Part 5:

Failure modes and models

Failure modes

The *way* a system can fail is called its failure mode. Failure modes are defined through the behavior that is perceived at the system's interface.

Classification of failure modes

- **Byzantine** or **arbitrary failures:**
  there is no restriction on the behavior at the system interface, this mode is often called *fail-uncontrolled*
  ("two-faced" behavior, forging of messages)
- **Authentification detectable byzantine failures:**
  the only restriction on the behavior at the system interface is that messages of other systems cannot be forged
  (this failure mode applies only to distributed systems)

Classification of failure modes (cont.)

- **Performance failures:**
  under this failure mode systems deliver correct results in the value domain, in the time domain results may be **early** or **late**
  (early or late failures)

- **Omission failures:**
  a special class of performance failures where results are either correct or infinitely late
  (for distributed systems subdivision in send and receive omission failures)

- **Crash failures:**
  a special class of omission failures where a system does not deliver any subsequent results if it has exhibited an omission failure once
  (the system is said to have crashed)

Classification of failure modes (cont.)

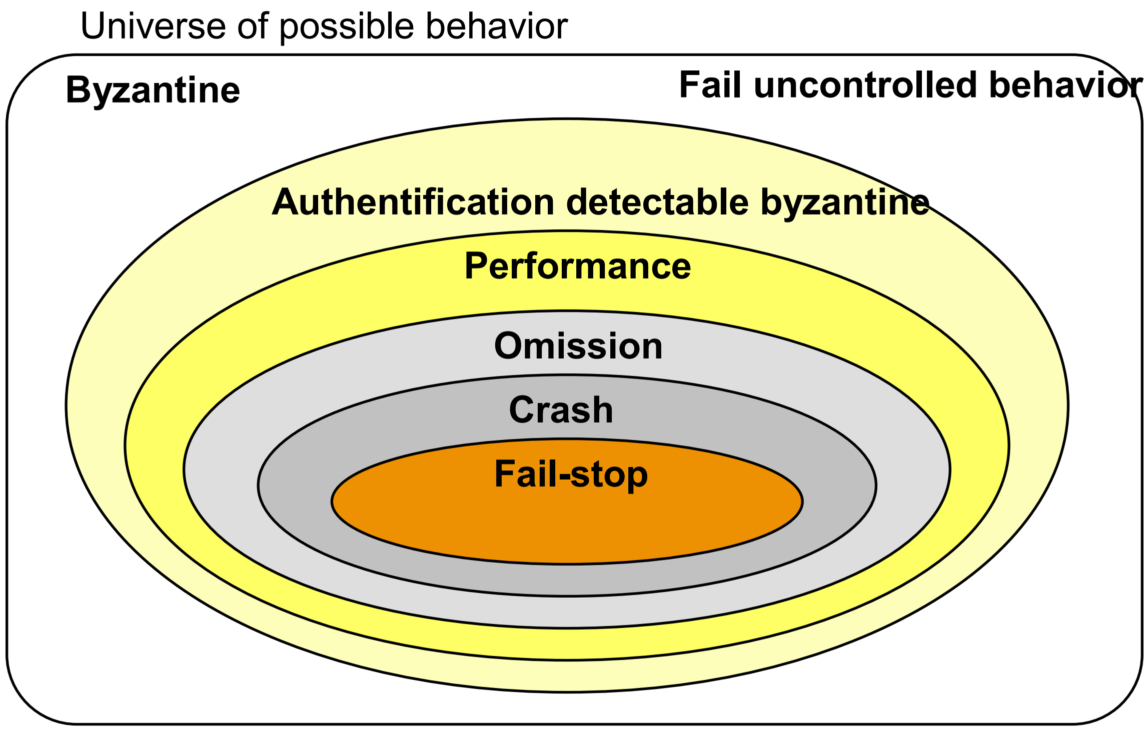
- **fail-stop failures:**
  besides the restriction to crash failures it is required that other (correct) systems can detect whether the system has failed or not and can read the last correct state from a stable storage

Hierarchy of failure modes

Based on the *strength* of the assumptions the failure modes form a hierarchy.

