### Hashtags: #asteroid, #asteroidsim

Contact: [[email protected]](http://www.cloudflare.com/email-protection)

### Tags: Model

**Challenge Description**

Asteroid missions come together thanks to space mission simulators, physics-based game engines, asteroid databases, and data visualizers. Each application excels at certain aspects of a mission: one simulator is great at propulsion, another provides the capability to create the spacecraft cockpit, and the asteroid databases provide details about the ephemera. However, mission simulations can be limited to a single computer. Though a few simulators and game engines provide a networking capability, they can only exchange data with other copies of the same application. This challenge is to create a solution that would integrate a variety of simulators, game engines, databases, and data visualizers into a mission simulation that is distributed across the Internet.

The High Level Architecture (HLA) protocol enables exchange of data among computer models at locations around the Internet, and the production of distributed simulations. Create space mission simulators, asteroid databases, and data visualization applications with an HLA interface.

**Background**

Originally developed by the Department of Defense (DoD), standardized by the Institute of Electrical and Electronics Engineers (IEEE), and maintained by the Simulation Interoperability Standards Organization (SISO), the IEEE 1516-2010 High Level Architecture (HLA) Evolved standard enables communication among computer models, or “federates”, that join a “federation” via a Run Time Infrastructure (RTI). Most university libraries have a copy of IEEE-1516-2010. The PoRTIco Project provides a free 2000 version of the HLA RTI. Open HLA (OHLA) is another free RTI.

Space mission simulators, asteroid databases, and data visualizers provide Application Programming Interfaces (API), or telecommunication functions. Using code libraries provided by the RTI and commands specified by the API or documentation, developers can produce a reusable HLA interface between the application and a federate.

Activities associated with this challenge include identifying space mission simulators, game engines, asteroid databases, data visualizers that offer an Application Programming Interface (API) or a communications link that could benefit from an HLA interface. You can use NASA’s General Mission Analysis Tool (GMAT), other simulators and physics-based game engines, and asteroid databases such as the Minor Planets Center Orbit Database (MPCORB). Developing HLA interfaces for these systems would enable the integration of distributed space mission simulations.

Write Java, C++, or modeling code to implement an interface between the simulator, asteroid database, or data visualizer, and an HLA federate. Demonstrate the HLA Interface by producing computer simulations or “federates”, a federate publishes and sends data and another federate subscribes to and receives the data. If the HLA interface is for a space flight simulator then an associated federate can get data from the simulator and send the data to another federate that displays the data as text, or a graph, or drive the position of 3D objects in a display. If the application is a data visualizer, the associated federate receives data from another federate and updates the visualization.

**Solution Ideas**

Here are some ways for you to frame this solution:

Source code for the HLA interface to space mission simulator, asteroid database, or data visualizer which includes an interface with a modular design to enable reusability; data with location, orientation, heading, velocity; source code with comments and documentation; and language compatible with the RTI, e.g., Java or C++. Integrated demonstration with two HLA federates, if possible, that send and receive federates; exchange data between federates with an object name, an identifier, position, orientation, heading, and velocity; and receiving federate displaying incoming data either as text, a graph, or animation of a 3D model; and procedure documentation.

**Sample Resources**

## IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules: <https://standards.ieee.org/findstds/standard/1516-2010.html>

* <http://en.wikipedia.org/wiki/High_Level_Architecture>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20010016107>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20120015629>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20070006472>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20070030309>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20110014007>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20110011924>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20120016767>
* <http://ntrs.nasa.gov/search.jsp?print=yes&R=20120016768> <http://ohla.sourceforge.net/>
* <http://porticoproject.org/index.php?title=Main_Page>
* <http://pitch.se/downloads>
* <http://ptgrogan.scripts.mit.edu/fundms/day4.php>

Space mission simulators, game engines, and asteroid databases:

* <http://gmat.gsfc.nasa.gov/>
* <http://ssd.jpl.nasa.gov/sbdb_query.cgi>
* <http://neo.jpl.nasa.gov/orbits/>
* <http://minorplanetcenter.net/physical_db>