

# ***Künstliche Intelligenz***

Sebastian Iwanowski  
FH Wedel

## **Kap. 4:**

KI-Architektur: Wissensbasierte Systeme

4.2: Die verschiedenen Verarbeitungstechniken im Vergleich

# Anwendungsbeispiel Technische Diagnose

## Laufzeitsystem:

(heißt in wissensbasierten Systemen **Problemlösungsmaschine / Inferenzmaschine**)

### **Eingabe:**

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

### **Ausgabe:**

- Eine eindeutige Anweisung, welche Komponenten wie repariert werden sollen

***Darin unterscheiden sich verschiedene Diagnosesysteme***

***nicht voneinander !***

# Anwendungsbeispiel Technische Diagnose

## Wissensverarbeitung:

### 1) Wissenserwerb: Eingabe in die Wissensbasis

- fallbasiert
  - regelbasiert (symptombasiert)
  - modellbasiert
- } als Alternativen

### 2) Struktur der Wissensbasis

- abhängig von Wissenserwerb

### 3) Verarbeitung des Wissens in der Problemlösungsmaschine

- abhängig von Struktur der Wissensbasis

***Darin unterscheiden sich verschiedene Diagnosesysteme voneinander !***

# 1. Symptombasierte Diagnose

## Eingabe in die Wissensbasis:

- Verursachende bzw. sich manifestierende Fehler im Gesamtsystem
- Mögliche Symptome (Messwerte)
- Zusammenhang zwischen Fehlern und Symptomen (Regeln)
  - Symptome können das Vorliegen eines Fehlers erhärten bzw. sogar erklären
  - Symptome können das Vorliegen eines Fehlers ausschließen.

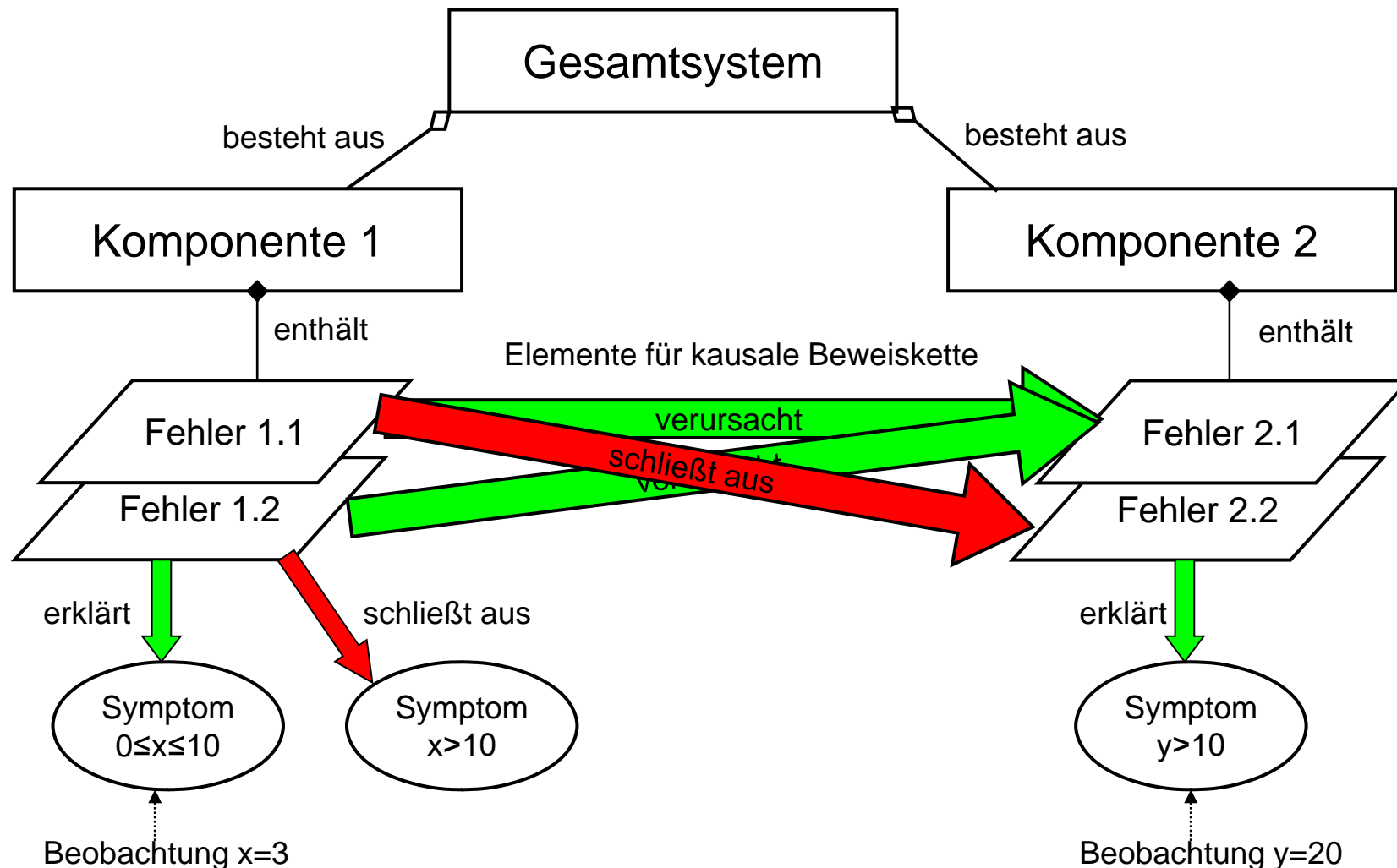
## Struktur der Wissensbasis:

- Semantisches Netz
- Mögliche Strukturen:
  - Fehlernetzwerke
  - Entscheidungsbäume

***Das ist die „klassische Expertensystemtechnik“***

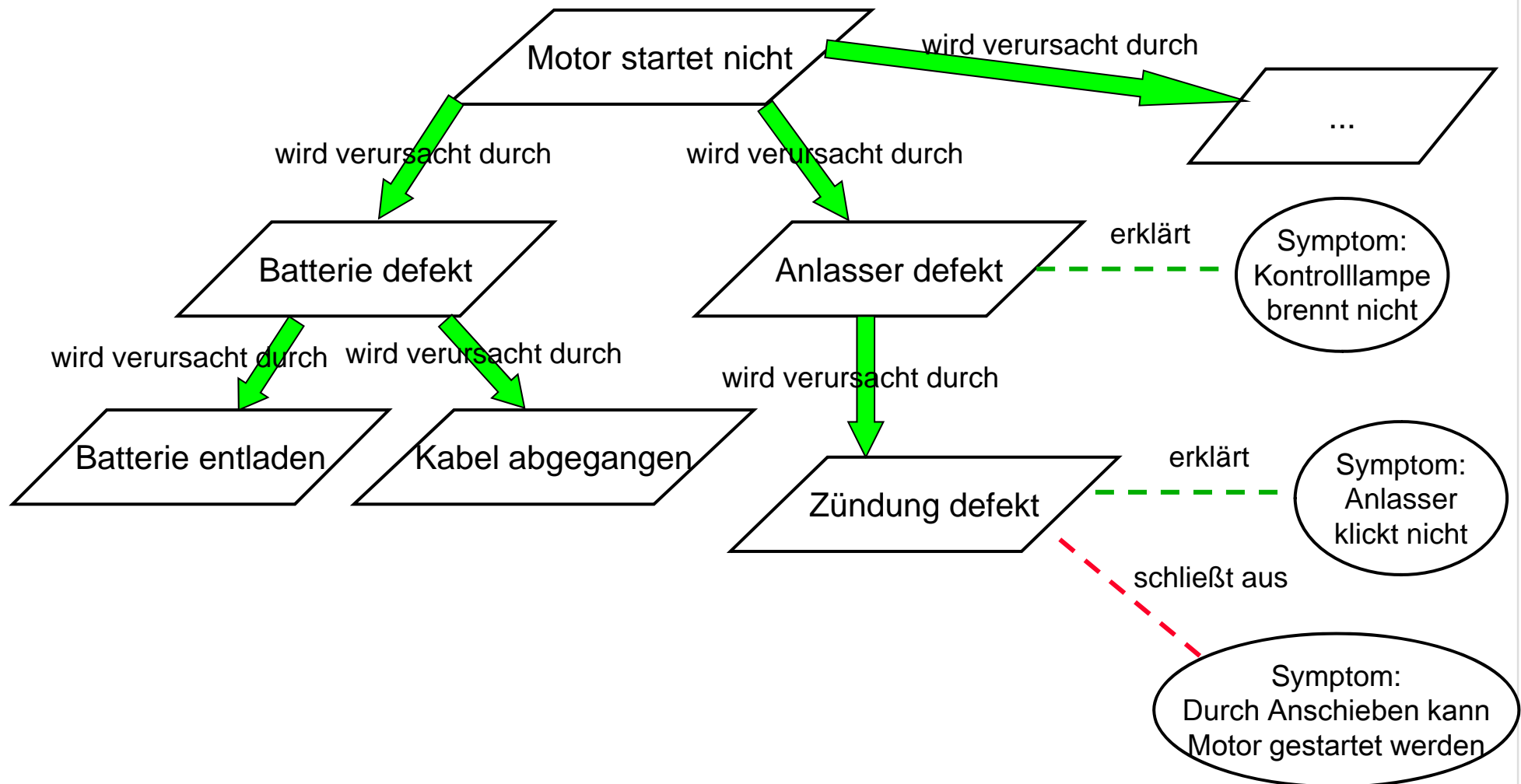
# 1. Symptombasierte Diagnose

Beispiel