- byzantine failures are based on the weakest assumption
  (a non-assumption)
- fail-stop failures are based on the strongest assumptions
  (only correct results, information about the last correct state in case of a failure)

Hierarchy of failure modes



Classification of failure modes

The failure modes can be characterized according to the viewpoints

- domain of failure occurrence
- perception by the system users
- consequences on the environment.

Domain of failure occurrence

- **Value failures:**
  the value of the service response does not agree with the service specification. (Byzantine and authentification detectable byzantine failures)
- **Timing failures:**
  the timing of the service response does not agree with the service specification. (performance, omission, crash, and fail-stop failures)

Perception by the system users

In a distributed system with several users, the viewpoint of failure perception leads one to distinguish:

- **Consistent failures:**
  All system users have the same perception of the failure.
  (performance, omission, crash, and fail-stop failures)
- **Inconsistent failures:**
  Different system users obtain different perceptions of the failure.
  (byzantine and authentification detectable byzantine failures)

Consequences on the environment

- **Benign failures:**
  The consequences of a service failure are of the same order of magnitude as the benefit provided by a correct service delivery.
- **Catastrophic failures:**
  The consequences of a service failure are vastly more severe than the benefit provided by a correct service delivery. Especially, this includes severe consequence to health and human live.

Failure semantics, fault hypothesis and assumption coverage

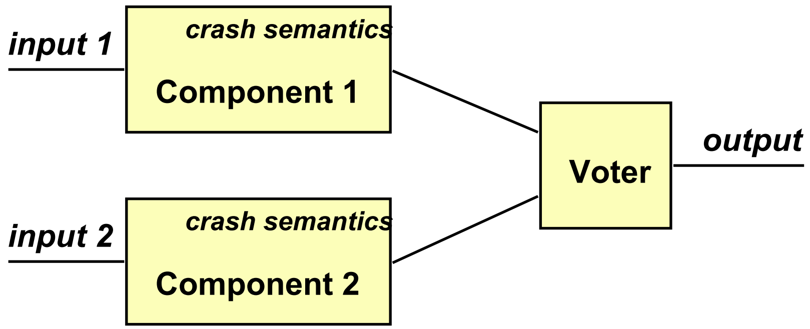
- **Fault hypothesis:**
  The fault hypothesis specifies anticipated faults which a server must be able to handle (also fault assumption).
- **Failure semantics:**
  A server exhibits a given failure semantics if the probability of failure modes which are not covered by the failure semantics is *sufficiently* low.
- **Assumption coverage:**
  Assumption coverage is defined as the probability that the possible failure modes defined by the failure semantics of a server proves to be true in practice conditions on the fact that the server has failed.

Importance of assumption coverage

- The definition of a proper fault hypothesis, failure semantics and achievement of *sufficient* coverage is one of the most important factors.
- If the fault hypothesis (or failure semantics) is violated a system may fail as a whole.

An example

If component 1 or 2 violates its failure semantics the system fails, although it was designed to tolerate 1 component failure.

