Software Safety

Hazard and Failure Analysis

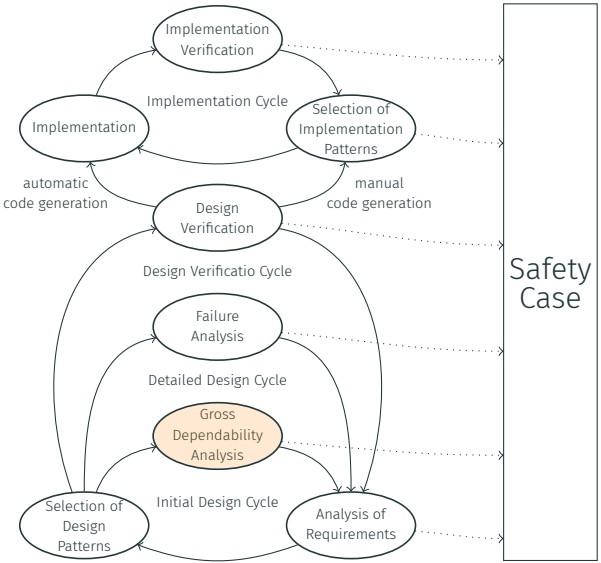
Prof. Dr.-Ing. Patrick Mäder, MSc. Martin Rabe

Copyright

- This lecture is based on Chris Hobbs's book "Embedded Software Development for Safety-Critical Systems" [Ho16], heavily inspired by his course on Embedded Safety-Critical System Development, by Dr. Anton Setzer's course on Critical Systems at University of Wales Swansea, and Nancy Leveson's course on "System Safety Engineering" at MIT.

Contents

Software Lifecycle



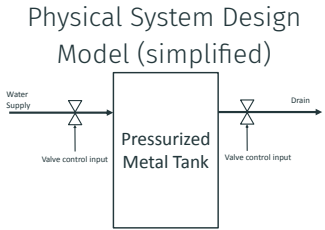
Hazard Analysis Overview

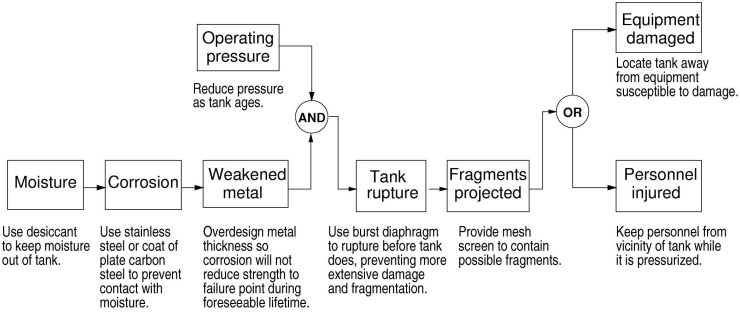
Hazard (Causal) Analysis

- “Investigating an accident before it happens”
- Goal is to identify causes of accidents (before they occur) in order to eliminate or control them in
  - Design
  - Operations

- **Requires** (even if only in the mind of the analyst)
  - A system design model
  - An accident model

Chain-of-events Example

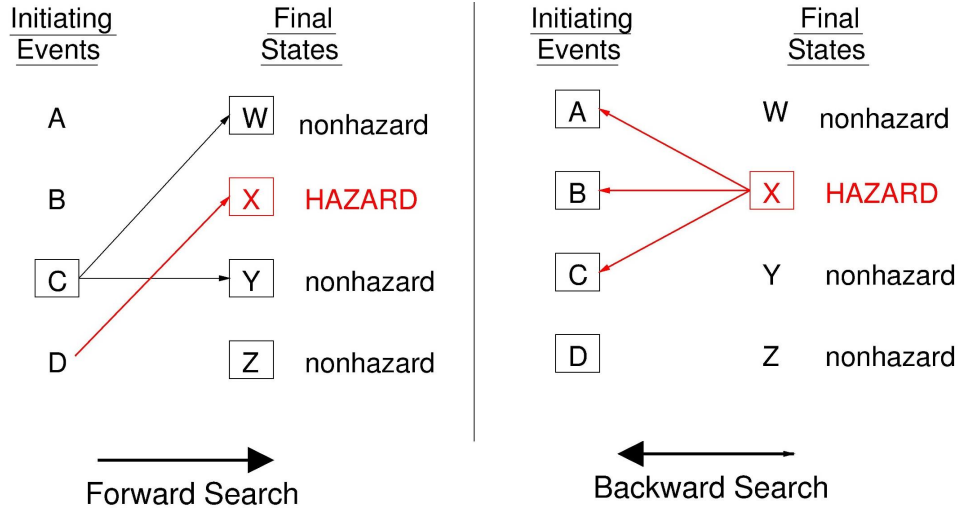




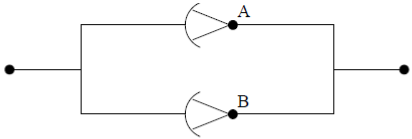
From Leveson, Nancy (2012). Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, © Massachusetts Institute of Technology. Used with permission.

How do you find the chain of events before an accident?

Hazard Perspective: Forward vs. Backward Search



A Forward Search Example (FMEA Analysis)



- FMEA for a system of two amplifiers in parallel

Component		Failure mode		Effects	
				Critical	Noncritical
A		Open			X
		Short		X	
		Other		X	
B		Open			X
		Short		X	
		Other		X	

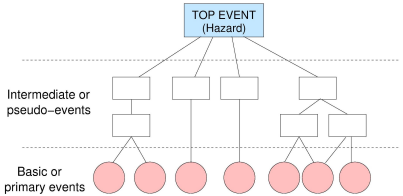
Vesley et al.: **Fault Tree Handbook**, 1981.

A Backward Search Example (Five Whys Analysis)

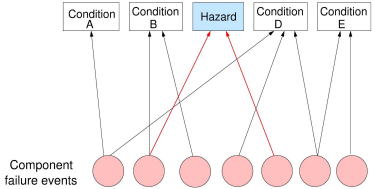
- **Problem**: The Washington Monument is disintegrating.
  - **Why** is it disintegrating?
    Because we use harsh chemicals
  - **Why** do we use harsh chemicals?
    To clean pigeon droppings off the monument
  - **Why** are there so many pigeons?
    They eat spiders and there are a lot of spiders at monument
  - **Why** are there so many spiders?
    They eat gnats and lots of gnats at monument
  - **Why** so many gnats?
    They are attracted to the lights at dusk
- **Solution**:
  - → Turn on the lights at a later time.

