MATLAB Aerospace & Defense Technical Briefing

Modeling in the Stateflow® Environment to Support Launch Vehicle Verification Testing for Mission and Fault Management Algorithms in the NASA Space Launch System*

Luis Trevino, Ph.D.
Jacobs ESSSA Group – Marshall Space Flight Center
Mission & Fault Management (M&FM), EV43
Spacecraft and Vehicle Systems Department
04/27/2017
Huntsville, AL

* AIAA Paper previously presented at SPACE 2016, Long Beach CA
MATLAB Aerospace & Defense Technical Briefing

AIAA Paper Co-Authors

- **Peter Berg**
  SLS State Flow Lead, M&FM Team
  Stinger Ghaffarian Technologies, Inc.
  Intelligent Systems Division
  NASA Ames Research Center

- **Dwight England**
  Chief, Integrated Systems Health Management & Automation Branch, EV43
  NASA Marshall Space Flight Center

- **Stephen B. Johnson, Ph.D.**
  Analysis Lead, M&FM Team
  Dependable System Technologies, LLC
  Jacobs ESSSA Group
  University of Colorado, Colorado Springs
Mission & Fault Management - SLS

- Fault Management Software
  - Error Prone
  - Requirements and Design Phase
  - Other Factors

- Model Based Systems Engineering
  - Rich graphical constructs
  - Deterministic
  - Standards

- Previous NASA Stateflow® Applications
  - LADEE
  - Ares – Orion Command Abort
  - NESC – Toyota, Commercial Crew Program
MATLAB Aerospace & Defense Technical Briefing

State Analysis Model (SAM)

State Flow Environment

Input Processing Constructs → SAM (Plant, Controller) → Output Processing Constructs

SLS Subsystems
- MPS
- CCSE
- CLSS
- TVC
- Flight Computers
- SLS Avionics
- Boosters

M&FM Algorithms
- MPS
- Booster
- System Management
- Fault Management
- CSE
- EPS
- SDQC
- TVC
MATLAB Aerospace & Defense
Technical Briefing

UML Modeling and Stateflow for M&FM

function determine_if_failed_high_reg_is_only_remaining_source

{functrace["determine_if_failed_high_reg_is_only_remaining_source"] = true;}

function deenergize_the_valve.supplying_over_pressure

function set_above_limit_event
SAM Testing

- **Script Driven → Ground Operations Timeline → Nominal Sequence Generator → Fault Generator**
- **Rule Checker → Analysis Report Generator → Timeline & State Report scripts → SAM Test Report**

- **User GUI**
- **Test Cases: Nominal, Off-Nominal, VMET, MCaRT, SIL**
- **TRAC Trouble Ticket System Summaries**
User GUI
VMET, MCaRT, SIL Test Cases for the SAM

MATLAB Aerospace & Defense Technical Briefing

### Test Case ID: MPS_Helium

**Test Objective:**
- Test failure of helium isolation valve.
- "EVT_HeliumValve.Redundancy_Reduced" becomes "True" at Mission.Elapsed_Time = [T] sec

**Success Criteria:**
- Test duration is from Mission.Elapsed_Time = [T1] sec to [T2] sec

---

### Table: Test Case Details

<table>
<thead>
<tr>
<th>Element</th>
<th>System</th>
<th>Response</th>
<th>Monitored Condition Name</th>
<th>Monitored Condition Description</th>
<th>Start Monitoring</th>
<th>Stop Monitoring</th>
<th>Units</th>
<th>Lower Trigger Limit (TBD)</th>
<th>Upper Trigger Limit (TBD)</th>
<th>Number of Indicators Needed to Generate Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Booster</td>
<td>Igniter</td>
<td>Rail</td>
<td>DualBoostersIgnitionFailure</td>
<td>Both Boosters fail to ignite after [T] is reached</td>
<td>T- [T] msec</td>
<td>T+ [T] msec</td>
<td>psia</td>
<td>[ ]</td>
<td>[ ]</td>
<td>2012</td>
</tr>
</tbody>
</table>
Findings: VMET & SAM

MCaRT & SIL
19% of MCaRT entries tested
85.5% passed
45% of SIL test cases executed
27% passed

Finding Types
Logic Interpretation 30%
Editorials 55%
Logic Update 15%
SAM Forward Directions / Summaries

• Interactive Failures
• Prelaunch procedures → OMRs → LCCs → Rule Checker
• Hazardous State Identification
• Post Flight Analysis
• Other: EUS, crew habitat, payloads, proximity ops, rovers, robotic deep space missions, EDL ops
• MBE → M&FM Algorithms → FSW → Testing
• Challenges
• Questions