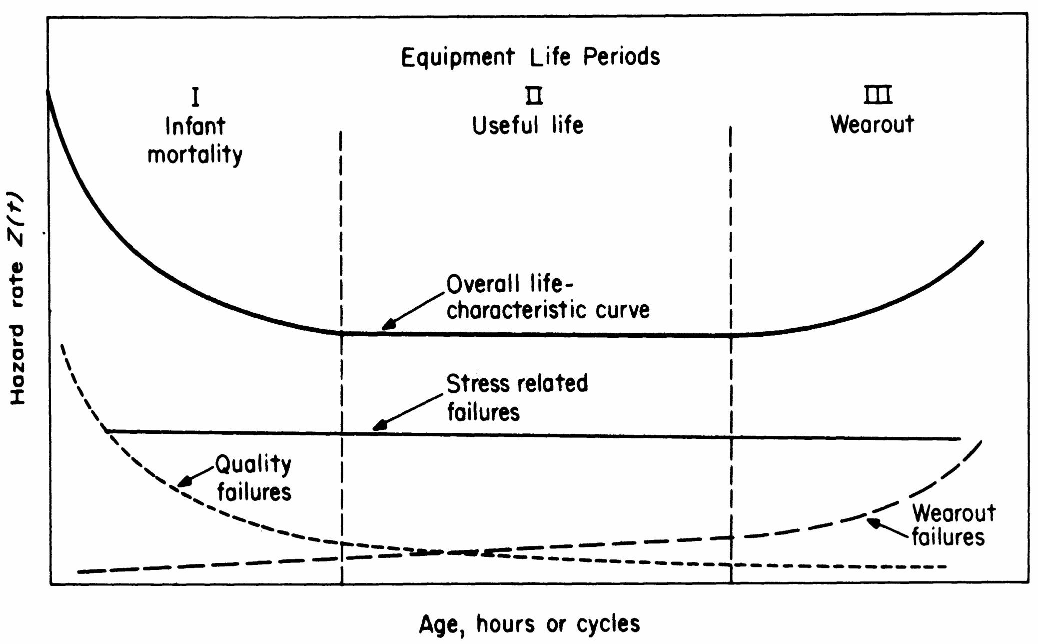


The *Titanic* or: violated assumption coverage

- **The fault hypothesis:**
  The Titanic was built to stay afloat if less or equal to 4 of the underwater departments were flooded.
- **Rationale of fault hypothesis:**
  This assumption was reasonable since previously there had never been an incident in which more than four compartments of a ship were damaged.
- **But:**
  Unfortunately, the iceberg ruptured five spaces, and the following events went down to history.

Life-characteristics curve (Bathtub curve)

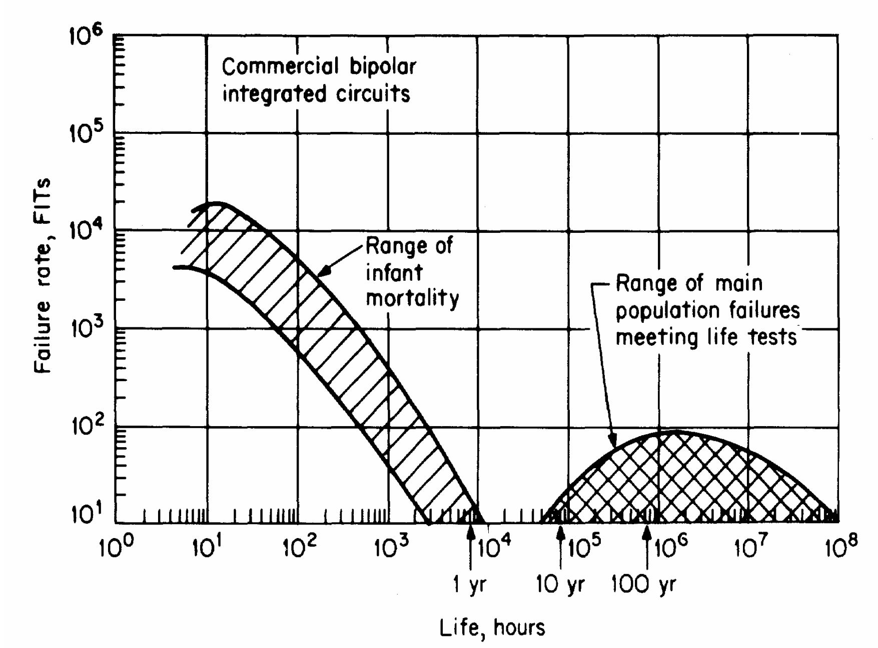
- For semiconductors, out of three terms describing the life-characteristics only *infant mortality* and the *constant-failure-rate* region are of concern



Life-characteristics curve, showing the three components of failure

Semiconductor failure rate

- a typical failure rate distribution for semiconductors shows that wear out is of no concern



Semiconductor failure rate

Stress

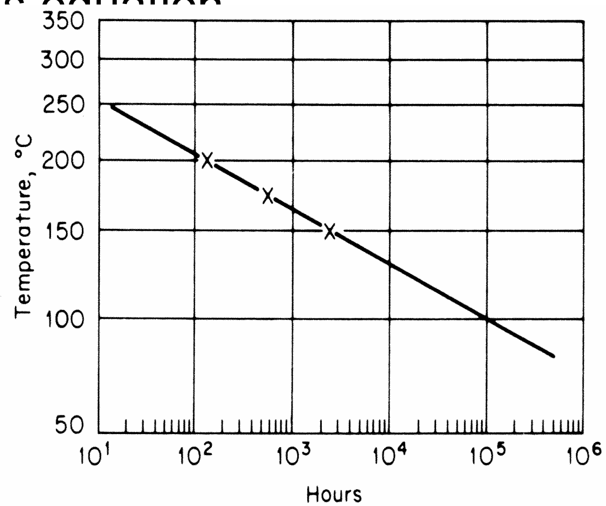
- semiconductor failures are stress dependent
- the most influential stress factor is temperature