für die Elemente einer Wissensbasis:



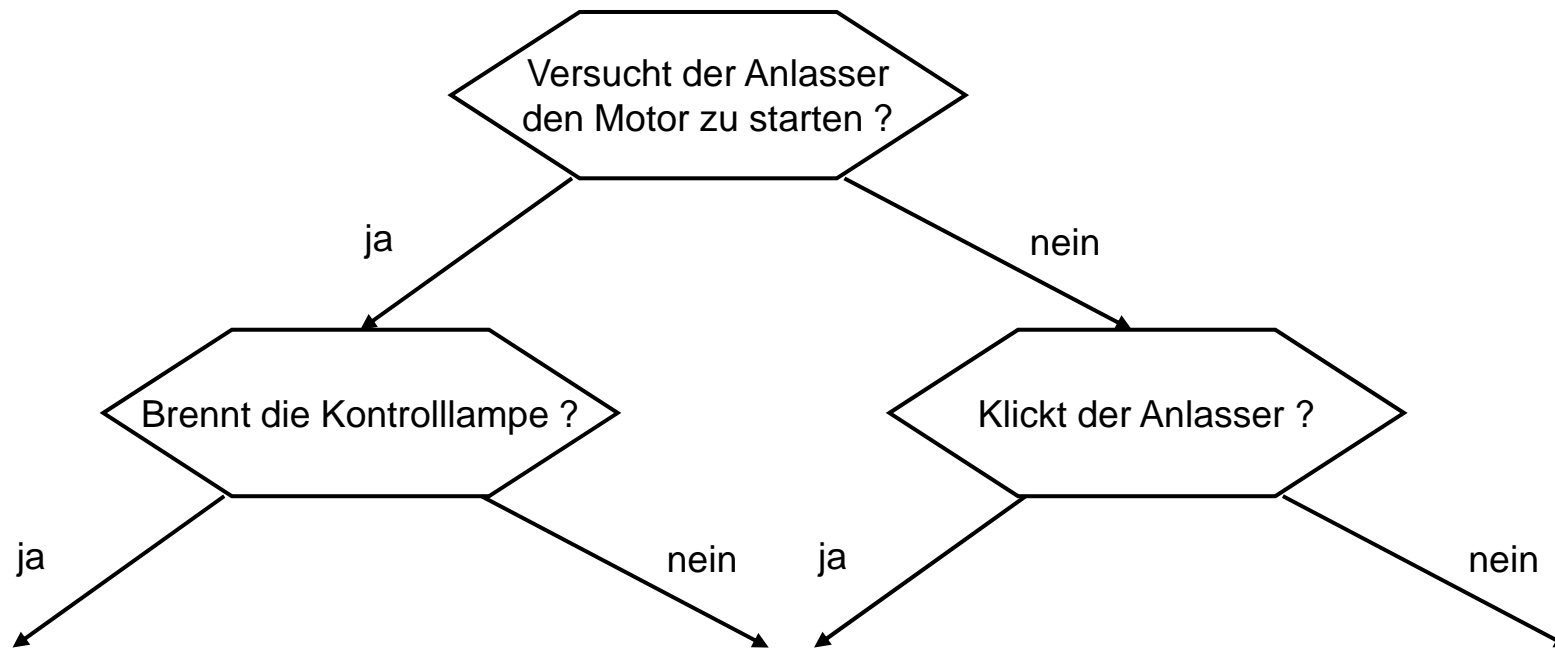
# 1. Symptombasierte Diagnose

## Beispiel für einen Fehlerbaum:



# 1. Symptombasierte Diagnose

Beispiel für einen Entscheidungsbaum:



# 1. Symptombasierte Diagnose

## Arbeit der Problemlösungsmaschine:

- **Navigation im semantischen Netz**  
(z.B. Fehlerbaum oder Entscheidungsbaum)
- **Mögliche Startpunkte der Navigation:**
  - Vermutete Fehler
  - Beobachtete Symptome
- **Kernarbeit ist das Auswerten und Feuern von Regeln:**
  - Setze berechnete Resultate in die Prämissen von anderen Regeln ein.
  - Arbeite mit Wahrscheinlichkeiten bzw. Unschärfe für Regeln

↓  
*Eine solche Eingabe muss im Wissenserwerbssystem vorgesehen sein.*



# 1. Symptombasierte Diagnose

## Vorteile und Nachteile:

- **Die Wissensstruktur entspricht der Ausdruckweise vieler Experten.**
  - Ein Experte kann die Wissenserwerbskomponente leicht bedienen.
  - Die Füllung der Wissensbasis kostet sehr viel Zeit.
- **Das Wissen ist sehr zielgerichtet abgelegt.**
  - Die Diagnose der Laufzeitkomponente ist schnell.
  - Die Wissensbasis kann nicht leicht geändert werden.
  - Wiederverwendbarkeit ist grundsätzlich ein Problem.
  - Für wiederverwendbare Wissensbasen gibt es aber Lösungsansätze.

# 1. Symptombasierte Diagnose

## Vorteile und Nachteile:

- **Die Wissensbasis enthält keine tiefere Struktur.**
  - Alle Anwendungsbereiche sind prinzipiell möglich.
  - Die Wissensbasis ist häufig unvollständig.
  - Die Wissensbasis ist unübersichtlich und kann daher schwer verifiziert werden.

↓  
*Viele Wissensbasen enthalten Fehler.*

## 2. Fallbasierte Diagnose

### Eingabe in die Wissensbasis:

- Fälle mit vollständigem Symptomvektor und zugehörigen Fehlern (eindeutig klassifiziert)

### a) KI-klassisch, mit Ähnlichkeitsmaß:

- Ähnlichkeitsmaß für unvollständige Symptomvektoren (gewichtet häufig zwischen den verschiedenen Symptomtypen)

### Struktur der Wissensbasis:

- Punkte im Vektorraum
- Ähnlichkeitsmetrik

### Arbeit der Problemlösungsmaschine:

- Finde zu gegebenem neuen Vektor den ähnlichsten Symptomvektor in der Wissensbasis.
- Weise dem neuen Fall dieselben Fehler zu, die dem Referenzvektor in der Wissensbasis zugeordnet sind (u.U. mit Angabe der Treffsicherheit).

## 2. Fallbasierte Diagnose

### Eingabe in die Wissensbasis:

- Fälle mit vollständigem Symptomvektor und zugehörigen Fehlern (eindeutig klassifiziert)

### b) mit Neuronalen Netzen

- Neuronales Netz mit Eingabeschicht (für Symptomvektor) und Ausgabeschicht (für Fehler) sowie (optional) Zwischenschicht aus Knoten und Kanten, jeweils mit veränderbaren Gewichten.

### Struktur der Wissensbasis:

- Punkte im Vektorraum
- Neuronales Netz mit konkret eingestellten Gewichten (abhängig von Symptomvektoren und zugehörigen Fehlern der Eingabe)

### Arbeit der Problemlösungsmaschine:

- Lege gegebenen neuen Symptomvektor an die Eingabeschicht des neuronalen Netzes an.
- Lies die zugeordneten Fehler aus der Ausgabeschicht ab.