System Perspective: Top-Down vs. Bottom-Up Search

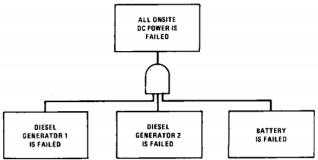
Top-down search



Bottom-up search



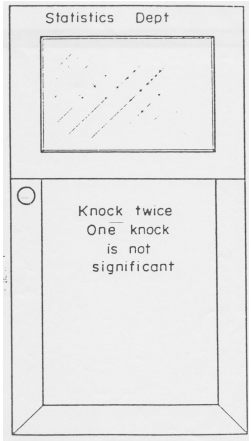
Example



Vesley et al.: **Fault Tree Handbook**, 1981.

Quantitative Methods

- The quantification is usually based on probability theory and statistics
- Common assumptions
  - Behavior is random
  - Each behavior independent
  - Identical distributions
- Are these good assumptions?
  - Hardware?
  - Humans?
  - Software?

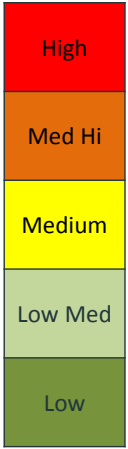


Hazard Level Assessment

- Hazard = combination of severity and likelihood
- Difficult for complex, human/computer controlled systems
- Challenging to determine likelihood for these systems
  - Software behaves exactly the same way every time
    - Not random
  - Humans adapt, and can change behavior over time
    - Adaptation is not random
    - Different humans behave differently
    - Not independent and identically distributed (I.I.D)
  - Modern systems almost always involve new designs and new technology
    - Historical data may be irrelevant

→ **Severity is usually adequate** to determine effort to spend on eliminating or mitigating hazard.

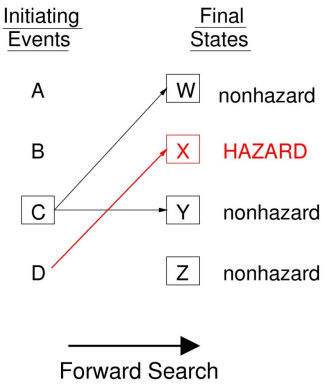
Hazard Level or Risk Level:



FMEA

Failure Modes and Effects Analysis (FMEA)

- **FMEA**: a systematic method for identifying and preventing product and process problems before they occur
  - Primarily safety, but also quality in general
- **Forward search technique**
  - Initiating event: component failure
  - Goal: identify effect of each failure
- **History**
  - First proposed: 1949 [MIL-P-1629]
  - First industrial usages in aerospace industry (mid-1960s) → studying safety issues



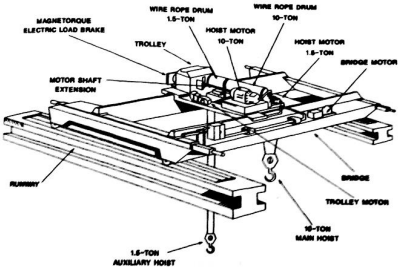
The General FMEA Process

1. Identify individual components
   break the main system into subsystems
2. Identify failure modes
   per subsystem: determine whether its failure would affect the main system
   - If not: ignore that subsystem
   - If yes: iteratively break this subsystem into further subsystems until the component level is reached
3. Identify failure mechanisms (causes)
4. Identify failure effects

Identifying Failure Modes, Mechanisms, and Effects

- For each component identified as above, do the following:
  - Study the component's failure modes = the ways, the component can fail
  - Identify the mechanisms causes that may cause each failure mode
  - Assess each failure's effects locally and globally
    - Usually the worst-credible case with consequence severity and probability of occurrence is assessed, if this is possible to calculate.
    - Determine its mission phase (installation, operation, maintenance, repair).
    - Identify, whether the failure is a single-point failure.
      (Single point failure = failure of a single component that could bring down the entire system.)

FMEA Worksheet and Bridge Crane Example



Failure Mode and Effect Analysis

Program:_____ System:_____ Facility:_____
Engineer:_____ Date:_____ Sheet:_____

Component Name	Failure Modes	Failure Mechanisms	Failure effects (local)	Failure effects (system)
Main hoist motor	Inoperative, does not move	Defective bearings Motor brushes worn Broken springs	Main hoist cannot be raised. Brake will hold hoist stationary	Load held stationary, cannot be raised or lowered.

FMEA Follows an Accident Model

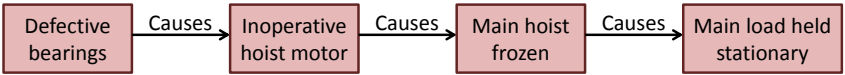
- FMEA method

Failure Mode and Effect Analysis

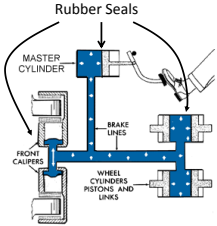
Program:_____ System:_____ Facility:_____
Engineer:_____ Date:_____ Sheet:_____

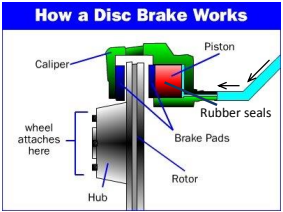
Component Name	Failure Modes	Failure Mechanisms	Failure effects (local)	Failure effects (system)
Main Hoist Motor	Inoperative, does not move	Defective bearings Loss of power Broken springs	Main hoist cannot be raised. Brake will hold hoist stationary	Load held stationary, cannot be raised or lowered.

