The state of the art in fault tree modeling and analysis

Enno Ruijters

November 5, 2014
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance
About me

- Enno Ruijters
- PhD Student at University of Twente (Formal Methods and Tools)
- ArRangeer project
  - ProRail / STW
  - Improving railroad maintenance using Dynamic Fault Trees and Stochastic Model Checking
Introduction to fault trees

- Developed in 1961 by Nuclear Regulatory Agency
- Question: How reliable is your system?
- Now used by:
Introduction to fault trees

- Developed in 1961 by Nuclear Regulatory Agency
- Question: How reliable is your system?
- Now used by:

  - NASA
  - ESA
  - ProRail
  - Honeywell
  - BoeIng
Why fault trees?

- Some things really should not fail
- Risk assessment is sometimes mandatory
  - Probability of catastrophic failures?
  - Biggest risk factors?
  - Possible mitigations?
Why fault trees?

- Some things really should not fail

  **Reliability** Probability of failing within given time
Why fault trees?

- Some things really should not fail
  - **Reliability** Probability of failing within given time
  - **Availability** Proportion of time in functioning state
What do we want to know?

Qualitative:
- Insight into biggest risks
- Relatively fast to perform
- Easy to understand
- Limited information

Quantitative:
- Quantify total risk
- Quantify effect of mitigation
- Time consuming
- Hard to estimate numbers
What to we want to know?

Quantitative:

- **Reliability** \( \equiv \) Probability of failure within time \( t \)
  
  *Example*: Probability of containment failure within 25 year nuclear plant lifetime
What to we want to know?

Quantitative:

- **Reliability** $\equiv$ Probability of failure within time $t$
  
  *Example:* Probability of containment failure within 25 year nuclear plant lifetime

- **Availability** $\equiv$ Proportion of time (in $[0, \infty)$ or $[0, t]$) spent not failed
  
  *Example:* Amazon EC2 cloud offers SLA of 99.95% uptime
What to we want to know?

Quantitative:

- **Reliability** $\equiv$ Probability of failure within time $t$
  
  *Example*: Probability of containment failure within 25 year nuclear plant lifetime

- **Availability** $\equiv$ Proportion of time (in $[0, \infty)$ or $[0, t]$) spent not failed
  
  *Example*: Amazon EC2 cloud offers SLA of 99.95% uptime

- **MTBF** $\equiv$ Expected time between two successive failures (in finite or infinite horizon)
  
  *Example*: How frequently will my car break down?

- Others (MTTF, ENF, etc.)
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance
Fault tree example

- Redundant CPUs
- 1 shared spare memory unit
Example of fault tree failure propagation

- No failures
Example of fault tree failure propagation

- Failure of M1
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
- Failure of M2
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
- Failure of M2
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
- Failure of M2
## Fault tree types

<table>
<thead>
<tr>
<th>Model</th>
<th>Reliability</th>
<th>Availability</th>
<th>MTTFF</th>
<th>MTTF</th>
<th>MTBF</th>
<th>MTTR</th>
<th>ENF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete-time</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Continuous-time</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Repairable cont.-time</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Table:** Applicability of stochastic measures to different FT types
Quantitative analysis of static fault trees

<table>
<thead>
<tr>
<th>Method</th>
<th>Reliability</th>
<th>Availability</th>
<th>MTBF</th>
<th>Exact</th>
<th>Speed</th>
<th>Computable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom-up method</td>
<td>+</td>
<td>+</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Rare-event approximation</td>
<td>+</td>
<td>+</td>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Bayesian networks</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Monte Carlo Simulation</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Algebraic analysis</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>±</td>
</tr>
<tr>
<td>Algebraic approximation</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Table:** Quantitative analysis for static fault trees
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance
Fuzzy numbers

- Uncertainty and variation in BE probabilities
- Expert judgement not exact
- Possible solution: BE probabilities in fuzzy sets
- Several frameworks for computations on fuzzy numbers
- Can compute same measures as for non-fuzzy FTs.
Other uncertain FTs

- ‘Intuitionistic fuzzy set theory’: Membership function uncertain
- Probability distribution for BE failure rates
- Multi-state BE with uncertain states
- Normal distribution approximation
Normal FTs assume independent BEs
Not always realistic (‘valve stuck open’ and ‘valve stuck closed’ are not independent)
Component failures and degradation may propagate
Dependent event extensions

- Specifying mutually exclusive events
- Extended FTs
- Multiple FTs for different failure modes
- Replace BEs by Petri nets
- Boolean Driven Markov Processes
Repairable fault trees

- Simple repair model: Simultaneous independent repairs
- Problem: Limited resources for repairs in real life
- Problem: Hidden failures
- Solution method: Repairable Fault Trees
- Add repair boxes that specify when to repair what
Fault trees with temporal properties

- Static FTs do not consider timing information
- Phased systems
- Delays
- Failure sequences
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance
Shortcomings of fault trees

- No information about failure sequences
- Poor modeling of shared spare components
- Dependencies cause large trees
- One solution: Dynamic fault trees (DFTs)
Dynamic fault trees

Three new gates:

- **PAND gate**
- **FDEP gate**
- **SPARE gate**
DFT Example

Power Supply

Bus

CPU 1

MEM 1

CPU 2

MEM 2

MEM 3

G3

C3

FDEP

PS

C1

SPARE

M1

C2

SPARE

M2

M3
DFT Example

Power Supply

Bus

CPU 1
MEM 1
MEM 3

CPU 2
MEM 2

MEM 3

G3

C1
SPARE

M1

C2
SPARE

M2

M3

FDEP

PS
DFT Example

Power Supply

Bus

CPU 1

MEM 1

CPU 2

MEM 2

MEM 3
DFT Example

- Power Supply
  - Bus
    - CPU 1
      - MEM 1
        - MEM 3
    - CPU 2
      - MEM 2
  - MEM 3

