

Multi-Simulation Interface (MSI) for Complex Simulation

April 5, 2006



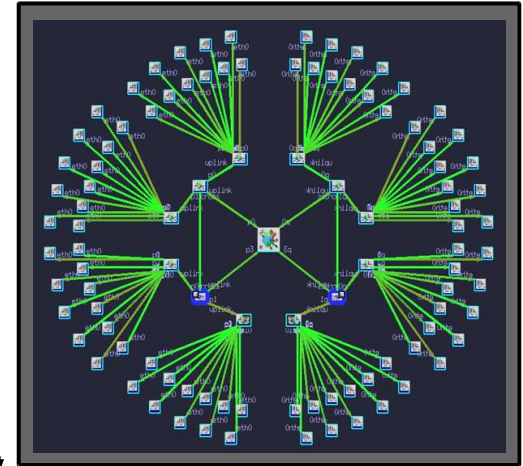
LOCKHEED MARTIN 

Aron Rubin, Carl Hein, and Guru Prasad
Advanced Technology Laboratories
Cherry Hill, NJ
{arubin, chein, gprasad}@atl.lmco.com

Overview

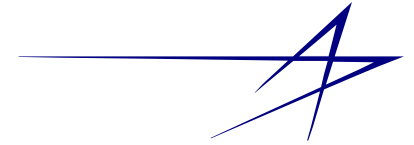
Distributed Simulations

- **Complex simulations are often realized by interconnecting multiple distinct simulators.**
- **Distributed interactive simulation has long history of methods.**
- **In 2001 Lockheed Martin Advanced Technology Laboratories (LM ATL) used DMSO HLA for distributed faster-than-real-time (FTRT) simulation of 50,000-node communication network.**
 - **HLA traffic saturated LAN, preventing performance objective.**
 - **Light-weight alternative, MSI, developed to meet requirements.**

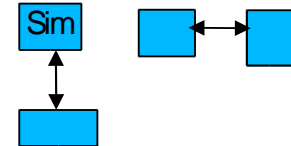


Background

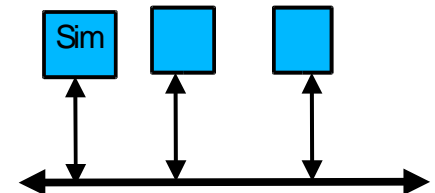
Simulation Interconnection Frameworks



- **Ad-hoc/point-to-point: From before 1980's through current day.**
 - Custom interfaces for each pair of simulators.
 - Not easily maintained, nor scalable.

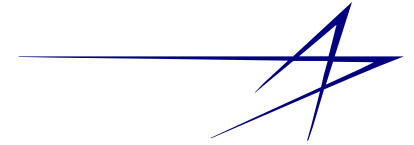


- **SIMNET and DIS (Distributed Interactive Simulation): 1980s**
 - Standard message formats broadcast to disseminate state/event information.
 - Aimed at loosely synchronized training simulations.
 - Real-time only.
 - Each simulator processes every message.
 - Not efficiently scalable to large federations.

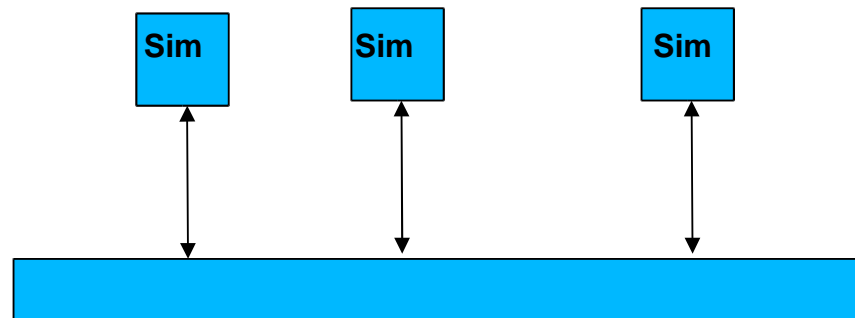


Background (cont.)

