Hazard Analysis for Software Safety

Technology Maturation

17-939A
Jennifer Morris
4/19/2005

Where Software Hazard Analysis Began

☐ Software Safety
  ■ Different from hardware (failure modes, complexity)
  ■ Hardware is covered, let’s tackle software

☐ Design/Code Analysis
  ■ Artifact of hardware analysis
    ○ Software Fault-Tree Analysis [Leveson & Harvey 83, Cha et al. 88]
    ○ Software Failure-Modes & Effects Analysis [Bowles & Wan 01]

☐ Searching for the right things in the right places?

Lutz 92
Analyzing Software Requirements Errors in Safety-Critical, Embedded Systems

☐ Question: Generalization/Characterization
  ■ What are the most common causes of safety-related software errors in safety-critical, embedded systems?

☐ Result: Result from specific analysis
  ■ Few errors in code
  ■ Recognizing requirements
  ■ Interfaces with rest of system (hardware)

☐ Validation: Experience
  ■ Two systems (Voyager & Galileo)
  ■ 387 errors in integration & system testing, 209 safety-related

☐ Focused on safety-related errors
☐ Requirements primary source of safety-critical errors
☐ Interaction of software with the rest of the system

☐ Is this enough to show generality?
Requirements Hazard Analysis: Targeting Requirements Errors

- Set of criteria (failure modes, hazards)
- Interfaces & system effects
- Validation by example & experience

- State-machine modeling [Jaffe, et. al 91]
  - Safety criteria for state machines (determinism)
  - Formal model of software (not system)
  - Validation by example

- Safety checklist [Lutz, 93]
  - Criteria for interfaces & robustness
  - System causes, software effects
  - Validation by experience (two existing spacecraft, one new)

Requirements Hazard Analysis: Deriving Safety Requirements

- FTA & FMEA [Maier 95]
  - System-level FTA to derive safety requirements
  - Followed up by SFMEA & SFTA of software requirements
  - Validation by experience (remote handling robot)

- SFTA to derive safety requirements [Hansen et. al 98]
  - Formalized fault-trees in duration calculus
  - System hazards & events
  - Validation by example

- SFTA for product lines [Dehlinger & Lutz 04]
  - Fault-trees + variability/commonality
  - Partial re-use of fault-trees
  - System hazards & events
  - Validation by example

Requirements Hazard Analysis: Targeting Requirements Errors

- HAZOP for software [McDermid & Pumfrey 94]
  - Data type & flow faults using guidewords
  - Software causes, system effects
  - Validation by example

- Petri nets & FMEA [Goddard 96]
  - Petri net model of system
  - Table of failures in Petri net, causes & effects
  - Validation by example

- SFMEA [Lutz 97]
  - Table of failures (inputs, outputs, & processes) & effects
  - System hazards & causes
  - Validation by experience (new spacecraft)

Requirements Hazard Analysis: Integrated Safety Analysis of Requirements Specifications

- Question: *Feasibility/Analysis of an instance*
  - Is it possible to do apply multiple hazard analysis techniques to the same safety-critical software requirements specification?
  - How do three different hazard analysis techniques compare?

- Result: *Result from specific analysis*
  - State machine hazard analysis
    - Forward simulation, forward search, backward search
    - Each finds different errors

- Validation: *Experience*
  - Guidance system from NASA Ames (research stage)
  - Academic researchers

- Comparison of different techniques
- Preparing for the analyses found many errors
  - Creating the model, completeness & consistency checks

4/19/2005 Jennifer Morris 17-939A
Where Software Hazard Analysis is Now

- System safety
  - Hazards arise from interactions
- Requirements-based analysis
  - Data flows
  - System processes
- Validation methods weak
  - Enhanced method, but no comparison
  - Real-world systems, but no real-world analysts

Bibliography [1 of 2]