Lockheed Martin ATL
Aron Rubin <arubin@atl.lmco.com>
February 5, 2003

What is the MSI?

The Multi-Simulator Interface (MSI) is a simulation interconnection engine. In other words it is a program that connects simulations together by synchronizing their clocks and data. The MSI serves the same purpose as HLA and supports most of HLA's functionality (and more). In that respect it can be considered an HLA alternative. The major motivating factors in the design of the MSI are speed, interoperability, and ease of use.

How does MSI compare to HLA?

The MSI was originally created to be just a light weight HLA RTI. However, as it was written, limitations in HLA were discovered. The MSI is an improvement on both the design and implementation of HLA. Some highlights include:

- A 1,536 to 1 reduction in size over the publicly available (until late 2002) HLA RTI
- The ability to subscribe to a object name in addition to a type
- Time synchronization that allows for proper causality when used with discrete event simulators.

MSI Setup and Use

The MSI uses a XML stream through a direct socket connection for communications. This enables the MSI to be used from any programming language that can use sockets (C, C++, Java, Ada, Lisp, Perl, etc.). Also, the MSI was written with cross-platform libraries that make it portable to all the major OS platforms (Linux, Solaris, Mac OS X, Microsoft Windows, IRIX, HPUX, etc.).

The MSI is a single executable file and is distributed with example code for the simulator/federate side interface. The MSI requires GLib (http://gtk.org) and GNet (http://gnetlibrary.org) to run.

MSI Time Synchronization

The MSI time synchronizer can mix unconstrained with time constrained simulations. When synchronizing multiple constrained simulations the MSI follows this procedure:

1. Each simulation/federate processes the events for the current time.
2. Each constrained simulation reports the time of the next event that will occur in that simulation/federate. This time may be artificially inflated to cause loose synchronization (less overhead but less guarantee of accuracy).
3. The MSI determines the time of the next event for all federates by taking the minimum of the reported. The MSI then announces the time to all simulations/federates.
4. The simulations/federates will advance to the announced time.
5. Go to step 1.

MSI Data Synchronization

The MSI implements a publish/subscribe data broker. The MSI is presently not validating, therefore it does not require a separate data format specification (like the HLA FOM). When data format validation is implemented, it will be an optional feature and not written in Lisp. This greatly reduces MSI's setup time. Also, not being locked to a predetermined data format allows for dynamic data types.

There are four commands associated with the MSI data broker – publish, subscribe, update, destroy. Simulations/federates may subscribe to object names in addition to object types. This allows simulations to subscribe to specific objects of a type without needing to receive updates of all objects of that type. The update command is both an incoming and outgoing command. When a simulation/federate receives an update command, it is expected to reflect the new values of that object.

MSI Messaging

The MSI allows simulations/federates to send messages (interactions in HLA) to each other. These messages can contain multiple attributes and be multicast to a specific group of simulations.

Recently Added Features

- Changed the XML parser to allow objects and messages to contain hierarchal data structures.
- Limited subscriptions – subscriptions to specific attributes of an object

Planned Features

- IP multicast option
- Breakpoint handling
- Federate-in-a-file to support canned scenarios and static data
- Conditional subscriptions – allows a federate to specify a predicate that the MSI will reevaluate as object values change.