- Compare: accident model (chain-of-events)



FMEA Exercise: Automotive brakes



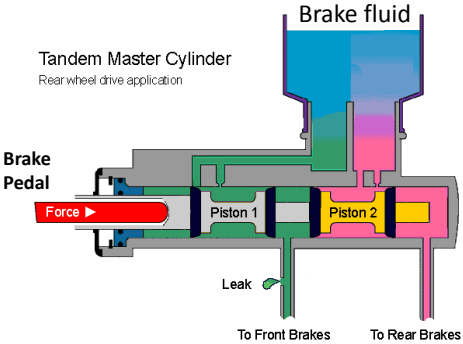


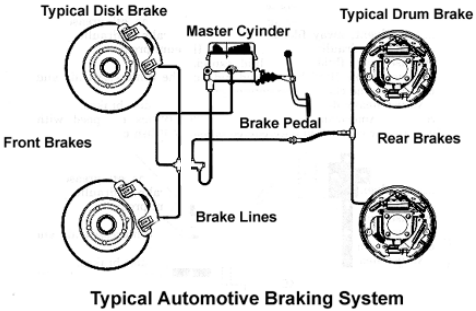
- System components
  - Brake pedal
  - Brake lines
  - Rubber seals
  - Master cylinder
  - Brake pads

- FMEA worksheet columns
  - Component
  - Failure mode
  - Failure mechanism
  - Failure effect (local)
  - Failure effect (system)

How would you make this system safe?

FMEA Exercise: Actual Automotive Brakes





Typical Automotive Braking System

- FMEA heavily used in mechanical engineering
- Tends to promote redundancy

→ Useful for physical/mechanical systems to identify single points of failure

Toyota's Unintended Acceleration Accidents

- 2004–2009
  - 102 incidents of stuck accelerators
  - Speeds exceed 100 mph despite stomping on the brake
  - 30 crashes
  - 20 injuries
- 2009, August
  - Car accelerates to 120 mph
  - Passenger calls 911, reports stuck accelerator
  - Some witnesses report red glow / fire behind wheels
  - Car crashes killing 4 people
- 2010, July
  - Investigated over 2,000 cases of unintended acceleration

Captured by FMEA?

Limitations of FMEA

- Captures component failure incidents only
  - Unsafe interactions? Design issues? Requirements issues?
- Single component failures only
  - Multiple failure combinations not considered
- Requires detailed system design
  - Limits how early analysis can be applied
- Works best on hardware/mechanical components
  - Human operators? (Driver? Pilot?)
  - Software failure?
  - Organizational factors (management pressure? culture?)
- Inefficient, analyzes unimportant + important failures
  - Can result in 1,000s of pages of worksheets
- Tends to encourage redundancy → often leads to inefficient solutions
- Failure modes must already be known
  - Best for standard parts with few and well-known failure modes

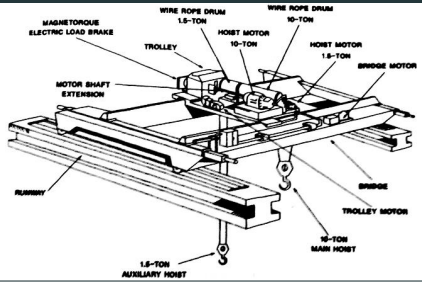
FMECA

Failure Modes Effects and Criticality Analysis (FMECA)

- Same as FMEA, but but with “criticality” information per failure
- Criticality
  - Can be ordinal severity values
  - Can be likelihood probabilities
  - An expression of concern over the effects of failure in the system [Vincoli, 2006, Basic Guide to System Safety]

FEMCA: Bridge Crane Example Revisited

- Could also specify likelihood (e.g. probability of occurrence)



Failure Mode and Effect Analysis

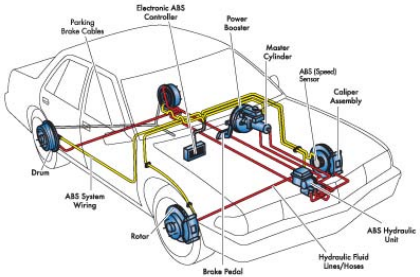
Program:_____ System:_____ Facility:_____
Engineer:_____ Date:_____ Sheet:_____

Component Name	Failure Modes	Failure Mechanisms	Failure effects (local)	Failure effects (system)	Criticality Level
Main hoist motor	Inoperative, does not move	Defective bearings Loss of power Broken springs	Main hoist cannot be raised. Brake will hold hoist stationary	Load held stationary, cannot be raised or lowered.	(5) High, customers dissatisfied

Safety $\neq$ Reliability

Safety vs. Reliability

- Common assumption: Safety = reliability
- How to improve safety?
  - Make everything more reliable!
- Making car brakes safe
  - Make every component reliable
  - Include redundant components



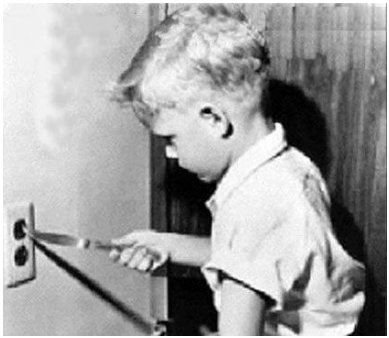
Is this a good assumption?

A simpler example

Safe or unsafe?

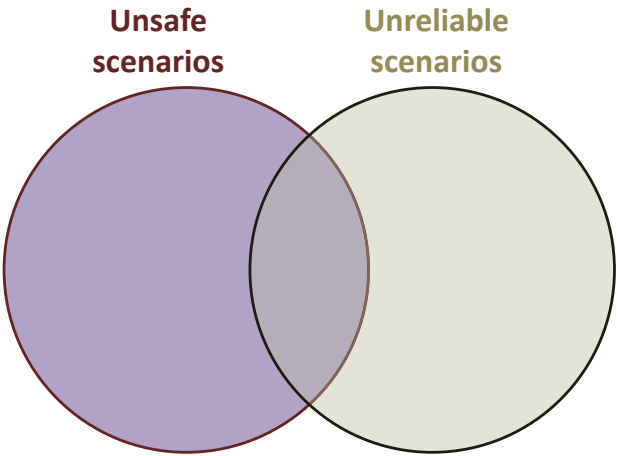
Safety Is Not a Component Property

- Safety is an emergent property of the system
  - Depends on context and environment



Individual components are not inherently safe or unsafe

Safety vs. Reliability



Safety ≠ Reliability

- Safety often means making sure X never happens
- Reliability usually means making sure Y always happens

