

Semantic Web

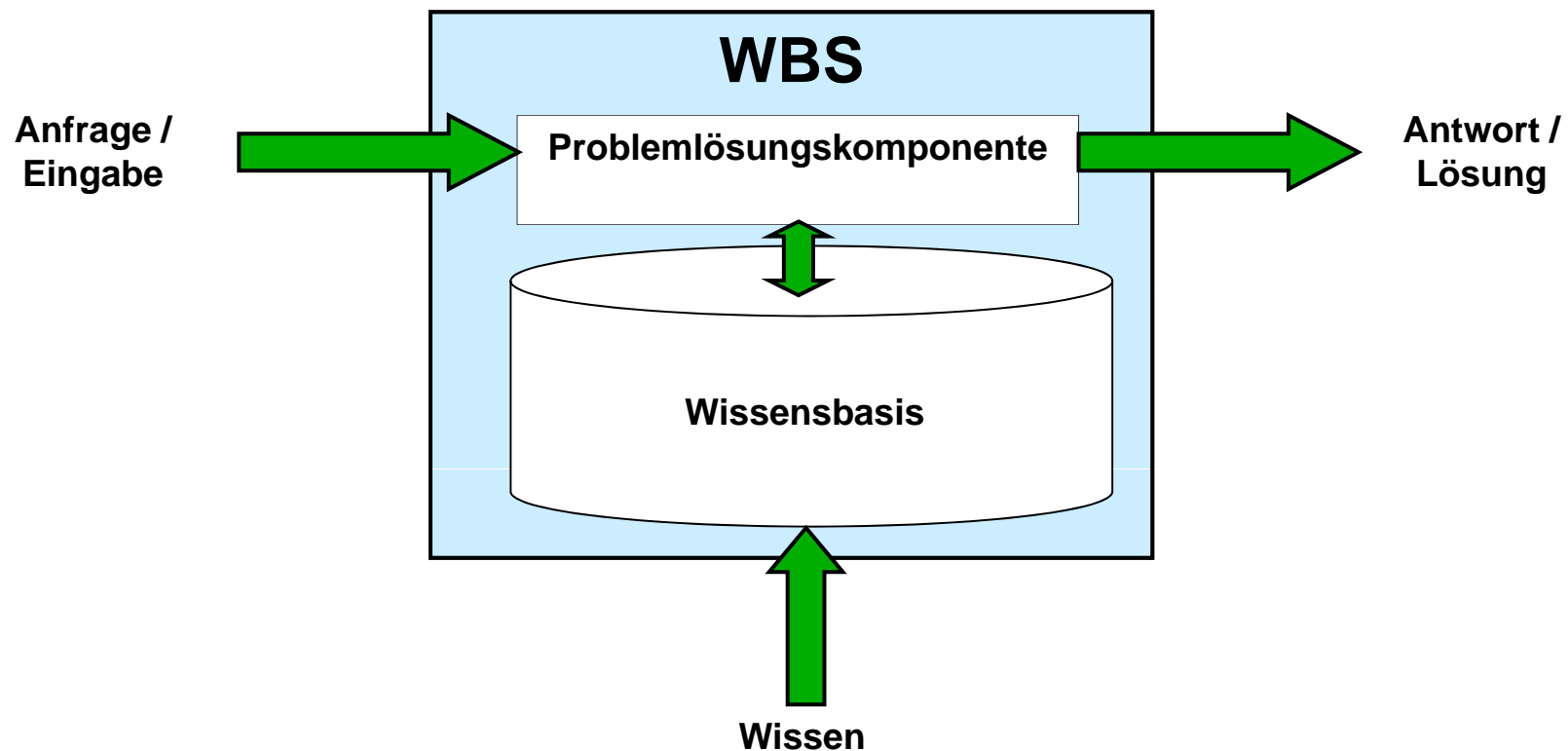
Sebastian Iwanowski
FH Wedel

Exkurs:
Semantische Wissensverarbeitung: Techniken aus der KI

Basistechnologie: Wissensbasiertes System



Architektur WBS (allgemeinste Form)



Repräsentation von Wissen: **Wie ?**



Logisches Wissen:

Atome **Regeln** **Ableitungsregeln**
Fakten wenn ... dann ... Theorembeweistechniken

Funktionales Wissen:

Daten **Funktionen** **Funktionsauswertung**

Objektorientiertes Wissen:

Objekte **Methoden** **Compiler / Interpreter**

deklaratives **prozedurales** **Steuerungs-** **Wissen**

Klassifizierung von Wissen: Was ?

Die folgenden Kriterien sind unabhängig voneinander:

- **tief vs. flach** (betrachte die Zusammensetzung einer Aussage aus kleineren Einheiten)
modellbasiert vs. allgemeingültig
- **sicher vs. unsicher** (betrachte die Wahrscheinlichkeit einer Aussage)
deterministisch vs. probabilistisch
- **exakt vs. unscharf** (betrachte die Genauigkeit einer Aussage)
quantitativ vs. qualitativ

Klassifizierung von Wissen: Was ?

Repräsentation von zeitlichem Wissen

Allens Intervalllogik für die qualitative Repräsentation von Zeitintervallen

1. STARTS(t_1, t_2)

t_1 hat denselben Anfang wie t_2 , endet aber vor dem Ende von t_2

2. FINISHES(t_1, t_2)

*t_1 hat dasselbe Ende wie t_2 , beginnt aber **nach** dem Anfang von t_2*

3. DURING(t_1, t_2)

t_1 ist vollständig in t_2 enthalten

4. BEFORE(t_1, t_2)

t_1 ist vor t_2 und t_1 und t_2 überlappen sich nicht

5. OVERLAP(t_1, t_2)

t_1 beginnt vor t_2 und endet nach Anfang von t_2

6. MEETS(t_1, t_2)

t_1 ist vor t_2 und es gibt kein Intervall zwischen t_1 und t_2 , d.h. t_1 endet, wenn t_2 beginnt

7. EQUAL(t_1, t_2)

t_1 und t_2 sind dasselbe Intervall

aus Handbuch der KI, Wissen über Raum und Zeit, S. 367

Klassifizierung von Wissen: Was ?