## 2. Fallbasierte Diagnose

### Vorteile und Nachteile:

- **Das Verfahren ist einfach.**
  - Die Diagnose der Laufzeitkomponente ist schnell.
  - Der Wissenserwerb ist leicht automatisierbar.
  - Die Wissensbasis kann nur für Systeme erstellt werden, für die Erfahrungswissen vorhanden ist.
  - Die Wissensbasis nimmt sehr viel Speicherplatz ein (nur bei Ähnlichkeitsmetrik).

## 2. Fallbasierte Diagnose

### Vorteile und Nachteile:

- **Die Wissensbasis enthält überhaupt keine andere Strukturinformation als das Ähnlichkeitsmaß.**
  - Alle Anwendungsbereiche sind gleich geeignet.
  - Die Problemlösungsmaschine kann unverändert für alle Anwendungsbereiche eingesetzt werden.
  - Die Wissensbasis ist schon bei einer kleinen Änderung des Systems nicht mehr zuverlässig einsetzbar.
  - Das Ähnlichkeitsmaß bzw. Neuronale Netz ist häufig willkürlich.
  - Jede Laufzeitdiagnose kann falsch sein.
  - Die Diagnose ist überhaupt nicht nachvollziehbar (nur für Neuronale Netze).

# 3. Modellbasierte Diagnose

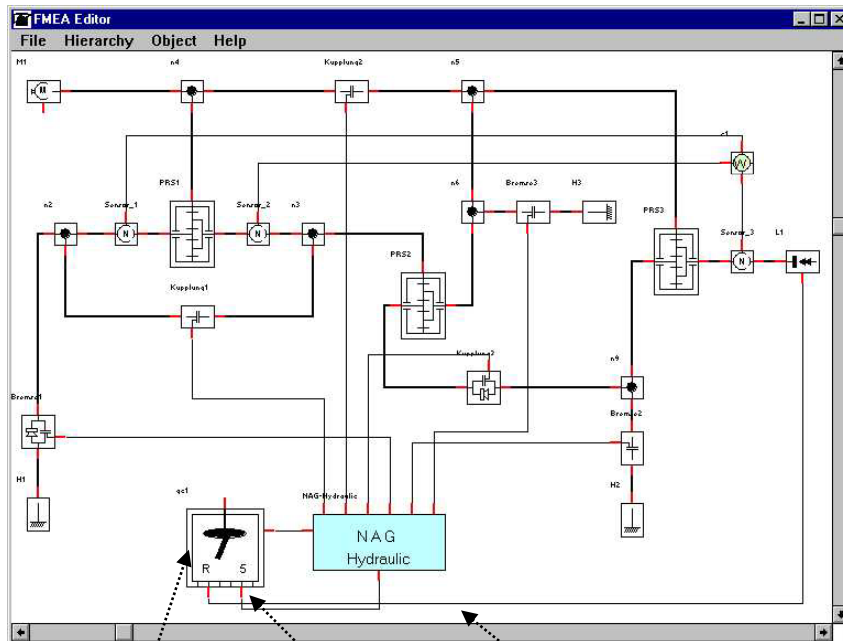
## Ziel:

- schneller Wissenserwerb
- exaktes und nachvollziehbares Ergebnis der Problemlösungsmaschine

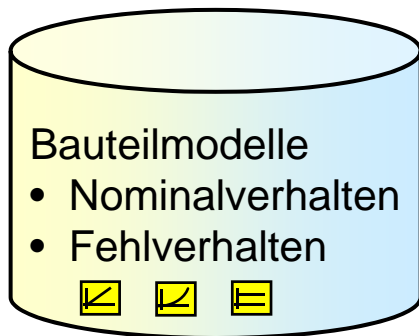
## Schwierigkeit:

- 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

# Vergleich der Wissensverarbeitungstechniken

## Klassifizierung wissensbasierten Schließens



# Vergleich der Wissensverarbeitungstechniken

	fallbasiert	symptombasiert	modellbasiert
schnell zur Laufzeit	<b>++</b>	<b>++</b>	<b>o</b>
schneller Wissenserwerb	<b>++</b>	<b>o</b>	<b>+</b>
steht zur Produkteinführung zur Verfügung	<b>a) -- b) -</b>	<b>o</b>	<b>++</b>
Aussagekraft der Diagnosen	<b>a) o b) --</b>	<b>+</b>	<b>++</b>