Testing Net-Centric Systems of Systems: Applying Lessons Learned from Distributed Simulation

Presentation to the NDIA Systems Engineering Conference

October 26, 2005

Doug Flournoy
rflourno@mitre.org

Elizabeth Lee
elee@mitre.org

Robert Mikula
rmikula@mitre.org

MITRE

Outline

- Motivation
- Net-Centric Computing & Testing Challenges
- Distributed Simulation Computing & Similarities to Net-Centric Computing
- Distributed Simulation Test Experiences and Tools
- Applicability to Net-Centric Services Testing
- Findings & Recommendations

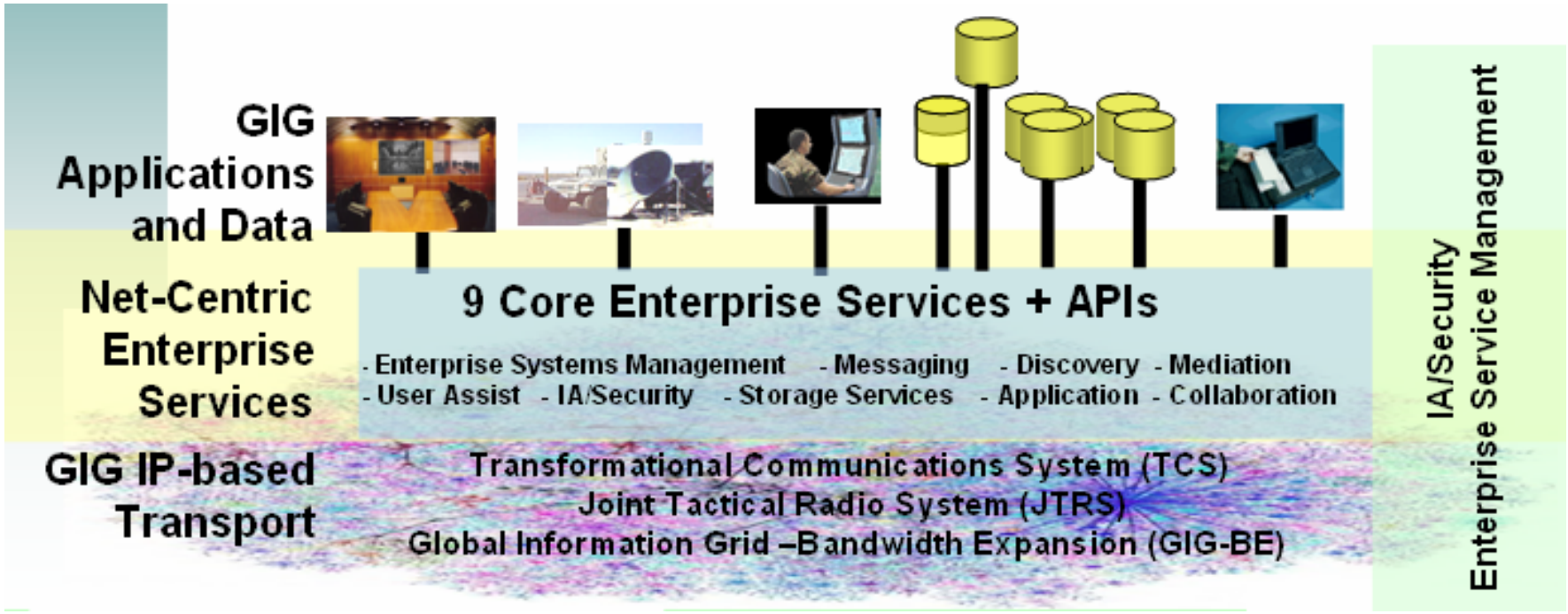
MITRE

Motivation

- Over the next several years, DoD will begin fielding components of the Net-Centric Enterprise Services (NCES) and Global Information Grid (GIG)
  - Testing these Service Oriented Architecture (SOA)-based capabilities will require new techniques and tools beyond those used for traditional platform-based systems