- G3
  - FDEP
    - PS
      - C1
        - SPARE
          - M1
      - C2
        - SPARE
          - M2
      - M3
DFT Example
DFT Example

- Power Supply
- Bus
  - CPU 1
  - CPU 2
  - MEM 1
  - MEM 2
  - MEM 3
- G3
- FDEP
- PS
- C1
  - SPARE
  - M1
- C2
  - SPARE
  - M2
- M3

The diagram illustrates a dynamic fault tree example with components like CPUs, memory modules, and power supplies interconnected.
DFT Example

Power Supply

Bus

CPU 1

MEM 1

MEM 3

CPU 2

MEM 2

MEM 3

G3

PS

FDEP

C1

SPARE

M1

M3

M2

C2

SPARE

M1

M2

M3

C3

PS 38 / 55
DFT Example

Power Supply

Bus

CPU 1

MEM 1

CPU 2

MEM 2

MEM 3

Power Supply

CPU 1

MEM 1

CPU 2

MEM 2

MEM 3

FDEP

G3

SPARE

M1

M2

M3

SPARE

C1

C2

C3

39 / 55
DFT Example

- Power Supply
- Bus
- CPU 1
- MEM 1
- CPU 2
- MEM 2
- MEM 3

Fault Tree Analysis

1. **DFT Example**
2. **Power Supply**
3. **Bus**
4. **CPU 1**
5. **MEM 1**
6. **CPU 2**
7. **MEM 2**
8. **MEM 3**

Fault Tree Diagram:

- **G3**
- **PS**
- **C1**
- **SPARE**
- **M1**
- **C2**
- **SPARE**
- **M2**
- **M3**

Diagram:

- **G3**
- **PS**
- **C1**
- **SPARE**
- **M1**
- **C2**
- **SPARE**
- **M2**
- **M3**
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance
### Quantitative analysis of dynamic fault trees

<table>
<thead>
<tr>
<th>Method</th>
<th>Reliability</th>
<th>Availability</th>
<th>MTBF</th>
<th>Exact</th>
<th>Deterministic</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov Chains</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>?</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>I/O IMC</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Petri Nets</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>?</td>
<td>+</td>
</tr>
<tr>
<td>Dynamic Bayesian Networks</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>?</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>Monte Carlo Simulation</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Algebraic analysis</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table:** Quantitative analysis for dynamic fault trees
DFT analysis: Markov chain

Analysis by markov chain:

\[ \lambda_1 = 1 \]
\[ \lambda_2 = 2 \lambda_3 = 3 \]
Advantages:
- Exact semantics
- No nondeterminacy
- Reuse of existing modelcheckers (PRISM, etc.)
DFT analysis: Markov chain

**Advantages:**
- Exact semantics
- No nondeterminacy
- Reuse of existing modelcheckers (PRISM, etc.)

**Disadvantages:**
- Semantics are ca. 20 pages long
- Combinatorial explosion
DFT analysis: Compositional Markov Analysis

- **Input/Output Interactive Markov Chains** exist of gates and basic events
- Input/Output signals allow parallel composition
- Models of FT elements are composed into one large model
DFT analysis: I/O IMC example

\[
\begin{align*}
M_{E_2} &= \begin{array}{c}
\lambda = 2 \\
E_2 \quad f_{E_2}!
\end{array} \\
M_{B} &= \begin{array}{c}
\lambda = 2 \\
E_3 \quad f_{E_3}! \\
E_2 \quad f_{E_2}! \\
B \quad f_{B}!
\end{array} \\
M_{E_2} \parallel M_{B} &= \begin{array}{c}
\lambda = 2 \\
E_3 \quad f_{E_3}! \\
E_2 \quad f_{E_2}! \\
B \quad f_{B}!
\end{array}
\end{align*}
\]
DFT analysis: Compositional Markov Analysis

**Advantages:**
- Semantics easier to understand
- Intermediate minimization reduces state-space explosion
- Easy to add new gates or events
- Can model nondeterminacy
DFT analysis: Compositional Markov Analysis

**Advantages:**
- Semantics easier to understand
- Intermediate minimization reduces state-space explosion
- Easy to add new gates or events
- Can model nondeterminacy

**Disadvantages:**
- Still has state-space explosion
- Nondeterminacy
Importance of maintenance
Importance of maintenance
When to do maintenance

- Preventive maintenance
- Corrective maintenance
Effect of maintenance

On component:
- 'As good as new' replacement
  - example: Replace battery
- Reduced failure rate
  - example: Oil change
Effect of maintenance on system

Positive:

- Correct failure (corrective)
- Reduce failure rate (preventive)
Effect of maintenance on system

Positive:
- Correct failure (corrective)
- Reduce failure rate (preventive)

Negative:
- Cost
- Downtime
Maintenance strategy

- What maintenance actions to do on which components?
- When to perform preventive maintenance?
  - Type of schedule (clock based, etc.)
  - Frequency
- How to react to failures?
Project goal

Fault tree

Data

Options

Maintenance

Availability

Reliability

Cost

Strategy
Outline

1. Introduction
2. Fault tree analysis
3. FT extensions
4. Dynamic fault trees
5. DFT analysis
6. Maintenance