1 The Topic
2 Tasks
3 Modeling
4 Diagnosis
   4.3 Component-oriented Diagnosis
      4.3.2 Application: On-board Diagnosis of Kfz

Goal:
- Prototype Applications
- Real-Time-Requirements
- Ref: [Struss-Price 04]

Diagnosis and Fault Analysis - Requirements

- Increasing complexity of systems
- Stronger requirements
  - Legal restrictions
  - Customers
- Variant problem
  - versions of subsystems
- Safety critical application
  - completeness of results
- Diagnostics during design
- Representation and re-use of knowledge
Diagnosis and Fault Analysis - The Opportunity

- Computational power available
  - During design
  - In the workshop
  - On-board
- Computer support possible

- Knowledge-intensive tasks
- require knowledge-based systems

Project Vehicle Model Based Diagnosis (2/97-1/99)

![Map of Europe with project participants]

- GenRad
- Aberystwyth
- Università degli Studi di Torino
- DASSAULT ELECTRONIQUE
- AMT MARELLI
- MANN-FILTER
VMBD Guiding Applications: Drive Train

Demonstrator: Turbo Charger System

- On-board detection and localization of
- faults related to black smoke
- under realistic conditions (e.g. sensors)
- with model-based techniques from Artificial Intelligence
Demonstration Turbo Control Subsystem

Scenario 1
Leakage

Scenario 2
Air flow sensor

Scenario 3
Boost pressure

Demonstrator Vehicle Set-Up

MAC 2
serial line

ETK
ECU

VS100
RAZ'R
The VMBD Demonstrator Cars

Demonstrator Car (Volvo) with RAZ’R
Demonstrator Car (Volvo) Switchboard for Fault Injection

RAZ’R Development System

Modeling Primitives → Ontology Definition
Component Types → Model Definition
Structure → Scenario Definition
Model Composition → Observations
System Model → Diagnosis
RAZ’R Runtime System

Modeling Primitives → Component Types → Structure

Ontology Definition → Model Definition → Scenario Definition → Model Composition

Observations → System Model

VS100 → Signal Abstraction → Diagnosis RTS

Demonstration Turbo Control Subsystem

Scenario 1: Leakage

Scenario 2: Air flow sensor

Scenario 3: Boost pressure
Diagnosis and Fault Analysis of Vehicles

- Increasing complexity of systems
- Stronger requirements
  - Legal restrictions
  - Customers
- Variant problem
  - versions of subsystems
- Safety critical application
  - completeness of results
- Diagnostics during design
- Representation and re-use of knowledge

Increasing Complexity ...

Abbildung 1: Fahrzeugverkabelung Baureihe 170, Baujahr 1949

Source: Hoffmann et al. (DaimlerChrysler), VDI'01
Increasing Complexity ......

Abbildung 2: Fahrzeugverkabelung S-Klasse, Baujahr 1991

Source: Hoffmann et al. (DaimlerChrysler), VDI'01

Vehicles: A Mobile Hw/Sw Platform

- Kombi
- EPB
- Airbag
- ESP
- Gateway
- Auxiliary Heating
- VL381
- SCU/SMLS
- EDC17
- Simos 8.2
- MED9
Diagnosis Problems through Interacting ECU’s

Example:
- The AC does not come on
- Reason: defective tank level sensor!

Explanation:
- AC ECU send a request to Drive Train ECU
- Drive Train ECU checks fuel level
- Defective tank level sensor signals low fuel
- Drive Train ECU denies AC request

- Relevant
  - to on-board diagnosis
  - to diagnosability analysis during design

Diagnosis Problems through Interacting ECU’s
Knowledge-based Systems for Industrial Applications

1 The Topic
2 Tasks
3 Modeling
4 Diagnosis
   4.3 Component-oriented Diagnosis
    4.3.3 Fault localization - a second glance

Goal:
- Problem of fault localization
- Script: Chap. 10.4.1

Yet Another Simple Example

- Head lights work
- Starter and rear light don’t
- Obvious diagnosis: Starter and rear light are broken
Fault Localization for the Simple Example - Conflicts 1 and 2

