Automated Generation of Failure Modes and Effects Analyses from AADL Models

Alexander Lam, Christopher Vogl, Myron Hecht, and Chris Dimpfl
The Aerospace Corporation

Presented to
2011 Flight Software Workshop
Applied Physics Laboratory
October 2011
Outline

• Motivation
• Background on FMEAs
• Introduction to AADL
• Automated FMEA Generation Process
• Algorithm Description
• Results
• MDDA Workbench Description
• Discussion
• Conclusions
Failure Modes and Effects Analysis (FMEA) Background

• Purpose
  – To determine the effect of hardware and software failures upon the system and equipment failures.
    • Classify effects by impact on mission success and personnel/equipment safety.
    • Identify single points of failure
• A standard of practice in a wide variety of industries: Examples
  • DoD: MIL-STD-1629A (introduced as MIL-P-1629A in the 1960s)
  • Industrial: IEC 60812 (1985)
  • Aviation: SAE ARP 5580 (2001)
  • Automotive: SAE J1739 (2002)
  • Space (ESA): ECSS-Q-30-02A (2001)
Motivation

• Failure Modes and Effects Analyses (and related Criticality Analyses) are rigorous and comprehensive reliability and safety design evaluations
  – Generally required either by industry standards or Government policies
  – A fundamental element of defense in many product liability lawsuits

• When performed manually, FMEAs are usually done only once during the detailed design phase because of cost and schedule constraints
  – Labor intensive
  – Require senior level; analysts

• If automated, FMEAs would have significant benefits
  – Multiple iterations from conceptual to detailed design
  – Enables early identification of potential problems
    • Single points of failure
    • Unanticipated effects
  – Facilitates tradeoff studies and evaluations of alternatives
Manual vs. Automated Output

Format defined in MIL STD 1629A

From FMEA Generator and Postprocessor
## FMEA Development Method Comparison

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Generated Manually</th>
<th>Generated Using Automated Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Identification Label or number</td>
<td>Same</td>
</tr>
<tr>
<td>Item</td>
<td>Item (un-failed state only)</td>
<td>Item (in un-failed and multiple degraded states – allows for consideration of multiple simultaneous failures)</td>
</tr>
<tr>
<td>Failure Mode</td>
<td>Description of how the item failed</td>
<td>Name and description of error model transition (description of how failed contained in properties of transition)</td>
</tr>
<tr>
<td>Immediate Effect</td>
<td>Manual description of immediate effect</td>
<td>Name and properties of state to which transitioned (description contained in properties of destination state)</td>
</tr>
<tr>
<td>Subsequent Effects</td>
<td>Limited to only two additional levels (immediate and end effects)</td>
<td>Not limited to only two levels– could be tens or hundreds of effects (subsequent states and transitions)</td>
</tr>
<tr>
<td>Detection</td>
<td>Description of any detection methods</td>
<td>Detection is described in the context of subsequent transitions (contained in transition properties)</td>
</tr>
<tr>
<td>Compensating Provisions</td>
<td>Description of any mitigation methods</td>
<td>Compensating provisions (i.e., recovery) described in the context of subsequent transitions (contained in transition properties)</td>
</tr>
<tr>
<td>Severity level:</td>
<td>Assigned at time of analysis</td>
<td>Contained in state property</td>
</tr>
</tbody>
</table>
Architecture Analysis & Design Language (AADL)

- Society of Automotive Engineers (SAE) Aerospace Standard AS5506 (2006 for v 2)
- Provides a standardized textual and graphical notation for describing software and hardware system architectures and their functional interfaces
  - architectures (using standard language).
  - expected program behavior (using behavior annex)
  - Failure and recovery behavior (using error annex)
- Representation of failure propagation through system components
  - Event Ports
  - Guards
  - Propagations
AADL Error Model Annex

• AADL annex that supports stochastic analysis

• Defines error model
  – State transition diagram that represents normal and failed states
  – Error models can be associated with hardware components, software components, connections, and “system” (composite) components

• Error model consists of
  – State definitions
  – Propagations from and to other components
  – Probability distribution and parameter definitions
  – Allowed state transitions and probabilities

