

Grundlagen der Künstlichen Intelligenz

Sebastian Iwanowski
FH Wedel

Kap. 4:
Wissensbasierte Systeme

4.4: Modellbasierte Verarbeitung

3. Modellbasierte Diagnose

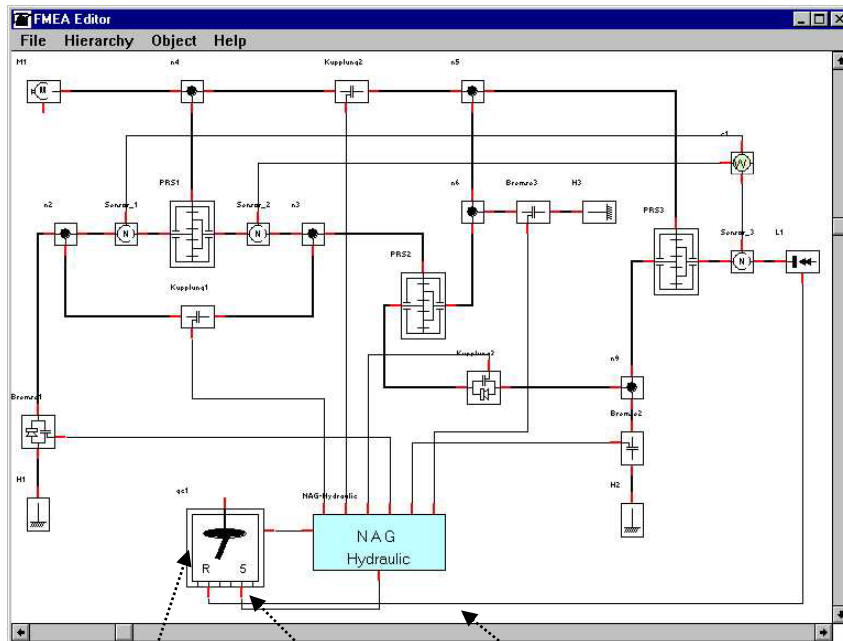
Ziel:

- schneller Wissenserwerb
- exaktes und nachvollziehbares Ergebnis der Problemlösungsmaschine

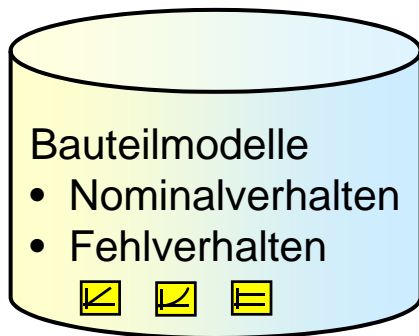
Schwierigkeit:

- schnelle Antwort der Problemlösungsmaschine zur Laufzeit

3. Modellbasierte Diagnose



Komponente Port Verbindung



Systemstruktur:

Welche Komponenten von welchem Typ sind wie miteinander verbunden ?

→ erhältlich aus CAD-Daten

Komponentenmodelle:

Wie ist die Abhängigkeit zwischen den Werten, die an den Verbindungspunkten einer Komponente anliegen ?

→ pro Komponententyp einmal zu modellieren

→ Modell ist wiederverwendbar für alle Systeme, in denen Komponenten dieses Typs enthalten sind

3. Modellbasierte Diagnose

Eingabe in die Wissensbasis:

- Hierarchische Struktur des Systems (Aufbau aus Komponenten)
- Komponentenmodelle