Repräsentation von räumlichen Wissen

Exaktes Wissen

- Geokoordinaten
- relative Entfernung von Gegenständen in m

Qualitatives Wissen

- relative Himmelsrichtung (im Westen von, ...)
- relative Anordnung (vor, hinter, links von, rechts von, etc.)

Klassifizierung von Wissen: Was ?

Praktisches Problem für zeitliches und räumliches Wissen:

Wie exakt soll das Wissen sein ?

- Jahr, Monat, Tag, Stunde, Sekunde, Millisekunde, ...
- Land, Stadt, Adresse, exakte Geokoordinaten, ...

Anwendungsbeispiel Technische Diagnose

Laufzeitsystem:

(heißt in wissensbasierten Systemen **Problemlösungskomponente / 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
- modellbasiert

} als Alternativen

2) Struktur der Wissensbasis

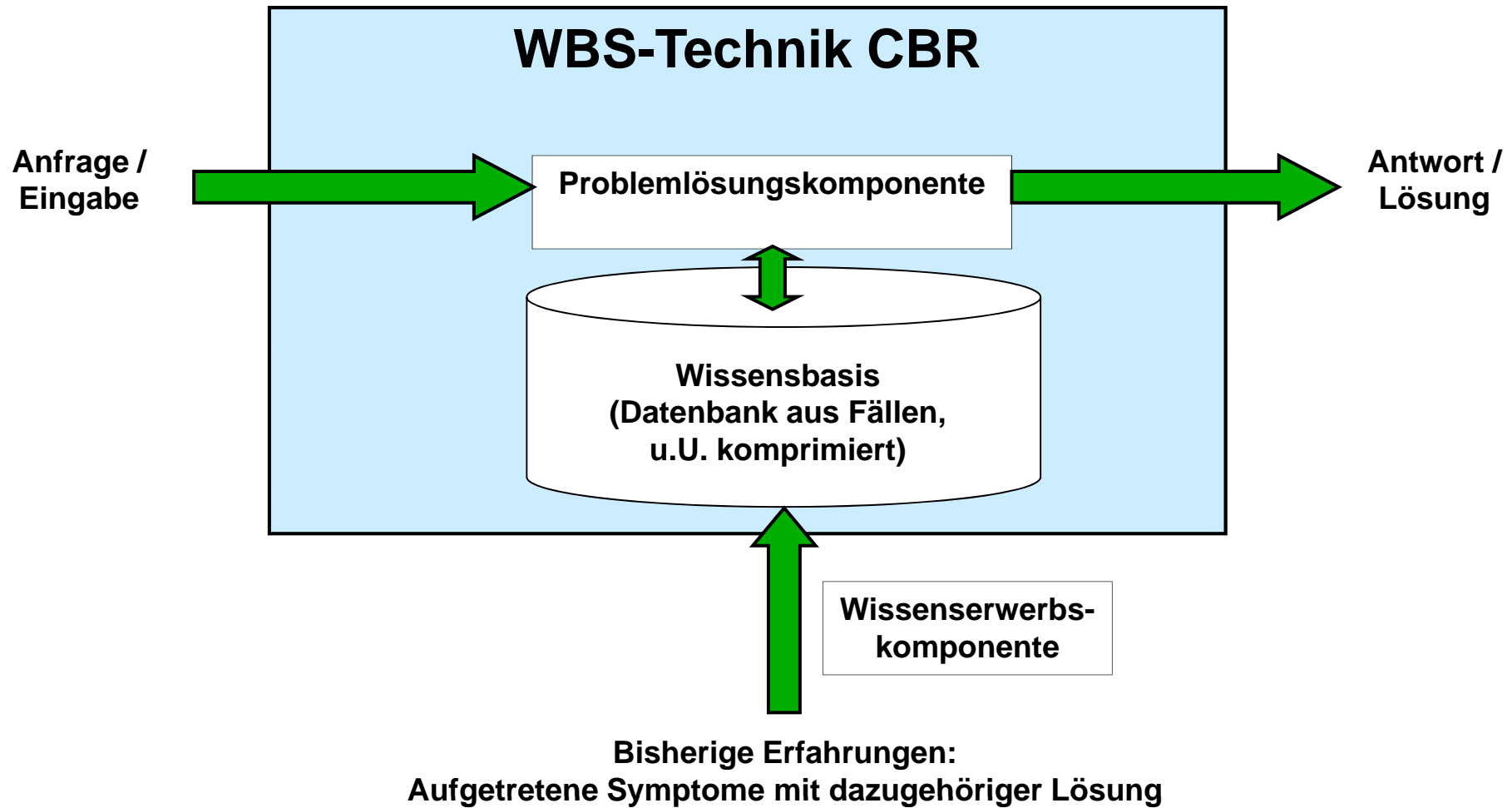
- abhängig von Wissenserwerb

3) Verarbeitung des Wissens in der Problemlösungskomponente

- abhängig von Struktur der Wissensbasis

Darin unterscheiden sich verschiedene Diagnosesysteme voneinander !