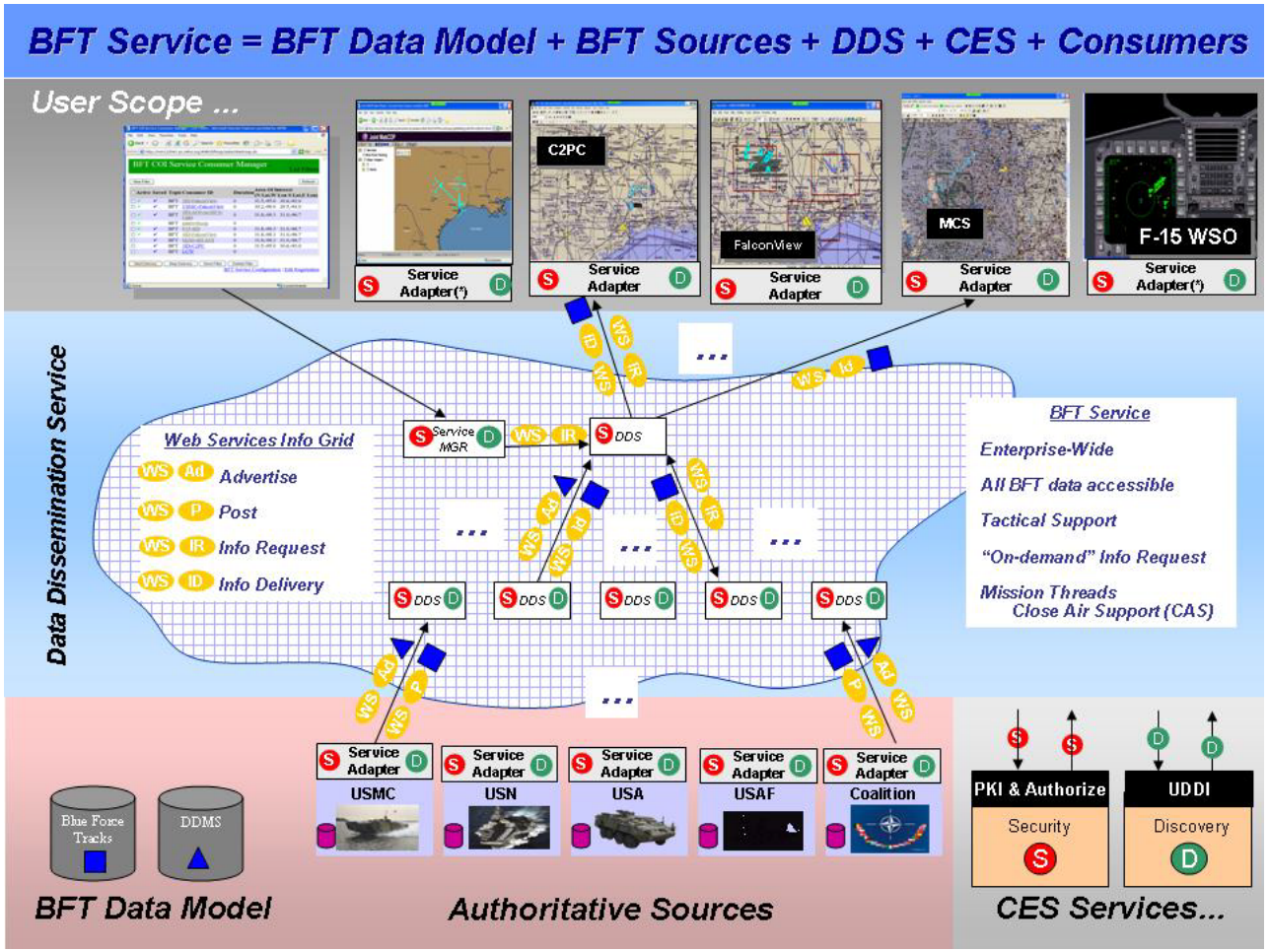
- Hypothesis: *an approach to testing net-centric systems can be formed based on successful experiences testing service oriented distributed simulation systems.*
  - This briefing presents the findings of a MITRE IR&D study that examined the potential for simulation test methods and tools to address net-centric test challenges

Trends in Net-Centric Computing: The GIG and NCES



MITRE

Trends in Net-Centric Computing: Communities of Interest (COIs) and the Blue Force Tracking (BFT) COI



BFT COI Architecture (from *Blue Force Tracking (BFT) Community of Interest (COI) Service (v1.0);* F. Wildes, K. Kelley, and P. Kim; MITRE Working Note: WN 05W0000001, Dec. 2004.)

MITRE

Challenges Associated with Testing SOAs: Why is it hard?

- Rapidly Evolving Standards
  - limits potential choreography between services
- Rapidly Evolving Core Services
  - many still in early prototyping phase
- Organization of Registries
  - the right service is out there somewhere, can you find it?
- Service Pedigree
  - once you find a service, can you trust it?

