Synthesizing FDIR Recovery Strategies
Safety of Future Systems Workshop 2018

Sascha Müller
DLR German Aerospace Center
Institute of Simulation and Software Technology
Software for Space Systems and Interactive Visualization
Braunschweig

April 9 - 13
Fault Detection, Isolation and Recovery

**FDIR**

Even well designed systems cannot avoid the existence of faults

⇒ But not every fault is a **failure**

⇒ FDIR tries to prevent faults from turning into failures
Fault Detection, Isolation and Recovery

FDIR

Even well designed systems cannot avoid the existence of faults
⇒ But not every fault is a **failure**
⇒ FDIR tries to prevent faults from turning into failures

Detection
⇒ Fault detected?

(Isolation)
⇒ Where?
⇒ Why?

Recovery
⇒ Can we recover?
⇒ How?
Fault Detection, Isolation and Recovery

FDIR

Even well designed systems cannot avoid the existence of faults
- But not every fault is a failure
- FDIR tries to prevent faults from turning into failures

Detection
- Fault detected?

(Isolation)
- Where?
- Why?

Recovery
- Can we recover?
- How?
Modeling the F in FDIR

Fault Model

Relationship between basic faults and how they lead to failures

- **Fault Tree Analysis**
- Failure Modes and Effects Analysis
- ... and many more
Modeling the F in FDIR

Fault Model

Relationship between basic faults and how they lead to failures

- Fault Tree Analysis
- Failure Modes and Effects Analysis
- ... and many more

Problem

Input: Given a fault model...
Output: ...compute a Recovery Strategy.
Dynamic Fault Trees - Spare Gate

Issues with default semantics

- Order is statically fixed
- Spare order may not be optimal
- Semantic issues with concurrent spare claims
Dynamic Fault Trees - Spare Gate

Issues with default semantics

- Order is statically fixed
- Spare order may not be optimal
- Semantic issues with concurrent spare claims
Non-Deterministic Fault Trees

**Idea**

Split Fault Tree up into...
- Non-Deterministic Fault Tree (NdDFT)
  - No Fixed spare ordering
- Deterministic Recovery Strategy (Recovery Automaton)
  - Recovery actions: Claim spare gate, do nothing
Non-Deterministic Fault Trees

Idea

Split Fault Tree up into...
- Non-Deterministic Fault Tree (NdDFT)
  - No Fixed spare ordering
- Deterministic Recovery Strategy (Recovery Automaton)
  - Recovery actions: Claim spare gate, do nothing

Encode nondeterministic decision for applying recovery actions in a Markov Automaton model.
Transformation Road Map

Markov Automaton

FAIL

\[ B_1: \text{CLAIM (System; Spare)} \]

\[ B_2: \text{Recovery Automaton} \]

Markov Chain

FAIL

\[ q_0: \text{start} \]

\[ q_1: \text{CLAIM (System; Spare)} \]

Reliability

NdDFT
Transformation Road Map

NdDFT ➔ Markov Automaton

- **System**
- **Primary**
- **Spare**

$B_1$, $B_2$

**CLAIM**

- **FAIL**

$B_1 : \lambda$

$B_2 : \mu$

CLAIM(System, Spare)
Transformation Road Map

- NdDFT
- Markov Automaton
- Recovery Automaton
Transformation Road Map

NdDFT → Markov Automaton → Recovery Automaton

Markov Chain

B₁: λ → CLAIM(System, Spare) → B₂: μ → FAIL

start → q₀ → B₁: CLAIM(System, Spare) → q₁
Transformation Road Map

- NdDFT
- Markov Automaton
- Recovery Automaton

Markov Chain

Reliability

CLAIM (System; Spare)

FAIL

start → $q_0$ → $B_1 : \text{CLAIM}(\text{System}; \text{Spare})$ → $q_1$

$B_1 : \lambda$

$B_2 : \mu$

$\lambda$

$\mu$
Example
Example - Results

\[ q_0 \rightarrow q_1 \rightarrow q_3 \]

- \( B_1 : \emptyset \)
- \( B_2 : \text{CLAIM}(S_1, \text{Spare}) \)
- \( B_3 : \text{CLAIM}(S_2, \text{Spare}) \)
- \( B_4 : \emptyset \)

Reliability of DFT

Reliability of NdDFT with Recovery Automaton

\[
\begin{align*}
\text{Reliability} & \quad \text{Time} \\
0 & \quad 0 \\
0.2 & \quad 0.2 \\
0.4 & \quad 0.4 \\
0.6 & \quad 0.6 \\
0.8 & \quad 0.8 \\
1 & \quad 1
\end{align*}
\]
Formalization of Recovery Actions

Future...

Other actions that are relevant:

- Repair
  - "Reset failed sensor"
Future...

Other actions that are relevant:

- Repair
  ("Reset failed sensor")

- **Mode changes**
  ("Switch to Safe Mode")
Formalization of Recovery Actions

Future...

Other actions that are relevant:

- Repair
  ("Reset failed sensor")

- **Mode changes**
  ("Switch to Safe Mode")

- Maintenance
  ("Flush memory to clean data corruptions")

Thank You!!