1. Fallbasiertes Schließen (CBR)



1. Fallbasiertes Schließen (CBR)

Eingabe in die Wissensbasis:

- Fälle mit Eingabevektor und zugehörigem Ausgabevektor (Referenzvektoren)

a) KI-klassisch, mit Ähnlichkeitsmaß:

- Ähnlichkeitsmaß für Vektoren (gewichtet häufig zwischen den verschiedenen Komponenten)

Struktur der Wissensbasis:

- Punkte im Vektorraum (Referenzvektoren)
- Ähnlichkeitsmetrik

Arbeit der Problemlösungskomponente:

- Finde zum neuen Eingabevektor den ähnlichsten Referenzvektor in Wissensbasis.
- Weise dem neuen Eingabevektor denselben Ausgabevektor zu, der dem Referenzvektor in der Wissensbasis zugeordnet ist (u.U. mit Angabe der Treffsicherheit).

1. Fallbasiertes Schließen (CBR)

Eingabe in die Wissensbasis:

- Fälle mit Eingabevektor und zugehörigem Ausgabevektor (Referenzvektoren)

b) mit Neuronalen Netzen:

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

Struktur der Wissensbasis:

- Neuronales Netz mit konkret eingestellten Gewichten (trainiert durch die Referenzvektoren)

Arbeit der Problemlösungskomponente:

- Lege neuen Eingabevektor an die Eingabeschicht des neuronalen Netzes an.
- Lies den zugeordneten Ausgabevektor aus der Ausgabeschicht ab.

1. Fallbasiertes Schließen (CBR)

Verallgemeinerung auf beliebige fallbasierte Auswertungsstrategien:

Prinzip:

- Gegeben seien Fälle in Form von Vektoren (*vollständige Referenzvektoren*): Diese werden “gelernt” und bilden die Wissensbasis.
- Gegeben seien neue Vektoren, in denen nicht alle Komponenten bekannt sind: Diese sind zu klassifizieren.
- Ordne den unbekanntem Komponenten der neuen Vektoren Werte zu.

Arbeit des Problemlösers (einfache Variante):

- Finde zu gegebenem neuen Vektor den “ähnlichsten” Symptomvektor in der Wissensbasis.
- Weise den unbekanntem Komponenten des neuen Vektors dieselben Werte zu, die den entsprechenden Komponenten des Referenzvektors in der Wissensbasis zugeordnet sind.

Dieses Verfahren ist nur sinnvoll, wenn die unbekanntem Werte aus einem diskreten (besser: endlichen) Wertebereich kommen !

1. Fallbasiertes Schließen (CBR)

Verbesserung für kontinuierliche Wertebereiche:

Arbeit des Problemlösers (bessere Variante):

- Weise den unbekannt Komponenten des neuen Vektors Werte zu, die zwischen den Werten der entsprechenden Komponenten von "in der Nähe liegenden" Vektoren der Wissensbasis liegen.

Andere (mathematische) Formulierung dieser Methode:

- Fasse die unbekannt Werte der neuen Vektoren als Funktionswerte der bekannten auf: Finde stetige Funktion, von der alle Referenzvektoren der Wissensbasis eine Lösung sind.
- Weise den unbekannt Komponenten der neuen Vektoren die Funktionswerte der stetigen Funktion zu.

Frage: Wie findet man die Funktion zu einer gegebenen Menge von Referenzvektoren ?

Antwort:

- Gib eine Funktionsklasse vor, innerhalb der sich die Funktionen durch Parameter unterscheiden.
- Bestimme die Parameter als Lösung eines Gleichungssystems aus den bekannten Referenzvektoren.

1. Fallbasiertes Schließen (CBR)

Parameterbestimmung in Funktionsklassen (Regression):

Lineare Regression:

- Finde die Gewichte in linearen Funktionsgleichungen der Form $f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n W_i x_i$

Verallgemeinerung:

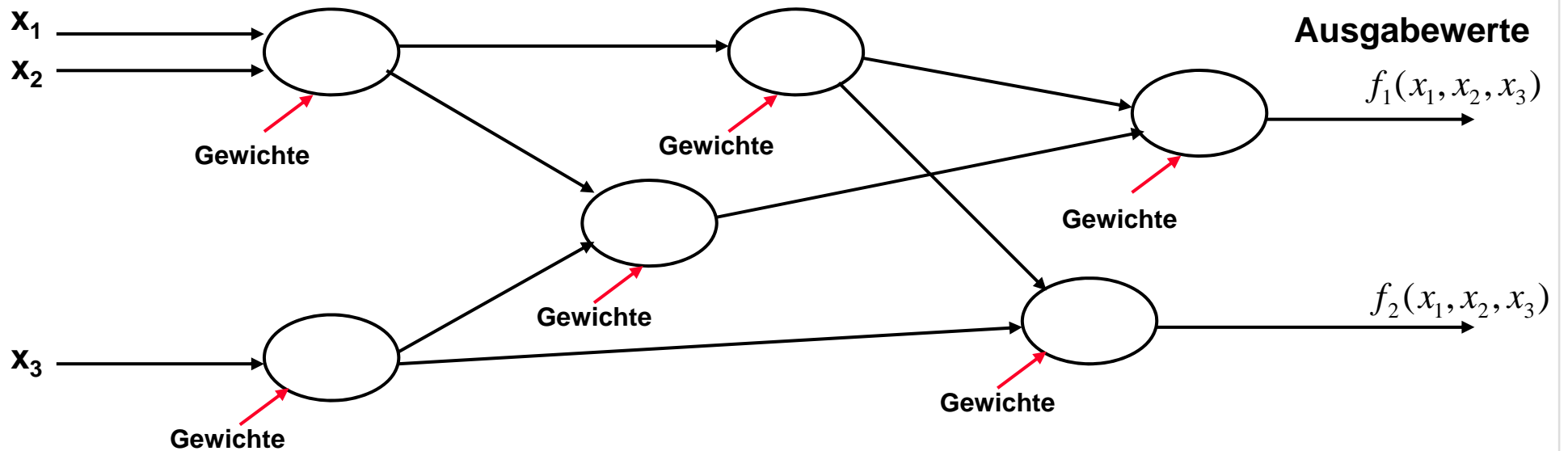
1. Finde die Gewichte in einem linearen Gleichungssystem.
 2. Finde die Gewichte in Gleichungssystemen höherer Ordnung.
 3. Finde die Gewichte in parametrisierten Ungleichungssystemen.
- **Der fallbasierte Ansatz ist gerade für Systeme gedacht, die man *nicht* leicht modellieren kann.**
 - **Daher bringt der Ansatz mit höheren Gleichungssystemen nicht viel.**
 - **Besser ist es, mit vielen nur lose miteinander gekoppelten Gleichungssystemen zu arbeiten und das unsichere Wissen zu verteilen.**

