NASA Operational Simulator (NOS)
A Reusable Software-Only Verification & Validation (V&V) Architecture

http://www.nasa.gov/centers/ivv/JSTAR/ITC.html

Justin R Morris
Justin.R.Morris@nasa.gov

NASA IV&V Facility
100 University Drive
Fairmont, WV 26554
Agenda

• Independent Test Capability (ITC)
  • Jon McBride Software Testing & Research Lab (JSTAR)

• NASA Operational Simulator (NOS)
  • Architecture
  • Middleware

• NOS Utilization
  • Global Precipitation Measurement (GPM) Operational Simulator (GO-SIM)
  • James Webb Space Telescope (JWST) Integrated Simulation and Test (JIST)

• Closing Remarks
Charter

Acquire, develop, and manage adaptable test environments that enables the dynamic analysis of software behaviors for multiple NASA missions
• ITC Develops System Simulators
  – Experts in Hardware Modeling and Distributed Simulation
  – Experts in Simulator & Software Integration

• NOS Architecture
  – Reusable Hardware Models
  – Custom Middleware

• System Test Automation

• Typical NOS Users
  – V&V and IV&V Engineers
  – Project Developers
  – Operators and Testers
• Cloud-based infrastructure using server and desktop virtualization

• Large scale simulator deployments

• Hardware-in-the-loop and software-only test environments

• Integration of COTS and GOTS software tools to support V&V activities
Mission Support

Global Precipitation Measurement (GPM) Operational Simulator (GO-SIM)

Closed-loop simulator including unmodified operational ground system, unmodified flight software, environmental simulator, and science instrument simulators

James Webb Space Telescope (JWST) Integrated Simulation and Test (JIST)

Simulator that demonstrates reusable NOS technologies can be applied to other NASA missions
NASA Operational Simulator (NOS)
NASA Operational Simulator (NOS)

- Software-only simulation architecture (common components from in-house software simulation development)
- Capable of executing unmodified flight software executable(s)
- Custom layered-architecture middleware
- Dynamic interception capability
- Reusable software modules and scripts
- Virtual machine deployment

Typical NOS Architecture (Space Domain)
NOS Feature Set

Plug-and-Play Hardware Models
- Processors, Boards, Racks

Use of Operational Ground Systems Software
- Instrument Model Framework
  - Instrument1
    - Subaddress HandlerA → FunctionA
    - Subaddress HandlerB → FunctionB
    - ...
    - Subaddress HandlerN → FunctionN
  - InstrumentX
    - Subaddress HandlerA → FunctionA
    - Subaddress HandlerB → FunctionB
    - ...
    - Subaddress HandlerN → FunctionN

Internal Bus Monitoring

NOS Middleware
- Specialized Layers
- Base Layer Communications

Deployment & Maintenance
- Virtualization
### Overview

- Offers re-usable communication mechanism
  - Ensures consistent and correct data passing

- Provides synchronization between distributed applications

- Flexible and extensible design
  - Can be extended to incorporate any communication protocol

### Features

- Transport agnostic
- Cross platform C++ implementation
- Robust User API
- Specialized User API Layers
  - MIL-STD-1553B
  - ESA SpaceWire
  - Discrete Signals
  - Time Synchronization
- Interception allows for V&V analysis
  - No modification to software-under-test
NOS Middleware Architecture

- System Under Test
  - MIL-STD-1553
  - SpaceWire
  - Discrete
  - Time Synchronization
  - Additional Protocols as Needed

- NOS Core Middleware with Interception Capability

- I/O Interface Layer

- System Monitoring
  - Bus Analyzer
NOS Dynamic Interception

Normal Data Flow

Node A → NOS → Node B

Interceptor

Block

Interception Points:
- Interceptor modifies data
- Interceptor blocks data

NOS Dynamic Interception is a capability provided by NASA IV&V Independent Test Capability.
NOS User Interfaces
MIL-STD-1553
NOS User Interfaces

SpaceWire
NOS Software Utilities

• Virtual Oscilloscope
  – Virtual CompactPCI (cPCI) Analysis
  – Board-Level Signal Analysis

• Virtual MIL-STD-1553 Bus
  – Bus Controller with XML Defined Schedules
  – Remote Terminal
  – Bus Monitor/Logger
  – PASS3200 Software Emulator

• Virtual SpaceWire Router
NOS Utilization
NOS Utilization

**GO-SIM**

1. **GPM GSFC Flight Software Testers**
   - Dry run test procedures; reduce required use of lab resources

2. **GPM Software Safety**
   - Tool kit to support safety studies

3. **GPM IV&V Engineers**
   - Provides flexible testing platform for IV&V personnel
   - Independent Testing & Risk Reduction

**JIST**

4. **JIST Development Team**
   - Spacecraft simulation environment setup in ½ time due to GO-SIM architecture

5. **JWST IV&V Engineers**
   - Risk reduction simulator under development
   - Supported processor offline mode test efforts
NOS Utilization

GPM Operational Simulator (GO-SIM)
## GPM Operational Simulator
### GO-SIM

### Components
- COTS Emulator
- Primary Instrument Simulations (GMI/DPR)
- GPM Ground System
- GSFC Goddard Dynamic Simulator (GDS)
- NOS Middleware
- GPM Hardware Models

### Capabilities
- Load and run unmodified flight software binaries
- Execute test flight scripts
- Single-step debugging
- Inject errors via ground system and NOS middleware
- Stress system under test

**NASA Software of the Year Honorable Mention 2012**
NOS Utilization

James Webb Space Telescope (JWST)
Integrated Simulation and Test (JIST)
JWST Integrated Simulation and Test (JIST)

- Software-only spacecraft simulator
- Flexible environment to support V&V activities
- Unmodified ground system and scripts
- Unmodified software-under-test binaries
- Integration of COTS, GOTS and in-house developed components
- Custom hardware models
- Automated Testing Framework
## Virtualized Deployment

**GO-SIM_3.5** Running

<table>
<thead>
<tr>
<th>vApp Diagram</th>
<th>Virtual Machines</th>
<th>Networking</th>
</tr>
</thead>
</table>

### GO-SIM_NET

- ASIST
- GDS
- GPM-Build-Env
- GPM-FSW
Closing Remarks

• NOS provides a generic software-only simulation architecture that has been utilized on NASA missions
• NOS architecture is transparent to user
• New instantiations of NOS require customization for missions/projects → NOS has demonstrated significant cost and time savings
• NOS provides reusable hardware models
• NOS provides custom-developed middleware with user APIs and interception
• NOS extends to other domains
  – Large complex systems
  – Distributed components
Contact Information

• Web Page
  – http://www.nasa.gov/centers/ivv/jstar/JSTAR.html

• E-Mail
  – Justin.R.Morris@nasa.gov
  – Team Mailing List: ivv-itc@lists.nasa.gov