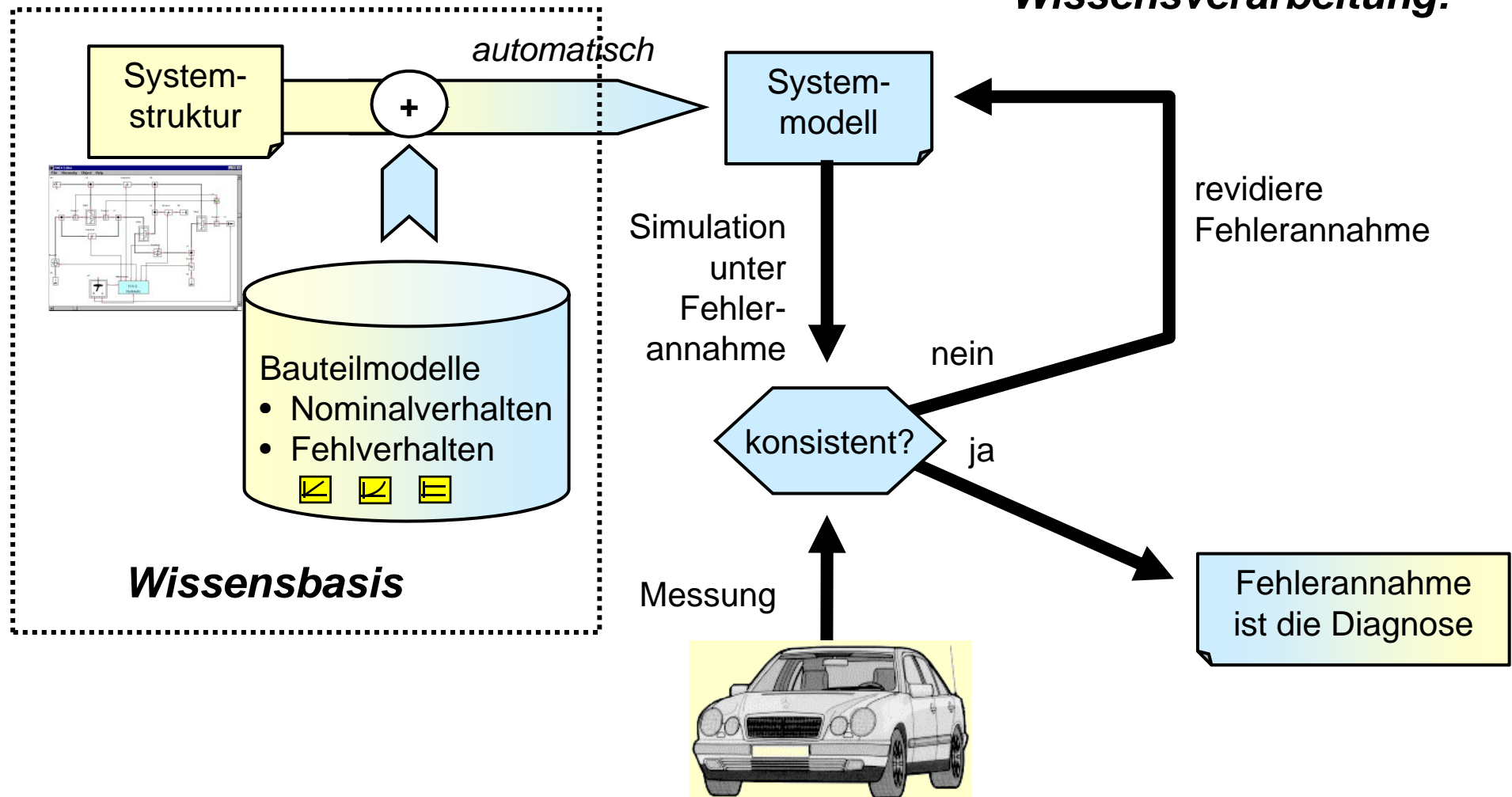
Struktur der Wissensbasis:

- Constraint-Netzwerk (automatisch zusammengebaut)
- Gliederung des Constraint-Netzwerks durch:
 - Zuordnung der Constraints zu Komponenten bzw. Ports
 - Zuordnung der Variablen zu Komponenten bzw. Ports

Anwendungsgebiet: Technische Diagnose

Basisfunktionalität der modellbasierten Diagnose

Wissensverarbeitung:



Anwendungsgebiet: Technische Diagnose

Basisfunktionalität der modellbasierten Diagnose

Die GDE 1987: *Der Prototyp* für die modellbasierte Diagnose

Problem:

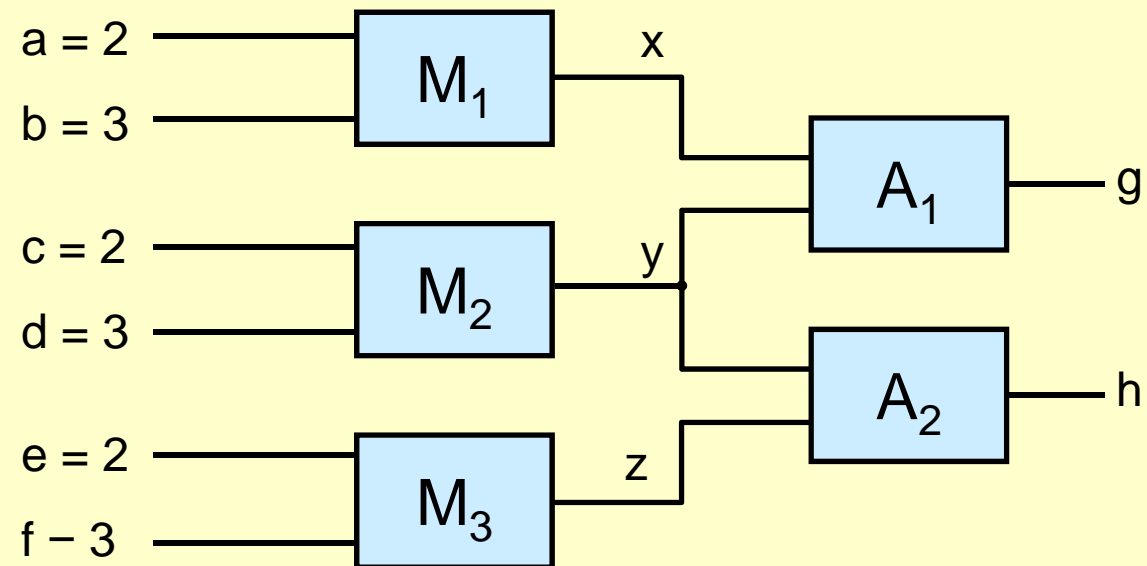
- ‚brute-force‘ Simulation **aller** Fehlerannahmen
kombinatorisch nicht realisierbar

Idee: General Diagnostic Engine GDE, deKleer & Williams 1987

- intelligente Suche im Raum der möglichen Fehlerannahmen
- nutzt inkonsistente Annahmen zum Verkleinern des Suchraums
- Prinzip: **konfliktgesteuerte Suche**

GDE - Beispiel

Systemstruktur

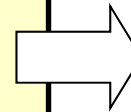
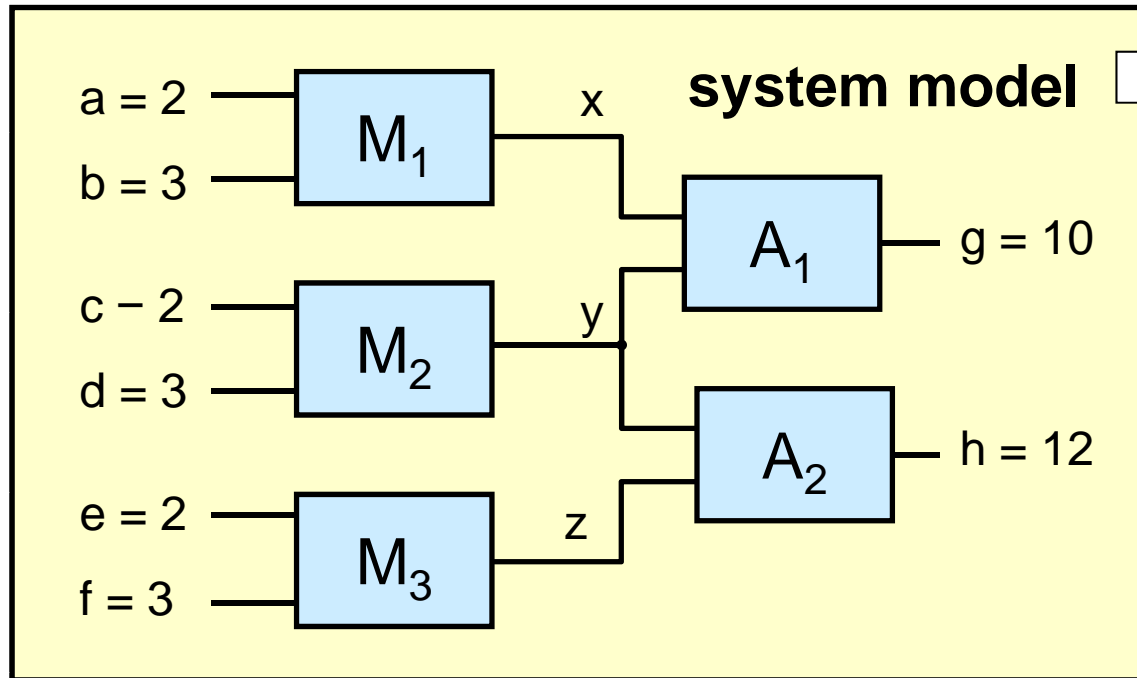