Arrhenius equation

- the basic relationship between the activation rate of failures and temperature is described by the Arrhenius equation

$$R = R_0 e^{-\frac{E_A}{kT}}$$

R_0 .. constant
T .. absolute temperature (K)
E_A .. activation energy (eV)
k .. Boltzmann's constant 8.6 10^{-5} eV/K



Arrhenius plot (E_A = 1 eV)

Accelerated stress testing of semiconductors

- to remove freaks and infant-mortality failures (screening)
- to determine the expected failure rate

Accelerated conditions

- accelerated temperature
- cycling of temperature
- temperature and voltage stress
- temperature, voltage and humidity stress

- lowering of temperature
- high temperature and current
- α particles
- high voltage gradients

Software stress

For software there is no sound empirical and mathematical basis to use stress as a method to characterize the behavior of components.

- it is currently unknown how to characterize stress for software
- it is impossible to carry out accelerated stress tests to examine failure rates for software
- for software there is no such relation as the Arrhenius equation which describes the activation rate of failures
- there is no general possibility to “over-engineer” a system to handle conditions
  which are more stressful

Hardware/software interdependence

- **software depends on hardware:**
  - software requires hardware to execute (e.g. Intel's Pentium bug)

- **hardware depends on software:**
  - VLSI design uses software tools
  - PCB layout and routing by software tools
  - EMC analysis by software tools
  - hardware testers are software driven

System Safety

- is a subdiscipline of system engineering that applies scientific, management, and engineering principles to ensure adequate safety, throughout the operational life cycle, within the constraints of operational effectiveness, time and cost.

Safety

- has been defined as “freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property”.
- safety has to be regarded as a relative term

Software Safety

- to ensure that the software will execute within a system context without resulting in unacceptable risk

Safety analysis

- includes complete life cycle of project/product (specification, design, maintenance, modification, ...)
- definition of responsibilities
- communication with other groups
- complete documentation
- analysis of complex processes
- management procedures (specialists, meetings, action reviews, time schedule, ...)

Failure modes and models

Major topics of Safety analysis

- which (hazard analysis)
- how (accident sequencing)
- how likely (quantitative analysis)

Safety analysis methodologies

- Preliminary Hazards Analysis (PHA)
- Hazards and Operability Study (HAZOP)
- Action Error Analysis (AEA)
- Fault Tree Analysis (FTA)
- Event Tree Analysis (ETA)
- Failure Modes and Effect Analysis (FMEA)
  Failure Modes, Effect and Criticality Analysis (FMECA)
- Cause-consequence analysis

Preliminary hazard analysis (PHA)

The first step in any safety program is to identify hazards and to categorize them with respect to criticality and probability

- define system hazards
- define critical states and failure modes
- identify critical elements
- determine consequences of hazardous events
- estimate likelihood of hazardous events
- issues to be analyzed in more detail

Hazards and Operability Study (HAZOP)

Based on a systematic search to identify deviations that may cause hazards during system operation

- **Intention:**
  for each part of the system a specification of the “intention” is made
- **Deviation:**
  a search for deviations from intended behavior which may lead to hazards
- **Guide Words:**
  Guide words on a check list are employed to uncover different types of deviations
  (NO, NOT, MORE, LESS, AS WELL AS, PART OF, REVERSE, OTHER THAN)
- **Team:**
  the analysis is conducted by a team, comprising different specialists

Example for HAZOP

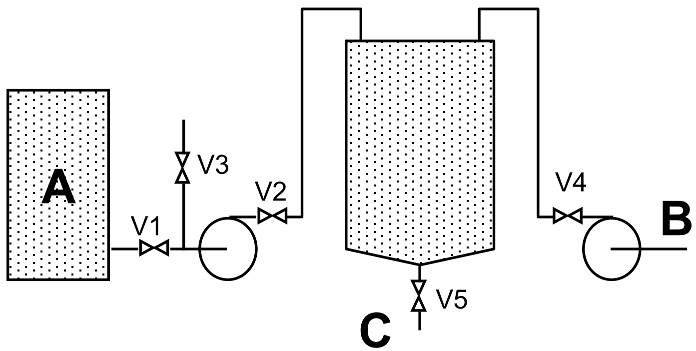
- **Intention:** pump a specified amount of A to reaction tank B. Pumping of A is complete before B is pumped over.

