

0101seda010100

software engineering dependability

Safety and Reliability of Embedded Systems

(Sicherheit und Zuverlässigkeit eingebetteter Systeme)

Fault Tree Analysis

Conducting a Fault Tree Analysis

- FTA in the Process Context
- The FTA Procedure

- FTA is *one* technique for probabilistic risk assessment
- It must be embedded in a safety respecting process, assuring
 - construction of correct, reliable and safe hardware and software
 - analysis and validation of safety and reliability of the whole system in its operation environment throughout all process phases
- It should be accompanied / preceded by
 - Preliminary Hazard Analysis
 - FMEA
 - Event Tree Analysis

1. Identify the objective
2. Get familiar with operation and success criteria of the system
3. Define the top-event
4. Define the scope
5. Define resolution
6. Define ground rules
7. Construct the FT
8. Evaluate the FT
9. Interpret and present the results

adapted from:

- FT Handbook with Aerospace Applications
- IEC 61025
- DIN 25424

1.Objective

- All stakeholders should agree on what is to be examined
- The objective should be stated in written
- The objective should refer to a failure of the system in application domain vocabulary
- The objective determines the top-event, the scope, the resolution

 **An FTA is a big effort: You should know what it's for!**

2. Operation and success criteria of the system

- System functions
- System structure / components
- Environmental conditions
- Auxiliary supplies

Use block diagrams, software models, requirement specifications!

A hierarchical schema of the system is helpful

System structure and correlated failures can be found by an FMEA

Domain experts should participate

 **Be sure to understand what you are examining!**

3. Define top-event

- Tight cooperation with customer / system integrator
- If in doubt, try several possibilities and select later
- Possibly more than one top-event

 **A wrong or unclear top-event makes the analysis useless!**


4. Define the scope

- Can be system / component boundary (if well defined)
- Can be broader than that (e.g. including power supply, operator...)
- Write down assumptions about the parts that are not supposed to fail

 **Before starting make clear what to examine and what not**


5. Define resolution

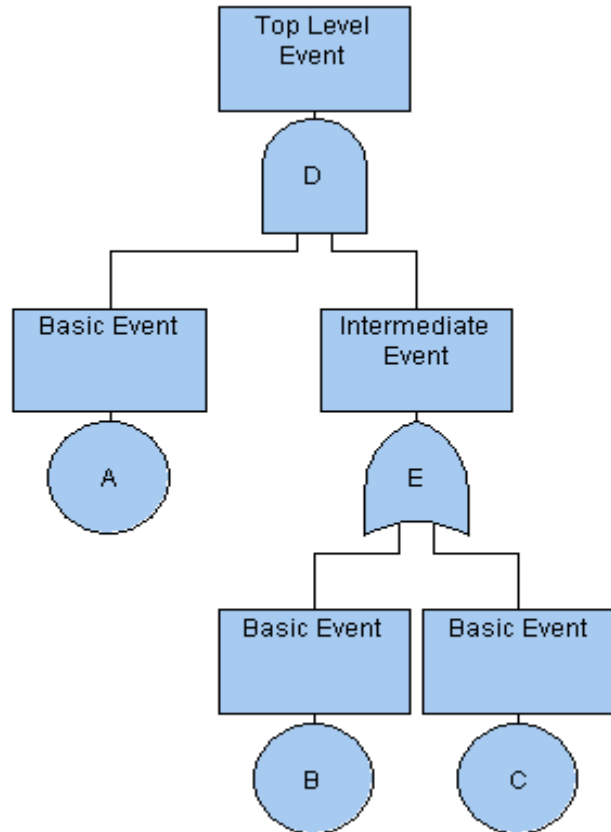
- In complex systems it is impossible to model each detail
- Before starting, define where to stop
- For system level FTA, it may be useful to stop at the components

 **In practice, often too many events are considered that actually play no role!**

6. Define ground rules

- Naming of events
- Modeling of recurring structures
- Write down rules and train all participants

 **If cooperation between different companies or departments or later reuse of FTs is an issue, it is worthwhile to set up rules**



7. Construct the fault tree

- Go backward in small steps
- Always ask for *all immediate* predecessors of an event
- Predecessors are necessary and sufficient causes
- Name intermediate events
- Take care of repeated events, distinguish the equal from the same

 **The goal is to depict an uninterrupted causal chain, not to (try to) find the most basic causes quickly**

8. Evaluate the fault tree

- Set parameters correctly (e.g. resolution)
- Apply qualitative analysis to find minimal cut sets
- Apply quantitative analysis to get top-event probability and importances of minimal cut sets

 **Evaluation is a mechanical job and should be left to the computer**

9. Interpret and present the results

- Is the probability in the expected / tolerable range?
- What are the main influences to the top-event?
- Where should corrective actions be applied?
- What can be learned about the system structure?

 **A probability figure alone is not useful**