributed simulation through an interface that developed in Microsoft Visual Studio 2010 C++. We had to use the full version of VS 2010 rather than to be compatible with the RTI libraries. The full version of visual studio provided the capability to debug DLLs (Dynamic Linked Libraries) by attaching to the process. This allows the programmer to use breakpoints and step through the code as if it were a standalone program. In the Express edition, this capability was not available. This would have slowed development time significantly since all custom modules (or specifically ours) in orbiter are DLLs.

Orbiter provides very limited debug capability. We were able to write a string to the Orbiter screen. We used this to display of the LMD in X,Y,Z coordinate system in order to compare it to the LMD Payload when we needed to correct reference frameworks.

We used Visual Studio 2010 because the libraries for Pitch and MäK were compiled with 2010. They would not work with 2012. At the time of the Smackdown, Pitch had not released a set of libraries compiled with 2010.

The Orbiter 2012 LMD that we reused is a scenario developed in the orbiter community. It is designed to launch a payload into low lunar orbit. This scenario will run and publish its position updates for the LMD station and for the LMD payload to the LCANSat2013 federate for visualization of the location of the federate in their 3D viewer. The LMD Federate also publishes the location of the station and payload as the scenario ran to the 3D lunar surface visualization federate. The need to ensure insuring that the coordinates published by federate were in a reference frame that could be use an available transformation matrix to convert to the reference frame used by each of each visualization federates. Orbiter used Moon center Inertial but the visualization tools r used the moon centered fixed reference frame. When converting from Moon center Inertial or any other reference frame it is important that to use the same date and time for the reference frame. This problem was further complicated by discrepancies about how reference frames were labeled in different applications.
Because Orbiter was programed using C++ and the source code was made available the API used to allow the Orbiter program to talk to the MAK and Pitch RTIs was programmed in Visual Studio 2010 using C++.

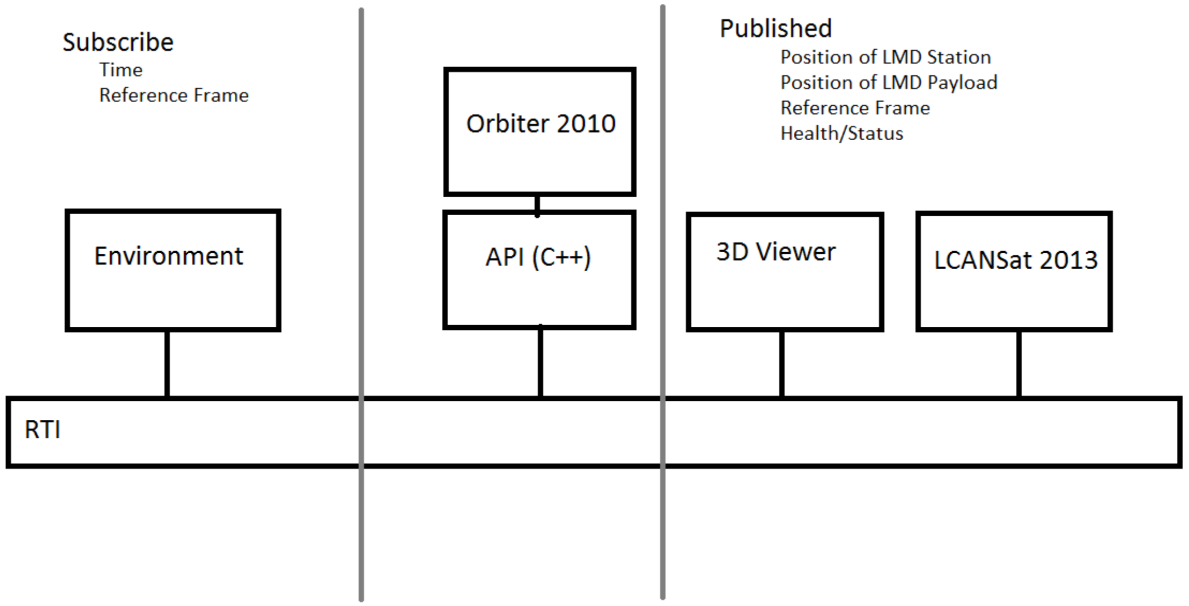


Figure 4-11 Lunar Mass Driver and relationship to other federates

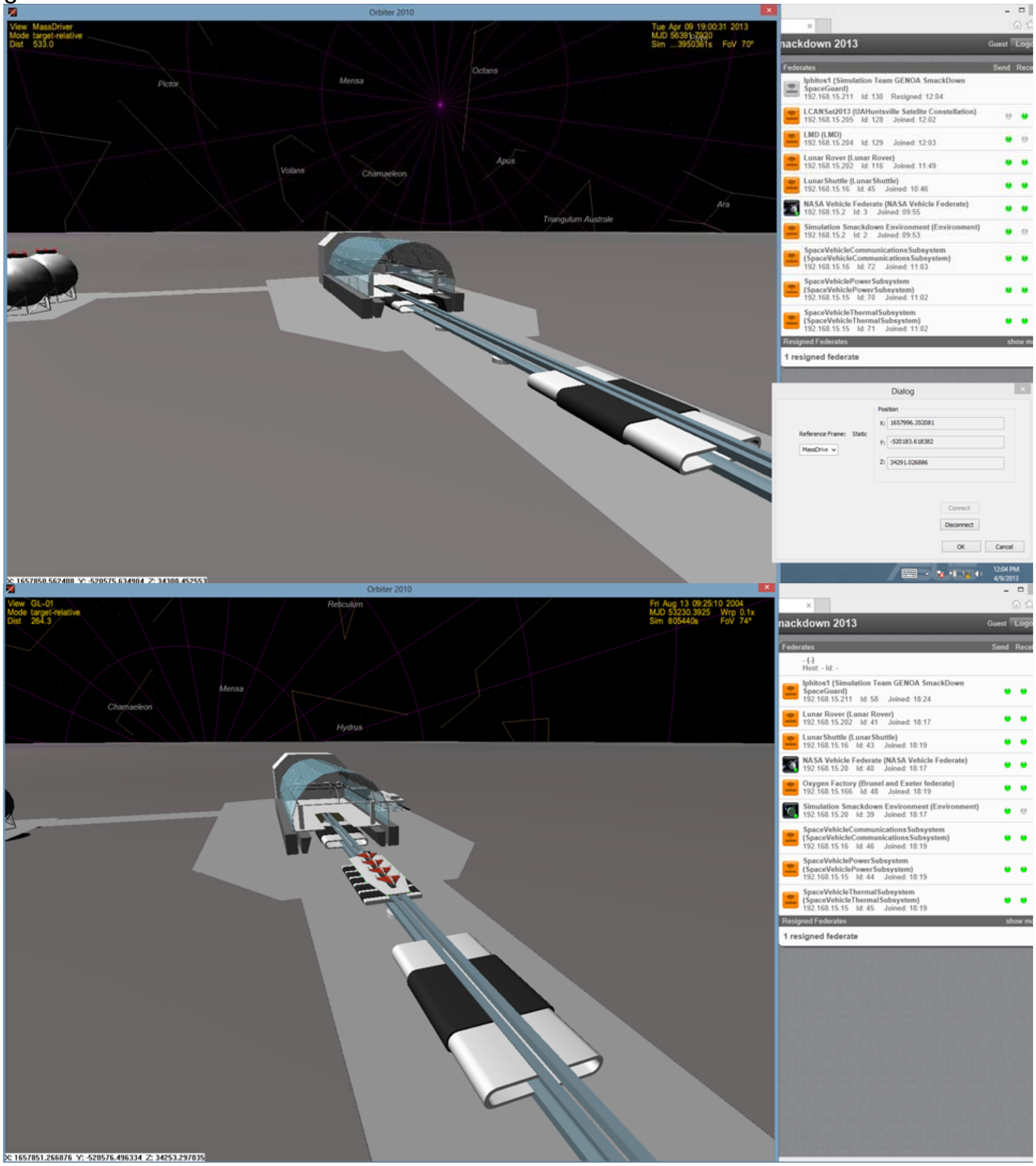
The Goals for the LMD were:

- Modeling
  - To apply the principle of LMD to launch a payload into a low lunar orbit
- Simulation/Reuse
  - Use a scenario developed by Yuri Kulchitsky from the Orbiter 2010 community
- Programing
  - Use the developed environment of Orbiter 2010 to account for the lunar environment variables.
  - Correctly connect to both RTIs
  - Publish positions of both the LMD station and the LMD payload.
  - Ensure the multiple reference frames in use are correctly coordinated

Our LMD federate used only a portion of the capability of Orbiter 2010 and the LMD scenario. Several opportunities for future work were seen. The scenario in Orbiter 2010, there are four delta gliders that can be programed to do a variety of different missions, interact with different federates, and even preform an ownership transfer such as docking with an orbiting space ship, placing an additional satellite into orbit, retrieving a satellite for maintenance then returning it to orbit, or just to return to earth. Due to our Due to our limited time and our need to focus on the coordinate transforms and RTI connections, we were unable to implement these additional missions. Thus, our federate was limited to the launch of a payload into low lunar orbit.

Lessons Learned

During the SISO Smackdown event the LMD successfully connected to the Federation and simulated the launch of a payload into low lunar orbit using the Orbiter 2010 program. Position and Health/Status where published for the federate. The LMD federate subscribed to the environment federate for reference frame, and time advance grants.



View MassDriver
Mode target-relative
Dist 533.0

Tue Apr 09 19:00:31 2013
MJD 56391.7920
Sim ...3950361s FoV 70°

X: 1657850.562408 Y: -520575.634904 Z: 34300.452553

nackdown 2013

Federates	Send	Rece
Iphitos1 (Simulation Team GENOA SmackDown SpaceGuard) 192.168.15.211 Id: 130 Resigned: 12:04		
LCANSat2013 (UAHuntsville Satelite Constellation) 192.168.15.205 Id: 128 Joined: 12:02	○	●
LMD (LMD) 192.168.15.204 Id: 129 Joined: 12:03	●	○
Lunar Rover (Lunar Rover) 192.168.15.202 Id: 116 Joined: 11:49	●	●
LunarShuttle (LunarShuttle) 192.168.15.16 Id: 45 Joined: 10:46	●	●
NASA Vehicle Federate (NASA Vehicle Federate) 192.168.15.2 Id: 3 Joined: 09:55	●	●
Simulation Smackdown Environment (Environment) 192.168.15.2 Id: 2 Joined: 09:53	●	○
SpaceVehicleCommunicationsSubsystem (SpaceVehicleCommunicationsSubsystem) 192.168.15.16 Id: 72 Joined: 11:03	●	●
SpaceVehiclePowerSubsystem (SpaceVehiclePowerSubsystem) 192.168.15.15 Id: 70 Joined: 11:02	●	●
SpaceVehicleThermalSubsystem (SpaceVehicleThermalSubsystem) 192.168.15.15 Id: 71 Joined: 11:02	●	●

Resigned Federates show m

1 resigned federate

Position
Reference Frame: Static
MassDrive
X: 1657996.352081
Y: -520183.618382
Z: 34291.026886

12:04 PM
4/9/2013

View GL-01
Mode target-relative
Dist 264.3

Fri Aug 13 09:25:10 2004
MJD 53230.3925 Wrp 0.1x
Sim 805440s FoV 74°

X: 1657851.266876 Y: -520576.496334 Z: 34253.297035

nackdown 2013

Federates	Send	Recei
- (-) Host: - Id: -		
Iphitos1 (Simulation Team GENOA SmackDown SpaceGuard) 192.168.15.211 Id: 58 Joined: 18:24	●	●
Lunar Rover (Lunar Rover) 192.168.15.202 Id: 41 Joined: 18:17	●	●
LunarShuttle (LunarShuttle) 192.168.15.16 Id: 43 Joined: 18:19	●	●
NASA Vehicle Federate (NASA Vehicle Federate) 192.168.15.20 Id: 40 Joined: 18:17	●	●
Oxygen Factory (Brunel and Exeter federate) 192.168.15.166 Id: 48 Joined: 18:19	●	●
Simulation Smackdown Environment (Environment) 192.168.15.20 Id: 39 Joined: 18:17	●	○
SpaceVehicleCommunicationsSubsystem (SpaceVehicleCommunicationsSubsystem) 192.168.15.16 Id: 46 Joined: 18:19	●	●
SpaceVehiclePowerSubsystem (SpaceVehiclePowerSubsystem) 192.168.15.15 Id: 44 Joined: 18:19	●	●
SpaceVehicleThermalSubsystem (SpaceVehicleThermalSubsystem) 192.168.15.15 Id: 45 Joined: 18:19	●	●

Resigned Federates show m

1 resigned federate

Figure 4-12 Orbiter 2010 screenshots of Lunar Mass Driver

5 INTEGRATION AND TEST

Six full up integration tests were conducted between March 29th and April 5th just prior to the Smackdown Event. All the integration tests were supported by communication with the remote sites via by teleconference and VSee video teleconferencing. During each integration test, each teams' federates joined in the federation in turn based on a roll call.

In the integration test, teams worked together to verify the correctness of interactions between different federates. People from JSC monitored all the joined federates to make sure each of them properly advanced time and published data. As LCANSat2013 is supposed to represent all the physical entities in the federation in the Insight3D viewer, our federate actually verified all the location and motion information of different entities either in the orbit of the moon or on the surface.

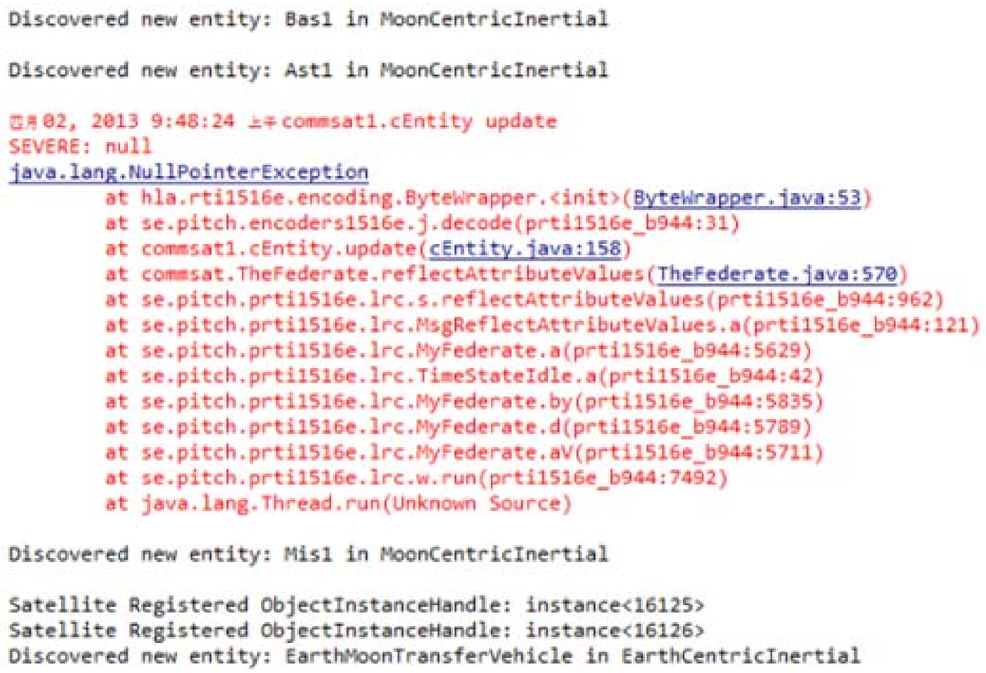
In the same time, several problems about our LCANSat2013 federate were found during the six integration test:

Problem 1: LCANSat2013 could not join the federation after LMD federate joined.
Solution: UAHuntsville team made sure that LMD federate joined the federation after LCANSat2013. This problem was later completely solved by JSC assigning more IP addresses to the team.