NO or NOT
– the tank containing A is empty
– one of the pipe's two valves V1 or V2 is closed
– the pump is blocked, e.g. with frozen liquid
– the pump does not work (switched off, no power, ...)
– the pipe is broken
CONSEQUENCE is serious, a possible explosion

MORE
– the pump has a too high capacity
– the opening of the control valve is too large
CONSEQUENCE not serious, tank gets overfilled

AS WELL AS
– valve V3 is open, another liquid or gas gets pumped
– contaminants in the tank
–A is pumped to another place (leak in the connecting pipe)
CONSEQUENCE is serious, a possible explosion

. . .



Action Error Analysis (AEA)

Considers the operational, maintenance, control and supervision actions performed by human beings. The potential mistakes in individual actions are studied.

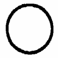
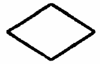
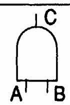
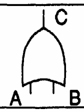
- list steps in operational procedures (e.g. “press button A”)
- identification of possible errors for each step, using a check-list of errors
- assessment of the consequences of the errors
- investigations of causes of important errors
  (action not taken, actions taken in wrong order, erroneous actions, actions applied to wrong object, late or early actions, ...)
- analysis of possible actions designed to gain control over these process
- relevant for software in the area of user interface design

Fault Tree Analysis (FTA)

A graphical representation of logical combinations of causes that may lead to a hazard (top-event). Can be used as a quantitative method.

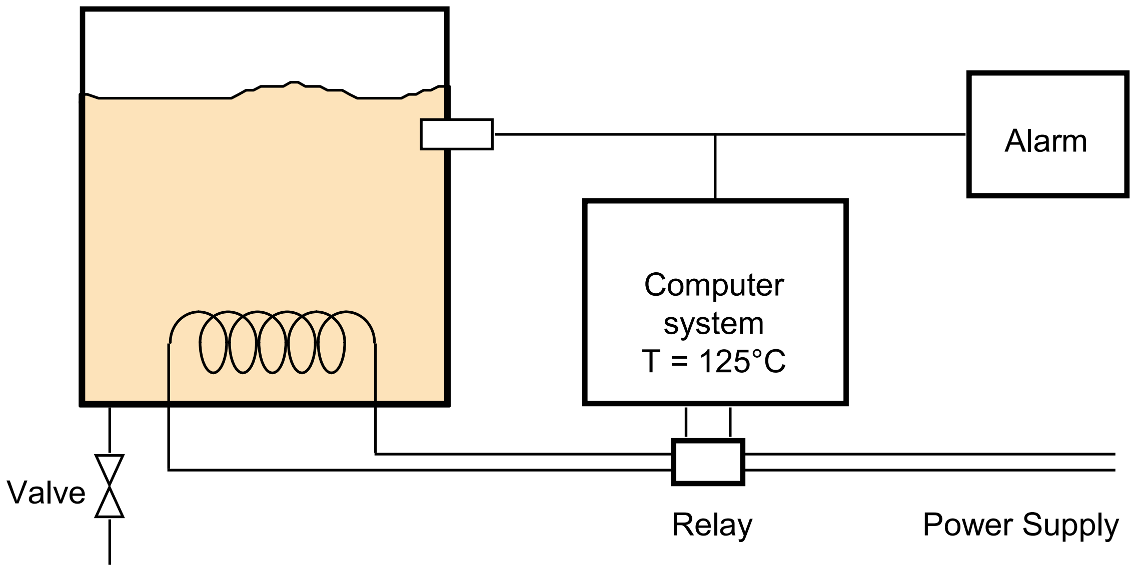
- identification of hazards (top-events)
- analysis to find credible combinations which can lead to the top-event
- graphical tree model of parallel and sequential faults
- uses a standardized set of symbols for boolean logic
- expresses top-event as a consequence of AND/OR combination of basic events
- minimal cut set is used for quantitative analysis