Simulation Interconnection Frameworks



- **ADS and ALSP: Early 1990s**
 - **Advanced Distributed Simulation project led to Aggregate Level Simulation Protocol.**
 - **Three advantages over DIS:**
 - **Centralized clock management.**
 - **Common data representation system.**
 - **Architecture independence.**

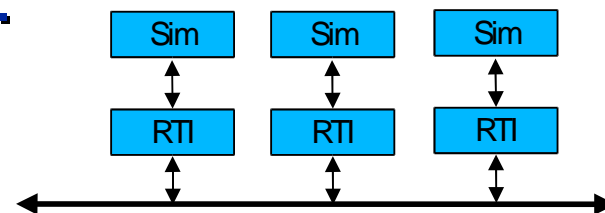


Background (cont.)

Simulation Interconnection Frameworks (cont.)

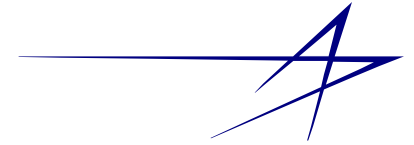
- **HLA (High-Level Architecture): Circa 1996**

- **Developed by Defense Modeling Simulation Office (DMSO).**
- **Sophisticated re-usable technologies encapsulated in Run-Time-Interface (RTI).**
- **Includes time and data management.**
- **Publish/Subscribe method distributes event data only where needed; ==> more scalable.**
- **Initial DMSO-funded reference release was binary-only, quickly obsolete.**
- **OMG and IEEE adopted; IEEE Std. 1516.**
- **Several commercial vendors now support licensed versions, but cost can be prohibitive.**



Background (cont.)

Simulation Interconnection Frameworks (cont.)



- **MSI (Multi-Simulation Interface): 2001**

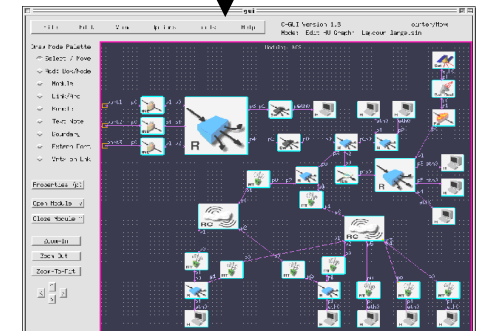
- Although HLA was well-designed, some usability, performance, and availability issues.
- MSI addresses some of the issues.
- Light-weight HLA-like simulation-interconnect engine.
- Free Open-Source, Un-encumbered Lessor-GPL (LGPL); hosted on SourceForge.
- XML based.
- <http://msi.sourceforge.net/>



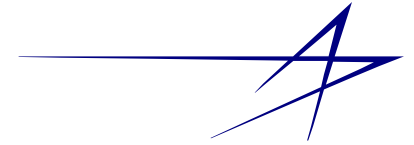
Overview (continued)

MSI (Multi-Simulator Interface)

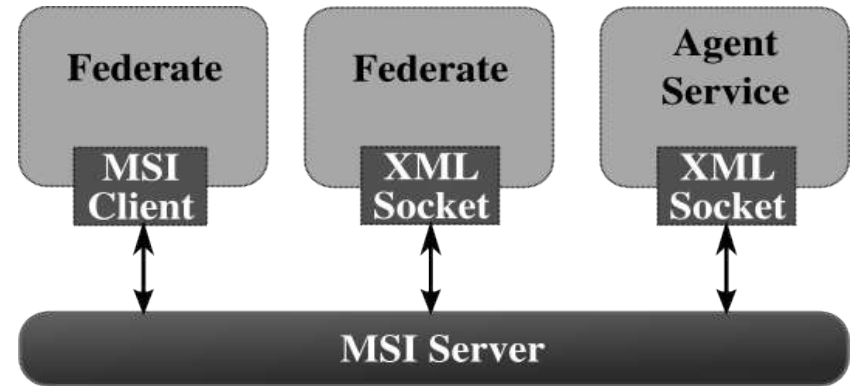
- **MSI (Multi-Simulator Interface)**
 - Three order of magnitude reduction in code size/complexity.
 - Exhibited order-of-magnitude reduction in LAN traffic.
 - Supports new and diverse simulation types, such as: agent-based, social/economic modeling, and system-of-system simulations.
- Continued MSI developments provide an alternative addressing: complexity, availability, scalability, support, standardization, and most importantly—cost of building complex multi-domain systems.
- Presentation reviews distribution technologies, MSI features.



