End-to-End Fault Management Analysis Method, Results and Future Improvements

Presented by Ryan Starn
September 11, 2014

Completion and Conflicts,
Identifying the Problematic “Unknown Unknowns”
Agenda

- Challenges
  - Overview
  - FM Assurance Goals
  - IV&V Assurance Questions
  - 2D State-Space Representation

- Proposed Approach
  - Overview
  - Traceability Process
  - System Model Generation Process
  - Conflict Analysis Process
  - Example

- Status and Results

- Future Efforts and Conclusions

- Acronyms/Abbreviations
Challenges: Overview

- Provide SW Assurance for a large, distributed and fault-tolerant system
  - Local and Global responses
  - Need to balance asset safety with mission objectives
- Subsystems have varying lifecycles and development organizations
- Providing and maintaining assurance up to launch (and beyond)
- Automated or scripted responses/sequences are not governed by “requirements” and are implemented late in system lifecycle.
- 100% Verification/Assurance/Success cannot be guaranteed
  - How much assurance is enough?
  - How do we provide efficient and effective SW assurance?
Challenges: FM Assurance Goals

**FM Validation Goals**
- List of Failures are complete
- List of Effects (Impacts) are complete
- List of Controls/Mitigation complete for each Fault
- Able to meet mission/user needs
- List of (FM) requirements complete

**FM Verification Goals**
- Every FM requirement met
- FM system responds in time to prevent Failure Propagation (i.e. domino effect)
- FM system isolates failures (i.e. allow other objectives to continue, Failure Containment)
- FM system effectively diagnoses the problem and limits effects to healthy systems
- Known dependencies for each Control to other systems, other controls, etc
- FM system is free from conflicts within controls and other systems
Challenges: IV&V Assurance Questions

• Is the (FM) system implemented correctly (Q1/Q2)?
  – Assuming that every component of the FM system is tied to a requirement, normal IV&V activities will answer this question as it relates to Q1 for each component.
  – Q2 is harder to address because we must check the entire system as a whole to determine if there are conflicts between the components.

• Does the FM system behave correctly under adverse conditions?
  – Difficult to answer without determining all credible adverse conditions (off-nominal/failure scenarios). Two options exist:
    – 1) Brute Force/Bottoms-Up/Testing - Throw every possible scenario at the system and “catch” problems.
    – 2) Top-Down - Identify the conflicts you want to prevent (assume this is complete) see if any credible scenario exists to reach this point.
    – Proposal - 3) Intermediate – Using limited bottoms-up and top-down approach, determine potential conflicts and work backwards to find a credible scenario or eliminate the conflict as not possible.

Goal: Identify more problem scenarios than top-down approach without the cost of full bottoms-up approach
Challenge: 2D State-Space Representation

- Nominal (Req) Path
- FTA/FM Path
- Random State Cov.
- Unknown “Conflict”
- “Conflict” Potential

Beginning of Mission

End of Mission
Proposed Approach: Overview

- Top-Down: develop conflict rules based on user/mission needs and known failure modes
- Bottoms-Up: develop low to medium fidelity model of the system (Event Network) (emphasis on software perspective) (Reduced state-space)
- Apply automated/scripted principles to establish scenario coverage of the system model
- Automated static analysis will eliminate scenarios where conflict is not possible and highlight those with potential/suspected conflict
- Manual analysis will determine scenario credibility of suspected conflicts
- High-fidelity simulations, tests, and analysis used to confirm conflicts
- Approach Relies on:
  - Relational Database, FM Traceability, System State Models, Event Network, Conflict Rules, and High-Fidelity Simulator / test bed
End-to-End FM analysis requires End-to-End traceability

Proposed Approach: System Model Generation Process

Example Event Chain

FM HW and SC dependency will aid in Change Impact Analysis
Proposed Approach: Conflict Analysis Process

Full End-to-End conflict and Change Impact Analysis is possible
Proposed Approach: Example

### Components/States

1. **System**
   - **Light**
     1. On
     2. Off
     3. *Failed Off*
   - **Switch**
     1. Open
     2. Closed
     3. Failed Open
     4. Failed Closed
   - **Power Supply**
     1. Nominal
     2. Failed

2. **Event Network**

   - Power Supply Failed
   - Light Off
   - Switch Closed
   - Light Off Fault
   - Dead Bulb Fault
   - Change Bulb
   - Toggle Switch
   - Switch Open
   - Switch Closed

3. **Conflict Rules:**
   1) >1 Bulb changed / day
Status and Results

- FM Database (FTA, FMEA) Integrity (i.e. Traceability Analysis) has uncovered severe issues in one portion of the system:
  - Missing Sensor Collection/Processing Requirements – to support FM
  - Missing Fault Requirement / Implementation
  - ~300 lower severity issues impacted developer FM assurance

- SW Lifecycle Traces under active development
  - Mixture of present and past traces through flow/activity diagrams and artifact mapping tables

- First Iteration of Event Modeling of one subsystem
  - Late Lifecycle: Most software behaviors are verified
  - Custom IV&V event network modeling/visualization tools and database design
  - Custom IV&V network parsing and conflict violation highlighting
Future Efforts and Conclusions

- Expand FM Database and Lifecycle Traceability to all subsystems
  - Integrate and automate Change Impact Analysis

- Refine Event Network and Conflict Rules
  - Iterate Event Network and expand to multiple subsystems
  - Build comprehensive set of Conflict Rules and execute tools to identify potential conflict scenarios

- Develop test cases and perform Independent Testing using IV&V test bed
  - Confirm absence of conflicts as well as confirm their presence

- Stretch Efforts:
  - Integrate Portfolio Based Risk Assessments (PBRA) and Risk Based Assessments (RBA) to more clearly define and document the criticality
  - Integrate Assurance Case Goal Structured Network (GSN)

- Conclusions:
  - Early results provided ~300 findings to improve developer (and IV&V) FM Assurance
  - Significant findings with <10% additional effort
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARD</td>
<td>Algorithm Requirement Document</td>
</tr>
<tr>
<td>CMD</td>
<td>Command</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>FM</td>
<td>Fault Management</td>
</tr>
<tr>
<td>FMEA</td>
<td>Failure Modes and Effects Analysis</td>
</tr>
<tr>
<td>FTA</td>
<td>Fault Tree Analysis</td>
</tr>
<tr>
<td>GSN</td>
<td>Goal Structured Network</td>
</tr>
<tr>
<td>IV&amp;V</td>
<td>Independent Verification and Validation</td>
</tr>
<tr>
<td>PBRA</td>
<td>Portfolio Based Risk Assessment</td>
</tr>
<tr>
<td>Q1</td>
<td>Question 1: Does the system do what its supposed to do</td>
</tr>
<tr>
<td>Q2</td>
<td>Question 2: Does the system not do what its not supposed to do</td>
</tr>
<tr>
<td>Q3</td>
<td>Question 3: Does the system behave correctly under adverse conditions</td>
</tr>
<tr>
<td>RBA</td>
<td>Risk Based Assessment</td>
</tr>
<tr>
<td>REQ</td>
<td>Requirement</td>
</tr>
<tr>
<td>SC</td>
<td>Stored Command</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>TLM</td>
<td>Telemetry</td>
</tr>
</tbody>
</table>