Symbols used in fault tree analysis

Symbol	Designation	Function
	BASIC EVENT	Basic event or failure
	UNDEVELOPED EVENT	Causes are not developed
	EVENT	Event resulting from more basic events
	CONDITIONAL EVENT	Event that can occur normally
	AND gate	Output event occurs only if all input events occur simultaneously
	OR gate	Output event occurs if any one of the input events occurs
	TRANSFER SYMBOL	Represents an event which comes from another lower-order fault tree or which is to be transferred to a higher-order tree

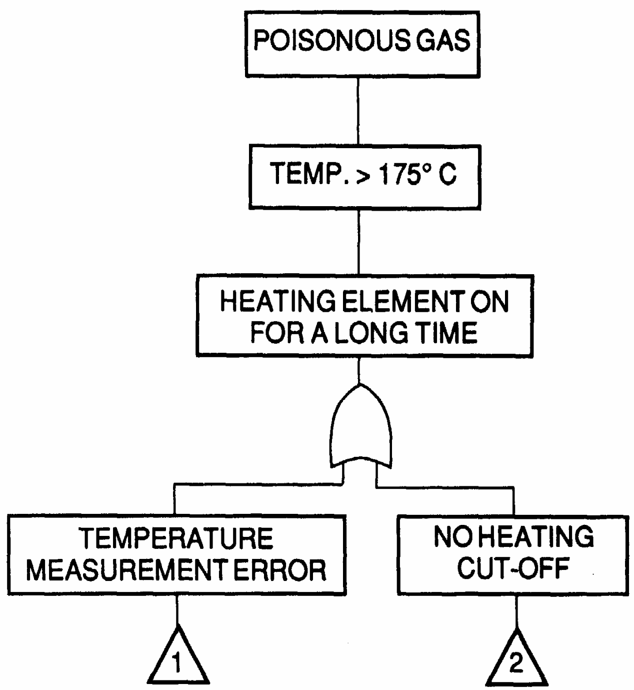
An Example for fault tree analysis

In a container two chemicals react with each other over a period of 10 hours at a temperature of 125 *°C.* If the temperature exceeds 175 *°C* toxic gas is emitted. The temperature is controlled by a computer system.



