JDEP Information Brief
Understanding HLA and DIS

February 2003

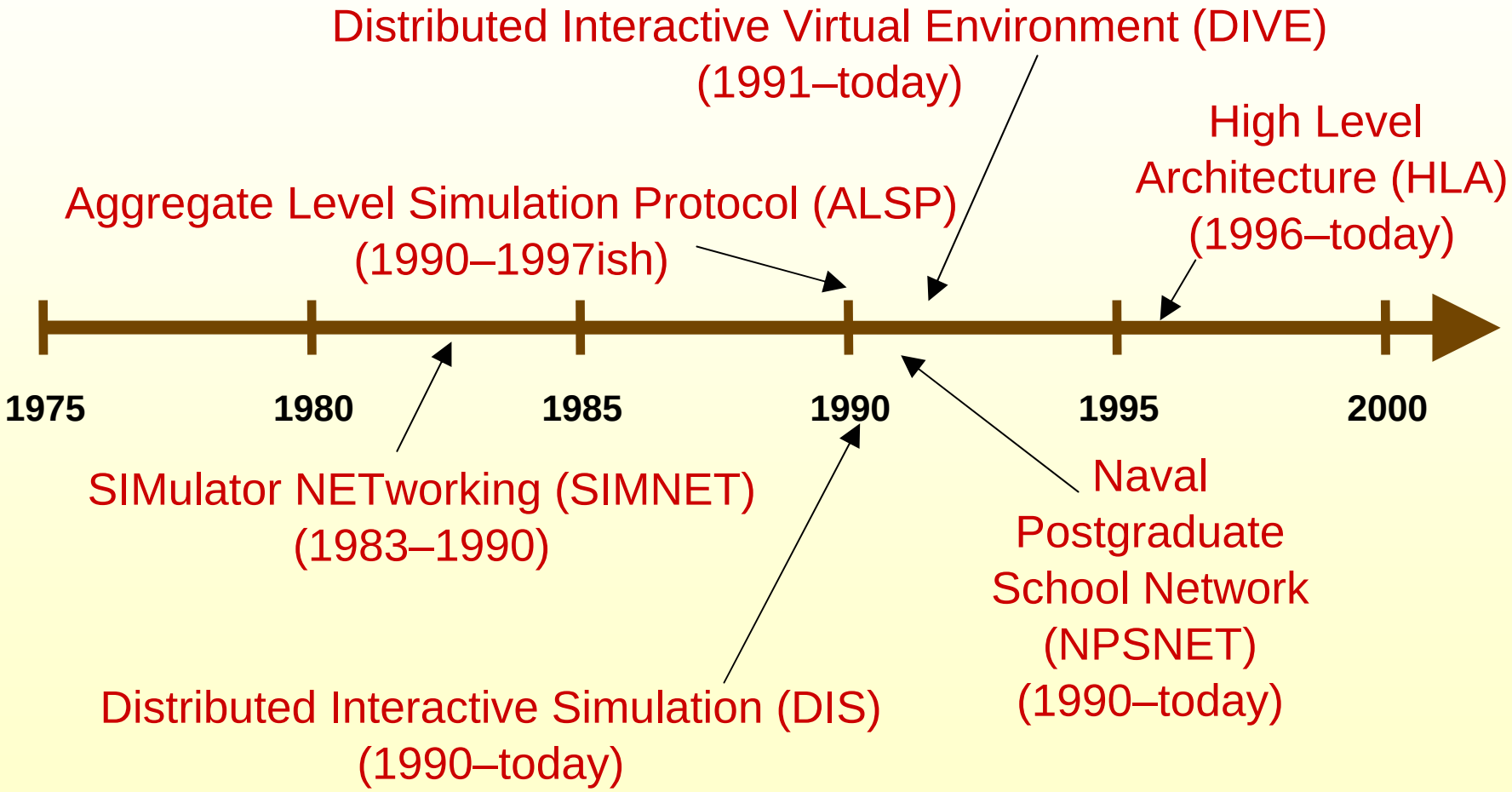
Contents

- History of Simulation Technologies
- Comparison of Distributed Interactive Simulation (DIS) and High Level Architecture (HLA)
- HLA Features and Issues
- JDEP Framework Work-in-Progress
- Future Uses of HLA in JDEP