MSI Features



- **Based on XML socket stream.**
- **Usable from any programming language – C, C++, Java, Ada, Perl, etc.**
- **No library dependencies.**
- **Cross-platform code, portable to all major OS platforms (Linux, Solaris, Irix, HP-UX, Mac OS X, Microsoft Windows, FreeBSD, etc.).**
- **Provides managed federation start-up (join) control.**
- **MSI is a single executable file.**
- **Distributed with example code for simulator (federate) side interface.**



MSI Description

Concepts

- **Data exchange formats**

- Many prior frameworks force clients to convert all shared data to neutral formats.
- Meta-data is added to each exchange, enabling envelope parsing; ==> Verbose.
- High processing and bandwidth costs to transform and pack all data.

- **MSI Rosetta Stone approach**

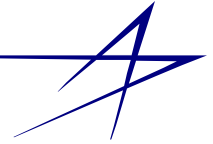
- Instead of transforming data on input, MSI leaves data in original form for efficient transfer.
- MSI client-side library transforms data to best local form.
- If receiving simulation can parse incoming data, then no translation overhead incurred.
- XML enables context insensitive parsers and transformability.
- XSLT can specify arbitrary transforms; ==> Supports direct inter-ontology mappings .



Direct face-to-face mappings

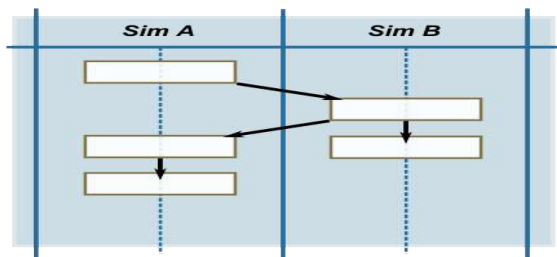


Lean Time & Data Management



- **Maintaining Causality**

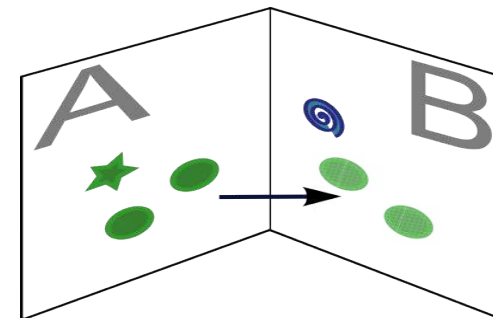
- Temporal ordering of events must be managed among distributed federation.
- Data and events may be cached to simplify federation start-up; asynchronous joining.
- MSI supports conservative and loose time management.
- Each federate reports time-of-next-event (TONE).
- Global Virtual Time (GVT) enables advancement as it is safe to do so.



Temporal Causality

- **State Data Management: Reduces Network Bandwidth**

- MSI enables subscription to object states by subfields and/or instances, not just types.
- Data caching of existing states also minimizes network traffic, as well as federate accesses.
- Dynamic data culling by clients; filtering changes with time and conditions.