Komponentenmodelle

- Multiplizierer: $\text{mode=ok} \Rightarrow \text{out} = \text{in}_1 * \text{in}_2$
- Addierer: $\text{mode=ok} \Rightarrow \text{out} = \text{in}_1 + \text{in}_2$

Messungen $g = 10 \wedge h = 12$

GDE - Beispiel



simulation

$x = 6$ {M1}

$y = 6$ {M2}

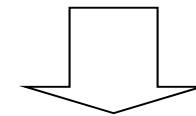
$z = 6$ {M3}

$g = 12$ {M1 M2 A1}, $g = 10$

$y = 4$ {M1 A1}

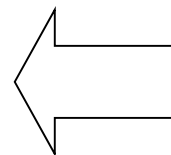
$h = 10$ {M1 A1 A2 M3}, $h = 12$

$y = 6$ {A2 M3}



two conflicts

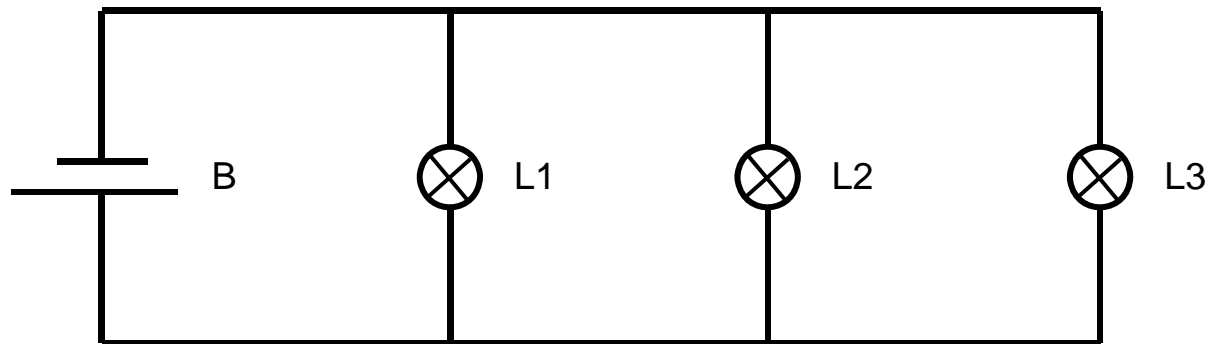
diagnoses:
 single-fault **M1**
 single-fault **A1**
 double fault **M2 M3**
 :



M1	M2	M3	A1	A2
X	X		X	
X		X	X	X

Modellbasierte Diagnose: Basisfunktionalität

Beispiel, warum Addierer-/Multiplizierer-Beispiel nicht alle Schwierigkeiten des GDE-Ansatzes aufzeigt:



Beobachtung:

L1, L2 leuchten nicht, L3 leuchtet

GDE-Diagnosen:

1. (B ok, L1 defekt, L2 defekt, L3 ok)

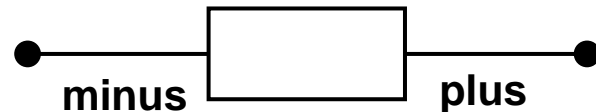
2. (B defekt, L1 ok, L2 ok, L3 defekt) ???

3. (B defekt, L1 ok, L2 ok, L3 ok) ???