1. Fallbasiertes Schließen (CBR)

Idee der neuronalen Netze:

Gegeben sei eine mehrdimensionale Funktion f (Schreibweise: $f_i(x_1, x_2, \dots, x_n)$)

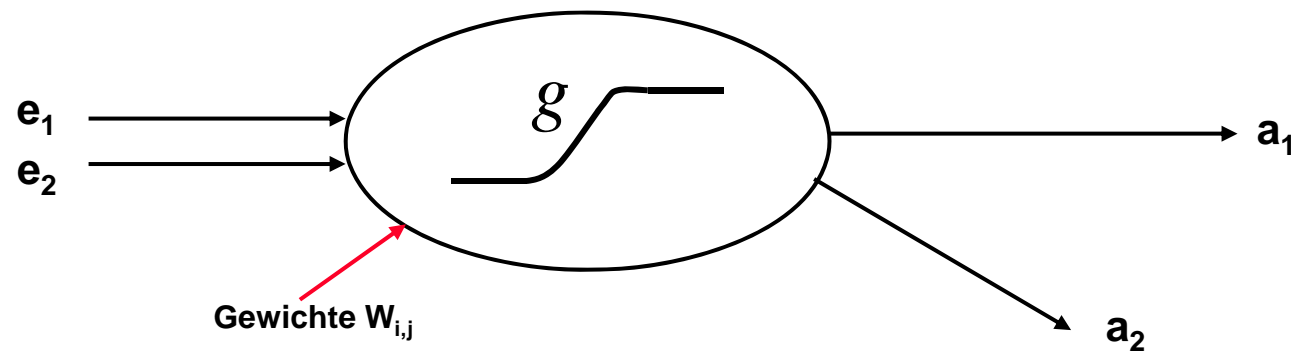
Eingabewerte



- Die Gewichte können voreingestellt werden, werden aber an die gelernten Beispiele angepasst.
- Neue Funktionswerte werden durch Durchlauf des Netzes ausgerechnet.

1. Fallbasiertes Schließen (CBR)

Funktionsweise eines einzelnen Neurons:



$$a_i(e_1, e_2, \dots, e_n) = g \left(\sum_{j=1}^n W_{i,j} e_j \right)$$

- g ist eine verallgemeinerte Schwellenwertfunktion, die für alle Ausgaben desselben Neurons gleich ist.

1. Fallbasiertes Schließen (CBR)

Verschiedene Stufen neuronaler Netze:

Neuronale Netze ohne Zwischenschichten:

- Es gibt Neuronen in einer ersten Schicht, die mit der Eingabe verbunden sind und das Ergebnis an Neuronen einer zweiten Schicht, der Ausgabeschicht leiten, welche mit der Ausgabe verbunden sind.

Neuronale Netze mit Zwischenschichten

- Eingabe- und Ausgabeschicht sind über weitere “verborgene Schichten von Neuronen verbunden.

Neuronale Netze mit Rückkopplung:

- Ausbildung eines “Gedächtnisses”

1. Fallbasiertes Schließen (CBR)

Was ist der entscheidende Unterschied zwischen neuronalen Netzen und „klassischen“ fallbasierten Systemen ?

- **In neuronalen Netzen ist das Wissen über die gelernten Fälle **verteilt**.**

Theoretische Vorteile der Verteilung:

- Willkürlichkeit des Funktionsklassenansatzes spielt nicht so eine große Rolle
- Undurchschaubare Fälle bekommen einen undurchschaubaren Ansatz:
Der verteilte Ansatz “reguliert sich selbst”.

In der Praxis hat sich gezeigt:

- Für gut funktionierende neuronale Netze benötigt man weniger Lernbeispiele als in klassischen fallbasierten Systemen.
- Neuronale Netze liefern bessere Resultate bei der Klassifizierung.

Zusammenfassung: Fallbasiertes Schließen

Vorteile und Nachteile:

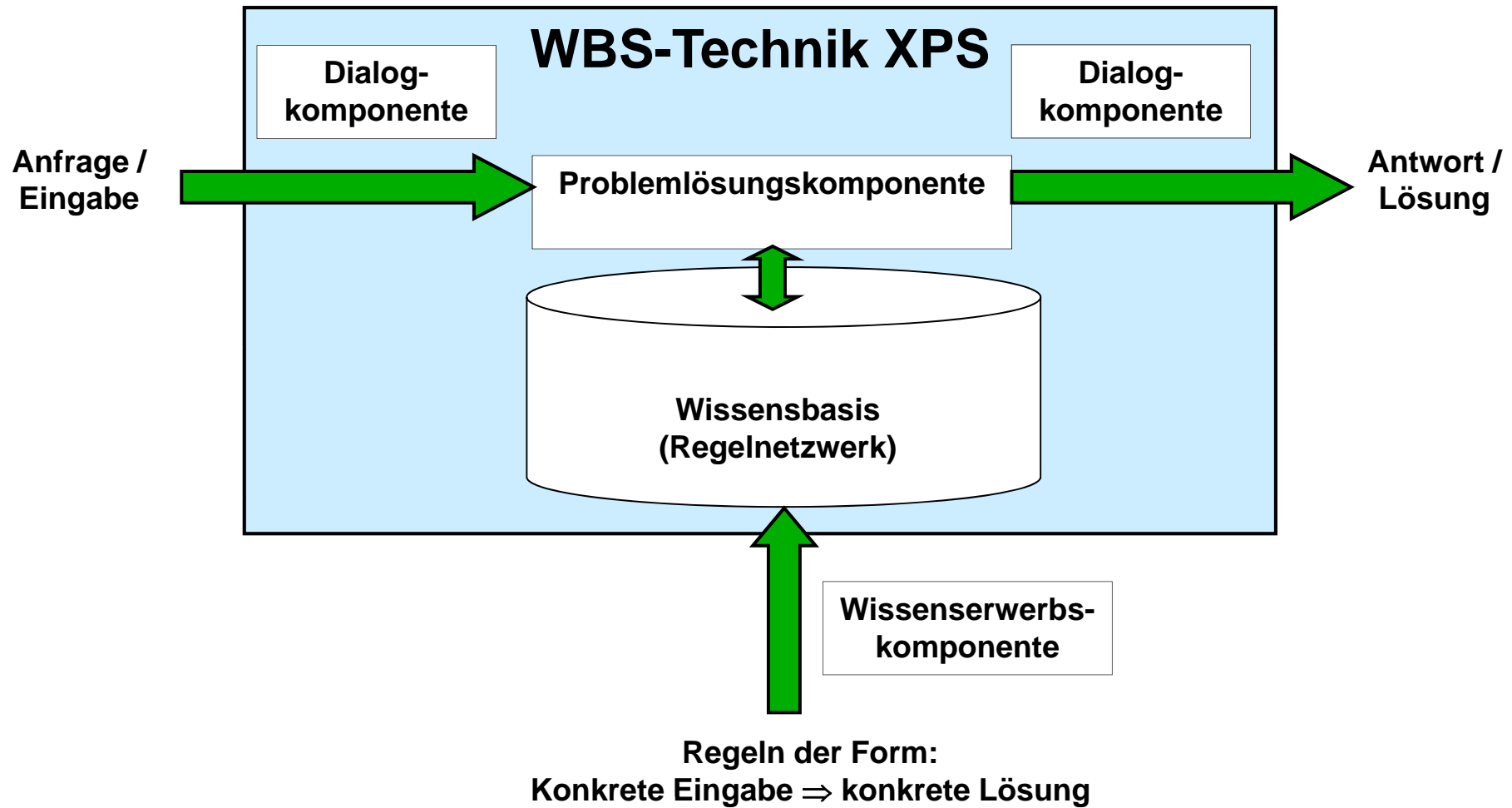
- **Das Verfahren ist einfach.**
 - Die Problemlösungskomponente arbeitet sehr schnell.
 - Der Wissenserwerb ist leicht automatisierbar.
 - Die Wissensbasis kann nur für Systeme erstellt werden, für die Erfahrungswissen vorhanden ist.
 - Die Wissensbasis nimmt sehr viel Speicherplatz ein (nur bei Ähnlichkeitsmetrik).

Zusammenfassung: Fallbasiertes Schließen

Vorteile und Nachteile:

- **Die Wissensbasis enthält überhaupt keine andere Strukturinformation als das Ähnlichkeitsmaß bzw. das NN.**
 - Alle Anwendungsbereiche sind gleich geeignet.
 - Die Problemlösungskomponente 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 Lösung der Problemlösungskomponente kann falsch sein.
 - Die Lösung ist überhaupt nicht nachvollziehbar (nur für Neuronale Netze).

2. Regelbasiertes Schließen



2. Regelbasiertes Schließen

Eingabe in die Wissensbasis (Bsp. Technische Diagnose):

- 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.