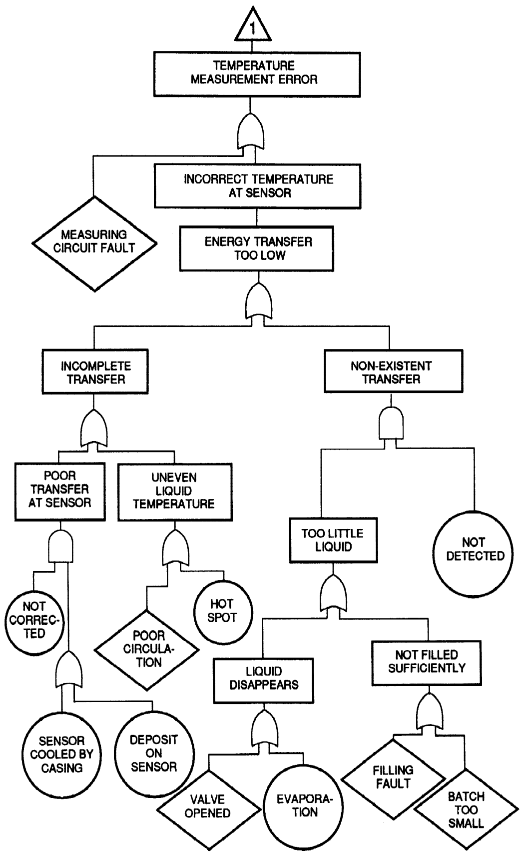
Identification of the top-event

Emission of poisonous gas is the top event

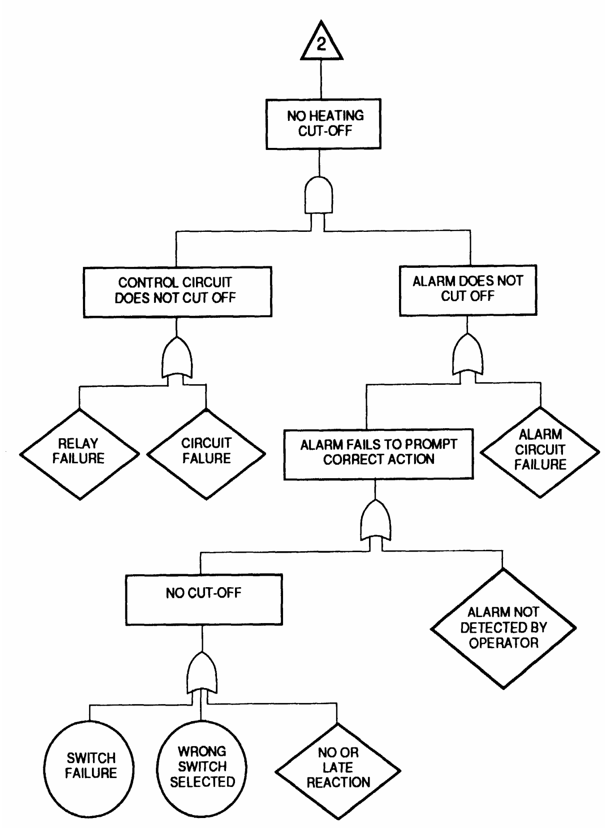


The upper part of the fault tree

Subtree for temperature measurement failure



Subtree for heating cut off failure



Event Tree Analysis (ETA)

Models the potential consequences of faults which are considered as events.

Can be used as a quantitative method.

- identification of basic events
- start with basic events and describe possible consequences of this event
- binary decision for consequences of events
- opposite of FTA which starts with top events

Failure Modes and Effect Analysis (FMEA)

A common method where the designer in a systematical way has to answer the questions “How can the component fail?” and “What happens then?”.

- the system is dived up into different components in the form of a block diagram
- failure modes are identified for all components
- causes, consequences and the significance of failures are assessed for each
failure mode
- an investigation is made into how the failure can be detected
- if necessary, recommendations for suitable control measures are made
- analysis is supported by tabular sheets (e.g. IEC standard 1985)
- failure mode, effects and criticality analysis (FMECA) puts special emphasis on the criticality aspect

Failure modes and models

An example FMEA hazard assessment

Severity of consequence		*Probability of occurrence*			*Probability of detection*	
10	**Very severe**	10	**High**	$500\ 10^{-6}$	10	**Unprobable**
	System operation has to be		It is almost certain that the			It is impossible or at very in-
9	abandoned or even a safety	9	failure will occure with high			probable that the failure can
	critical state may be reached		probability			be detected
8	**Severe**	8	**Moderate**	$50\ 10^{-6}$	9	**Very low**
	Failure causes disturbance of		The component is similiar to com-			It is possible to detect the fault
	end user (no safety critical	7	ponent designs which already have			before the system fails
7	failures or violations of regu-		caused problems in the past			
	lations)				8	**Small**
6	**Moderate**	6	**Small**	$5\ 10^{-6}$	7	
	Failure causes inconvenience of		The component is similiar to com-			
5	the end user, restricted system	5	ponent designs which have caused		6	
	operation will be perceived by		problems in the past, but the extend			
4	the customer	4	of problems was relatively low		5	**Moderate**
3	**Minor**	3	**Very small**	$100\ 10^{-9}$	4	
	Failure causes only minor incon-		The component is similiar to com-		3	
	venience of the end user, only		ponent designs which had very			
2	minor restrictions of the system		low failure rates in the past			
	operation are perceiveable				2	**High**
1	**Improbable**	1	**Improbable**	$1\ 10^{-9}$	1	**Very High**
	It very improbable that the failure		It is very improbable that a failure			It is certain that the faults gets de-
	will be perceived by the end user		ocurrs			tected before the system fails