- ok(Battery) ∧ ok(Wire₁) ∧ ok(Wire₂) ∧ ok(Starter) ⇒ active(Starter)
- OBS ⇒ ¬ active(Starter)
- → Conflict
  ¬ ok(Battery) ∨ ¬ ok(Wire₁) ∨ ¬ ok(Wire₂) ∨ ¬ ok(Starter)

Analogously:
- ¬ ok(Battery) ∨ ¬ ok(Wire₁) ∨ ¬ ok(Wire₂) ∨ ¬ ok(Wire₃) ∨ ¬ ok(Wire₄) ∨ ¬ ok(RLight)

Fault Localization for the Simple Example - Conflicts 3 and 4

- lit(HLight) ∧ ok(HLight) ∧ ok(Wire₅) ∧ ok(Wire₆) ∧ ok(RLight) ⇒ lit(RLight)
- OBS ⇒ ¬ lit(RLight)
- → Conflict
  ¬ ok(HLight) ∨ ¬ ok(Wire₅) ∨ ¬ ok(Wire₆) ∨ ¬ ok(RLight)

Analogously:
- ¬ ok(HLight) ∨ ¬ ok(Wire₅) ∨ ¬ ok(Wire₆) ∨ ¬ ok(Wire₃) ∨ ¬ ok(Wire₄) ∨ ¬ ok(Starter)
Fault Localization for the Simple Example - Hitting Sets

- \{\text{Battery, Wire}_1, \text{Wire}_2, \text{Starter}\}
- \{\text{Battery, Wire}_1, \text{Wire}_2, \text{Wire}_3, \text{Rlight}\}
- \{\text{Rlight, Wire}_5, \text{Wire}_6, \text{Rlight}\}
- \{\text{Hlight, Wire}_5, \text{Wire}_6, \text{Wire}_3, \text{Wire}_4, \text{Starter}\}
- \{\text{Starter, Rlight}\}
- \{\text{Battery, HLight}\}
- \{\text{Wire}_1, \text{Wire}_3\}

...+ 19 more!

What Makes Most of the Fault Localizations Implausible?

- If the battery were broken, the headlights would not be lit
- Broken headlights cannot be lit
- \(\rightarrow\) Knowledge about faults can reduce the set of fault localizations

- \{\text{Starter, Rlight}\}
- \{\text{Battery, HLight}\}
- \{\text{Wire}_1, \text{Wire}_3\}

...+ 19 more!
Fault Models - “Physical Negation”

- If the battery were broken, the headlights would not be lit
- Broken headlights cannot be lit
- → Knowledge about faults can reduce the set of fault localizations

- ¬ok(C_i) Negation logically: Anything but the normal behavior holds
- “Physical negation”: The broken component still behaves in a restricted way
- Captured by fault models
- ¬ok(C_i) \(\Rightarrow\) \(\lor\) fault\_ij(C_i)
- \(\land\) ¬fault\_ij(C_i) \(\Rightarrow\) ok(C_i)

Refuting Fault Modes

- Flat (Battery) \(\land\) ok(Wire_1)
  \(\land\) ok(Wire_2) \(\land\) ok(Wire_3)
  \(\land\) ok(Wire_4) \(\land\) ok(Wire_5)
  \(\land\) ok(Wire_6) \(\land\) ok(HLight)
  \(\Rightarrow\) ¬lit(HLight)
- OBS \(\Rightarrow\) lit(HLight)
- \(\Rightarrow\) ok(Wire_1) \(\land\) ok(Wire_2)
  \(\land\) ok(Wire_3) \(\land\) ok(Wire_4)
  \(\land\) ok(Wire_5) \(\land\) ok(Wire_6)
  \(\land\) ok(HLight) \(\Rightarrow\) ¬Flat (Battery)
- \(\Rightarrow\) ok(Battery)

- Physical negation eliminates all but one minimal fault localizations
- {Starter, RLight}