Problem 2: It took too much time to propagate six satellites.
Solution: This problem is understandable, since in order to achieve high fidelity, STK numerical propagator calculates a lot of data for one satellite. By using background calculation capability, we paralleled the six propagations instead of propagating six satellites one by one. This greatly reduced the six satellites propagation time.

Problem 3: LCANSat2013 worked well with PITCH RTI but failed to run on MAK RTI.
Solution: By working with Shagoto Nandi from MAK and after a lot of debugging and comparing programming environments, we found that java application developed using Java Development Kit 1.7 on Windows could cause the federate to throw UnsatisfiedLinkError exception every time it tried to connect to MAK RTI. We solved this problem by completely uninstalling JDK 1.7 and installing JDK 1.6.

Problem 4: LCANSat2013 federate kept throwing NullPointer exceptions due to other teams failing to provide the latest Attribute Handle Value Map or providing Attribute Handle Value Map which missed certain attribute values.



```
Discovered new entity: Bas1 in MoonCentricInertial

Discovered new entity: Ast1 in MoonCentricInertial

四月 02, 2013 9:48:24 上午 commsat1.cEntity update
SEVERE: null
java.lang.NullPointerException
	at hla.rti1516e.encoding.ByteWrapper.<init>(ByteWrapper.java:53)
	at se.pitch.encoders1516e.j.decode(prti1516e_b944:31)
	at commsat1.cEntity.update(cEntity.java:158)
	at commsat.TheFederate.reflectAttributeValues(TheFederate.java:570)
	at se.pitch.prti1516e.lrc.s.reflectAttributeValues(prti1516e_b944:962)
	at se.pitch.prti1516e.lrc.MsgReflectAttributeValues.a(prti1516e_b944:121)
	at se.pitch.prti1516e.lrc.MyFederate.a(prti1516e_b944:5629)
	at se.pitch.prti1516e.lrc.TimeStateIdle.a(prti1516e_b944:42)
	at se.pitch.prti1516e.lrc.MyFederate.by(prti1516e_b944:5835)
	at se.pitch.prti1516e.lrc.MyFederate.d(prti1516e_b944:5789)
	at se.pitch.prti1516e.lrc.MyFederate.aV(prti1516e_b944:5711)
	at se.pitch.prti1516e.lrc.w.run(prti1516e_b944:7492)
	at java.lang.Thread.run(Unknown Source)

Discovered new entity: Mis1 in MoonCentricInertial

Satellite Registered ObjectInstanceHandle: instance<16125>
Satellite Registered ObjectInstanceHandle: instance<16126>
Discovered new entity: EarthMoonTransferVehicle in EarthCentricInertial
```

Figure 5-1 LCANSat2013's decoder trying to decode "null" attribute

Solution: It is good software engineering principle to not expect federate behaves exactly as what you want. Proper and robust exception handling mechanism was added to help solve this problem.

Problem 5: For our Insight3D viewer, we used makers and texts to represent entities on the moon surface and orbiting the moon. Since there are so many of them, the texts and markers could easily bloat the viewer and thus hindered its readability.
Solution: We add a distance constraint on the visibility of the marker so that a marker would only become visible when the distance between camera and the entity is less than 1000km. If we look at the moon at a rather far away distance, we could only see the text representing the name of the entity. This makes the viewer much cleaner.

Problem 6: During the testing, we found that some HLA callbacks never get called when they were supposed to be called.
Solution: HLA provides more than one different versions of the same callback, for instance, there are three reflectAttributeValues callbacks:

```
public void reflectAttributeValues(ObjectInstanceHandle theObject,
               AttributeHandleValueMap theAttributes, byte[]
userSuppliedTag,
               OrderType sentOrdering, TransportationTypeHandle
theTransport,
               LogicalTime theTime, OrderType receivedOrdering,
               SupplementalReflectInfo reflectInfo) throws
FederateInternalError
public void reflectAttributeValues(ObjectInstanceHandle theObject,
               AttributeHandleValueMap theAttributes, byte[] 
userSuppliedTag,
               OrderType sentOrdering, TransportationTypeHandle 
theTransport,
               SupplementalReflectInfo reflectInfo)
public void reflectAttributeValues(ObjectInstanceHandle theObject,
               AttributeHandleValueMap theAttributes, byte[] 
userSuppliedTag,
               OrderType sentOrdering, TransportationTypeHandle 
theTransport,
               LogicalTime theTime, OrderType receivedOrdering,
               MessageRetractionHandle retractionHandle,
               SupplementalReflectInfo reflectInfo)
```

We never know which version the RTI will invoke, so instead of just writing one version, it is better to write all the above three in TheFederate class and put the same implementation code in the three reflectAttributeValues().

Problem 7: When a federate resigned from the federation, the insight3D viewer still kept the marker and text representing the physical entity of that resigned federate.
Solution: Clear marker and text when the callback method removeObjectInstance gets called.

Problem 8: There is a synchronization exception during the demonstration in insight3DTimeChanged() method of class LCANSatManager due to different threads competing otherEntitiesToBeDrawn list. This would cause the Insight3D viewer to crash!
Solution: Our current solution is not good since we only surround the trouble code with try-catch block in order to prevent the entire Insight3D viewer from crashing during the event. Future teams may wish to add a mutual exclusion lock on this resource to resolve this. The future team should be aware that this problem has not been solved properly.

6 DEMONSTRATE SIMULATIONS AT THE EVENT

The team made the final selection of computers to be used at the event, and used them at the last two test events to make sure that the software was correctly loaded and that the machines were capable of executing the software with acceptable speed. A “battle box” containing tools, spare cables, etc. was packed and shipped along with the computers to San Diego. The computers used in the lab for testing were not used in San Diego because of the difficulty of shipping large desktop machines. The simulations wound up running on individual student laptops.

On Sunday afternoon, before the event on Tuesday evening, the UAHuntsville team began setting up their computers, networking, and power at the same time as the NASA representatives began to set up their computers, power and networking in order to provide the maximum amount of time for addressing the problems that would inevitably arise. Figure 6-1 was taken during this period and shows some of the intensity and complexity of the event.