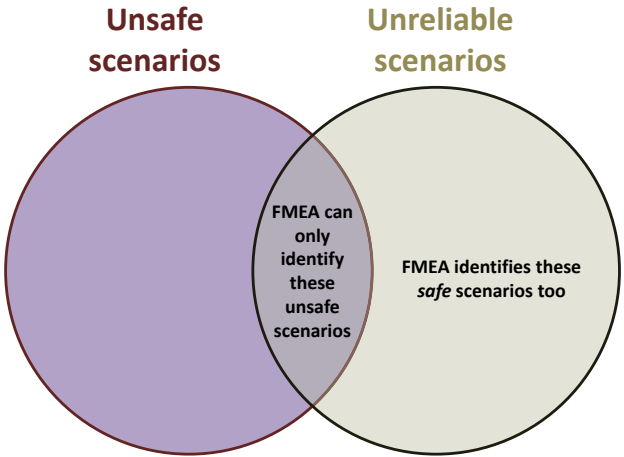
	Safe	**Unsafe**
Reliable	•Typical commercial flight	
Unreliable		•Aircraft engine fails in flight

Safety ≠ Reliability

- Safety often means making sure X never happens
- Reliability usually means making sure Y always happens

	Safe	**Unsafe**
Reliable	•Typical commercial flight	•Computer reliably executes unsafe commands •Increasing tank burst pressure •A nail gun without safety lockout
Unreliable	•Aircraft engine won't start on ground •Missile won't fire	•Aircraft engine fails in flight

Safety vs. Reliability

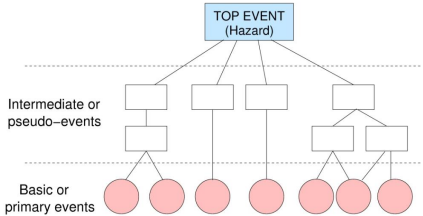


- FMEA is a reliability technique → explains the inefficiency
- FMEA sometimes used to identify unsafe outcomes

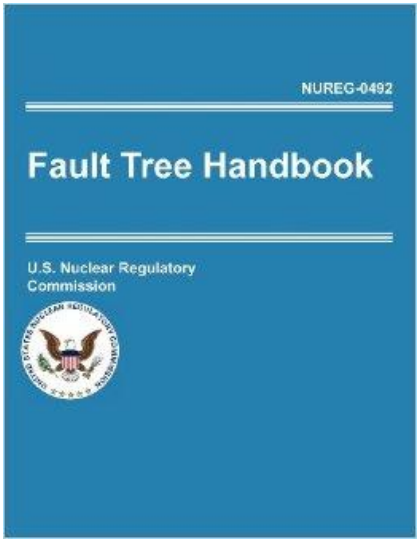
Fault Tree Analysis

Fault Tree Analysis (FTA)

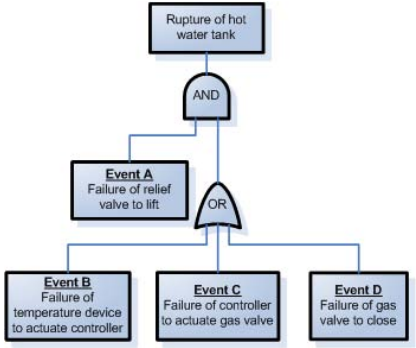
- 1961: Bell labs analysis of Minuteman missile system
- Today one of the most popular hazard analysis techniques
- Top-down search method
  - Top event: undesirable event
  - Goal is to identify causes of hazardous event



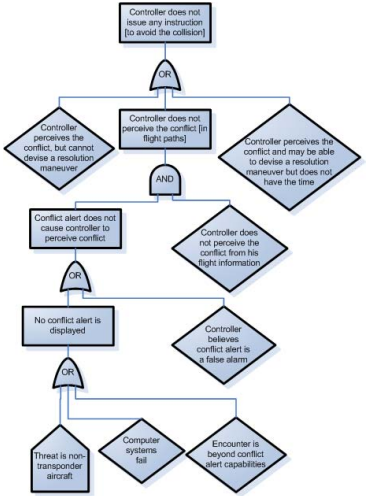
1. Definitions
   - Define top event
   - Define initial state/conditions
2. Fault tree construction
3. Identify cut-sets and minimal cut-sets



Fault Tree Examples

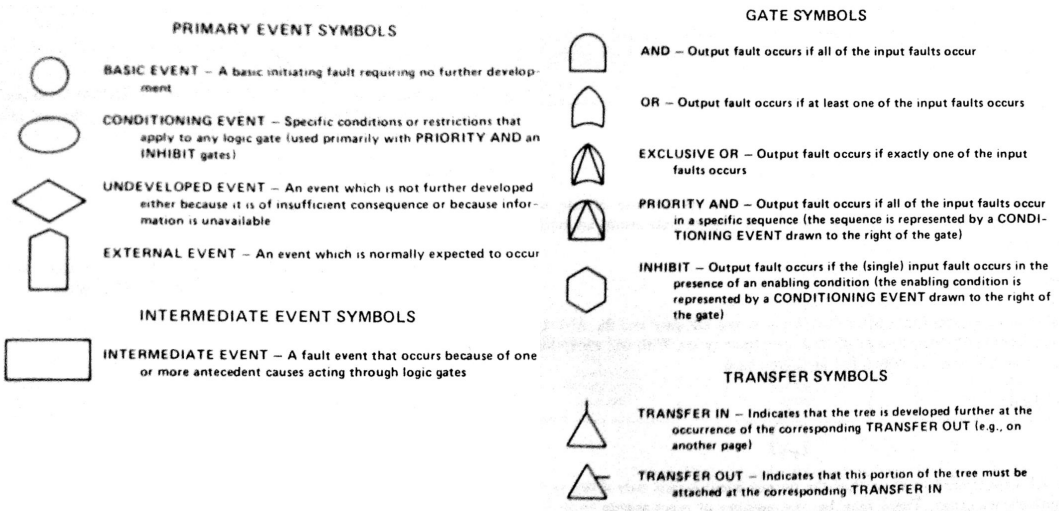


Example from original 1961 Bell Labs study



Part of an actual TCAS fault tree (MITRE, 1983)

