Lecture outline – Safety engineering

Learning outcomes:

1. Define the term "safety" as it related to software and systems, and demonstrate the difference between safety and other qualities, such as correctness and reliability.

2. Understand the high-level process used for safety engineering.

3. Explain how accidents relate to and influence safety engineering.

4. Use an event chain to describe an accident.

5. Explain the concepts of risk, integrity, hazard, accident, and causation.

6. Apply HAZOPS to identify preliminary hazards.

7. Apply FMEA to analyse the safety concerns of a design.

8. Apply fault-tree analysis to a design.

Lecture plan

Safety and accidents

1. Define *safety* and how it differs to correctness, reliability, etc. A product is safe if it does not cause unacceptable harm or damage; e.g. to a person or to the environment.

2. NOTE: Safety must be considered outside of the boundaries of the software — software alone cannot harm anything.

3. Examples of failures:

   Therac-25 – Between June 1985 and January 1987, six people died after receiving more than 100 times the intended radiation dose from Therac-25 radiation therapy machine. Investigation concluded *poor software design.*

   A320 Airbus Accidents – The Lufthansa A320 accident at Warsaw airport, and the Mulhouse-Habsheim Airport accident (discussed in more detail below).

   London Ambulance – October 1992, the failure CAD system resulted in over 30 deaths and problems in dealing with life-threatening incidents. Up to 11 hours of waiting. Investigation concluded poor design and implementation.

   July 2006 upgrade: repeated crashes, so back to pen and paper methods.

   June 2011 new system: developed tech problems, so back to pen and paper methods, and revert to previous system on the same day. Not re-instated until May 2012.

4. A320 Airbus: a *fly by wire* aircraft, described a *"network with an aircraft wrapped around it"*. 150 ECUs.

5. Mulhouse-Habsheim Airport accident (France):

   (a) Demonstration of the aircraft's capability at low altitude.

   (b) First demo at 10m with landing gear down, second demo at 30m without landing gear.

   (c) Crashed during second demo, killing 3 of the 130 passengers and injuring 34 others.

   (d) Investigation: low fly-over, low speed (flight idle), late application of thrust, no malfunction → *pilot error*.

   (e) Observation: pilot error far more likely to be concluded when pilot dies (not there to defend themselves).

   (f) If pilot e.g. does not notice a signal, is this entirely the pilot's fault, or is the greater system to blame?

   (g) Back to Habsheim — pilot's version:

      i. company procedures not followed (by others): fly-over lower than specified, flight dossier about airport arrived too late for crew to study; trees 30-40ft high not known about until 15 seconds away.

      ii. Altimeter faulty: displays 67ft before take off.

      iii. Runway shorter than normal runway – had never landed at this airport before..

   (h) Technical problems: two guidance systems, not communicating as they should. Caused problems, so entered data into only one.

   (i) Pilot claims engines took almost 9 seconds to spool rather than 5 seconds they should.

   (j) Conclusion: accident cause is system-wide: bad reporting, fault systems, procedures not followed, unexpected environment, and *maybe* some pilot error.

6. Accidents in safety: accidents and incidents are the most useful way to identify potential hazard. By looking at older systems in the same domain, we can get an idea of the factors that contributed to accidents and incidents.

7. Causality: *When can it be said that event A causes event B*?:

   (a) Reason with *counterfactuals*: what could have happened under different conditions.

   (b) Example (and hint): "They would have done better on their assignments, if they had first done the workshops'

   (c) For accidents, "*A causes B*" if we assume that *A* did not occur then *B* would not have occurred.

   (d) NOTE: this makes event *A* just *one* cause, not (necessarily) the sole cause.

   (e) Be careful not to confuse counterfactuals with correlation: *If the were the case that the barometer falls then a storm would occur.*

Safety engineering

1. Three relevant phases in a SE lifecycle:

   (a) *preliminary hazard analysis*: identify what can go wrong (hazards).

   (b) *design*: design the system to mitigate the hazards.

   (c) *implement and test*: implement the design correctly.

2. PHA: input is system data and previous accident info. Method is to brainstorm. Output is a *hazard log*: lists the hazards, their causes and their severity, as well as other data such as target frequencies or hazard types.

3. Show risk & frequency values, and their classes (below).

HAZOPS

1. *Exploratory* hazard analysis. Well established and used.

2. Basic outline: take intended behaviour of a single design item, vary that behaviour, and brainstorm what could happen. Group brainstorming with domain experts who know a lot about accidents.

3. Example: brake by wire system in a vehicle. Used by Mercedes Benz and Toyota in most new vehicles. Intended behaviour: push brake and car slows down. Chain is brake pedal → ECU → braking actuator. HAZOP:

   Get class to do NONE, EARLY, and LATE signals.