Failure modes and models

An example FMEA

Function	Failure Mode	Cause	Effect	Controls	Severity	Probability	Dedection	Product
speed sensor	open	connector or harness	no operation possible	supplier quality control and end of line testing	9	4	3	108
		computer	no operation possible	computer supplier quality control and end of line testing	9	3	3	81
		sensor	no operation possible	sensor supplier quality control, module and end of line testing	9	4	3	108
	short to supply	connector or harness	no operation possible	supplier quality control and end of line testing	9	2	3	54
		computer	no operation possible	computer supplier quality control and end of line testing	9	2	3	54
		sensor	no operation possible	sensor supplier quality control, module and end of line testing	9	2	3	54
	short to ground	connector or harness	no operation possible	supplier quality control and end of line testing	9	1	3	27
		computer	no operation possible	computer supplier quality control and end of line testing	9	1	3	27
		sensor	no operation possible	sensor supplier quality control, module and end of line testing	9	1	3	27

Cause-consequence analysis

Combination of fault tree analysis and event tree analysis

- starts at a critical event
- works forward by using event tree analysis (consequences)
- works backward by using fault tree analysis (causes)
- very flexible
- well documented method

Comparison of safety analysis methodologies

Method	Advantages	Restrictions and deficiencies
Preliminary hazards analysis	A required first step.	None.
Hazards and operability study	Suitable for large chemical plants. Results in a list of actions, design changes and cases identified for more detailed study. Enhances the information exchange between system designers, process designers and operating personnel.	Technique is not well standardized and described in the literature. Most often applied to continuos processes.
Action error analysis	Gives the computer system designer proposals for proper interface design. Helps the personnel or users to monitor the process during operation and helps to prevent operator mistakes.	AEA is an analysis of the technical system, and does not analyze the behavior of operators. The thoughts and intentions of human beings, i.e. the reasons for mistakes, are not considered.

Comparison of safety analysis methodologies (cont.)

Method	Advantages	Restrictions and deficiencies
Fault tree analysis	Well accepted technique. Very good for finding failure relationships. A fault oriented technique which looks for the ways a system can fail. Makes it possible to verify requirements, which are expressed as quantitative risk values.	Large fault trees are difficult to understand, bear no resemblance to system flow charts, and are mathematically not unique. It assumes that all failures are of binary nature, i.e. a component completes successfully or fails completely.
Event tree analysis	Can identify effect sequences and alternative consequences of failures. Allows analysis of systems with stable sequences of events and independent events.	Fails in case of parallel sequences. Not suitable for detailed analysis due to combinatorial explosion. Pays no attention to extraneous, incomplete, early or late actions.

Comparison of safety analysis methodologies (cont.)

Method	Advantages	Restrictions and deficiencies
Failure modes and effects analysis	Easily understood, well accepted, standardized technique. Non-controversial, non-mathematical. Studies potential failures and their effects on the function of the system.	Examines non-dangerous failures and is therefore time consuming. Often combinations of failures and human factors not considered. It is difficult to consider multiple and simultaneous failures.
Cause-consequence analysis	Extremely flexible and all-encompassing methodology. Well documented. Sequential paths for critical events are clearly shown.	Cause-consequence diagrams become too large very quickly (as FTA, ETA). They have many of the disadvantages of fault tree analysis.

Problems with software safety analysis

- relatively new field
- lack of systematic engineering discipline
- no agreed or proven methodologies
- time and cost
- complexity
  (understanding of the problem domain, separation of knowledge)
- discrete nature of software
  (difficulties with large discrete state spaces)
- real-time aspects
  (concurrency and synchronization)
- (partially) invalid assumption of independent failures

Summary

- hierarchical classification of failure modes (byzantine, authentification byzantine, performance, omission, crash, fail-stop)
- domain of failure occurrence (value domain, time domain)
- perception by the system users (consistent, inconsistent)
- consequences on the environment (benign, catastrophic)
- **fault hypothesis**, **failure semantics** and **assumption coverage** are one of the most important parameters when designing fault-tolerant systems (proper assessment of failure modes and likelihoods together with good design decisions are necessary)

Failure modes and models

Summary (cont.)

- semiconductor failure rate and bathtub curve
- stress dependency of semiconductor failures
- determination of failure rates by accelerated stress testing (Arrhenius equation)
- for software there is no sound empirical and mathematical basis for the determination of failure rates by stress testing
- hardware/software interdependence

Summary (cont.)

- safety analysis methodologies
  - Preliminary Hazards Analysis
  - Hazards and Operability Study
  - Action Error Analysis
  - Fault Tree Analysis
  - Event Tree Analysis
  - Failure Modes and Effect Analysis
  - Cause-consequence analysis