Modeling of Failure Propagation in Automotive Mechatronic Systems

Mattias Nyberg, Scania Predevelopment

mattias.nyberg@scania.com

MODPROD Workshop 2012-02-08
What is “failure propagation”? 
Why to model failure propagation? 
How to model failure propagation? 
How to obtain the model?
What is failure propagation?
What is failure propagation?
What is failure propagation?
What is failure propagation?
Why to model failure propagation?

- Fault signalling in on-board fault management
- Off-board diagnosis
- Dependability analysis
  - Safety (ISO26262 Automotive Functional Safety)
  - Reliability
  - Availability
On-Board Fault Management

Guarantee a certain level of performance even though the system is affected by faults.

- Uptime
- Functional safety
Without Fault Management
Without Fault Management

Diagram:
- SW Component
  - SW Component
    - SW Component
      - HW Component
Without Fault Management

- SW Component
- SW Component
- SW Component
- SW Component
- HW Component
Without Fault Management

Diagram showing the relationship between SW and HW components.
Without Fault Management
Without Fault Management

Accident / Breakdown

Component

SW Component

SW Component

SW Component

HW Component

SW Component

SW Component

SW Component
With Fault Management

- SW Component
  - SW Component
  - HW Component
  - SW Component
  - SW Component
  - SW Component
With Fault Management

**Diagram:**
- SW Component
  - SW Component
    - HW Component
  - SW Component
  - SW Component
  - SW Component

**Note:**
- HW Component indicates a hardware component.
With Fault Management

Diagram showing relationships between SW and HW components.
With Fault Management

![Diagram with warning signs and components]

- SW Component
- HW Component
- SW Component
- SW Component
- SW Component

† Indicates hierarchy or dependency relationship.

2/9/2012
Off-Board Diagnosis

Given symptoms, localize HW-components that need to be replaced or repaired.

- Computer support is needed for efficient fault localization.
Fault Causing a Symptom
Knowing the structure of the system, we can isolate the causing fault to a subset of HW-components.

The faulty component within the subset is unknown.
Diagnosis

Symptom

Symptom

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component

Component
Functional safety reliability / availability

Diagram:

- Accident / Breakdown
- P = 10^-7
- SW Component
- SW Component
- HW Component
- P = 10^-7
- SW Component
- SW Component
- SW Component
Key Knowledge

How does a fault propagate through the system?

Fault signalling in fault management

Off-board diagnosis

Functional safety reliability / availability

Fault in HW Component

- SW Component

Symptom

Fault in HW Component

- SW Component

Symptom

Fault in HW Component

- SW Component

Symptom

Accident / Breakdown

P = 10^-7

P = 10^-7
How to model failure propagation?

- "Classical" logical approach
- Probabilistic approach (Bayesian Networks)
"Classical" logical approach to model failure propagation

- Failure Propagation and Transformation Notation (FPTN) (Fenelon&McDermid:1993)
- Tabular Failure Annotation of the HiP HOPS methodology (Papadopoulos, McDermid, Sasse, Heiner:2001)
- Component Fault Trees (CFTs) (Kaiser, Liggesmeyer, Mäckel:2003)
- State Event Fault Trees (SEFTs) (Grunske, Kaiser, Papadopoulos:2005)
- Architecture Analysis & Design Language (AADL) with its Error Model Annex (Feiler&Rugina:2007)
- EAST-ADL with its Dependability package
"Classical" logical approach to model failure propagation

- Failure Propagation and Transformation Notation (FPTN) (Fenelon & McDermid: 1993)
- Tabular Failure Annotation of the HiP HOPS methodology (Papadopoulos, McDermid, Sasse, Heiner: 2001)
- Component Fault Trees (CFTs) (Kaiser, Liggesmeyer, Mäckel: 2003)
- State Event Fault Trees (SEFTs) (Grunske, Kaiser, Papadopoulos: 2005)
- Architecture Analysis & Design Language (AADL) with its Error Model Annex (Feiler & Rugina: 2007)
- EAST-ADL with its Dependability package

- Many effects involve uncertainties
- Limited inference possibilities
Probabilistic approach to model failure propagation

Bayesian Network

A
P(A)

B
P(B | A, E)

C
P(C | B)

D
P(D)

E
P(E | A, D)
How to obtain the model of failure propagation?

- Engineering intuition (informal)
- Derive from service dependencies (semiformal)
- Derive from requirements and specifications (formal)
A Software Component:
A Software Component: is a Service Provider

Customer

Service Provider

Suppliers

SW Component

SW Component

SW Component

Services
Building the Failure Propagation Graph
Prototype – SCR-System

- SCR System
- ECUer (SW+HW): EEC, EMS, COO
- Hardvara: sensorer, aktuatorer, tankar, rör
- Exempel…
Control Unit System Architecture
Bayesian Network of Failure Propagation in the SCR-system
Bayesian Network of Failure Dependencies in the SCR-system
How to obtain the model of failure propagation?