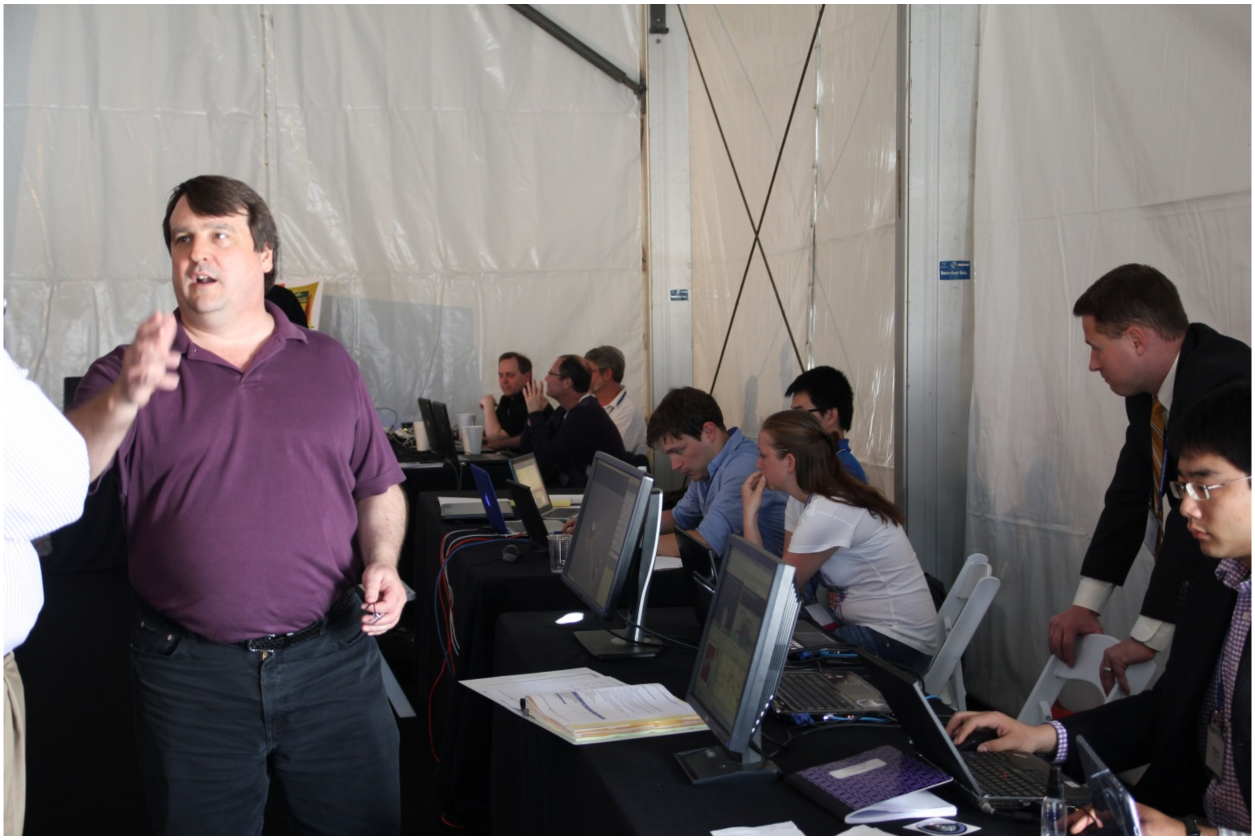


Figure 6-1 On-site Integration and Test

The schools teams worked closely together, with an impressive level of support from NASA and the tool vendors, to get the integrated simulation to reliably perform the demonstration scenario. This process was complicated by low network bandwidth, spotty wireless connectivity, and unreliable power. Since it was extremely difficult to download software form the internet, the team and vendors worked closely together to make sure everyone had the right software versions. The long days of continuous testing exposed a large number of problems that had been missed in the previous hour long, relatively simple test events.

Figure 6-2 shows a considerably neater set up, and significantly reduced stress as the team completes its preparations for the formal event. The monitor on the left shows the lunar mass driver federate. The monitor on the right shows the combined Matlab/STK tool used to analyze the communication coverage of alternate lunar satellite constellation orbits.

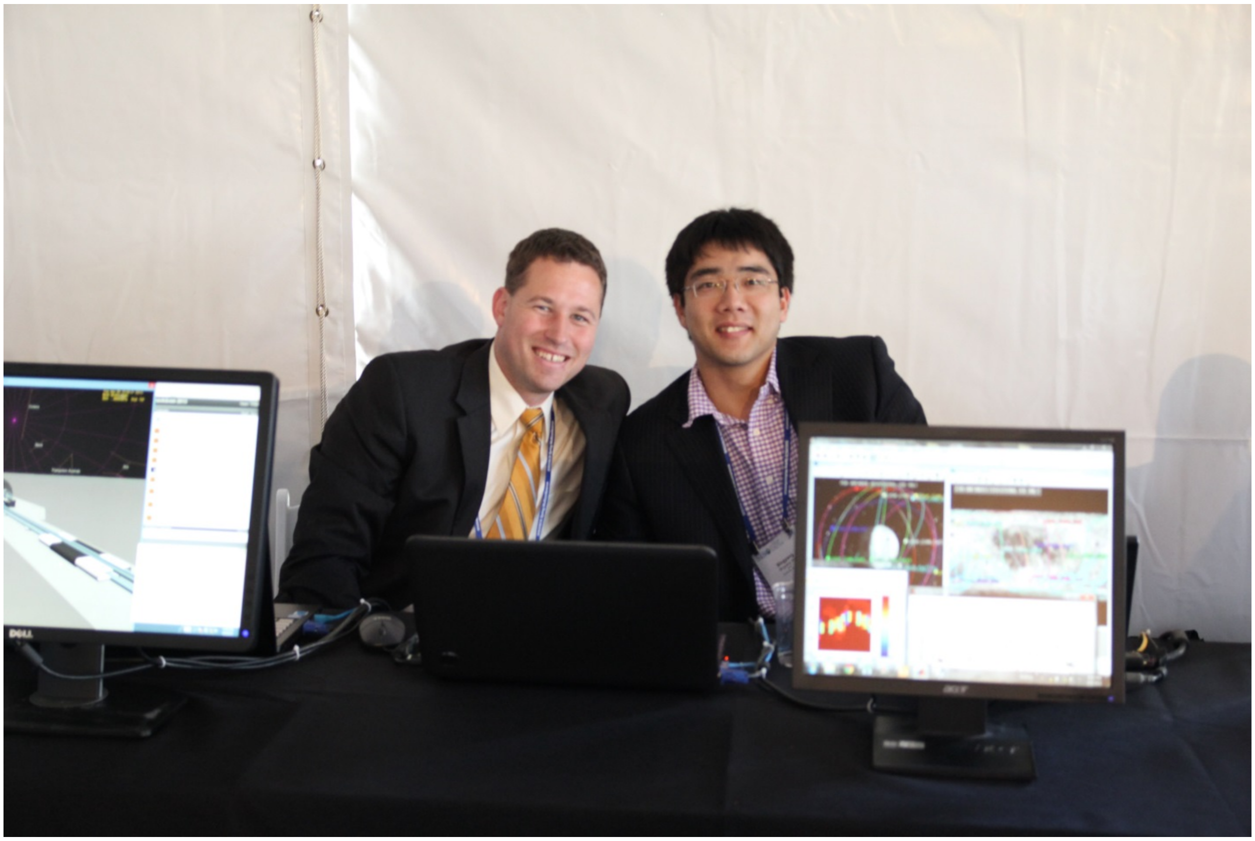


Figure 6-2 prepared for demonstration event.

Because so many teams were involved, the intense period of integration and testing was followed by a repetitious process of rehearsal and practice. The practice period was complicated by the fact that the audio visual equipment was not set up in time to support rehearsals, and when it was set up, it did not match that which was anticipated during the rehearsal, which in turn created some amount of confusion. This confusion was compounded by the fact that the person primarily responsible for planning the event was, at the last minute, not able to attend. This caused changes in the plan, and additional confusion.