Modellbasierte Diagnose: Basisfunktionalität

Modellierung der elektrischen Komponenten:

Batterie:



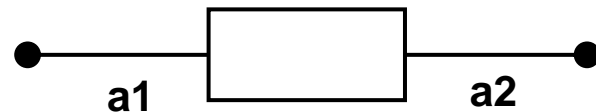
Wertebereiche: $\text{minus, plus} \in \{ \text{Masse, Versorgungsspannung} \}$

Regeln:

$\text{ok} \Rightarrow (\text{minus} = \text{Masse})$

$\text{ok} \Rightarrow (\text{plus} = \text{Versorgungsspannung})$

Kabel:



Wertebereiche: $\text{a1, a2} \in \{ \text{Masse, Versorgungsspannung} \}$

Regeln:

$\text{ok} \wedge (\text{a1} = \text{Masse}) \Rightarrow (\text{a2} = \text{Masse})$

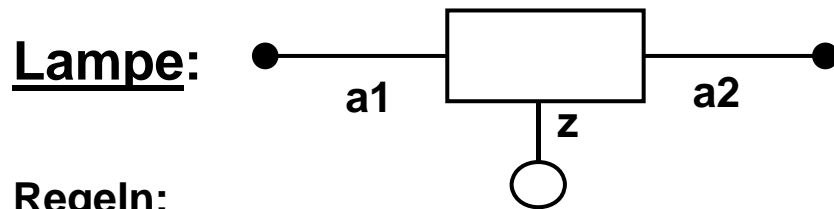
$\text{ok} \wedge (\text{a1} = \text{Versorgungsspannung}) \Rightarrow (\text{a2} = \text{Versorgungsspannung})$

$\text{ok} \wedge (\text{a2} = \text{Masse}) \Rightarrow (\text{a1} = \text{Masse})$

$\text{ok} \wedge (\text{a2} = \text{Versorgungsspannung}) \Rightarrow (\text{a1} = \text{Versorgungsspannung})$

Modellbasierte Diagnose: Basisfunktionalität

Modellierung der elektrischen Komponenten:



Wertebereiche:

$a1, a2 \in \{ \text{Masse, Versorgungsspannung} \}$
 $z \in \{ \text{hell, dunkel} \}$

$ok \wedge (a1 = \text{Versorgungsspannung}) \wedge (a2 = \text{Masse}) \Rightarrow (z = \text{hell})$

$ok \wedge (a2 = \text{Versorgungsspannung}) \wedge (a1 = \text{Masse}) \Rightarrow (z = \text{hell})$

$ok \wedge (a1 = \text{Versorgungsspannung}) \wedge (a2 = \text{Versorgungsspannung}) \Rightarrow (z = \text{dunkel})$

$ok \wedge (a1 = \text{Masse}) \wedge (a2 = \text{Masse}) \Rightarrow (z = \text{dunkel})$

$ok \wedge (a1 = \text{Masse}) \wedge (z = \text{hell}) \Rightarrow (a2 = \text{Versorgungsspannung})$

$ok \wedge (a1 = \text{Versorgungsspannung}) \wedge (z = \text{hell}) \Rightarrow (a2 = \text{Masse})$

$ok \wedge (a1 = \text{Masse}) \wedge (z = \text{dunkel}) \Rightarrow (a2 = \text{Masse})$

$ok \wedge (a1 = \text{Versorgungsspannung}) \wedge (z = \text{dunkel}) \Rightarrow (a2 = \text{Versorgungsspannung})$

$ok \wedge (a2 = \text{Masse}) \wedge (z = \text{hell}) \Rightarrow (a1 = \text{Versorgungsspannung})$