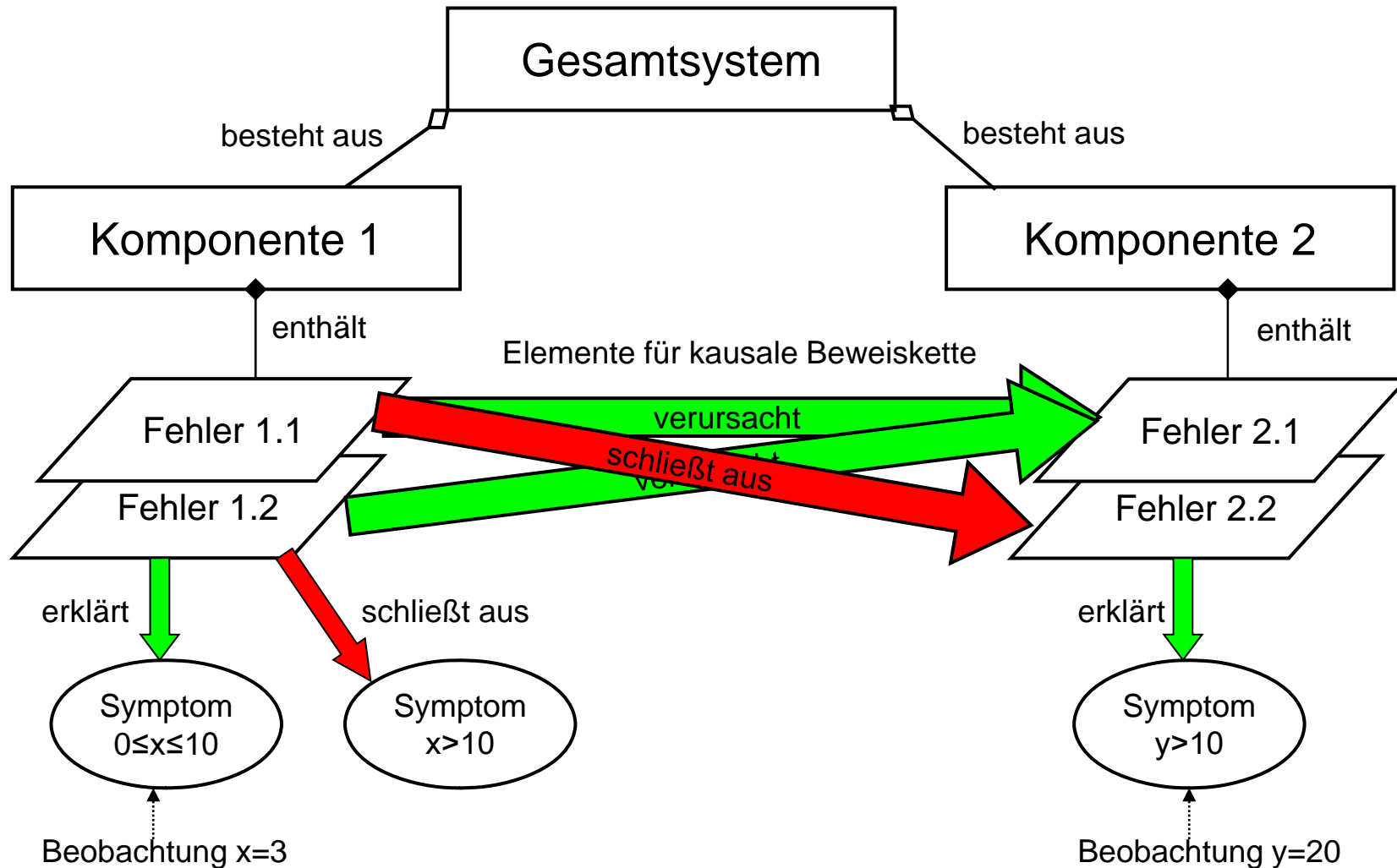
Struktur der Wissensbasis:

- Semantisches Netz
- Mögliche Strukturen:
 - Fehlernetzwerke
 - Entscheidungsbäume

Das ist die „klassische Expertensystemtechnik“

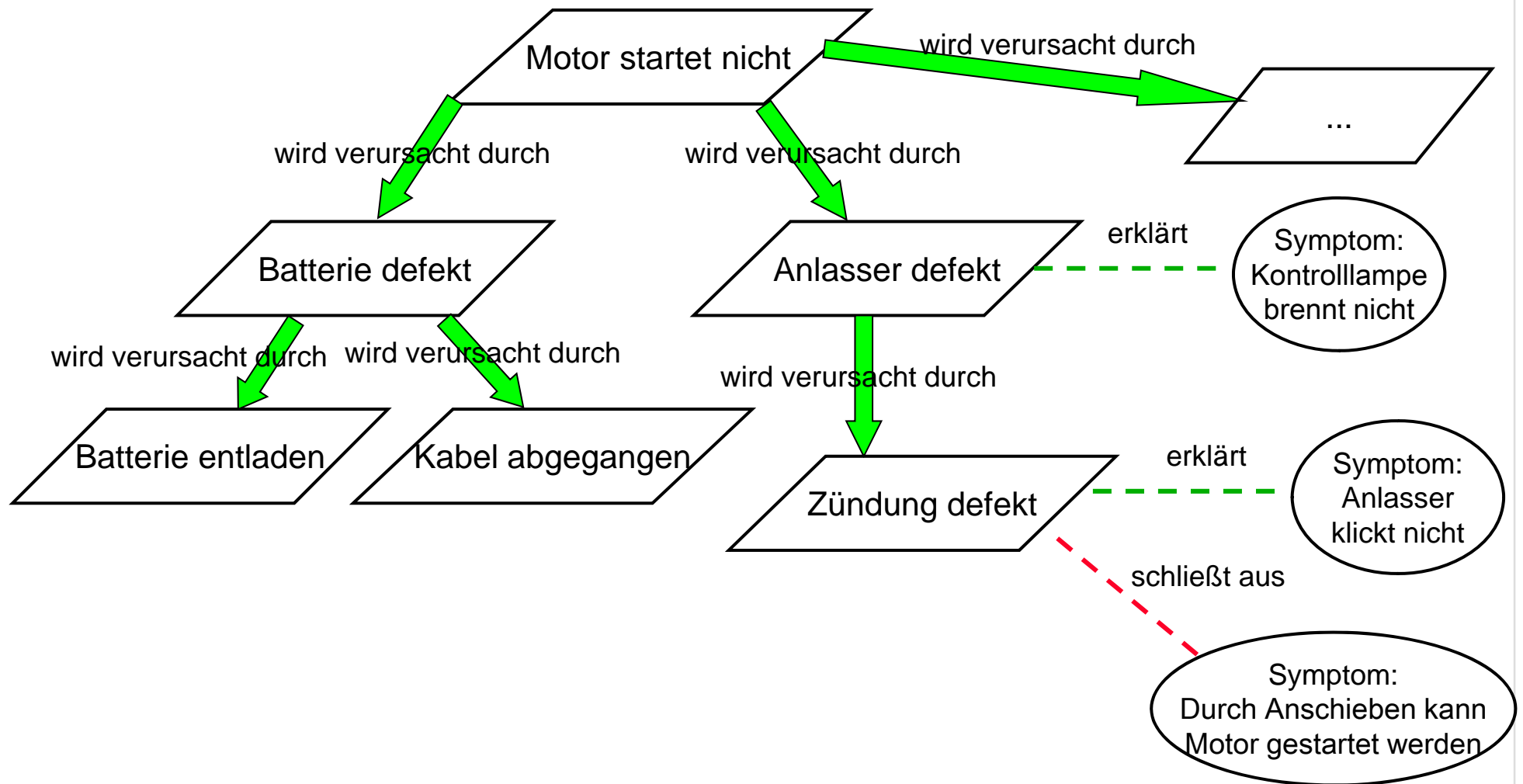
2. Regelbasiertes Schließen

Beispiel für die Elemente einer Wissensbasis:



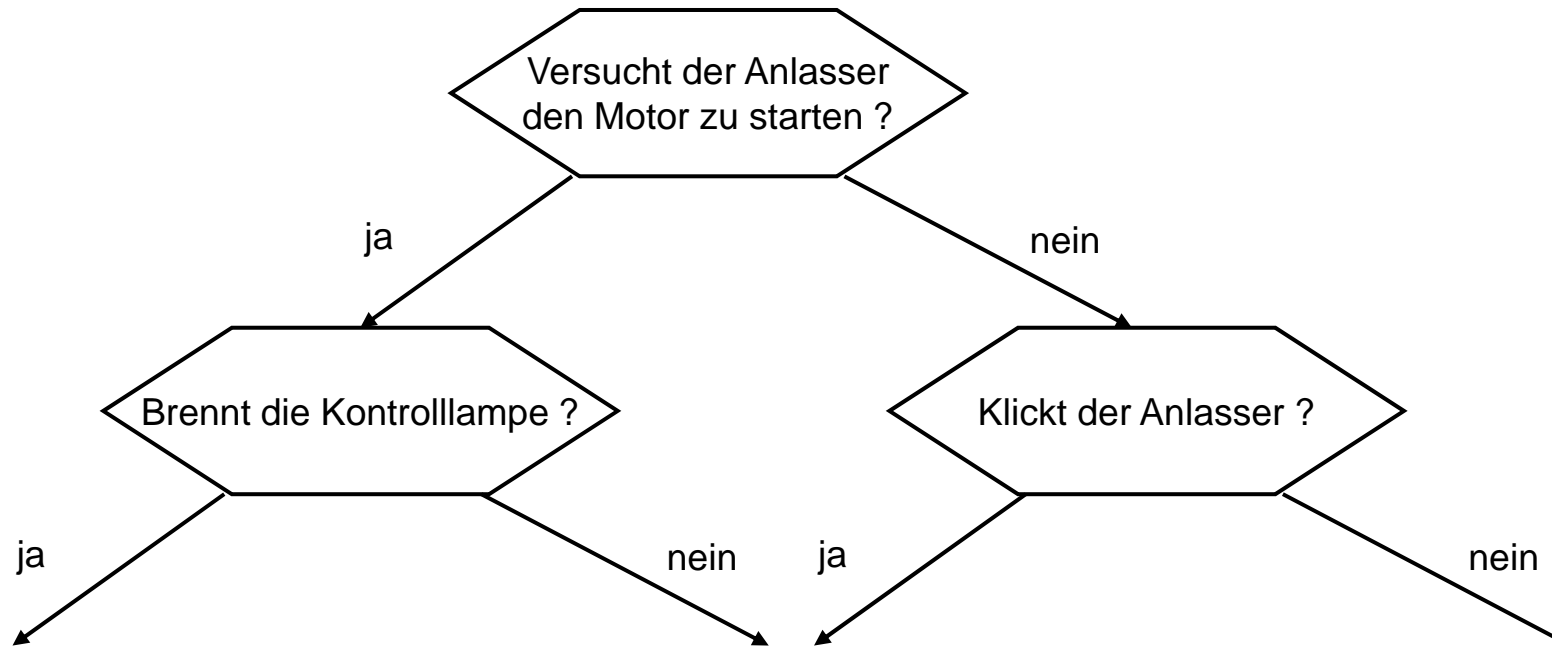
2. Regelbasiertes Schließen

Beispiel für einen Fehlerbaum:



2. Regelbasiertes Schließen

Beispiel für einen Entscheidungsbaum:



2. Regelbasiertes Schließen

Arbeit der Problemlösungskomponente:

- **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.

2. Regelbasiertes Schließen

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 Problemlösungskomponente arbeitet schnell.
 - Die Wissensbasis kann nicht leicht geändert werden.
 - Wiederverwendbarkeit ist grundsätzlich ein Problem.
 - Für wiederverwendbare Wissensbasen gibt es aber Lösungsansätze.

