Models Extracted from Text for System-Software Safety Analyses

Jane T. Malin, Ph.D.; NASA Johnson Space Center; Houston, Texas, USA

Abstract

This presentation describes extraction and integration of requirements information and safety information in visualizations to support early review of completeness, correctness, and consistency of lengthy and diverse system safety analyses. Software tools have been developed and extended to perform the following tasks: 1) extract model parts and safety information from text in interface requirements documents, failure modes and effects analyses and hazard reports; 2) map and integrate the information to develop system architecture models and visualizations for safety analysts; and 3) provide model output to support virtual system integration testing. This presentation illustrates the methods and products with a rocket motor initiation case.
Models Extracted from Text for System-Software Safety Analyses

Project:
Automated Tool and Method for System Safety Analysis

Jane T. Malin, Principal Investigator
NASA JSC Team: Land Fleming, Carroll Thronesbery, David Throop
Triakis Team: Ted Bennett and Paul Wennberg

Software Assurance Symposium
August 2010
Essential Early Safety Reviews

• Requirements and design problems are the source of most operational software defects
  – System integration, interfaces, failures and hazard causes

• Analysis of information for Preliminary Design Review (PDR) is needed

System Integration Operations and Stresses
Integrated Review of Scattered Information

• Need: Efficient system safety reviews of large sets of contractor documents
  – Make short-fuse reviews of requirements, safety analyses and plans manageable
  – Integrate key information from diverse uncoordinated and evolving documents
    • Interface Requirements Documents (IRD)
    • Failure Modes and Effects Analysis (FMEA)
    • Hazard Reports (HR)
    • Fault Detection, Isolation and Response (FDIR)
Automated Modeling Solution

- Models constructed from information extracted from text documents
- Visualizations for integrated insight into information scattered in large documents
- Output files and reports for model reuse in virtual testing and analysis for FDIR design

Model Graph

Connection Description Pop up from Arrow

DOCUMENT TITLE:
LAS FMEA

FMEA-0001

Type of thing sent: Energy_or_Power

Source Text: The initiators shall provide initiation energy to the rocket motor igniter.
NASA Application Cases

• Constellation Program (CxP) CEV cases
  – Launch Abort System (LAS) with focus on Ordnance
  – Crew Module (CM)
  – Service Model (SM) Propulsion

• Useful in other aerospace projects where safety engineers perform early reviews of contractor products
  – FMEAs, Hazard Reports, safety requirements
NASA Automated Modeling Tools

• Semantic Text Analysis Tool (STAT)
  – Parsing and information extraction from text to XML
  – Multiple information types and document types

• Hazard Identification Tool (HIT)
  – Model construction and visualization
  – Component-connection models and visualizations
  – Analyses of redundancy, dependencies, linkages

• Output for further modeling and analysis
  – Information from FMEAs and Hazard Reports for FDIR
  – Virtual System Integration Lab (VSIL) simulator
**STAT Linguistic Text Extraction**

- Powerful linguistic tagger and extractor
  - Advanced natural language processing
  - Extensive aerospace nomenclature
- Extractions for models

<table>
<thead>
<tr>
<th>Document</th>
<th>Section</th>
<th>Model Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMEA</td>
<td>Front matter section</td>
<td>System hierarchy</td>
</tr>
<tr>
<td></td>
<td>Worksheet hierarchy section</td>
<td>System hierarchy</td>
</tr>
<tr>
<td></td>
<td>Worksheet: Item Function</td>
<td>Components, connections</td>
</tr>
<tr>
<td></td>
<td>Worksheet: Failure modes and Causes</td>
<td>Failure mode descriptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cause descriptions</td>
</tr>
<tr>
<td>Hazard Report</td>
<td>Cause Descriptions, Cause Control Descriptions</td>
<td>Component types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Controls</td>
</tr>
<tr>
<td>IRD</td>
<td>Interface requirements</td>
<td>Components, connections</td>
</tr>
<tr>
<td>All Documents</td>
<td>Acronym Section</td>
<td>Acronyms</td>
</tr>
<tr>
<td></td>
<td>Titles, identifiers</td>
<td>Traceability Information</td>
</tr>
</tbody>
</table>
Graph Display: Redundant Components and Paths