• Error Model properties
  – Working status of states
  – Descriptive information for initial states, effects (subsequent states), and failure modes (transitions)
  – Initial states
  – Terminal States
AADL Components (graphical representation)

Architecture Model

Error Model

AS 5506 Annex A describes the graphical notation
Automated FMEA Generation Process

- Create AADL System Architecture and Error Models
  - *The Aerospace Corporation’s Model Driven Design and Analysis (MDDA) Workbench which is described later in this presentation*

- Transform the AADL Models into a Petri Net
  - *A Petri Net generation model generator component of the MDDA Workbench generates the Petri Net automatically*

- Produce a “raw” FMEA by tracing the Transitions in the Petri Net and creating a tree of these traces (“Petri Tree”)
  - *An FMEA generator component of the MDDA Workbench performs this automatically*

- Filter the FMEA to extract rows of relevance using a post-processing tool
  - *An FMEA post-processor component of the MDDA Workbench is an interactive tool for performing this process*
Overview of the FMEA Generation Algorithm

- Sample System in AADL
- Petri Net
- Generation of the Petri tree
- Final results
Sample System

Architecture Model

Error Model
Petri Net derived from AADL Model

**Bus**

**Events and Propagations**

**Payload**
Bus, Payload Working

Petri Net

Petri Tree

Bus Working
Payload Working
Bus Failed, Payload Working

**Petri Net**

- Bus working
- Bus Failure
- Bus Recovery
- Payload is Up
- Payload on standby
- Payload recovery
- Payload failed

**Petri Tree**

- Bus Working
- Payload Working
- Bus Failure
- Payload Failed
- Payload Working
Bus Failed, Payload on Standby

**Petri Net**

**Petri Tree**
Bus Working, Payload on Standby

**Petri Net**

**Petri Tree**
Bus Working, Payload Working (already visited)

**Petri Net**

**Petri Tree**

- Bus Working
- Payload Working
- Bus Failure
- Bus Recovery
- Payload Stby_Bus_Down
- Payload Stby_Bus_Down2
- Payload Working
- Payload Failure
- Payload_on_Stby
- Bus Working
- Payload Working
- Bus Working, Payload Working
- Bus Failure
- Bus Working, Payload Working
- Payload Stby_Bus_Down
- Bus Working, Payload Standby
- Bus Recovery
- Bus Working, Payload Standby

*Already Visited*
Bus Working, Payload Failed

Petri Net

Petri Tree
Bus Failed, Payload Failed

Petri Net

Petri Tree
Filtering the Results

Fragment of the resultant FMEA

Portion of the FMEA Post-processing tool user interface
### Postprocessing Tool Output Examples

#### FMEA Generator Original Output

<table>
<thead>
<tr>
<th>Failure</th>
<th>Component</th>
<th>State</th>
<th>Transition</th>
<th>Effect</th>
<th>Transition2</th>
<th>Effect2</th>
<th>Transition3</th>
<th>Effect3</th>
<th>Transition4</th>
<th>Effect4</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td></td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td>Working</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>Sat_Payload</td>
<td>Working</td>
<td>Working</td>
<td>Bus_is_Down(G)</td>
<td>Standby</td>
<td>Standby</td>
<td>Bus_is_up(G)</td>
<td>Working</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Working</td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Sat_Payload</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td></td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td>Working</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Sat_Payload</td>
<td>Failed</td>
<td>Failed</td>
<td>Bus_is_Down(G)</td>
<td>Standby</td>
<td>Standby</td>
<td>Bus_is_up(G)</td>
<td>Working</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Renaming Rule

<table>
<thead>
<tr>
<th>Failure</th>
<th>Component</th>
<th>State</th>
<th>Transition</th>
<th>Effect</th>
<th>Transition2</th>
<th>Effect2</th>
<th>Transition3</th>
<th>Effect3</th>
<th>Transition4</th>
<th>Effect4</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td></td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td>Working</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>Sat_Payload</td>
<td>Working</td>
<td>Working</td>
<td>Bus Failure Prop.</td>
<td>Standby</td>
<td>Standby</td>
<td>Bus Recover Prop</td>
<td>Working</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Working</td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Sat_Payload</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td></td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td>Working</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Sat_Payload</td>
<td>Failed</td>
<td>Failed</td>
<td>Bus Failure Prop.</td>
<td>Standby</td>
<td>Standby</td>
<td>Bus Recover Prop</td>
<td>Working</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Filtered Output