Evolution of the High Level Architecture (HLA)

- **HLA incorporates simulation technology developed over the last 20 years**
- **HLA's predecessors include**
  - **Distributed Interactive Simulation (DIS)**
  - **Aggregate Level Simulation Protocol (ALSP)**
  - **Many other R&D distributed virtual environments and parallel simulation systems**

Historical Perspective



Distributed Interactive Simulation (DIS)

- Evolved from SIMNET
- Designed to support loosely coupled training exercises on local area networks
  - Very successful in this arena
- Did not scale to support large exercises on wide area networks
- Close interactions are difficult to support

Simulation Complexity Scale

200+ ms	Infrequent and imprecise interaction (e.g., observing faraway and/or slow-moving objects)
100 ms	Interaction at moderate ranges or on moderate time scales (e.g., observing fast-moving tanks at 100+ meters)
50 ms	Interactions at close ranges on short time scales (e.g., formation flying and space station docking maneuvers)
Low latency and specialized network services	Tightly coupled close interactions and complex interactions between numerous systems (e.g., short-range multi-ship air-to-air engagements and constructive and interactive war games such as Combined Arms Support Task Force Evaluation Model [CASTFOREM] and Ground Warfare Simulation [GRWSIM])

Aggregate Level Simulation Protocol (ALSP)

- Incorporated time management technology to ensure a consistent world state in the face of
  - Out-of-order message delivery
  - Widely varying latency
- Interest management technology was incorporated to avoid over-loading networks and processors when there are numerous participants

HLA Services Specification

Service	Functionality
Federation Management (Implemented for JDEP)	**Create and delete federation executions.** **Join and resign federates.** **Control checkpoints, pause, resume, and restart.**
Declaration Management (Implemented for JDEP)	**Publish and subscribe to object attributes and interactions.**
Object Management (Implemented for JDEP)	**Create and delete object instances.** **Send object attribute updates and interactions.** **Create and delete object reflections.**
Ownership Management	**Transfer ownership of object attributes.**
Time Management	**Coordinate the advance of simulation time and its relationship to real time.**
Data Distribution Management	**Support efficient routing of data, usually by space-based interest management.**

Comparison of HLA/DIS

- DIS was (relatively) plug-and-play
  - Adhere to the DIS protocol & be interoperable
  - Not much interest in DIS/DIS gateways
- HLA is not plug-and-play! There is no HLA "protocol," only a standardized set of services
  - Standard defines how services behave and how they are invoked
  - There is no standard for service implementations
  - Use of services within a federation depends on federation design
  - Federations designed for a particular purpose

DIS Has Narrow Scope

- **Small-scale exercises on LANs**
- **DIS is inflexible**
  - **Use outside its domain requires careful engineering of special-purpose solutions**
- **Careful design choices need to be made about what can be distributed and what must be collocated**
- **In cases of very small latency requirements, critical systems must be collocated, e.g., TMDSE.**

What about TMDSE?