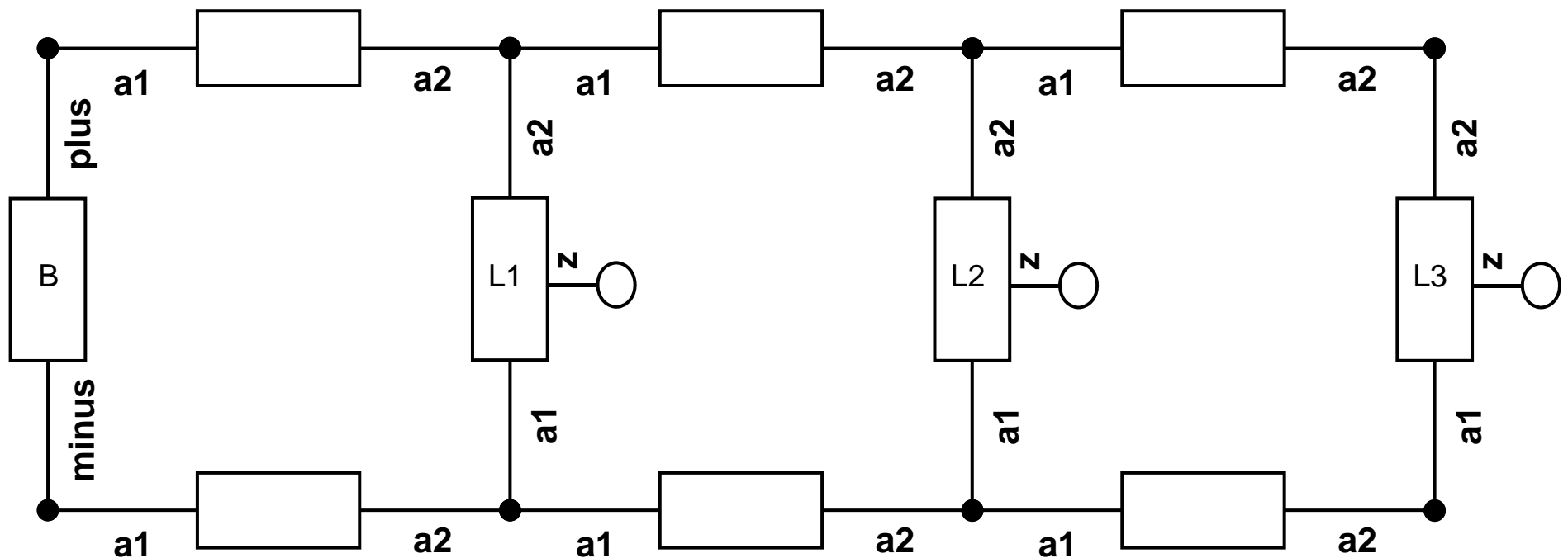
$ok \wedge (a2 = \text{Versorgungsspannung}) \wedge (z = \text{hell}) \Rightarrow (a1 = \text{Masse})$

$ok \wedge (a2 = \text{Masse}) \wedge (z = \text{dunkel}) \Rightarrow (a1 = \text{Masse})$

$ok \wedge (a2 = \text{Versorgungsspannung}) \wedge (z = \text{dunkel}) \Rightarrow (a1 = \text{Versorgungsspannung})$

Modellbasierte Diagnose: Basisfunktionalität

Zusammenbau des Systemmodells aus den Komponentenmodellen:



Werte an den Verbindungsknoten müssen gleich sein

Bei Widerspruch: Konflikt der den Werten zugrunde liegenden Verhaltensmodelle

Diagnosen sind Mengen von Verhaltensmodellen, die keinen Konflikt enthalten

Modellbasierte Diagnose: Basisfunktionalität

Fazit aus der bisher vorgenommenen Modellierung:

Es besteht kein logischer Widerspruch zu folgender Diagnose:

2. (B defekt, L1 ok, L2 ok, L3 defekt)

Grund:

L3 darf im Fehlerfall auch leuchten, wenn keine Spannungsdifferenz besteht

Unvollständigkeit der Wissensbasis !

Noch schlimmer:

Wenn eine Verhaltensregeln nur ausgewertet wird, wenn für ihre Voraussetzungen konkrete Werte vorliegen, dann kann auch kein Widerspruch zu folgender Diagnose gefunden werden:

3. (B defekt, L1 ok, L2 ok, L3 ok)

Grund:

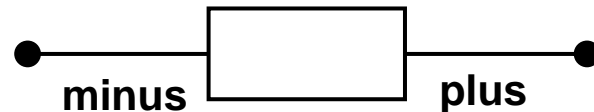
Es werden nirgendwo Spannungswerte berechnet

Mangelnde Beweisfähigkeit der Problemlösungskomponente !

Modellbasierte Diagnose: Basisfunktionalität

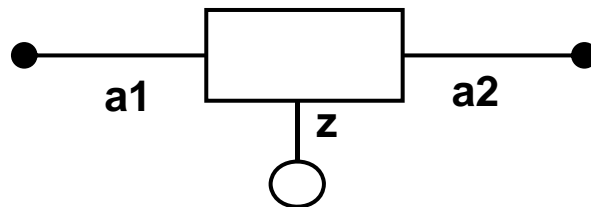
Zusätzliche Regeln für den Ausschluss der Diagnosen 2 / 3:

Batterie:



defekt \Rightarrow (minus = Masse)

Lampe:



defekt \wedge (a1 = Versorgungsspannung) \wedge (a2 = Versorgungsspannung) \Rightarrow (z = dunkel)

defekt \wedge (a1 = Masse) \wedge (a2 = Masse) \Rightarrow (z = dunkel)

Es müssen also auch Verhaltensmodelle für Fehler angegeben werden, um physikalisch unmögliches Verhalten auszuschließen.