- Engineering intuition (informal)
- Derive from service dependencies (semiformal)
- Derive from requirements and specifications (formal)
A Framework for Requirements, Architecture, and Design
- Challenges -

- Compatible with industrial practices (Scania)
- Multi-level compatibility
- Compatible with ISO26262
- User friendly
- Enough powerful for our aims
Framework

Contract Based Design: e.g. see EU project SPEEDS.

A system
- has a set of (intrinsic) port variables
- has an assumption expressed in the system ports
- has a promise expressed in the system ports

- may have an architecture, i.e. internal architecture:
  - a set of subsystems
- may have a set of extrinsic port variables
  (i.e. references to the environment)
**Example: Fuel Level Display**

Electrical System

Assumption: \( \text{tank\_level\_sensor\_level} = \text{fuel\_level} \) (i.e. sensor is placed in tank)

\[ \text{fuel\_level} = f(\text{fuel\_volume}) \] (i.e. correct tank volume-level profile)

Promise: \( \text{warning\_lamp} = \text{fuel\_volume} < J \)
Example: Fuel Level Display

Electrical System
Assumption: \( \text{tank\_level\_sensor\_level} = \text{fuel\_level} \) (i.e. sensor is placed in tank)
\[ \text{fuel\_level} = f(\text{fuel\_volume}) \] (i.e. correct tank volume-level profile)
Promise: \( \text{warning\_lamp} = \text{fuel\_volume} < J \)

Architecture
- ECU\_system\_COO
  Assumption:
  \begin{align*}
  & \text{CAN\_ICL\_in} = \text{CAN\_COO\_out} \\
  & \text{warning\_lamp} = \text{CAN\_ICL\_in} \\
  & \text{tank\_level\_sensor\_level} = \text{fuel\_level} \\
  & \text{fuel\_level} = f(\text{fuel\_volume})
  \end{align*}
  Promise: \( \text{warning\_lamp} = \text{fuel\_volume} < J \)
- CAN\_bus
  Promise: \( \text{CAN\_ICL\_in} = \text{CAN\_COO\_out} \)
- ECU\_system\_ICL
  Promise: \( \text{warning\_lamp} = \text{CAN\_ICL\_in} \)
Example: Fuel Level Display

Electrical System
Assumption: \( \text{tank\_level\_sensor\_level} = \text{fuel\_level} \) (i.e. sensor is placed in tank)
\[ \text{fuel\_level} = f(\text{fuel\_volume}) \] (i.e. correct tank volume-level profile)
Promise: \( \text{warning\_lamp} = \text{fuel\_volume} < J \)

Architecture
- **ECU\_system\_COO**
  Assumption: \( \text{CAN\_ICL\_in} = \text{CAN\_COO\_out} \)
  \( \text{warning\_lamp} = \text{CAN\_ICL\_in} \)
  \( \text{tank\_level\_sensor\_level} = \text{fuel\_level} \)
  \[ \text{fuel\_level} = f(\text{fuel\_volume}) \]
Promise: \( \text{warning\_lamp} = \text{fuel\_volume} < J \)
- **CAN\_bus**
  Promise: \( \text{CAN\_ICL\_in} = \text{CAN\_COO\_out} \)
- **ECU\_system\_ICL**
  Promise: \( \text{warning\_lamp} = \text{CAN\_ICL\_in} \)
Example: Fuel Level Display

ECU_system_COO:
Assumption:  CAN_ICL_in = CAN_COO_out
            warning_lamp = CAN_ICL_in
            tank_level_sensor_level = tank_level (i.e. sensor is placed in tank)
            fuel_level = f(fuel_volume)  (i.e. correct tank volume-level profile)
Promise:    warning_lamp = fuel_volume < J

Architecture
- Tank_level_sensor_system
  Assumption:  Tank_level_sensor_level = tank_level
  Promise:    port_in = tank_level
- COO
  Assumption:  port_in = tank_level
               CAN_ICL_in = CAN_COO_out
               warning_lamp = CAN_ICL_in
               fuel_level = f(fuel_volume)
  Promise:    warning_lamp = fuel_volume < J
Summary

- Model of failure propagation is a key for:
  - On-board fault management
  - Off-board diagnosis
  - Dependability analyses, e.g. functional safety in ISO26262

- Failure propagation can be modeled using
  - Classical logical approach
  - Bayesian networks (!)

- Failure propagation model can be obtained from
  - Engineering intuition (?)
  - Service dependencies
  - Requirements and specifications
References

A Service Based Approach to Decentralized Diagnosis and Fault Tolerant Control.

A Decentralized Service Based Architecture for Design and Modeling of Fault Tolerant Control Systems.