Tree Display: Orion and Subsystems in Model
HIT Models for Review and Reuse

• Integrated review of multiple FMEAs
  – Completeness, duplication, consistency and redundant components and paths
  – Compare versions and re-analyze
  – Upstream and downstream dependency paths
• Review of FMEAs and Hazard Reports that are related to the same components
• Reuse of model information
  – Models and failure modes for simulation tests
  – Output information for fault analysis and FDIR design
Source Information Pop ups

- Click on components and connections to pop up FMEA and Hazard Report Information

Highlighting shows components with Hazard Report references

---

**Initiators - FMEA**

DOCUMENT TITLE: LAS FMEA

FMEA-00001

**Item Function:**
The initiators provide initiation energy to the rocket motor igniter.

Failure Mode 1
Criticality: 1
Failure of initiator

---

**Initiators - Hazard Report**

**Initiator – Igniter Connection**

Type of thing sent: Energy_or_Power

Source Text: The initiators shall provide initiation energy to the rocket motor igniter.

**Hazard Report References for components of type Initiator**

HR Number: LAS-FLT-00001
HR Title: Failure to Function Rocket Motor Results in Loss of Vehicle Control

Cause F: Igniter Failure
  Cause Description ...
  Effect(s) ...
  Control(s)

  **Bullet:** 1
  Design. The igniter shall be designed to properly function upon receiving **initiator** stimulus.
  Verification(s) ...

Cause G: Inadequate motor performance
Model Reuse for Design and Test

- Automated generation of fault analysis spreadsheets with HIT model information (from FMEAs and HRs)
  - Progress in extending CxP functional fault analysis (FFA) spreadsheets with Hazard Report information

- Triakis Virtual System Integration Lab simulator uses HIT FMEA Output to test flight software
  - Automatic translation of HIT FMEA output into failure mode test framework files
  - Failure mode tests verify integrated system and software
    - Manifest component failures and record software response
    - Monitor the state of any simulated part or signals between parts
  - HIT displays link back to VSIL test results and methods, for analysis and review
Links to Test Plans and Results

- Clicking on components brings up test plans and results.
Evaluations by Safety Engineers

• CEV avionics/software safety engineers – basis for a great leap in productivity of reviews
  – One-stop rapid integrated review
    • Thousands of documents for review were beyond human capabilities
    • Linking to specific information in the source documents makes the information easily accessible
  – Graph display makes key information stand out
    • Highlights missing or inconsistent information or terminology
    • Easy to see architecture and components that have no outputs or insufficient inputs, indicating omissions in design or documentation
    • Helps engineers check redundancy and review potential hazard paths
  – Engineers can trace from HRs to linked FMEAs, to find more detailed FMEA information for the HR
    • Essential but not possible before
Software Technology Maturity

• Technology Readiness Level 7: Prototypes that fully demonstrate operational and engineering feasibility
  – Prototype software with all key functionality should be available for demo and test
    • Distribution image and CEV models delivered to SMA on notebook computer
  – Prototype code should be relatively clean – ‘tis
  – Limited documentation should be available
    • User documentation and the Concept of Operations
NASA-owned Prototypes

• FY10 milestone: Tools on installation CD
  – Triakis COTS VSIL LAS simulator (C++) and documentation will be delivered separately

• STAT implemented in Perl and open-source LISP
  – Parsers: open-source Stanford and University of Central Florida
  – Aerospace nomenclature implemented in Protégé ontology

• HIT implemented in Allegro Common LISP
Recommended Next Steps

• Another Aerospace Case
  – Safety engineers can use these tools to significantly increase productivity of reviews

• Further evaluate feasibility of model reuse
  – Fault analysis and FDIR design spreadsheets
  – Virtual system integration testing for safety

• Develop software tool products based on existing Concept of Operations and prototypes
  – Develop according to NPR 7150.2

• Extend to assist developers of FMEAs and Hazard Reports, with evaluation and guidance
Summary: Models Extracted from Text

• Coordinated data and documents are desirable, but virtually impossible

• Integrated information for efficient safety review requires text extraction
  – Important information stands out
  – Details are a click away

• Information for follow-on design and test
  – Model for virtual safety testing
  – Information output for FDIR design