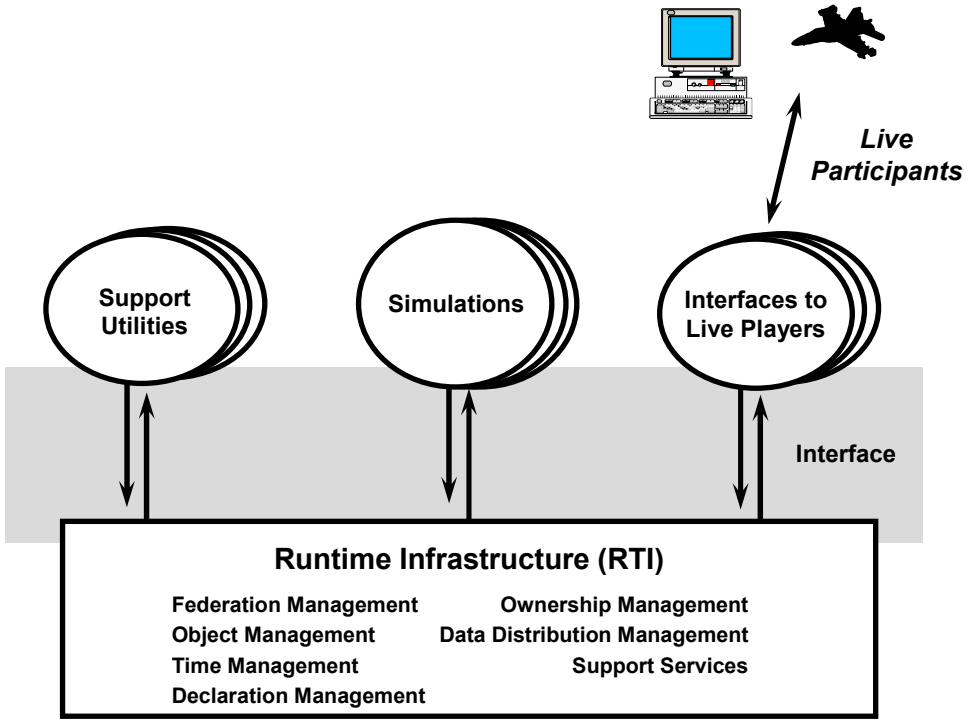
MITRE

Multiple Levels of Testing will be Required for NCES and the GIG

- Testing of each component
  - Does this node (database, consumer console, service provider) perform its function properly (as expected and according to specifications)?
- Testing services and transport components working together as different subnets
  - Do this/these services work in an integrated fashion on the network as expected?
- Testing each system's use of the network
  - Does this network architecture have bottlenecks and what is the maximum volume it can handle?
  - What is the network performance?
- Testing the end-to-end suite of systems over the network

Services Used in Distributed Simulation: The High Level Architecture

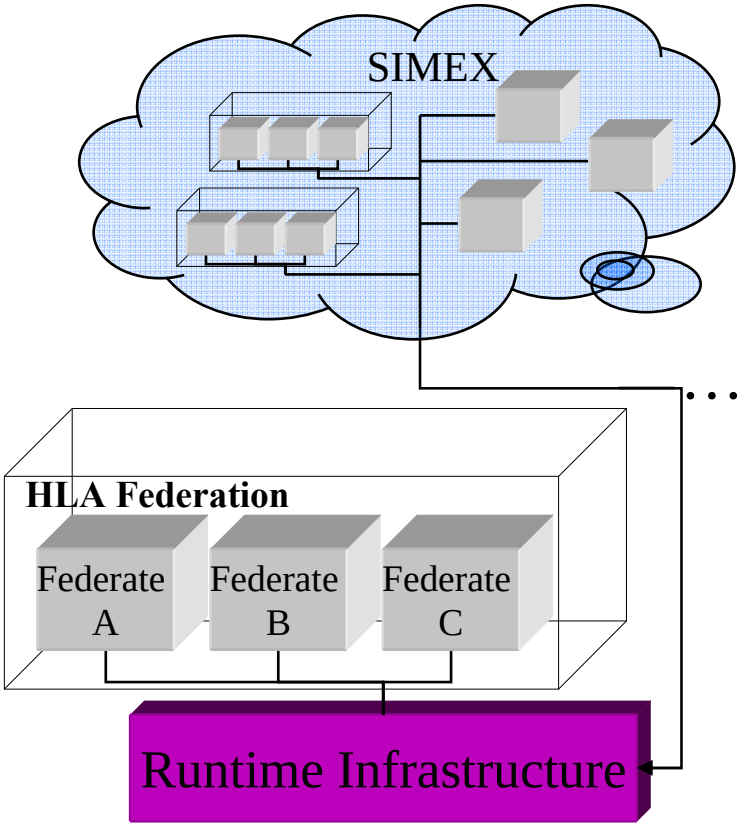
- Service Oriented Architecture for exchanging data among federated applications
  - Simulations, real-world systems and/oror system emulators, support utilities…
- Calls for an RTI which brokers data exchange via 7 service families
  - Includes publication & subscription services
  - Offers simulation time clock services



MITRE

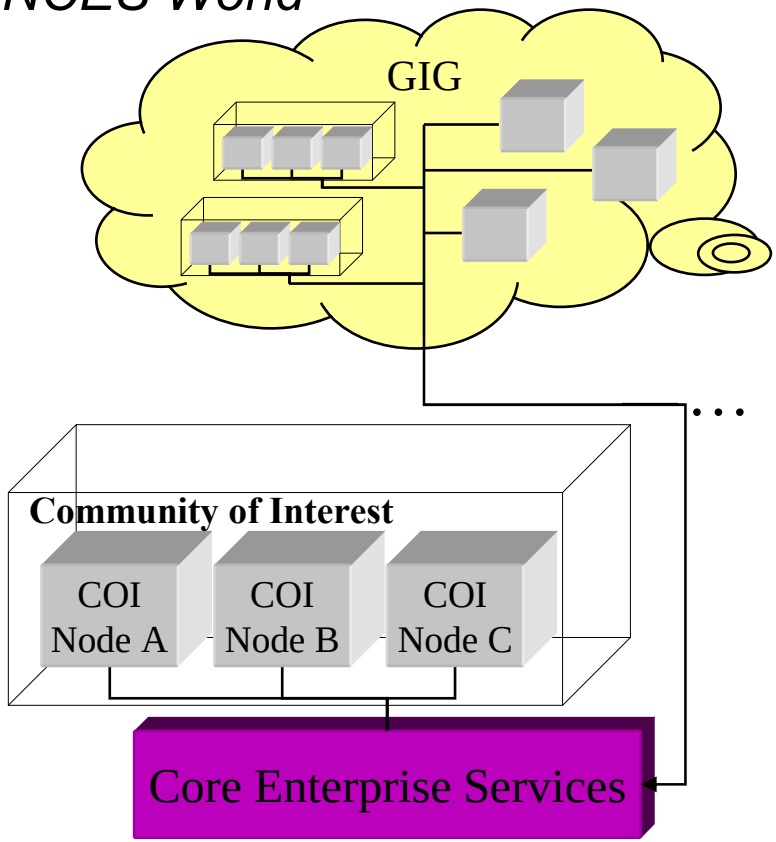
Parallels Between the Two Computing Worlds