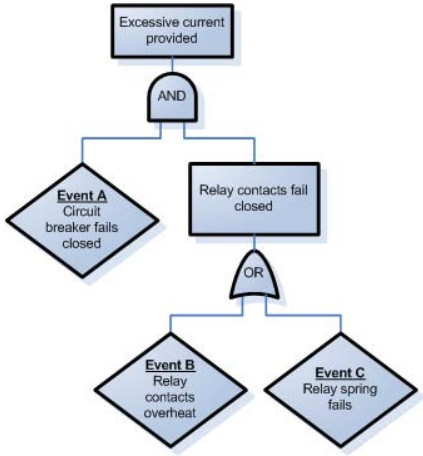
Fault Tree Symbols



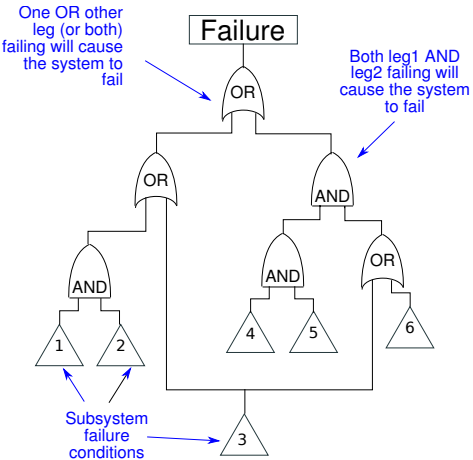
[From NUREG-0492 (Vesely, 1981)]

Fault Tree Cut-sets

- **Cut-set**: combination of basic events (leaf nodes) sufficient to cause the top-level event
  - Ex: (A and B and C)
- **Minimum cut-set**: a cut-set that does not contain another cut-set
  - Ex: (A and B)
  - Ex: (A and C)

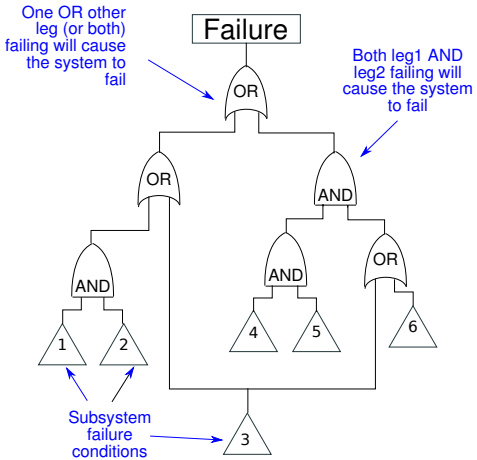


Minimum Cut-set Example



Minimum Cut Sets?

Minimum Cut-set Example

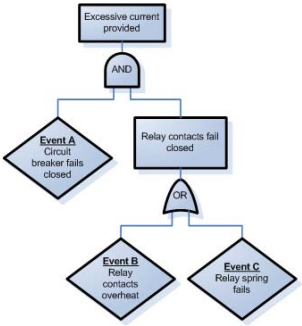


Minimum Cut Sets?

{ 3 }, { 1, 2 } and {4, 5, 6 }

FTA Follows an Accident Model

- Fault tree



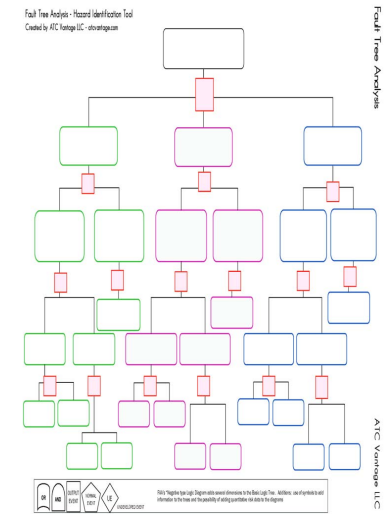
- Compare: accident model (chain-of-events)

FTA Strengths

- Captures **combinations** of failures
- More **efficient** than FMEA
  - Analyzes only failures relevant to top-level event
- Provides **graphical format** to help in understanding the system and the analysis
- Analyst has to think about the system in great detail during tree construction
- Finding minimum **cut sets** provides insight into weak points of complex systems

FTA Limitations (1/2)

- **Independence** between events is often assumed
- **Common-cause failures** not always obvious
- Difficult to capture **non-discrete** events
  - E.g. rate-dependent events, continuous variable changes
- Doesn't easily capture **systemic factors**



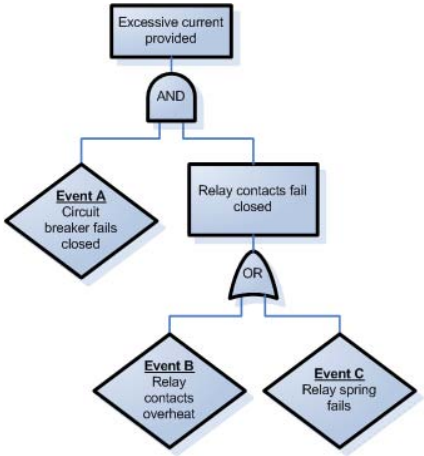
FTA Limitations (2/2)

- Difficult to capture delays and other **temporal factors**
- **Transitions** between states or operational phases not represented
- Can be **labor intensive**
  - In some cases, over 2,500 pages of fault trees
- Can become very complex very quickly, can be difficult to **review**

Quantitative FTA

Quantitative Fault Tree Analysis (FTA) (1/3)

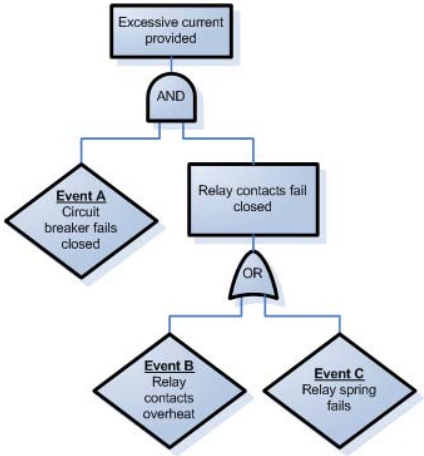
- If we can assign probabilities to lowest boxes ...
  - Can propagate up using probability theory
  - Can get overall total probability of hazard
- AND gate
  - P(A and B) = P(A) * P(B)
- OR gate
  - P(A or B) = P(A) + P(B)
- Which assumptions are being made?