Modellbasierte Diagnose: Erweiterte Funktionalität

Basisfunktionalität:

Eingabe:

- Einstellung bestimmter Werte im System
- Beobachtung davon abhängiger Werte im System

Ausgabe:

- Mehrere Diagnosen folgender Art:
 - Jede Diagnose weist jeder Komponente einen Verhaltensmodus zu: entweder ok oder ein definierter Fehlermodus
 - Die Regeln aller zugewiesenen Verhaltensmodi sind konsistent (mit allen eingestellten und beobachteten Werten)

Was braucht der Anwender ?

Eingabe: s.o.

Ausgabe:

- Eine eindeutige Anweisung, welche Komponenten wie repariert werden sollen

Modellbasierte Diagnose: Erweiterte Funktionalität

Erweiterte Funktionalität:

1) Vorschlag von Testeinstellungen (control inputs)

- Einstellung bestimmter Werte an bestimmten Stellen im System
(derart, dass die zu erwartenden Beobachtungen die bisher gültigen Diagnosen bestmöglich unterscheiden)

2) Vorschlag von Beobachtungspunkten

- Auswahl von Messstellen im System
(derart, dass die zu erwartenden Beobachtungen die bisher gültigen Diagnosen bestmöglich unterscheiden)

Test

Anforderung an die Modellierung:

- Definition von Testpunkten
- Definition von Testwerten, die an den Testpunkten eingestellt werden sollen
- Definition von Beobachtungspunkten, die gemessen werden sollen

Control actions

Observations

Komponentenmodellierung

Verhaltensmodi (*behavioural modes*)

- Werte der Komponente, nach denen im Diagnosevorgang gesucht wird
- Definitionsbereich immer endlich (in der Regel unter 10 verschiedene Werte möglich)

Variablen

- zum Abspeichern von Werten
- Die Variablenwerte werden in den Constraints benutzt.
- Die Constraints berechnen einen neuen Wert für eine Variable.

*Unterscheide
interne Variablen
von Portvariablen !*

Ports

- enthält die Variablen, die in den Verbindungen mit benachbarten Komponenten identifiziert werden sollen.

Constraints

- Menge von Verhaltensregeln, welche Variablen der Komponente miteinander verknüpfen
- Ein Constraint gilt normalerweise nur unter Voraussetzung, dass der Verhaltensmodus einen bestimmten Wert hat.