None-the-less, the program began on time, and the team leaders introduced their teams and described their simulations as shown in figure 6-3. The introduction process did not go as well as hoped because the audience was not able to hear all of the teams, especially those who were operating their simulations remotely.

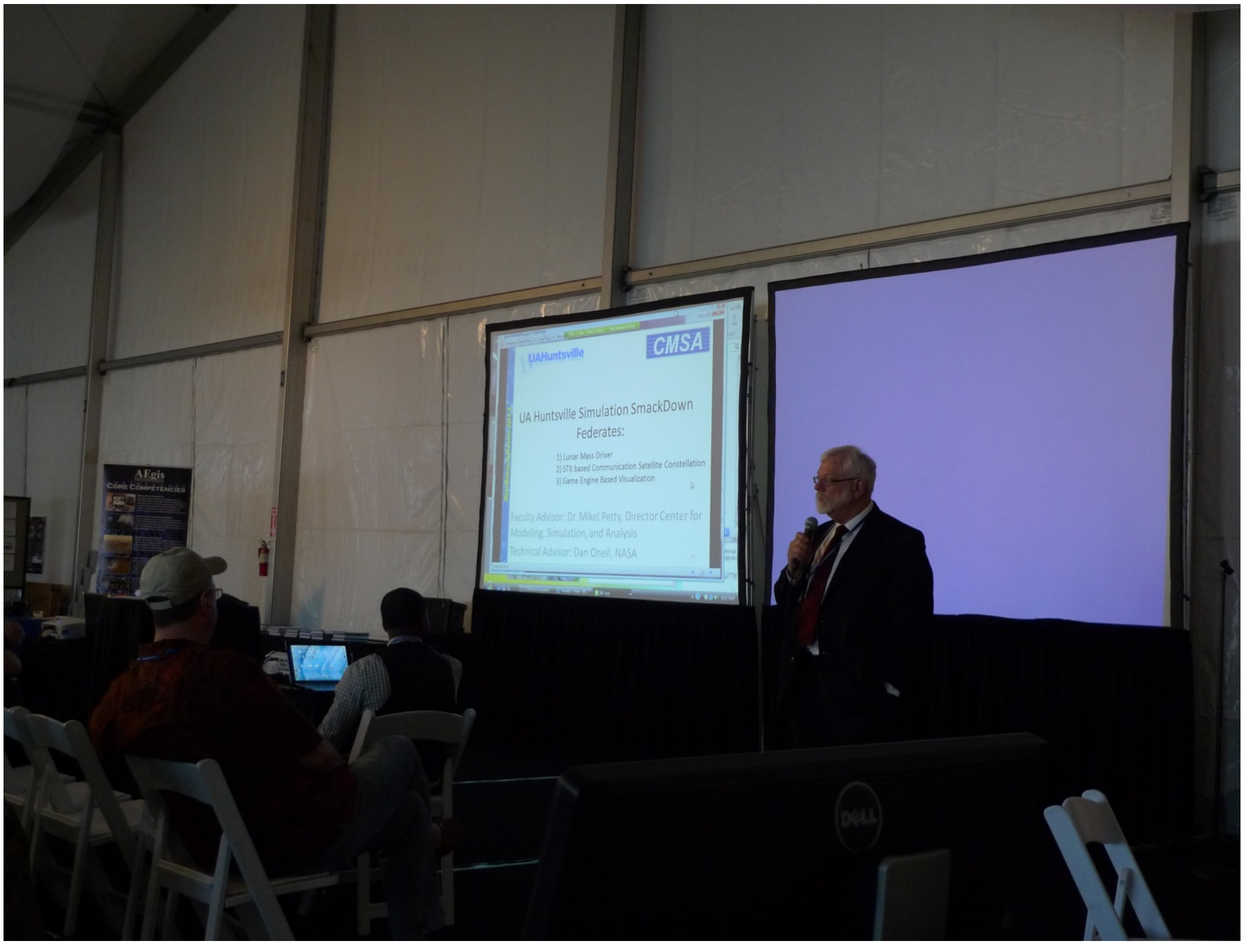

UAHuntsville
CMSA
SimSmackDown 2013
UA Huntsville Simulation SmackDown Federates:
1) Lunar Mass Driver
2) STK based Communication Satellite Constellation
3) Game Engine Based Visualization
Faculty Advisor: Dr. Mikel Petty, Director Center for Modeling, Simulation, and Analysis
Technical Advisor: Dan Oneil, NASA
AEgis
CORE COMPETENCIES
DELL

Figure 6.3 UAHuntsville Team introductions

When the introductions were complete, the actual demonstration of the integrated simulation began. The simulations themselves performed flawlessly. Figure 6.4 depicts a team's simulation joining into the federation. The image on the left is of their federate, displayed by the UAHuntsville Communication Satellite program. The person in the center is operating the UAHuntsville 3D visualization tool, preparing to display the operation of the simulation.
In figure 6.4, the image on the right is a traditional HLA "lollipop" diagram. Each white box represents a particular federate as a member of the federation.