"… a TMDSE test configuration [is] geographically distributed over great distances. Transmission times can introduce data latencies that would not exist in the real world. To reduce these latencies and afford real time operation, components … are collocated with the … systems they serve." *[Theater Missile Defense System Exerciser, Build 3, Version 3.0+, Allocated Requirements Document, Revision 1]*

HLA has more possibilities

- **HLA encapsulates sophisticated distributed computing services**
  - **Simplifies and speeds up the design and construction of complex distributed sims**
- **Federation designer can make trades**
  - **Consistency of global state/real-time performance**
  - **Federate Object Model (FOM) design can be simple (Real Time Platform Reference [RPR]-FOM), up to very complex (world divided into areas of interest to facilitate Data Distribution Management [DDM], partial object ownership, and object/interaction specific consistency rules)**

Comparison of HLA/DIS (cont)

- DIS is difficult to apply outside its intended area
  - NPSNET, DIVE, ALSP, and numerous other commercial, government, and R&D projects continuing research in consistency and scalability of virtual environments
- HLA provides technology to build simulation systems that could not be built using DIS
- HLA provides an advantage only if we build federations, not just collections of federates

Federation Management

- **DIS**
  - **Provides message formats and rules for their application. Burden is on the simulation developer to implement management features**
- **HLA**
  - **Provides some automated simulation management (e.g., unique federate names, synchronization points, and health and status information)**

Declaration Management

- **DIS broadcasts all messages**
  - **Controlling distribution is typically done at the network level (i.e., port assignments)**
  - **Bandwidth and computing resources are consumed when processing useless data**
- **HLA has built-in interest management based on object and interaction type descriptions**
  - **A federate can subscribe to object/interaction types it requires**
    - **Run Time Infrastructure (RTI) will only deliver message types that are of interest to the application**
    - **Can save computing power and network bandwidth**

Object Management

- DIS
  - Requires applications to generate entity identifiers using predefined Site and Application numbers plus an entity counter
  - Uses special messages or timeouts to remove entities from the game. Timeouts can result in late removal and special messages can be lost.
- HLA
  - Generates federation-wide unique entity identifiers at runtime
  - Provides reliable and rapid entity deletion services

Object Management (cont.)

- **HLA does not need periodic broadcasts of entity state to facilitate federate entry**
  - **New federates are provided with a list of known entity identifiers**
  - **Can request state information as needed**
  - **Potential savings in bandwidth and computing power**

Ownership Management

- **DIS is Ownership Management capable**
  - **Only through systems engineering agreements between federates**
- **HLA has it Built-in**
  - **Resolution of ownership conflicts**
  - **Services to initiate, negotiate, and resolve entity attribute ownership**
  - **Coordinated with object deletion services (ensures only owner can delete an entity)**

Time Management

- DIS only runs in real time
- HLA can run slower/faster or in real time
  - Federates consisting of pure simulations can run years' or weeks' worth of simulations within hours
  - Hardware-in-the-loop (HWIL) requires real time

Problems with Real-time

- HLA and DIS can be comparable in performance
- Biggest problems are with consistency of world view between distributed federates
  - Latency-induced errors
    - Reduce network loads via interest management
  - Causality errors due to out-of-order delivery (effects preceding causes)
    - HLA supports causally ordered message delivery
  - Lost data due to dropped packets
    - HLA supports reliable protocols

Data Distribution Management

- **Not a feature of DIS**
  - **Some experiments with use of multi-cast groups assigned to Areas of Interest (AOIs), difficult to implement – needs middleware**
- **HLA supports routing spaces**
  - **Geographic or logical areas of interest**
  - **Federates only receive events occurring in their AOI**
  - **AOI might be defined by sensor or communication limits**
  - **Save bandwidth and computing power!**

Run Time Infrastructure (RTI)

- **Basically acts as a distributed O/S**
  - **Is just one component of HLA**
- **RTI software**
  - **VTC/SAIC (based on DMSO RTI-Next Generation [NG])**
  - **Pitch's pRTI (portable RTI – only 1516 available)**
  - **MAK RTI (DMSO-based standard, proposed 1516)**
  - **Mitsubishi is building an RTI (proposed 1516)**

How to Choose an RTI

- **Different RTIs implement the HLA services in different ways**
  - **Optimized for particular types of networks**
    - **FDK (from Georgia Tech) developed for use on local area networks. Extensive use of multicast groups.**
    - **pRTI (from Pitch) is designed around point-to-point connections. May improve portability (TCP/IP is everywhere), e.g., for use on the Internet.**
  - **Optimized for particular applications (e.g., training, high-performance computing, and large distributed virtual environments)**