4. Record: deviation (guideword + design item), causes, consequences, safeguards in place, and list of recommendations.

   Show guideword table in appendix of this sheet.

5. Limitations: needs some described behaviour (so may come late in process), time and resource intensive, documentation heading, focuses on single failures.

Fault tree analysis

1. A *deductive* technique: starts with potential hazards and works backwards, trying to determine what could cause them. Done on design, not requirements.

2. Symbols (see below).

3. Identify *immediate*, *necessary* (linked with AND), and *sufficient* (linked with OR) events.

4. Example: chemical mixing plant.

FMEA – Failure modes and effects analysis

1. An *inductive* technique: starts with a failure mode and works forward to the consequence, using the system design. It is about identifying the *failure modes* of the system.

2. Typical failure modes are:

   (a) Premature operation.

   (b) Failure to operate at the required time.

   (c) Failure to stop operation at the required time.

(d) Failure during operation — and this is specific to the equipment.

(e) Degraded operational capacity.

(f) Excessive operational capacity.

3. Method: need a solid definition of the system.

4. Method: step 1 — what are the failure modes?

5. Method: step 2 — what are the failure effects?

6. Result: worksheet, documenting failure modes and effects, along with other items, such as causes, probabilities, severities, and recommendations.

7. Limitations: individual failures only (not chains), time and resource intensive, sometimes FMEA is done only to satisfy the altruistic urge to "do safety" (*How will the results be used?*).

IEC 61508 standard

Risks

catastrophic critical
marginal negligible

Frequencies

frequent probable occasional
remote improbable incredible

Risk classes

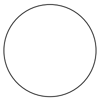
Frequency	Consequence			
	Catastrophic	Critical	Marginal	Negligible
Frequent	I	I	I	II
Probable	I	I	II	III
Occasional	I	II	III	III
Remote	II	III	III	IV
Incredible	IV	IV	IV	IV

Class I Intolerable risk.
Class II Undesirable risk and tolerable only if risk reduction is impractical.
Class III Tolerable risk if the cost of risk reduction would exceed the improvement gained.
Class IV Negligible risk.

HAZOPS guidewords

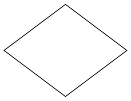
Guide Word	Deviation
NO or NONE	This is the complete negation of the design intention. No part of the intention is achieved and nothing else happens.
MORE	This is a quantitative increase.
LESS	This is a quantitative decrease.
AS WELL AS	All the design intention is achieved together with additions.
PART OF	Only some of the design intention is achieved.
REVERSE	The logical opposite of the intention is achieved.
OTHER THAN	Complete substitution, where no part of the original intention is achieved but something quite different happens.
EARLY	Something happens earlier than expected relative to clock time.
LATE	Something happens later than expected relative to clock time.
BEFORE	Something happens before it is expected, relating to order or sequence.
AFTER	Something happens after it is expected, relating to order or sequence.

Fault tree symbols



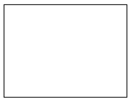
Basic Event

An initiating fault requiring no further action



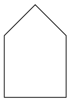
Undeveloped Event

An event which is not developed further, either because it is considered unnecessary or because insuficient information is available.



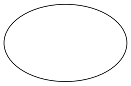
Intermediate Event

An event arising from the combination of other, more basic events.



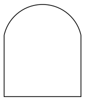
Normal Event

An event which is expected to occur as part of the normal operation of the system.



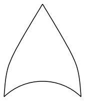
Conditioning Event

Specific restrictions that apply to some types of logic gates, for example the PRIORITY AND or INHIBIT gates.



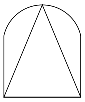
AND gate

All of the inputs must occur for the output to occur



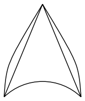
OR gate

One or more of the input events must occur for the output to occur



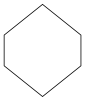
PRIORITY AND gate

The output occurs if the input events occur in a specific sequence which is described in a conditioning event.