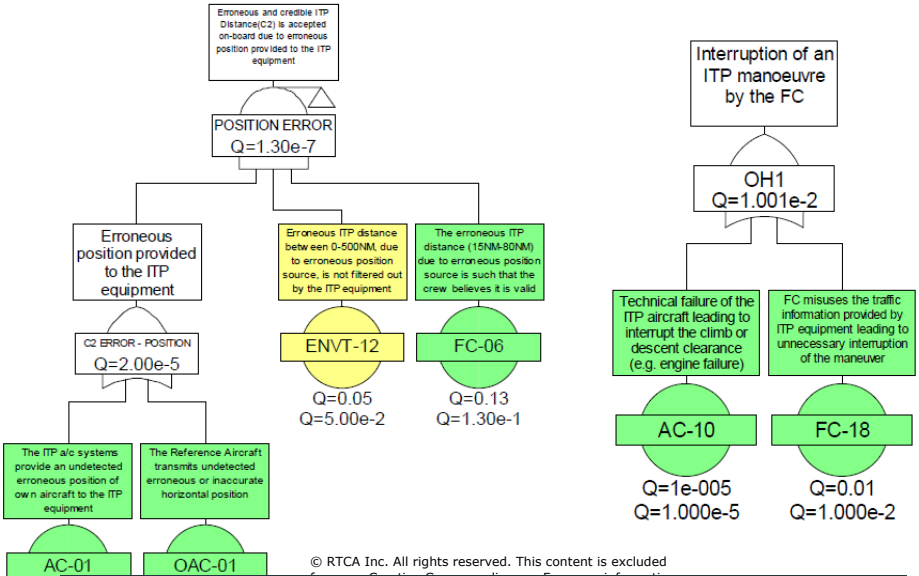
Quantitative Fault Tree Analysis (FTA) (1/3)

- If we can assign probabilities to lowest boxes …
  - Can propagate up using probability theory
  - Can get overall total probability of hazard
- AND gate
  - P(A and B) = P(A) * P(B)
- OR gate
  - P(A or B) = P(A) + P(B)

- Only if events A,B are independent – Good assumption?



Quantitative Fault Tree Analysis (FTA) (2/3)



Quantitative Fault Tree Analysis (FTA) (3/3)

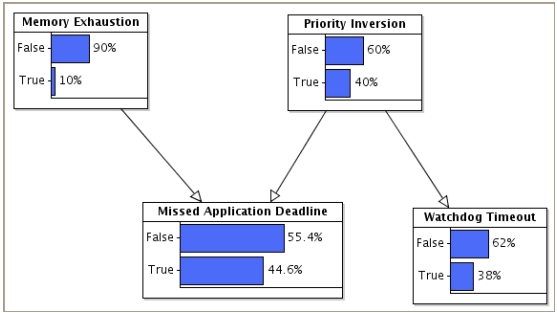
- Where do the probabilities come from?
  - Historical data
  - Simulations
  - Expert judgment

Qualitative Frequency	Quantitative Probability
Very Often	1E-01
Often	1E-02
Rare	1E-03
Very Rare	Less than 1E-04

Table 3.1 Qualitative Frequency and Relation to Quantitative Probability for Basic Causes

Bayesian Fault Tree

Forward Reasoning

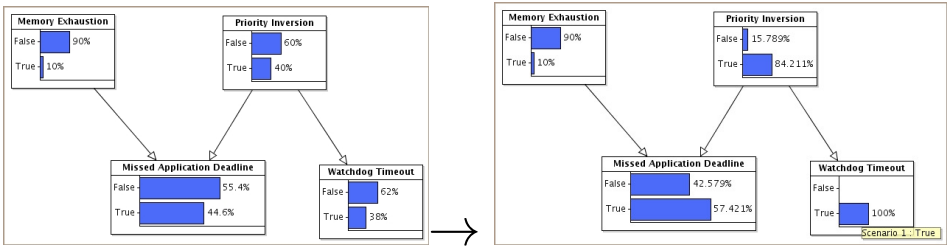


	Memory Exhaustion Priority Inversion	False False	False True	True False	True True
Missed Deadline	False	0.7	0.4	0.4	0.2
Missed Deadline	True	0.3	0.6	0.6	0.8

Bayesian Fault Tree

Bayes' Theorem $P(A \mid B) = \frac{P(B|A) \times P(A)}{P(B)}$

- Forward Reasoning as in any modeling tool.
- Backward Reasoning:



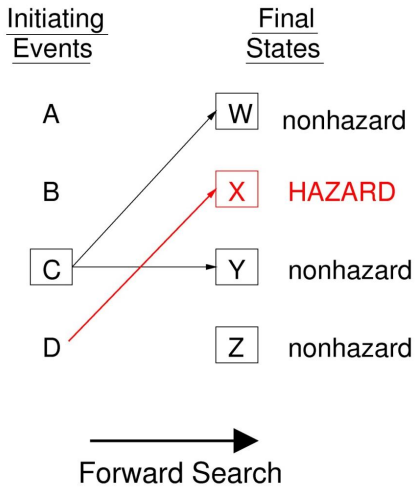
Advantages of Bayesian Networks for Fault Trees

- Mapping from binary tree to Bayesian network is easy.
- Conjunctions (AND/OR) are more expressive.
- A priori and a posteriori reasoning is supported.
- Empirical (measured) failure probability distributions can be handled.
- Common-cause failures don't significantly increase the size of the model.

Event Tree Analysis

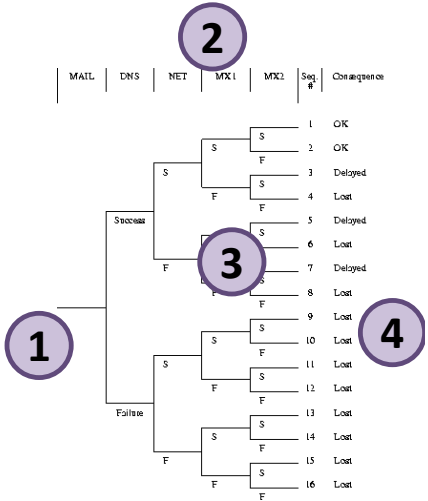
Event Tree Analysis (ETA)

- 1967: Nuclear power stations
- Forward search technique
  - Initiating event: component failure (e.g. pipe rupture)
  - Goal: Identify all possible outcomes