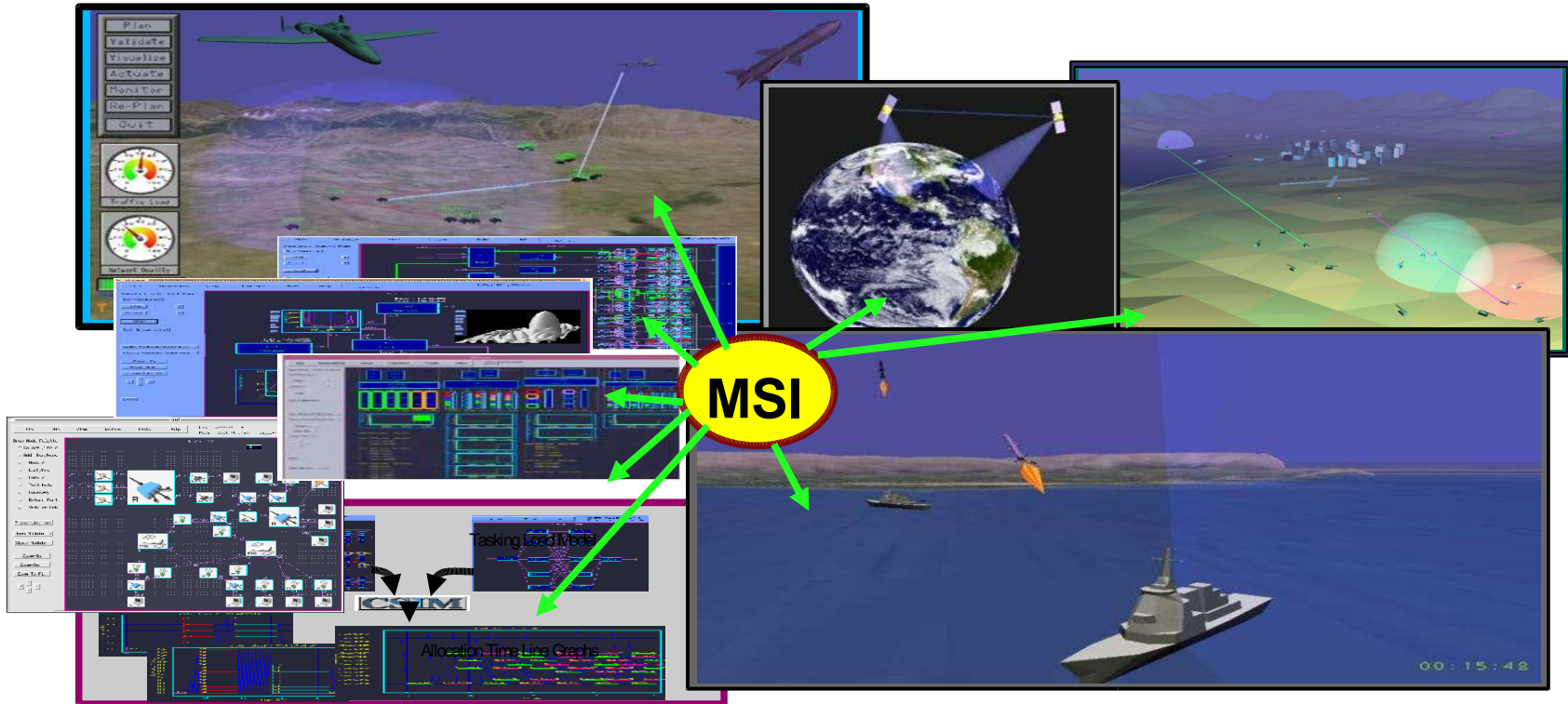


Data Coherency



MSI Conclusions

- Simple powerful tool for connecting simulations; small number of simple interfaces.
- Advances and updates prior efforts.
- Balances learning curve, complexity, and efficiency.
- Enables new kinds of distributed systems, mixed abstraction levels, systems-of-systems simulation.
- Data transformation approach facilitates agent-based, service oriented systems, etc..



<http://msi.sourceforge.net/>