<table>
<thead>
<tr>
<th>Failure</th>
<th>Component</th>
<th>State</th>
<th>Transition</th>
<th>Effect</th>
<th>Transition2</th>
<th>Effect2</th>
<th>Transition3</th>
<th>Effect3</th>
<th>Transition4</th>
<th>Effect4</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td></td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td>Working</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>Sat_Payload</td>
<td>Working</td>
<td>Working</td>
<td>Bus Failure Prop.</td>
<td>Standby</td>
<td>Standby</td>
<td>Bus Recover Prop</td>
<td>Working</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sat_Bus</td>
<td>Working</td>
<td>Working</td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Sat_Payload</td>
<td>Working</td>
<td>Failure</td>
<td>Failed</td>
<td>Recovery</td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

---

22
Model Driven Design and Analysis Workbench Tool Set

• Eclipse Development Environment (Ganymede) and Eclipse Modeling Framework (EMF)
• Component plug-ins
  – **TopCASED** graphical editor to create AADL architecture diagrams (SEI, Aerospace modifications)
  – **Error Model Editor** graphical editor to create AADL error model diagrams (The Aerospace Corporation newly developed)
  – **OSATE** AADL generator (SEI, The Aerospace Corporation modifications)
  – **ADAPT-M** Stochastic Petri net to MoBIUS stochastic analysis network tool (SEI/LAAS Toulouse and The Aerospace Corporation)
  – **MoBIUS** Quantitative Dependability modeling and prediction tool (University of Illinois, Champaign Urbana)
  – **FMEAGEN** FMEA Generator (The Aerospace Corporation newly developed)
  – **FMEA Post-Processor** (The Aerospace Corporation newly developed)
Model Driven Design and Analysis Data Flow

Qualitative Analysis Chain

- FMEAGEN
- FMEA
- FMEA Post-processor
- Qualitative Results

Quantitative Analysis Chain

- ADAPT-M
- SAN file
- MoBIUS
- Quantitative Results

AADL Architectural Model

AADL Error Model

TopCASED

OSATE

Error Model Editor

OSATE
Tool Set Capabilities for Quantitative Evaluation

AADL Architecture and Error Models

Mobius Stochastic Analysis Network Model

Results
Discussion: Recent Experience

• Largest analysis to date consists of 26,000 failure modes,
  - More detailed model of satellite bus
  - 500 Mbyte output file
  - 20 states perform failure mode
  - Longest failure mode sequences have 25 transitions (i.e., 25 effects)

• Care must be used in creation of models
  - Some legal constructs have unpredictable side effects
    • Multiple simultaneous instantaneous transitions
    • Propagations on transitions as opposed to states
    • Event propagation using name matching rather than explicit guard interfaces
  - Automated approach means that many technically insignificant failure mode sequences are produced
    • Example: failure/recovery sequences on some components in combination with termination sequences on others; termination sequence dominates irrespective of what other failure/recovery occurs.
Conclusions

• A new generation tool set for Failure Modes and Effects Analysis (FMEAs) for space systems is under development
  – Based on use of the Architecture Analysis and Design Language (AADL)
  – Graphically oriented
  – Modularized with reusable components

• Automated Generation of FMEA/CA enables multiple iterations analyses throughout all stages of the design
  – Allows design alternatives to be evaluated
    • Strategies for recovering from computing disruptions
    • Handling failure propagation and common mode failures
  – Enables safety and reliability problems to be identified early
    • Of critical importance to all users and stakeholders

• Additional work is needed to create an “industrial strength” capability
  – Proper representation of event and error propagations
  – Use of instantaneous transitions
  – Syntax checker to avoid errors
  – Model design rules to create more reasonably sized FMEA tables
References