Distributed Simulation World



Test Tools:
- SITH, RTI Verifier, …

NCES World



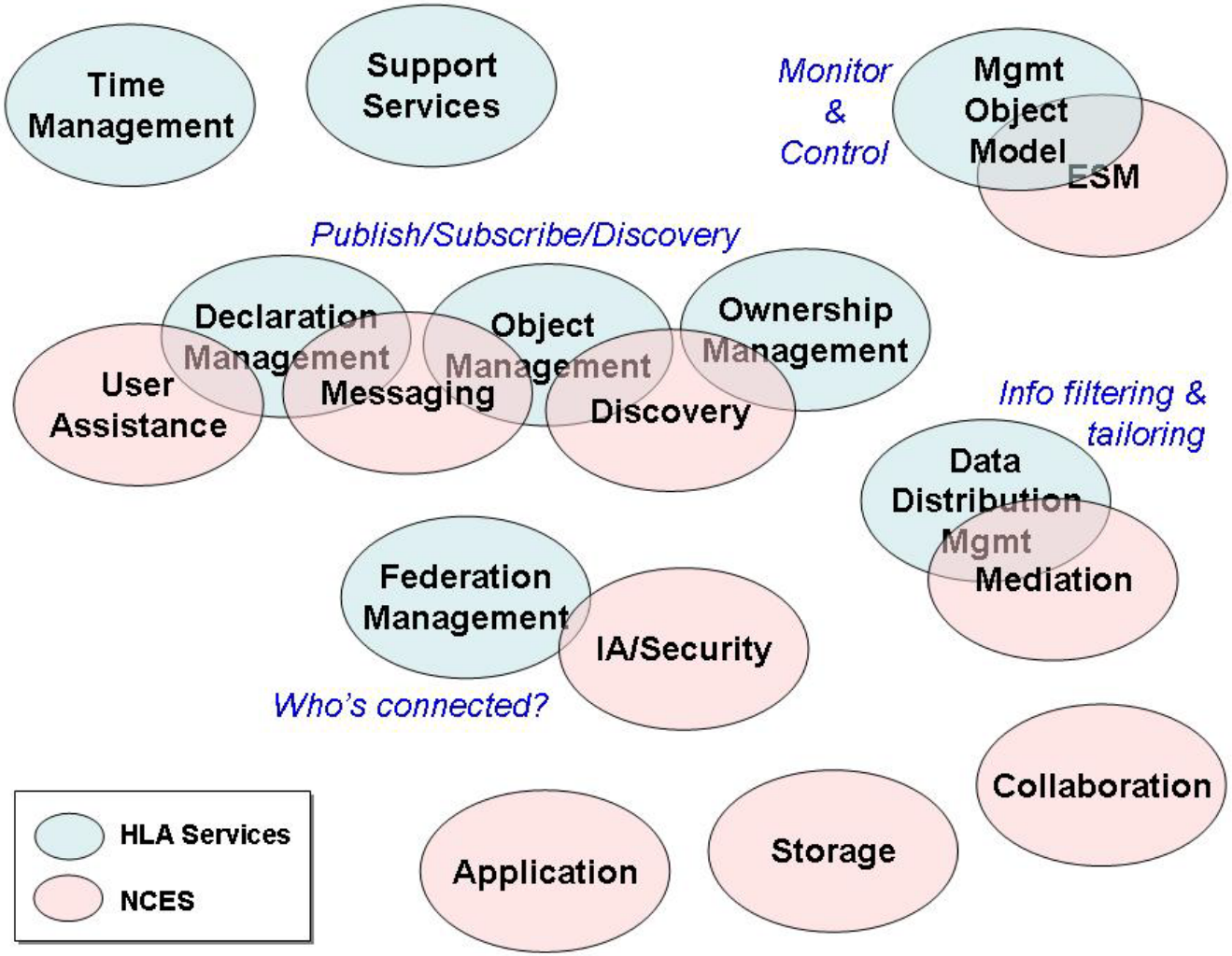
Test Tools:
- *SOA Test™, TestMaker™*

MITRE

Parallels Between the Two Computing Worlds

- Lessons learned from testing distributed simulation systems, including High Level Architecture (HLA) systems, have the potential to be leveraged to test net-centric systems
  - Both HLA and NCES are based on service oriented principles that have, at their core, a set of common infrastructure services that provide the basic connection mechanisms necessary for interoperability
  - Both worlds also embrace the concept that some subsets of the systems need to be tightly coupled together because their missions are strongly related or they share certain data exchange requirements
  - Both worlds also embrace a larger enterprise view whereby multiple nodes and multiple system subsets can interoperate as needed on a loosely coupled basis.

