Safe from the Start: Using MBSE for Safety Engineering

Ms. Daisy Bower & Ms. Kate Kovalovsky
System Architects
Systems Engineering Directorate
**SCOPE & CONTEXT**

**Safety Modeling makes for Happier Safety Engineers**

| These Model-Based Safety Analysis processes | • Analysis of Concept of Operations  
• Mishaps  
• Functional hazards  
• Software level-of-rigor  
• Safety messaging |
|--------------------------------------------|----------------------------------------------------------------------------------|
| generate products for Safety Boards & Certifications | • Tech Assists  
• AFSRB  
• JS-SSA  
• JSWSRB |
| by using a generic profile tailored for safety standards and applicable to projects using these standards. | • MagicDraw  
• SysML  
• MIL-STD-882E  
• AOP52 |
### STARTING POINT AND TEAMING

#### MIL-STD-882-E
- Establish Safety Criteria and Requirements
- Apply the Safety Order of Precedence
- Hazard Mitigation/Testing/Verification
- Guides Hazard Monitoring and Control

#### SSWG
- Safety Plans
- Safety Guidance and Design
- Safety Board Presentations & Certifications
- Identify & Mitigate Hazards
- Identify Safety Related & Safety Critical Functions
- Document Safety Analyses thoroughly

#### MBSE
- Support Safety Analysis using SSWG Deliverables and the MBSE Models
- Incorporate SSWG recommendations and supporting data into functional analysis
- Create and publish model based deliverables
  - Manage Mishaps
  - Manage Functional Hazards
  - Manage Safety Causes and Effects
  - Manage Function Safety Ratings
  - Manage Safety Requirements
  - Traceability to Standards and Guidelines
DEVELOPING THE SAFETY ARCHITECTURE

Inputs: System Need, Capability Gaps, ConOps, and/or Requirements

Functional Analysis

Safety Analysis

Outputs: Consistent views depicting the system and safety architecture

Activity Diagrams
Function & Signal Lists
Internal Block Diagrams
Mishaps & Hazards
System Function Ratings
Software Function Ratings
Signal Impact Analysis

The Bottom Line: Functional & Safety Analyses are aligned with one another using Model-Based Systems Engineering techniques
SAFETY MODEL ELEMENTS

• **Use Case Elements are informed by the ConOps:**
  – Functional Hazards
  – Safety Causes
  – Safety Effects
  – Top Level Mishaps

• **System-Level Functions must have an 882E rating and the following attributes:**
  • Safety Rating Rationale: Detailed documentation of why the rating was assigned and the date of assignment
  • Needs Safety Review: TRUE or FALSE for tracking purposes
  • Safety Status: Additional Tag for any Notes on open actions or open questions the SSWG has

• **Software-Level Functions:**
  – Use stereotype tags to capture all SSWG information
  – These are used to support the development and management of FSHA and FMEA

• **Signals:**
  – AOP-52 dictates safety criticality of information and how it is consumed and supplied
  – Table created to automatically search signal usages and their interactions with function ratings
Model & Profile Demonstration
FUNCTIONAL HAZARDS AND MISHAPS

Hazards can be discovered during system use case development. For each way users interact with the system, ask: "What could go wrong?", and, "What are the consequences?"

Track the mishaps as special types of use cases to prevent and mitigate them.

System of Interest

User X

Handle Scenario A

extension points
Functional Hazard 1

Handle Scenario B

extension points
Functional Hazard 2

«Mishap»
Something Dangerous

«extend»

«Mishap»
Something Very Dangerous

«extend»

«Mishap»
Another Dangerous Result

User Y
SYSTEM LEVEL FUNCTIONS

<table>
<thead>
<tr>
<th>FunctionID</th>
<th>Name</th>
<th>Safety Function Rating</th>
<th>Safety Rating Rationale</th>
<th>Safety Status</th>
<th>Needs Safety Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Process XYZ Algorithm</td>
<td>Safety-Related</td>
<td></td>
<td>17Oct2019: New input added t</td>
<td>true</td>
</tr>
<tr>
<td>101</td>
<td>Control System Safety</td>
<td>Safety-Critical</td>
<td></td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>102</td>
<td>Stow System</td>
<td>Not Safety Significant</td>
<td>17Oct2019: Stowing is a feature</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