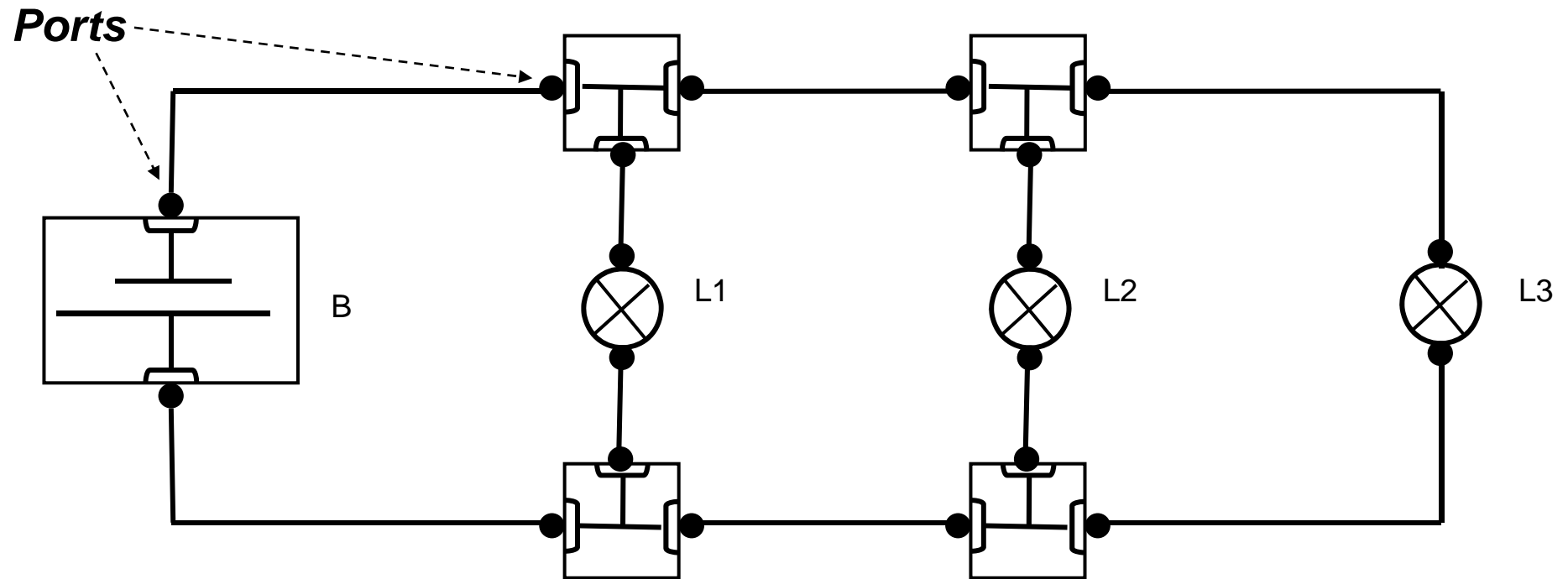
Maßnahmen (*control actions*)

- Variablen und einzustellende Werte dafür
- Maß für Zugänglichkeit und Schwierigkeit, bestimmte Werte einzustellen

Beobachtungen (*observations*)

- Variablen
- Maß für Zugänglichkeit

Modellierung eines einfachen elektrischen Systems



Komponententypen:

Batterie

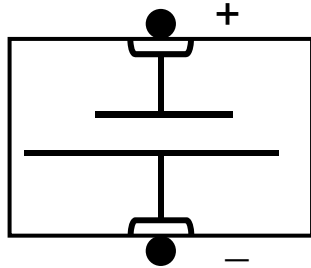
Lampe

Kabel

Steckverbindung (3)

Modellierung eines einfachen elektrischen Systems

Batterie



Fehlermodi:

entladen

Kontaktlücke bei +

Kontaktlücke bei -

Wackelkontakt bei +

Wackelkontakt bei -

korrodiert

Maßnahmen:

Klemme bei + lösen

Klemme bei - lösen

Klemme bei + befestigen

Klemme bei - befestigen

Beobachtungen:

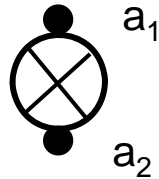
Klemmen ansehen

Spannung bei + messen

Spannung bei - messen

Modellierung eines einfachen elektrischen Systems

Lampe



Fehlermodi:

durchgebrannt
Lampe nicht eingesetzt
Wackelkontakt
korrodiert

Maßnahmen:

Lampe ausschrauben
Lampe einschrauben

Beobachtungen:

Lampe ansehen

Kabel



Fehlermodi:

unterbrochen
Kurzschluss gegen Masse
Kurzschluss gegen Spannung
korrodiert

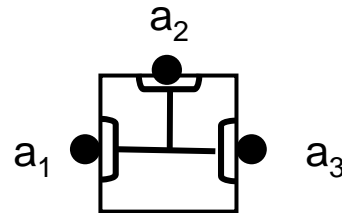
Maßnahmen:

Beobachtungen:

Spannung an a_1 messen
Spannung an a_2 messen
Kabel ansehen

Modellierung eines einfachen elektrischen Systems

Steckverbindung (3)



Fehlermodi:

Kontaktlücke bei a_1
Kontaktlücke bei a_2
Kontaktlücke bei a_3
Wackelkontakt bei a_1
Wackelkontakt bei a_2
Wackelkontakt bei a_3

Maßnahmen:

Kontakt bei a_1 schließen
Kontakt bei a_2 schließen
Kontakt bei a_3 schließen
Kontakt bei a_1 lösen
Kontakt bei a_2 lösen
Kontakt bei a_3 lösen

Beobachtungen: Kontakte ansehen