There are striking similarities between HLA Services and Core Enterprise Services



MITRE

Operational Differences between HLA and GIG Computing

- Routing of information exchanges
  - HLA: all data must pass through middleware (runtime infrastructure)
  - GIG: data routed "directly" between services
    - Most appropriate route between any two services likely to change over time
- Persistence of participants
  - HLA: static set of federates and data exchanges
    - Addition or deletion of services during execution is not the norm
  - GIG: can be open-ended
    - Dynamic addition or deletion of services is expected during normal operations

Testing Expertise in the Distributed Simulation Community

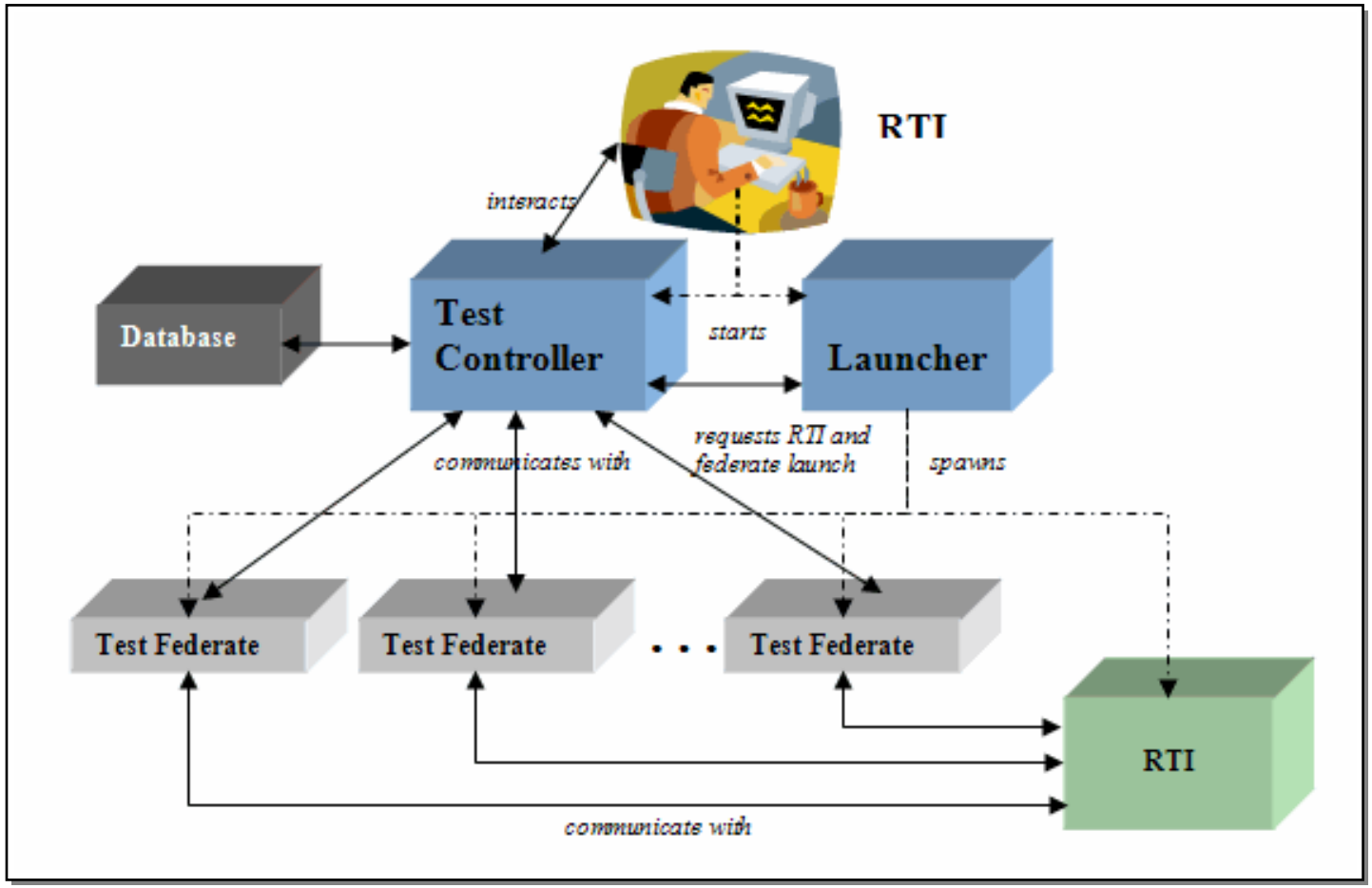
- The simulation community has years of successful experience testing complex distributed simulation systems, including High Level Architecture (HLA) systems:
  - Testing individual system functionality and performance.
  - Testing runtime infrastructure services for correct functionality and for performance.
  - Testing the simulations for their ability to use the runtime infrastructure services and to publish and subscribe to data as specified.
  - Testing subsets of the simulations working together over the RTI services.
  - Testing the end to end federation for functionality and performance.