EXCLUSIVE OR gate

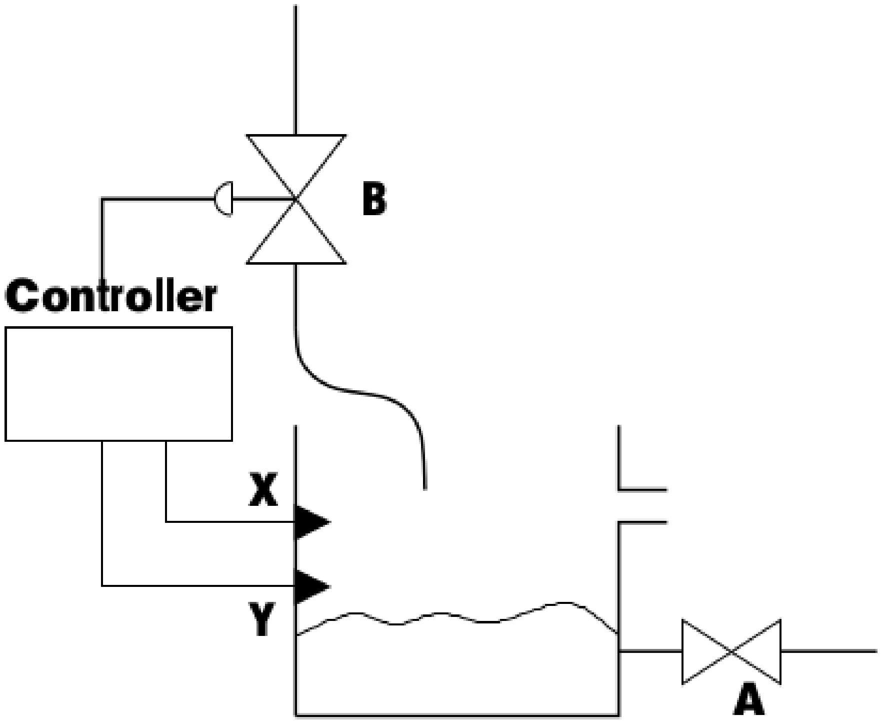
The output occurs if exactly one of the input events occurs.



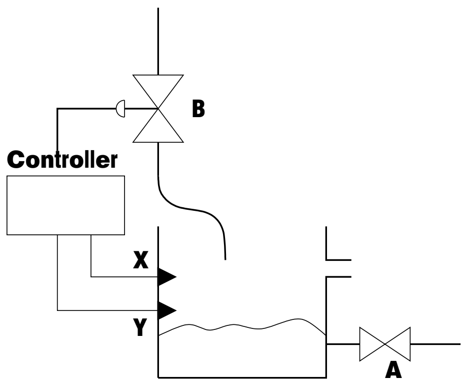
INHIBIT gate

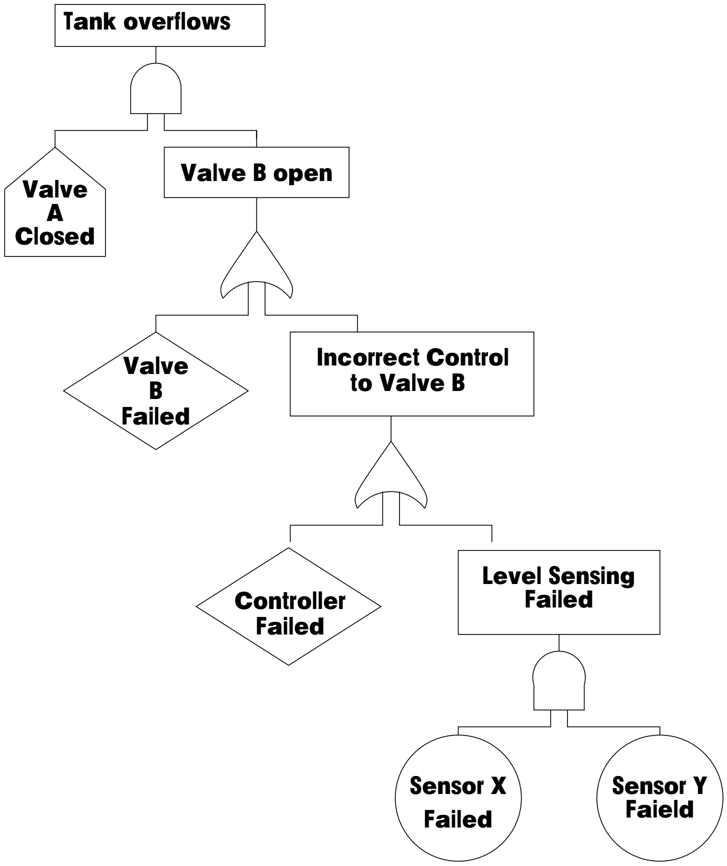
The output occurs if the single input event occurs in the presence of the enabling condition which is described in the conditioning event.

Fault tree example



Fault tree example


B
Controller
X
Y
A


Tank overflows
Valve A Closed
Valve B open
Valve B Failed
Incorrect Control to Valve B
Controller Failed
Level Sensing Failed
Sensor X Failed
Sensor Y Faield