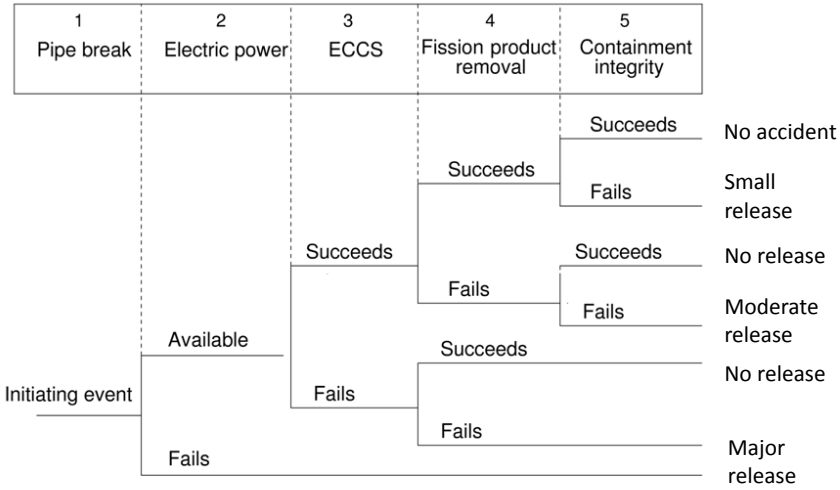


Event Tree Analysis: Process

1. Identify initiating event
2. Identify barriers
3. Create tree
4. Identify outcomes

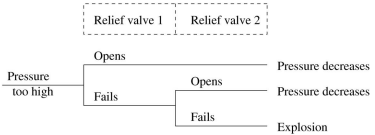


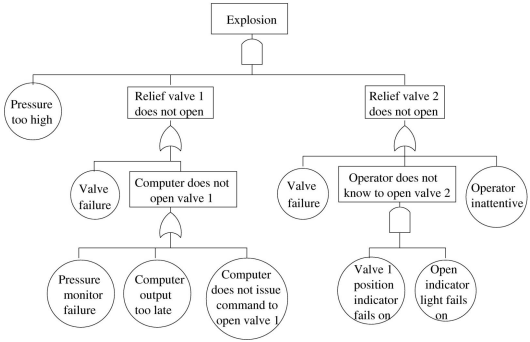
Event Tree Example



Event Trees vs. Fault Trees

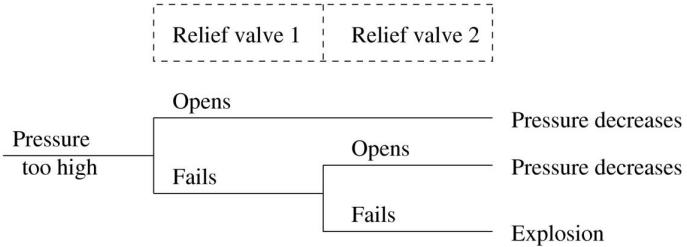
- Event Tree
  - Shows what failed, but not how.
  - Shows order of events
- Fault Tree
  - Complex, but shows how failure occurred
  - Does not show order of events



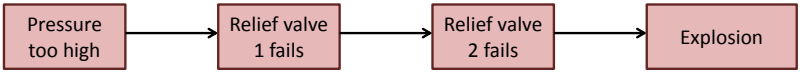


ETA Follows an Accident Model

- Event tree



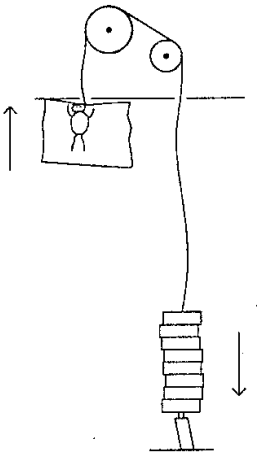
- Compare: accident model (chain-of-events)



Event Tree Analysis: Exercise

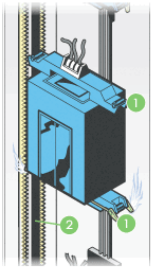
Elevator

1. Identify initiating event
   - Cable breaks
2. List Barriers
3. Create Tree
4. Identify outcomes



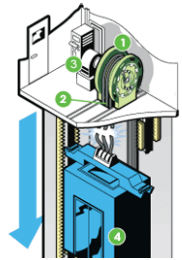
This image is in the public domain.

Event Tree Analysis: Exercise

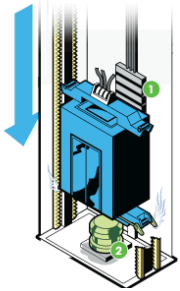


1. If the cables snap, the elevator's **safeties** would kick in. **Safeties** are braking systems on the elevator.
2. Some safeties clamp the **steel rails** running up and down the elevator shaft, while others drive a wedge into the notches in the **rails**.

What are the barriers?



1. Steel cables bolted to the the car loop over a **sheave**.
2. The sheave's grooves grip the **steel cables**.
3. The **electric motor** rotates the sheave, causing the cables to move, too.
4. As the cables move, the **car** is lifted.



1. The cables that lift the car are also connected to a **counterweight**, which hangs down on the other side of the sheave.
2. The built-in **shock absorber** at the bottom of the shaft - typically a piston in an oil-filled cylinder - helps cushion the imact in the event of snapping cables.

Event Tree Analysis: Strengths

- Handles ordering of events better than fault trees
- Most practical when events can be **ordered in time** (chronology of events is stable)
- Most practical when **events are independent** of each other
- Designed for use with **protection systems** (barriers)

Event Tree Analysis: Limitations (1/2)

- Not practical when chronology of events is not stable (e.g. when **order of columns may change**)
- Difficult to analyze **non-protection systems**
- Can become exceedingly **complex** and require simplification
- **Separate trees required** for each initiating event
  - Difficult to represent interactions among events
  - Difficult to consider effects of multiple initiating events

Event Tree Analysis: Limitations (2/2)

- Can be difficult to define functions across top of event tree and their order
- Requires ability to define set of initiating events that will produce all important accident sequences
- Most applicable to systems where:
  - All risk is associated with one hazard
    - (e.g. overheating of fuel)
  - Designs are fairly standard, very little change over time
  - Large reliance on protection and shutdown systems