System-level functions can be organized and rated per these categories: Safety Critical, Safety Related, Safety Significant, Not Safety Significant.
HARDWARE FUNCTION ANALYSIS

Table III: Risk assessment matrix

<table>
<thead>
<tr>
<th>SEVERITY</th>
<th>Catastrophic (1)</th>
<th>Critical (2)</th>
<th>Marginal (3)</th>
<th>Negligible (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequent (A)</td>
<td>High</td>
<td>High</td>
<td>Serious</td>
<td>Medium</td>
</tr>
<tr>
<td>Probable (B)</td>
<td>High</td>
<td>High</td>
<td>Serious</td>
<td>Medium</td>
</tr>
<tr>
<td>Occasional (C)</td>
<td>High</td>
<td>Serious</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Remote (D)</td>
<td>Serious</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Improbable (E)</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Eliminated (F)</td>
<td>Eliminated</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Model is aligned with the Risk Assessment Matrix for hardware items.
SOFTWARE FUNCTION ANALYSIS

TABLE V. Software safety criticality matrix

<table>
<thead>
<tr>
<th>SOFTWARE SAFETY CRITICALITY MATRIX</th>
<th>SEVERITY CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFTWARE CONTROL CATEGORY</td>
<td>Catastrophic (1)</td>
</tr>
<tr>
<td></td>
<td>Critical (2)</td>
</tr>
<tr>
<td></td>
<td>Marginal (3)</td>
</tr>
<tr>
<td></td>
<td>Negligible (4)</td>
</tr>
<tr>
<td>1</td>
<td>SwCl1</td>
</tr>
<tr>
<td>2</td>
<td>SwCl1</td>
</tr>
<tr>
<td>3</td>
<td>SwCl2</td>
</tr>
<tr>
<td>4</td>
<td>SwCl3</td>
</tr>
<tr>
<td>5</td>
<td>SwCl4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SwCl 1</th>
<th>Level of Rigor Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwCl</td>
<td>Program shall perform analysis of requirements, architecture, design, and code, and conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SwCl</td>
<td>Program shall perform analysis of requirements, architecture, and design, and conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SwCl</td>
<td>Program shall perform analysis of requirements and architecture, and conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SwCl</td>
<td>Program shall conduct safety-specific testing.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SwCl</td>
<td>Once assessed by safety engineering as Not Safety, then no safety-specific analysis or verification is required.</td>
</tr>
</tbody>
</table>

NOTE: Consult the Joint Software Systems Safety Engineering Handbook and AOP 52 for additional guidance on how to conduct required software analyses.

<table>
<thead>
<tr>
<th>Name</th>
<th>Software Control Category</th>
<th>Severity Category</th>
<th>SwCl</th>
<th>Safety Function Rating</th>
<th>Safety Status</th>
<th>Safety Rating Rationale</th>
<th>Needs Safety Review</th>
<th>Standard Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate Something</td>
<td>3</td>
<td>Critical</td>
<td>SwCl3</td>
<td>Safety-Critical</td>
<td></td>
<td>17Oct2019: Rating was</td>
<td>false</td>
<td>AOP 52 Edition 1</td>
</tr>
<tr>
<td>Set Some Result</td>
<td>5</td>
<td>Marginal</td>
<td>SwCl5</td>
<td>Safety-Related</td>
<td></td>
<td>17Oct2019: Need stan</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Provide Emergency Stop</td>
<td></td>
<td></td>
<td></td>
<td>Safety-Critical</td>
<td></td>
<td></td>
<td>true</td>
<td>MIL-STD-882E</td>
</tr>
</tbody>
</table>

Model is aligned with the Software Safety Criticality Matrix and supporting text for software items.
SIGNAL IMPACT ANALYSIS

Metachains in MagicDraw find usages and relationships in the model.
OUTCOMES OF ANALYSES

Re-usable profile can be leveraged to enable Safety Analyses

- Managing Functional Hazards and Mishaps
- Discovery of Safety Critical or Non-Safety Critical features
- Easy ID and quantification of safety criticality levels for system, hardware, and software level functions
- Clear delineation of Level of Rigor needed for SW development process: SWCI Level distinction
- Automated views that indicate potential problematic signal flows
- Clear communication of rationale
- Safety architecture → System architecture incorporation

Environment & Language: MagicDraw 18.5, SysML V1.4
Questions?