

Software Reliability and Safety

CSE 8317 — Fall 2012

Prof. Jeff Tian, tian@engr.smu.edu
CSE, SMU, Dallas, TX 75275
(214) 768-2861; Fax: (214) 768-3085
www.engr.smu.edu/~tian/class/8317.12f

SSE.2: Hazard Analysis & Resolution

- Hazard Analyses and Techniques
- HA Techniques: FTA and ETA
- Hazard Resolution
- Damage Reduction

Safety Analysis

- Hazard and risk identification:
 - ▷ Accident scenarios: actual/hypothetical
 - starting points for safety
 - ▷ Focus: operations and operational env.

- Hazard analysis and assessment:
 - ▷ Fault trees: (static) logical conditions
 - ▷ Event trees: dynamic sequences
 - ▷ Other analyses/assessment techniques

- Hazard and risk resolution
 - ▷ Hazard elimination
 - ▷ Hazard reduction
 - ▷ Hazard control
 - ▷ Damage control

Hazard Analyses: Types

- Sub-system hazard analyses (SSHA)
 - ▷ Hazard within individual sub-system
 - ▷ Component/sub-system in isolation

- System hazard analyses (SHA)
 - ▷ Focus: interface and interaction
 - ▷ Subsys/env/human effect on system
 - ▷ Throughout development process
 - ▷ Focus on early phases to provide info. for other activities (hazard resolution and safety verification)

- SHA/SSHA in software process
 - ▷ Throughout development process
 - ▷ Focus on early phases to provide info. for other activities (hazard resolution and safety verification)

Hazard Analyses: Techniques

- Primary techniques for SHA/SSHA:
 - ▷ Fault-tree analyses (FTA)
 - ▷ Event-tree analyses (ETA)
 - ▷ SQE Ch.16.4 and Safeware Ch.14.

- Other techniques:
 - ▷ Design reviews & checklists
 - ▷ Hazard indices
 - ▷ Risk trees
 - ▷ Cause-consequence analysis (CCA)
 - ▷ Hazard & operability analysis (HAZOP)
 - ▷ Failure modes and effect analysis (FMEA)
 - ▷ FMECA (FMEA + Criticality), etc.
 - ▷ Above: “Safeware” Ch.14.
 - ▷ Specific to software: “Safeware” Ch.15.

- FTA and ETA slides from SQE Ch.16.4.

Hazard Analysis: SFTA

- SFTA: Software FTA

- ▷ Same concept applied to software
- ▷ Actual implementation (white-box)
- ▷ Language elements (high-level):
 - assignment and function calls
 - branching statement, loops, etc.
- ▷ Also for specification/architecture
 - black-box control flow diagram
 - equivalent language representation

- SFTA construction:

- ▷ Templates/examples for diff. statements
- ▷ Safeware 18.2.2 (pp.497-507)

⇒ Additional work needed,
especially for system design/architecture

Hazard Analysis: ETA & CCA

- ETA alone: trace of accident.
May desire explanation also (from FTA)
- Cause-consequence diagram (CCA):
 - ▷ Combine ETA with FTA
 - ▷ Explaining decisions in ET
- Using ETA and CCA:
 - ▷ Partial vs. total ETA
 - ▷ Focus on main consequences
 - ▷ Details:
“Safeware” 14.5-14.6 (pp.327-pp.335)

Hazard and Risk Resolution

- Generic hazard resolution techniques (in order of their precedence):
 - ▷ Hazard elimination:
 - eliminate hazard sources
 - ▷ Hazard reduction:
 - reduce hazard likelihood/severity
 - ▷ Hazard control:
 - control hazard severity/scope
- Hazard resolution \Rightarrow prob(accident) \downarrow
- Related issues:
 - ▷ Basis: hazard identification and analysis via FTA, ETA, CCA, etc.
 - ▷ Many specific techniques
 - ▷ Related to QA and SRE
 - ▷ Risk resolution: damage reduction too

Hazard Elimination

- Elimination of hazard
 - ▷ Intrinsically safe (sub-)system
 - ▷ All eliminated: feasibility & cost?
 - ▷ Certain types of hazard eliminated
 - ▷ Direct use of hazard identification and analysis results.

- Specific techniques: “Good SE & SSE”
 - ▷ Component substitution (\Leftarrow FTA)
 - ▷ No single point of failure (\Leftarrow ETA)
 - ▷ Simplification of building blocks
 - ▷ Decoupling of system architecture
 - ▷ Human errors/hazardous material elim.
 - ▷ Component safety certification:
 - formal verification
 - components identified by FTA etc.
 - ▷ Link to testing/FT/QA activities

Hazard, Controllability, & Observability

- Related to hazard resolution, particularly hazard reduction and control.
- Controllability:
 - ▷ Between any two system states
 - ▷ Desirable/safe states: maintain
 - ▷ Fail \Rightarrow action \Rightarrow safe (haz. control)
 - ▷ Controllability limits:
 - system design/structure limit
 - energy/capacity limit
- Observability: observation of system states (and failures), basis for control.