AEgis
Core Competencies
MIT Hopper

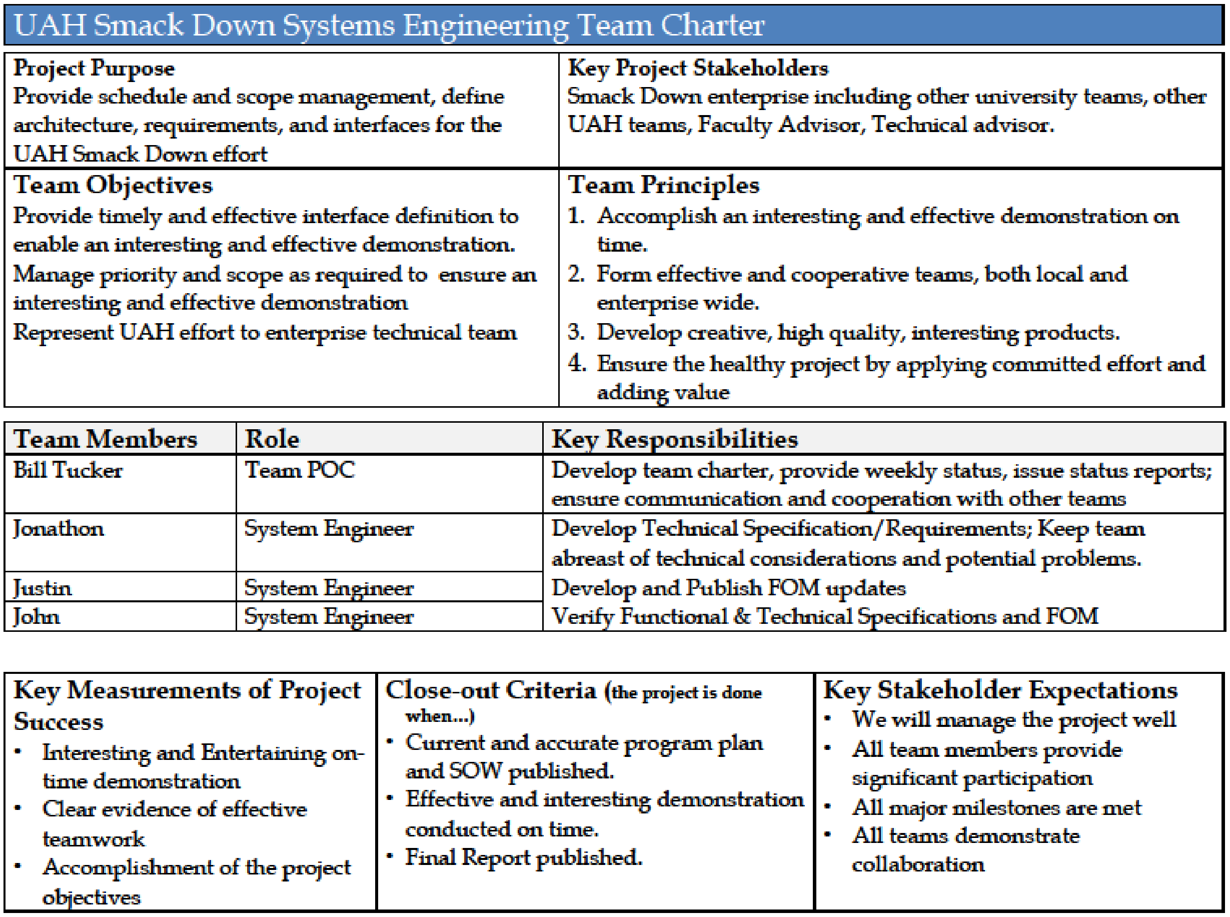
7 RECOMENDATIONS

A representative sample of participating schools and sponsoring organizations were interviewed to obtain observations of the event, and suggestions about how it might be improved. The comments collected during the interviews are paraphrased in Appendix C. The following synthesizes the comments and organizes them by theme, creating an actionable set of recommendations. The interviewees' names are withheld because the data was collected on a non-attribution basis. We do appreciate the insight they provided.

Table 7-1 Recommendations Summary

Summary of Recommendations	
Training & Education	Develop Syllabus…
Event Planning and Execution	Clear roles & responsibilities
…	
Establish Environment	Establish computers and network to be used in demonstration at beginning, and do not change them.

8 TEAM CHARTERS & PROGRAM PLAN



UAH Smack Down Systems Engineering Team Charter

Project Purpose	**Key Project Stakeholders**
Provide schedule and scope management, define architecture, requirements, and interfaces for the UAH Smack Down effort	Smack Down enterprise including other university teams, other UAH teams, Faculty Advisor, Technical advisor.
Team Objectives Provide timely and effective interface definition to enable an interesting and effective demonstration. Manage priority and scope as required to ensure an interesting and effective demonstration Represent UAH effort to enterprise technical team	**Team Principles** 1. Accomplish an interesting and effective demonstration on time. 2. Form effective and cooperative teams, both local and enterprise wide. 3. Develop creative, high quality, interesting products. 4. Ensure the healthy project by applying committed effort and adding value

Team Members	Role	Key Responsibilities
Bill Tucker	Team POC	Develop team charter, provide weekly status, issue status reports; ensure communication and cooperation with other teams
Jonathon	System Engineer	Develop Technical Specification/Requirements; Keep team abreast of technical considerations and potential problems.
Justin	System Engineer	Develop and Publish FOM updates
John	System Engineer	Verify Functional & Technical Specifications and FOM

Key Measurements of Project Success	Close-out Criteria (the project is done when…)	Key Stakeholder Expectations
• Interesting and Entertaining on-time demonstration • Clear evidence of effective teamwork • Accomplishment of the project objectives	• Current and accurate program plan and SOW published. • Effective and interesting demonstration conducted on time. • Final Report published.	• We will manage the project well • All team members provide significant participation • All major milestones are met • All teams demonstrate collaboration

UAH Smack Down LCANSat2013 Federate Team Charter

Project Purpose

Provide schedule and scope management, define architecture, requirements, and interfaces for the UAH Smack Down effort

Key Project Stakeholders

Smack Down enterprise including other university teams, other UAH teams, Faculty Advisor, Technical advisor.

Team Objectives

1. Build Lunar Communication System and Satellites Constellation federate(LCANSat2013) by using STK Components Java libraries.
2. Combine previous LCANServ federate and LCANSat federate together into a single federate.
3. Design and develop HLA interface for the above federate.

Team Principles

1. Accomplish an interesting and effective demonstration on time.
2. Form effective and cooperative teams, both local and enterprise wide.
3. Develop creative, high quality, interesting products.
4. Ensure the healthy project by applying committed effort and adding value

Team Members	Role	Key Responsibilities
Bingyang Wei	Team POC	Develop team charter, provide weekly status, issue status reports; ensure communication and cooperation with other teams. Design and build modules that simulate the Satellite constellation and lunar communication system (satellite propagation, satellite constellation visualization, communication animation).
Melissa Redmond	Software Engineer	Design and develop HLA interface to make LCANSat2013 connected to RTI. (This task will be done collaboratively by all the team members so everyone learns HLA interface development) Test federate on PITCH and MAK RTI.
Justin Brown	Software Engineer	

Key Measurements of Project Success

- Interesting and Entertaining on-time demonstration
- Clear evidence of effective teamwork and

Close-out Criteria (the project is done when…)

- Current and accurate program plan and SOW published.
- Effective and interesting demonstration conducted on time.
- Final Report published.

Key Stakeholder Expectations

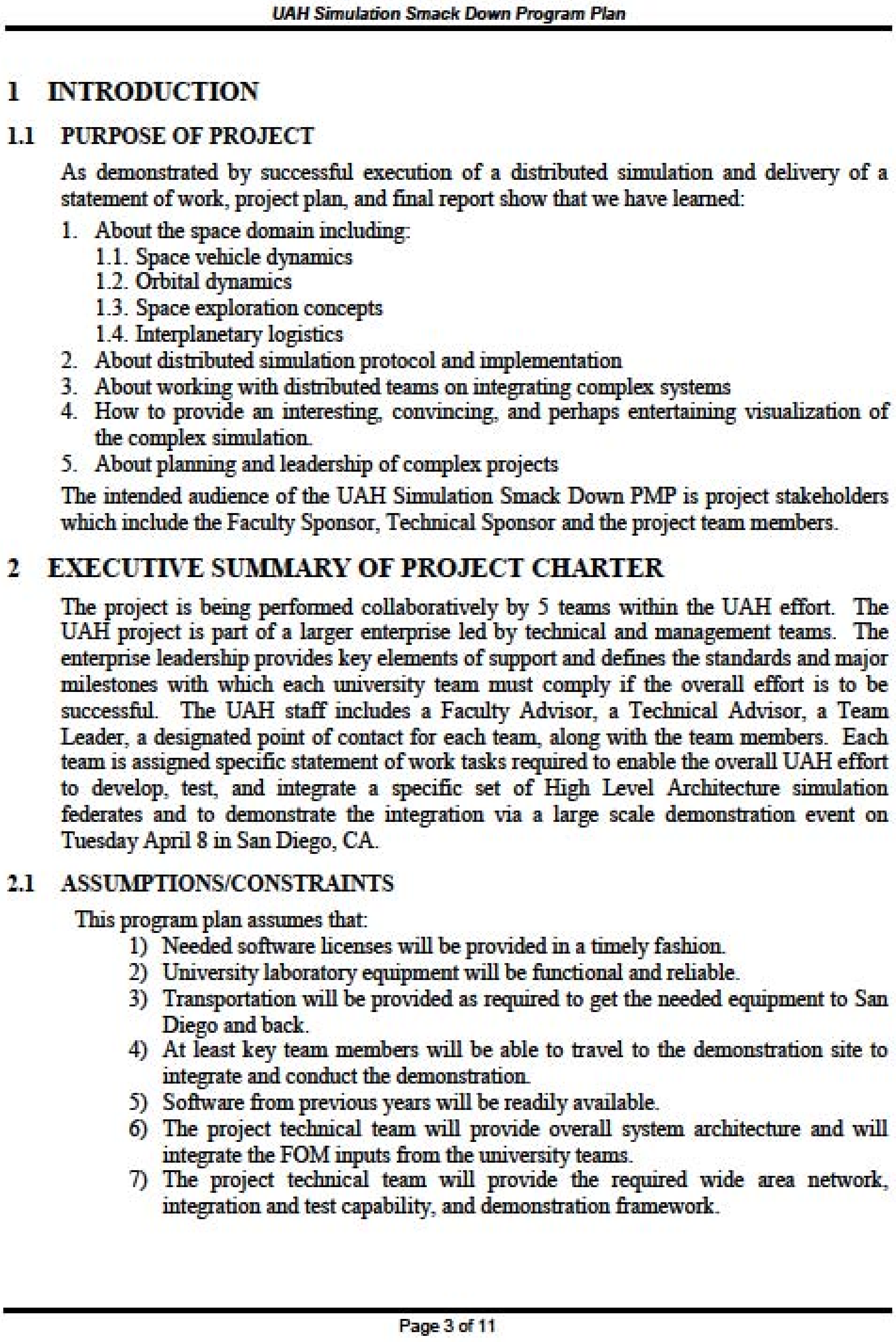
- We will manage the project well
- All team members provide significant participation
- All major milestones are met
- All teams demonstrate collaboration