Distributed Simulation Test Tools: RTI Verifier

- Created to certify compliance of RTIs with HLA Interface Spec
- RTI Verifier consists of:
  - Database of required tests
  - Launcher that starts federates and RTI
  - Test Controller that stimulates interplay between federates and RTI
- Key component: Script Definition Language
  - For specifying tests

MITRE

Distributed Simulation Test Tools: RTI Verifier (cont'd)



RTI Verifier Architecture (Ref. *Verifier3 User's Manual,* HLA RTI Verifier Team, MITRE Corporation, February 2005.)

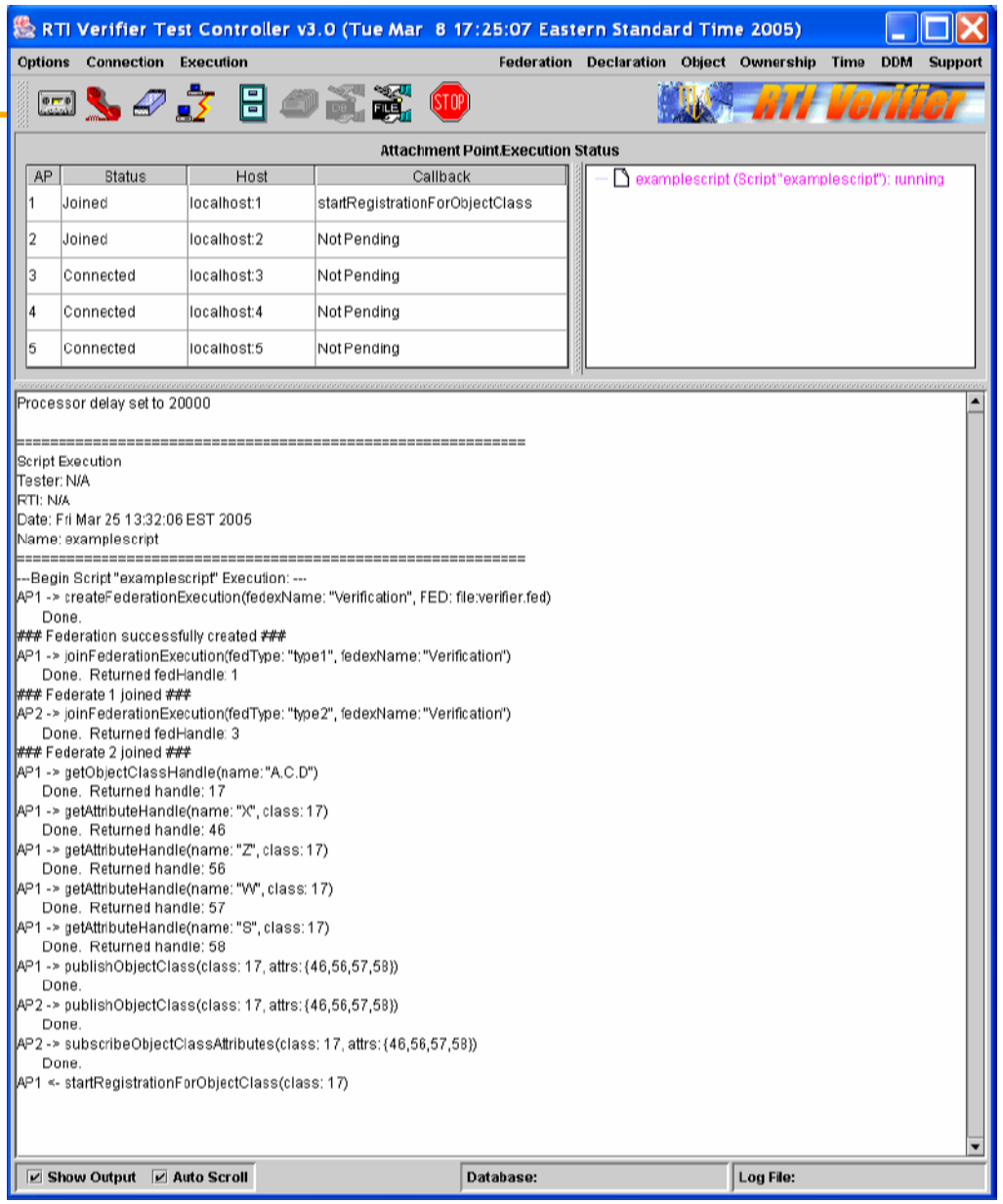
MITRE

Distributed Simulation Test Tools: RTI Verifier

- **RTI Verifier Test Controller GUI**

Status of federates

Test Controller activities



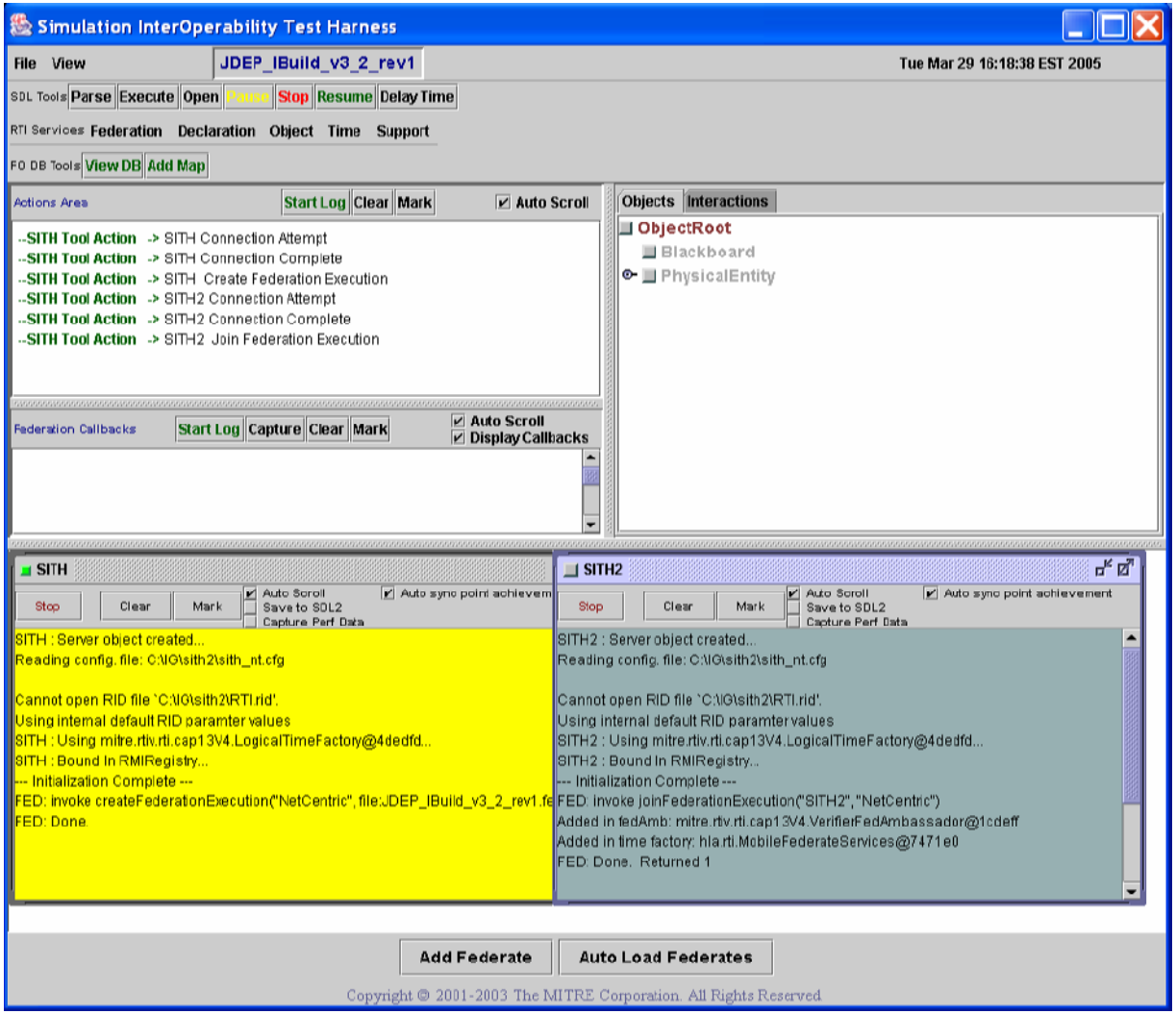
Distributed Simulation Test Tools: Simulation Interoperability Test Harness (SITH)

- Supports development and integration testing of HLA federations
  - general purpose tool that allows federate emulation for runtime data validation, functional testing, and performance testing
- Built around RTI Verifier core. Key add'l features:
  - Ability to create unlimited stand-in federates
  - Object Script Creator (OSC) for graphically creating/modifying SDL test scripts
- The SITH uses a sophisticated scripting capability to produce complex data exchanges, along with a data logging capability, to run tests that lead to quick diagnosis of problems
  - The SITH has been instrumental to the successful development and testing of several HLA federations

MITRE

Distributed Simulation Test Tools: SITH (cont'd)