Design for Controllability

- Maintain safe states
 - ▷ Use built-in control
 - ▷ Monitoring: observation \Rightarrow control
 - ▷ Multiple checks \Leftarrow monitoring
 - ▷ Mostly in hazard reduction

- Enhancing control opportunities:
 - ▷ Incremental control: more control points
 - ▷ Intermediate states: more obs. points
(\Rightarrow more control opportunities)
 - ▷ Decision aid: easier/more control points
 - ▷ Both in hazard reduction and especially in hazard control

Hazard Reduction

- Hazard reduction:
 - ▷ Severity reduction:
 - change failure characteristics
(failure $\wedge \neg$ hazard)
 - various locks/barriers
 - ▷ Likelihood reduction:
 - reduce failure probability
 - in combination with above
 - also: most QA/SRE related techniques

- Specific techniques:
 - ▷ Design for controllability
 - ▷ Barriers and locks (passive)
 - ▷ Failure/hazard probability/severity ↓
(accident probability↓)

Hazard Reduction: Techniques

- Monitoring and checks: Fig 16.2
 - ▷ Hardware checks: lowest level
 - ▷ Code-level checks: assertions
 - connection to PSC (SSE.4)
 - ▷ Audit checks: independent monitoring
 - ▷ Supervisory checks: system/highest level

- Locks and barriers (passive)
 - ▷ Lock-outs (preventing hazard)
 - ▷ Lock-ins (maintaining safety conditions)
 - ▷ Interlocks (correct order/combinations)
 - ▷ Other barriers (extra cap./redundancy/etc.)

Hazard Reduction: Techniques

- Hazard probability minimization:
 - ▷ Design with extra capacity:
 - safety factors/margins example
 - melt temp. T_m and margin M
 - \Rightarrow safety bound $T_s = T_m - M$
 - ▷ Redundancy: similar
 - ▷ QA and SRE: failure \downarrow
 - focused hazard probability min.
 - with FTA/ETA/etc. help

- Redundancy (FT etc.) \Rightarrow prob(hazard) \downarrow :
 - ▷ Hardware redundancy/backup
 - ▷ Software redundancy:
 - fault tolerance (NVP, & (?) RB)
 - anticipated input/env. enlargement
 - “fool-proof” software
 - ▷ Recovery: similar to RB in FT
 - ▷ Hardware/software interlocks

Hazard Resolution: Hw/Sw Interlock

- Interlock software
 - ▷ Software used as safety interlock
 - (s/w usage: data/control/safety)
 - example: emergency shut-down s/w
 - ▷ More stringent safety requirement:
 - most s/w function safety-related
 - should not rely solely on s/w
 - Therac-25 accident lessons

- Hardware/software interlock
 - ▷ Limitation of s/w backups:
 - diversity and independence problems
 - ▷ Hardware backups and interlocks:
 - different characteristics
 - different failure mechanisms
 - more likely to be *independent*
 - passive/active safety devices
 - ▷ Combine the advantages \Rightarrow safety \uparrow

Hazard Control

- Hazard control:
 - ▷ Detecting hazard, then control it
 - ▷ Built-in control: by design
 - ▷ Change after detection:
 - (passive) limits
(mostly outside system)
 - (active) control devices/sub-systems
- Specific techniques:
 - ▷ Limiting exposure (duration↓)
 - ▷ Isolation and containment
 - ▷ Protection systems
 - ▷ Fail-safe design

Hazard Control: Techniques

- Internal system change:
 - ▷ Isolation of hazard event
 - ▷ Containment around hazard event
 - ▷ Fail-safe design (passive)

- System augmentation:
 - ▷ Protection system (PS) added on:
 - hazard \Rightarrow PS action \Rightarrow safe
 - shut-down or partial shut-down
 - e.g., automatic coolant injection or pressure relief
 - ▷ Controllability limit (earlier)
 - ▷ Partial solution may be necessary:
 - reduce the severity
 - bring to a neighboring state

Risk Resolution: Damage Reduction

- Damage reduction: Why?
 - ▷ Risk factors:
 $f(\text{prob}(\text{haz}), \text{prob}(\text{haz} \rightarrow \text{acc}), \text{damage})$
 - ▷ All the hazard resolution techniques
 $\Rightarrow \text{risk} \neq 0$ still!
 - ▷ Damage reduction needed
 - ▷ Passed “point of no return”

- Specific techniques:
 - ▷ Escape routes (lifeboats, fire escapes, evacuation plans, etc.)
 - ▷ Safe abandonment (haz. waste disposal)
 - ▷ Devices for limiting damage:
 - auto safety devices
 - limited melt-down
 - collapsible signpost, etc.

Perspectives

- SSE: Augment S/w Eng.
 - ▷ Analysis to identify hazard
 - ▷ Design for safety
 - ▷ Verify safety constraints (next module)
 - ▷ Leveson's s/w safety program

- Dealing with hazard/risk in SSE:
 - ▷ Hazard identification and analysis
 - ▷ Design for safety/hazard resolution:
 - Hazard elimination/reduction/control
 - ▷ Damage reduction
 - ▷ Safety verification
 - ▷ All in SSE context: hazard focus.