UAH Smack Down Visualization Team Charter

Project Purpose Provide schedule and scope management, define architecture, requirements, and interfaces for the UAH Smack Down effort	**Key Project Stakeholders** Smack Down enterprise including other university teams, other UAH teams, Faculty Advisor, Technical advisor.
Team Objectives 1. Provide graphics and visualization to display all Federation objects, lunar surface and stellar background. 2. Develop a C++ interface between the existing MatLab federate and the Unity 3D MPC_Client.	**Team Principles** 1. Accomplish an interesting and effective demonstration on time. 2. Form effective and cooperative teams, both local and enterprise wide. 3. Develop creative, high quality, interesting products. 4. Ensure the healthy project by applying committed effort and adding value

Team Members	Role	Key Responsibilities
Tymaine Whitaker	Team POC Software Designer	Develop team charter, provide weekly status, issue status reports; ensure communication and cooperation with other teams. Implement Functional & Technical Specifications.
Daniel O'Neil	System Engineer	Develop Technical Specification/Requirements; Keep team abreast of technical considerations and potential problems.
Bill Tucker	Test Lead	Verify Functional & Technical Specifications

Key Measurements of Project Success	Close-out Criteria (the project is done when…)	Key Stakeholder Expectations
• Interesting and Entertaining on-time demonstration • Clear evidence of effective teamwork and •	• Current and accurate program plan and SOW published. • Effective and interesting demonstration conducted on time. • Final Report published.	• We will manage the project well • All team members provide significant participation • All major milestones are met • All teams demonstrate collaboration



1 INTRODUCTION

1.1 PURPOSE OF PROJECT

As demonstrated by successful execution of a distributed simulation and delivery of a statement of work, project plan, and final report show that we have learned:

1. About the space domain including:
   1.1. Space vehicle dynamics
   1.2. Orbital dynamics
   1.3. Space exploration concepts
   1.4. Interplanetary logistics
2. About distributed simulation protocol and implementation
3. About working with distributed teams on integrating complex systems
4. How to provide an interesting, convincing, and perhaps entertaining visualization of the complex simulation.
5. About planning and leadership of complex projects

The intended audience of the UAH Simulation Smack Down PMP is project stakeholders which include the Faculty Sponsor, Technical Sponsor and the project team members.

2 EXECUTIVE SUMMARY OF PROJECT CHARTER

The project is being performed collaboratively by 5 teams within the UAH effort. The UAH project is part of a larger enterprise led by technical and management teams. The enterprise leadership provides key elements of support and defines the standards and major milestones with which each university team must comply if the overall effort is to be successful. The UAH staff includes a Faculty Advisor, a Technical Advisor, a Team Leader, a designated point of contact for each team, along with the team members. Each team is assigned specific statement of work tasks required to enable the overall UAH effort to develop, test, and integrate a specific set of High Level Architecture simulation federates and to demonstrate the integration via a large scale demonstration event on Tuesday April 8 in San Diego, CA.

2.1 ASSUMPTIONS/CONSTRAINTS

This program plan assumes that:

1) Needed software licenses will be provided in a timely fashion.
2) University laboratory equipment will be functional and reliable.
3) Transportation will be provided as required to get the needed equipment to San Diego and back.
4) At least key team members will be able to travel to the demonstration site to integrate and conduct the demonstration.
5) Software from previous years will be readily available.
6) The project technical team will provide overall system architecture and will integrate the FOM inputs from the university teams.
7) The project technical team will provide the required wide area network, integration and test capability, and demonstration framework.

9 LESSONS LEARNED INTERVIEW RESULTS

Things I think would be useful to know:

1) HLAToolbox might be a nice option for those who have never worked with HLA.

2) When using Matlab, one can forcefully resign your federate by typing resignFederationExecution('DELETE_OBJECTS_THEN_DIVEST') into the Command Window. Likewise, one can destroy forcefully destroy a federate by typing destroyFederationExecution(“federationName”) into the Command Window.

3) Matlab has a problem ending a thread instance without terminating the program entirely. To prevent shutting down Matlab each time you want to end a run, type hlaDisconnect into the Command Window.

4) Federates should be designed to avoid hard dependencies on uncontrollable things like message ordering.

The Smackdown technical committee worked well. Issues were openly identified and responsively addressed leading to a technically successful event. They did the right thing with FOM modules. It would have been better if the technical team had been clear about what they students should expect from the committee, and what the committee expected from the students, including a description of the process. Their planning should have started with the operational scenario, recognizing it would like change. Perhaps a swim lane diagram of the planned entity interactions would be helpful. It should map to and from activities, assignments, and agents.

Participant presentations provided very little detail about what the team had actually accomplished.

Due to no fault of their own, the NASA representatives encountered last minute travel restrictions, placing the fundamental computing and networking basis for the event at risk. Plans should be developed to avoid this risk, as well as any other single point of failure – any dependencies that would lead total failure of the event.

Not all teams were aware of all educational opportunities. For example, last year Pitch offered to the teams a class on time management. Some universities were late in discovering the availability of training materials (and the need to use them!) A list of available training opportunities and materials as well as other support resources would have been helpful. A significant body of knowledge regarding prior year technical lessons learned was available to the students, but they did not seem to take advantage of it. At least two teams were not aware that their primary challenges were in the area of interoperability (vice model development) at the beginning of the process. Perhaps better guidelines are needed for schools thinking about participating. The federation agreement needs to be better and more publically documented. Better support for early off-line interoperability testing would be helpful, such as data driver federates. Test early and often.

The vendors were extremely supportive and responsive, including provision and support of licenses. However each vendors' process is a bit different, the students are not familiar and the computing environment was changing until the last minute. All of this combined to create confusion among the students, and an excessive version on the vendors. In the future it would be better if the computers to be used in the demonstration were selected very early in the process and did not change. Also, an integrated technical team coordinated single license request process would simplify the process. Obviously this integrated process would need to gather the signatures and other data required by each of the vendors, and need not require the active intervention of the technical committee. We do need to make the steps clear.