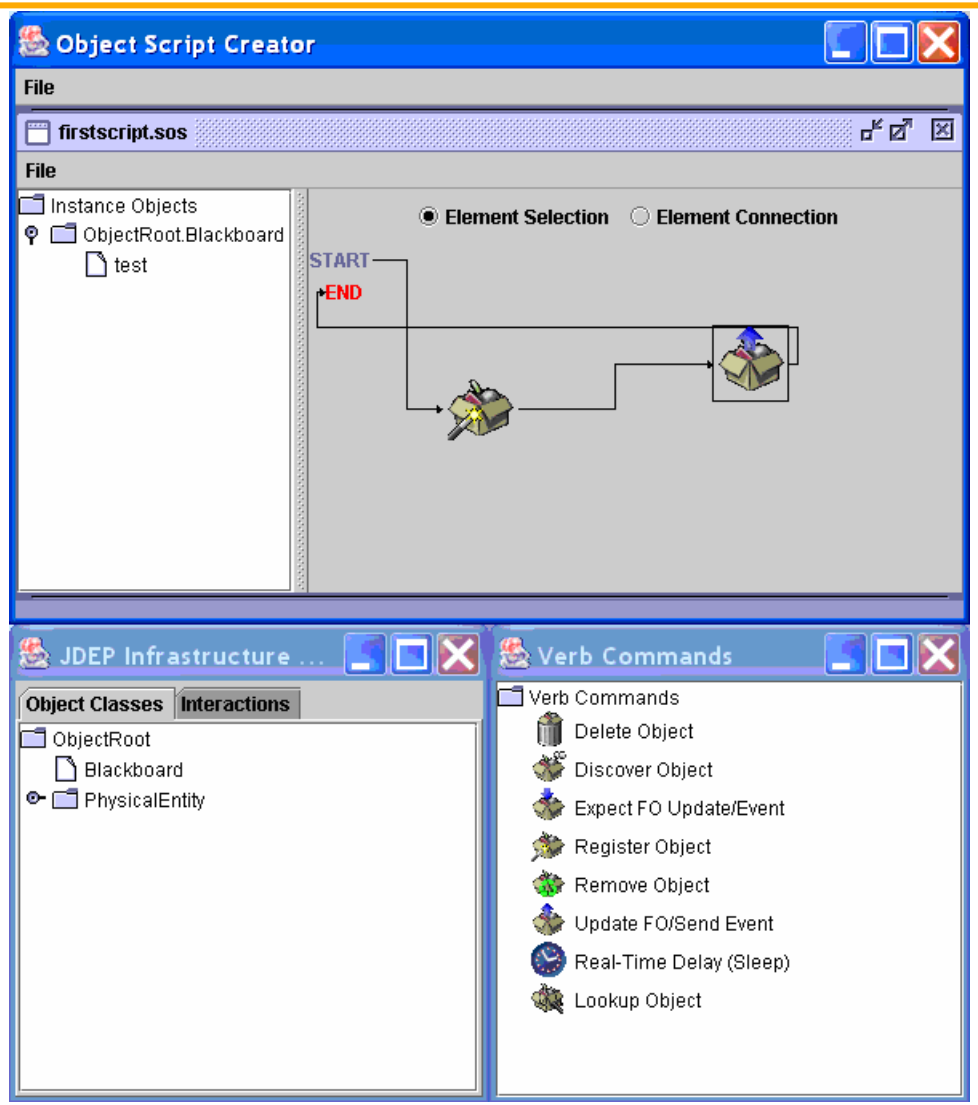
- **SITH GUI**



MITRE

Distributed Simulation Test Tools: SITH (cont'd)

- **Object Script Creator GUI**



MITRE

Applicability of Simulation Test Tools to Net-Centric Services Testing

- The following characteristics of the SITH and Verifier can be useful applied to testing net-centric SOAs:
  - GUI- this will be necessary to control the test environment which the tool will emulate
  - Test scripting capability (SDL)- this will be useful for setting up and repeating parts of the scenario relating to the network response and service behavior
  - Ability to see the entity states in the GUI
  - Record entity state changes for analysis
  - Run-time data validation capability
  - Service or system emulation
  - Network flooding capabilities
- However, fundamental differences in underlying core services will require reworking the SDL to control net-centric services

MITRE

SOA Test: An Emerging Automated Tool for Testing SOAs

- WSDL Verification
  - XML Validation
  - Tests interoperability against WS-I Standards
- Unit Testing
  - Verifies web service responses against valid and invalid data sets
    - Data sets can be composed of a range of values in legacy data stores
      - E.g. Microsoft Excel or Database queries
- Functional Testing
  - Scenario based testing using a chain of services
  - XML Databank used to map the output of a given web service to the input of another

MITRE

SOA Test: An Emerging Automated Tool for Testing SOAs

- Scripting
  - JavaScript, Jython (Java-enabled Python)
- Security
  - Message layer security
    - Username or SAML Tokens
  - Penetration testing
    - SQL Injections
      - Passing SQL Query Strings as parameters to the Web Service
    - Parameter fuzzing
      - Unbounded parameters leading to buffer overflow or explicit error messages
  - XML Encryption and Signature

MITRE

SOA Test: An Emerging Automated Tool for Testing SOAs

- Regression testing
  - Automated testing in continuous integration environments
  - Evaluate trends over time
- Load testing
  - How do multiple users affect timeliness, content

MITRE

Challenges

- Number and volatility of associated standards
  - E.g. Web services with attachments must account for:
    - Soap with Attachments, MIME, DIME, MTOM Recommendations
- Automated failover
- Federated registries
- Evaluation of service pedigree
- Semantic interoperability

Findings and Recommendations

- Significant overlap between HLA and GIG operations warrants closer look at simulation test tools and approaches
  - Examined SITH due to documented successes and extensible software design
- Recommend an exploratory prototype that reuses much of SITH for net-centric testing purposes
  - Replace SITH SDL with a new scripting language that leverages web services standards (WSDL, BPEL4WS)
- Apply the new SITH-like application first to interoperability testing of small groups of services
  - Then expand use for performance and behavior testing of larger groups of services in more complex internet-like environments

Contact Us

Doug Flournoy
rflourno@mitre.org
(781) 271-2774

Elizabeth Lee
elee@mitre.org
(703) 983-2692

Rob Mikula
rmikula@mitre.org
(703) 983-7168

MITRE