Quantitative ETA

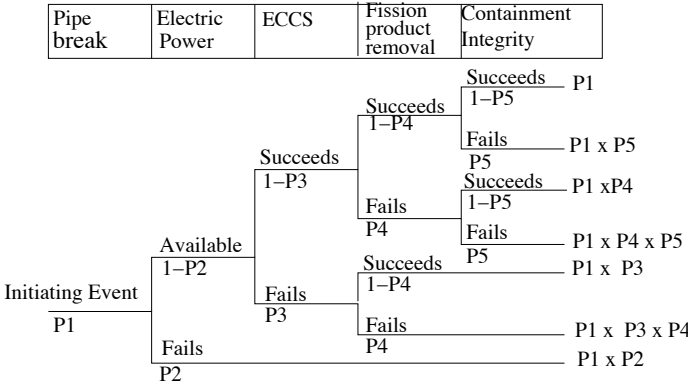
Quantitative Event Tree Analysis (ETA)

- Recap: ETA starts with faults that can cause accidents (e.g. broken pipe)
- Recap: then draw a decision tree in order to identify sequences of faults resulting in accidents
- Recap: for each such sequence one determines its outcome
- Probabilities are assigned to each event to determine the likelihood of that scenario
- Product of the failures on each path is the probability of that event sequence

→ **ETA draws a tree of possible sequences of unintended events (faults) and determines possible accidents as a result of these events**

Event Tree Analysis (ETA)

Example: Loss of cooleant accident in a nuclear power station
(ECCS = Emergency Core Cooling System)



- Since probability of failure is usually very low, probabilities of success are usually almost 1 and can be ignored in the product.

Event Tree Analysis (ETA)

- Quantify p(success) for each barrier
- Limitations
  - P(success) may not be random
  - May not be independent
  - May depend on order of events and context
  - Example: Fukushima Diesel Generators

OH	*Barrier 1a*	*Barrier 1b*	*Barrier 1c*	*Barrier 1d*	*Barrier 2*	*Barrier 3*	*OE Sev.*	Effects	*Pe*
	0.993116 A						5	No safety effect	
OH 2U-7		0.987384 B					4	Loss of separation 5 < x < 10 NM	6.80E-03 X & B
	6.88E-03 X		0.992699 C				3	Significant Reduction in separation 1 < x < 5 NM	8.62E-05 X&C&C
		1.26E-02 Y		0.93577236 D	0.90 E	0.80 F	2	Large reduction in safety margins x < 1 NM	6.21E-07 X&Y&Z& (D OR E OR F)
			7.30E-03 Z						
				5.36E-02 V	0.10 W	0.20 S	1	Near mid-air collision/ Collision	6.80E-10 X&Y&Z& V&W&S

Evaluation of Quantitative ETA

+ ETA handles continuity of events well.
+ ETA good for calculation of probability of events.
  - Most widely used method for quantification of system failures
- Tree usually contains many events that do not result in an accident
  - ETA becomes unnecessary big
  - $\rightarrow$ Cut away subtrees that do not result in an accident

HAZOP Analysis

Hazards and Operability Analysis (HAZOP)

- Developed by Imperial Chemical Industries in early 1960s
- Not only for safety, but efficient operations

- Accident model:
  - Chain of failure events (that involve deviations from design/operating intentions)

HAZOP Process (1/2)

HAZOP is carried out by a team

1. Define objectives and scope of the analysis
2. Select a HAZOP team
   - Requires a leader, who knows HAZOP well
   - Requires a recorder, who documents the process of HAZOP
3. Dissect design into nodes and identify lines into those nodes
4. Analyze deviations for each line and identify hazard control methods
5. Document results in a table
6. Track hazard control implementation

HAZOP Process (2/2)

- Guidewords applied to variables of interest
  - E.g. flow, temperature, pressure, tank levels, etc.
- Team considers potential causes and effects
- **Questions** generated from guidewords
  - Could there be no flow?
  - If so, how?
  - How will operators know there is no flow?
  - Are consequences hazardous or cause inefficiency?

Guidewords	Meaning
NO, NOT, NONE	The intended result is not achieved, but nothing else happens (such as no forward flow when there should be)
MORE	More of any relevant property than there should be (such as higher pressure, higher temperature, higher flow, or higher viscosity)
LESS	Less of a relevant physical property than there should be
AS WELL AS	An activity occurs in addition to what was intended, or more components are present in the system than there should be (such as extra vapors or solids or impurities, including air, water, acids, corrosive products)
PART OF	Only some of the design intentions are achieved (such as only one of two components in a mixture)
REVERSE	The logical opposite of what was intended occurs (such as backflow instead of forward flow)
OTHER THAN	No part of the intended result is achieved, and something completely different happens (such as the flow of the wrong material)

HAZOP: Generate the right questions, not just fill in a tree.

HAZOP Example

Line	Attribute	Guide word	Cause	Consequence	Recommend.
Sensor supply line	Supply voltage	No	Regulator or cable fault	Lack of sensor signal detected and system shuts down	
		More	Regulator fault	Damage to sensor	Consider overvoltage protection
		Less	Regulator fault	Incorrect temperature reading	Include voltage monitoring

HAZOP Strengths

- **Easy** to apply
  - A simple method that can uncover complex accidents
- Applicable to **new designs** and new design features
- Performed by **diverse study team**, facilitator
  - Method defines team composition, roles
  - Encourages cross-fertilization of different disciplines

HAZOP Limitations

- Requires **detailed plant information**
  - Flowsheets, piping and instrumentation diagrams, plant layout, etc.
  - Tends to result in protective devices rather than real design changes
- Developed/intended for **chemical industry**
- **Labor-intensive**
  - Significant time and effort due to search pattern
- Relies very heavily on judgment of engineers
- May leave out hazards caused by **stable factors**
- Unusual to consider deviations for **systemic factors**
  - E.g. organizational, managerial factors, management systems, etc.
- Difficult to apply to **software**
- **Human behavior** reduces to compliance/deviation from procedures
  - Ignores *why it made sense* to do the wrong thing

Summary

Summary

- Well-established methods
- Time-tested, work well for the problems they were designed to solve
- Strengths include
  - Ease of use
  - Graphical representation
  - Ability to analyze many failures and failure combinations
  - Application to well-understood mechanical or physical systems
- Limitations include
  - Inability to consider accidents without failures
  - Difficulty in incorporating systemic factors like managerial pressures, complex human behavior, and design/requirements flaws

→ Other methods may be better suited to deal with the challenges introduced with complex systems

Questions?