Planning for awards was ineffective. Requirements for things like type and number of awards and criteria by which teams would be selected were changing until right before the event. The quality of the awards (just paper) was not high. (Think about letting SCS or SISO do the awards). The judges had no opportunity to interact with the students before they were asked to make their decisions. As a result the judges had no way to assess the technical quality of the simulations, or the understanding level of the students. Consequently, the awards did not seem to emphasize the technical contributions and capabilities of the competing teams. The award process was conducted twice in order to accommodate last minute changes. We should provide an opportunity for the judges to meet with the teams

A full dress rehearsal complete with A/V and remote participants would have been very helpful.

A discussion of the scenario – mission to the moon, etc. – prior to the beginning of the demonstration would have been extremely helpful: Introduce the players and the simulations, and describe what is about to happen. The overview should address both pedagogical and scenario (mission) goals. An OV-1 describing the scenario would have been helpful. Perhaps a handout could be provided to give the judges (and attendees?) more background.

Simulation visualization was significantly improved compared to last year, but it could have been better used. For example the visualization of the asteroid mission was not available on the main screens.

Coordination between the Smackdown planners and the societies was not effective. The A/V requirements were not complete, and the requirements that were provided were not clearly configuration managed. Because the individuals involved in planning did not appear to have access to prior year information, prior years' experience seemed to have gotten lost. Requirements for venues need to be established months ahead of time. The names of the teams and student participants were not provided until after the conference had started.

Need to acquire A/V equipment, or include the requirement in the requirements used by the responsible society.

The allocation of responsibilities between and among the Societies and the Event planners was not clear. For example, SCS was responsible to manage the hotel contract, but they were not included in all discussions about the venue.

Although the vendors were extremely responsive, students had a hard time understanding the different vendor's needs, creating a great of extra work for the both.

The schools participating in the event did not have a common understanding of the educational resources available, creating unnecessary challenges. A list of suggested on line courses and references would be helpful, perhaps in the form of a syllabus.

A timeline with gates would help insure more compliance with the philosophy of testing early and often. Perhaps training materials could be identified to demonstrate ways to do testing very early in the software development process, such as test stubs and recorded data streams. Teams should record their data streams and send them to teams in other time zones to support their testing.

The venue was not suitable for the event. It was too big, cold and noisy. Audience members in the rear could barely see the two screens and hear the teams. For next year, we might want to consider a better venue and provide better audio/visual equipment.

The event should have a coordinator who understands every aspect of the program and of each team and who introduces the basic idea of the whole Smackdown event. Distribution of several-page handouts introducing HLA to them to the audience before the event might help. We need the audience to know we are doing something significant and worthwhile. Since graphics-wise, most of the simulation work does not appear very interesting and was not shown by the visualization federates, the event would require exposing more of the internal workings of the simulations to the audience. The audience might feel confused about the whole federation. So we definitely need to get the audience more involved in the event.

Now that the Smackdown event has grown to more than ten teams participating, we definitely need a better way to present our work to the audience. Here are my thoughts:

- Opening (Introduce teams, basic concepts of HLA and the two RTIs we are using: PITCH and MAK.)

- Start federates team by team (Let the audiences see how we populate the lollipop diagram by starting different federates from different teams so that they could get a big picture of the federation.)

- The coordinator provides a short description of the relationship and interactions between federates of different teams (UML diagrams are useful).

- Once their federation is executing and stable, each team in turn introduces its federate. This will help the audience can understand their work.(In the meantime, one projector shows the lollipop and the other one shows current team's the work)

- Wrap up by the coordinate by a brief summary.

- Break.

- Demonstrate on MäK (Maybe we need a different scenario on MäK so that the audiences don't feel bored.)

During the planning and preparation phase of the event, the tech committee should develop a detailed requirements specification for each candidate federate to be built. (The specification may be written in terms of a use case story so that each team can better understand what should be developed and what should not.)

The Simulation Smackdown Wiki site is really good but it could be better, I recommend the tech committee add a few more gadgets in it:

1. Set up a discussion forum where students from different teams can post their questions and exchange information. By publishing questions on the site, all the students can benefit from questions answered by experts from NASA, PITCH and MAK.

2. Each team should have a blog on the site where they can describe their federates, FOM and update their progress.

3. All the related materials (Time Management slides, HLA tutorials, PITCH installation manual and sample codes) should be uploaded and maintained in this wiki site. By doing so make all of the team members tightly connected by this Simulation Smackdown wiki site. Google Site service is really helpful in this perspective.

Last but not the least, simple but comprehensive sample code should be provided to each team at the beginning of the project and walked through by instructors or experts from NASA. For example, the environment federate code is really excellent. However the size of this program might intimidate new comers, so perhaps a simpler but still comprehensive example should be written for educational purposes. It should illustrate how to connect to RTI, how to advance time, how to separate HLA stuff and business logic, etc.)

We have used the Google site for several courses; one important thing about this is make sure students subscribe to the site using their frequently used email, so they can get all the notifications, updates, questions and answers from the site. (Students are not willing to check the site often, so directing all the changes on the site to their email and then follows the link to the site is better approach.)

The bar opened late and closed early.

Having waiters distribute the food was nice, but was distracting.

Consider having hosting Society manage creation of award certificates.

The decision making process for the executive team appeared to be opaque. It was not at all clear how decisions were made, or how one might present an issue for resolution. This combined with the lack of a clear definition of the relative responsibilities of the executive team and the technical team created unnecessary confusion. Further, communication from the executive team was not as effective as one would hope. People did not always seem to get the information they needed to accomplish their mission, particularly regarding changes in direction that affected their planning.

7.3 Lessons learned during the development of LCANSat2013 federate.

Needed improvements and recommendations for future development:

1. The development of LCANSat2013 benefits a lot from 2012 UAHuntsville team's legacy code. Building on top of those nicely written and documented code greatly reduced our developing effort.
2. The starting time of the animation of Insight3D should be obtained from the JSC's environment federate as soon as our federate joins the federation, however in this year's event, we fixed it at 7:00pm PT, April 10, 2013. This makes the positioning of moon centered inertial(MCI) physical entities inaccurate, since the translation of a Cartesian coordinate from MCI to MCF relies on the accurate federation time in Terrestrial Time.
3. Walker constellation is good enough to provide communication support over the entire moon. Next year's team could explore if STK provides the capability to find the right path among the 14 Walker constellation satellites in order to deliver a message from a source to destination.(Maybe the source is at Aitken Basin and destination is at Hadley Rille). If they do not provide, we could use a graph-related algorithm to implement this feature.
4. LCANSat2013 federate propagates the satellites over a period between 7:00pm PT, April 10, 2013 and 7:00pm PT, April 11, 2013 before joining the federation. Besides propagating over intervals in advance, it's also possible to manually step the propagator one integration step at a time by calling takeStep method every time the time advance permit is granted.
5. Next year's team could also explore if STK provide the terrain information of Aitken Basin, if so, we our LCANSat2013 could not only model each entity as a 2D marker, but use real 3D model to simulate the surface of the moon at Aitken Basin. In this year's Smackdown event, we assumed the moon to be an idealized, spheroidal model; actually it might also consider terrain features such as mountains or ditches.
6. If next year's team plans to use STK to model the physical entities in 3D model. The orientation of the entities must be considered, so besides the three attributes of PhysicalEntity, our federate also needs to subscribe to the orientation attribute.
7. Do integration test as early as possible. Both on PITCH and MAK. I really recommend MAK setting up a web viewer like PITCH does.