How to Choose an RTI

Different RTIs are not interoperable

- **RTI provides a standardized Application Programmer's Interface (API) and distributed computing services for use by application developers**
- **Standard API (both semantically and syntactically)**
- **No standards for implementing services (e.g., centralized vs. distributed implementations, TCP/IP vs. reliable multi-cast)**

RTI Evaluations

- **Benchmarking the various RTIs is a good idea.**
  - **Requires a careful understanding of a vendor's design in order to make a good implementation.**
  - **We have set up computers in the JDEP lab for testing RTIs and FOMs.**
  - **These configurations are then imported to NETWARS/OPNET for simulation. We can make changes quickly to the NETWARS/OPNET models and run simulations. These simulations are then the basis for new configuration trials.**

Reuse and Interoperability

Focus on flexible federate architectures

- **FOM agility**
  - **Software designs should make sharp distinction between internal and external data representations**
  - **FOM changes should not be difficult so long as they still reflect the underlying capabilities of the federate (days, not weeks or months)**
- **Separation of concerns**
  - **Internal vs. external data representation**
  - **Internal vs. external time management scheme**
- **Standard APIs**
  - **1516 specifies a normative API**
  - **HLA toolkits can reduce federate development time and promote flexibility**

RTI selection
Part of federation design process

- **Match RTI capabilities to federation requirements**
- **Flexible federates should be able to switch RTIs**
  - **Re-link or recompile**
  - **Standard APIs are key**

Reuse and Interoperability (cont)

- If they are going to be reusable, Federates must be adaptable
- Expect software maintenance costs over the lifetime of the federate
- Good systems engineering practices can significantly reduce maintenance costs

JDEP Event - MSCT

- **Used HLA 1.3 NG version 6 RTI**
  - **JRPR FOM variant**
    - **Derived from MC 02 FOM and RPR 1.1**
- **Nodes were "gatewayed" to HLA**
  - **Used HLA as a backbone to connect disparate DIS implementations**
  - **Gateways provided a limited solution**
    - **Provided connectivity to legacy equipment**
    - **Could not used capitalize on HLA services**
      - **Time management, Distributed Data Management, etc.**

JDEP Event E-2C

- **Top Down Federation Design**
  - **Used JDEP Reference FOM specifically designed to support Test Objectives**
  - **Event Planners followed FEDEP Process**
  - **Used HLA 1.3 NG version 6 RTI**
    - **Majority of federates were native HLA (I.e., Scenario Driver, HLA Results, IFF driver, E-2C Simulator)**
- **DLS "gatewayed" into Federation**
  - **The native HLA remote tracks were converted to DIS via a gateway. These remote track data were then inserted into the E2C mission computer from the DLS.**

Future use of Expanded RTI services

- **Time Management**
  - **In cases such as the CRS/D, we can use the RTI Time Management Services to increase the number of updates per second to address Theater Ballistic Missile (TBM) requirements.**
- **Distributed Data Management**
  - **Geographic and AOI to reduce bandwidth usage. Only certain entities would receive information based on either their location or need for information.**

Future Uses of HLA in JDEP (cont)

- **Exercises to date have been monologic**
  - **Need to consider the consequences of more dialogic interactions**
  - **Monologic refers to exercises in which information is essentially moving one-way.**
  - **Dialogic refers to many systems exchanging information. In general, dialogic exchanges will require careful planning and design to overcome the increased bandwidth and latency demands.**

Summary

- **HLA has many useful, unused capabilities**
  - **We are still learning and gaining experience on implementing HLA**
  - **We haven't yet take full advantage of everything HLA has to offer**
- **HLA and DIS offer varied functionality**
  - **The best technical solution needs to be engineered for each event**