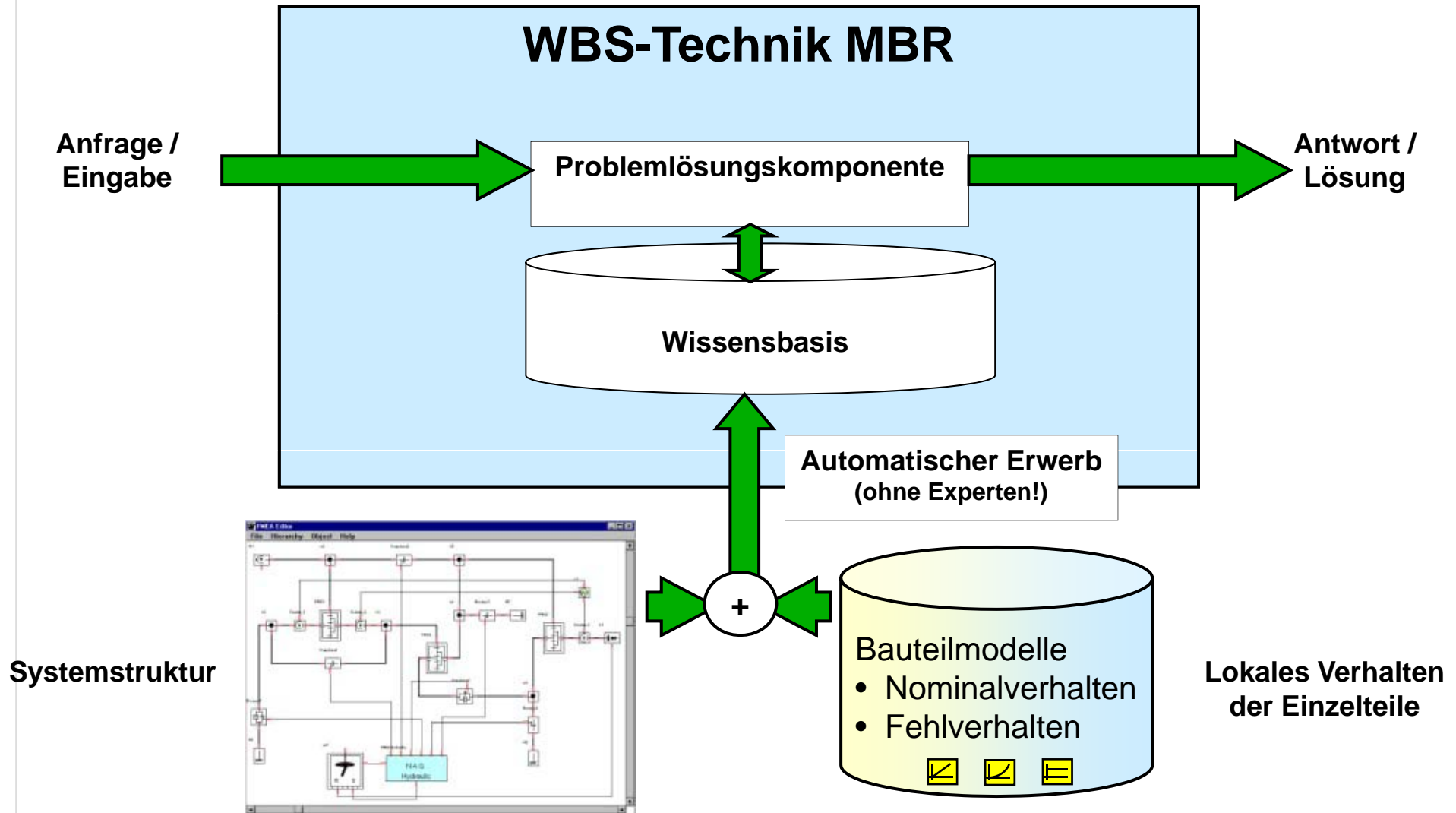
2. Regelbasiertes Schließen

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.

3. Modellbasiertes Schließen



3. Modellbasiertes Schließen

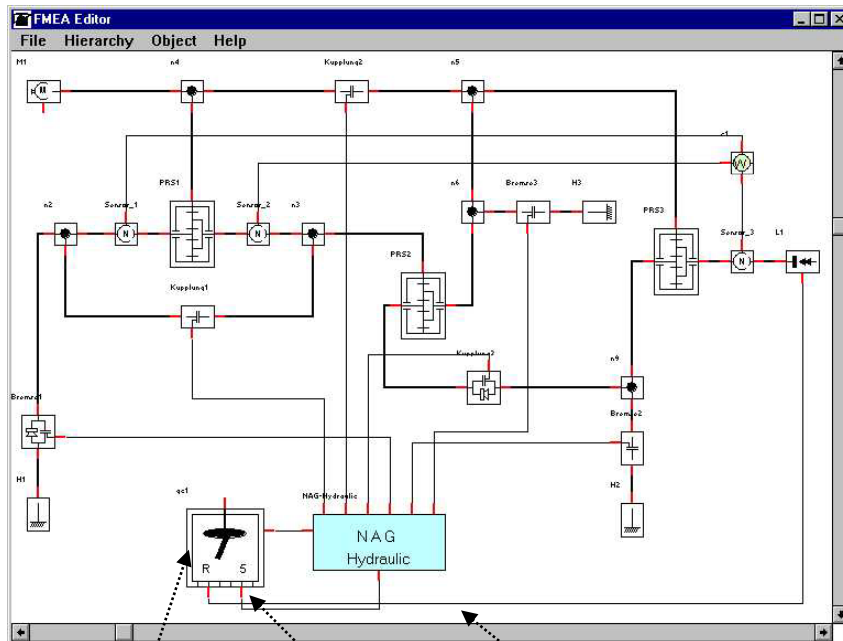
Ziel:

- schneller Wissenserwerb
- exaktes und nachvollziehbares Ergebnis der Problemlösungskomponente

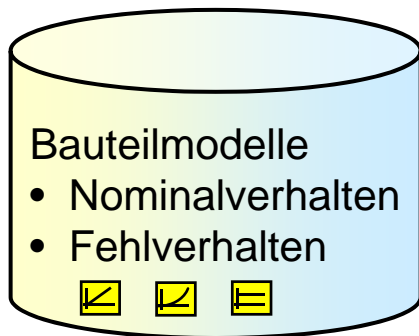
Schwierigkeit:

- schnelle Antwort der Problemlösungskomponente zur Laufzeit

3. Modellbasiertes Schließen



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. Modellbasiertes Schließen

Eingabe in die Wissensbasis:

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

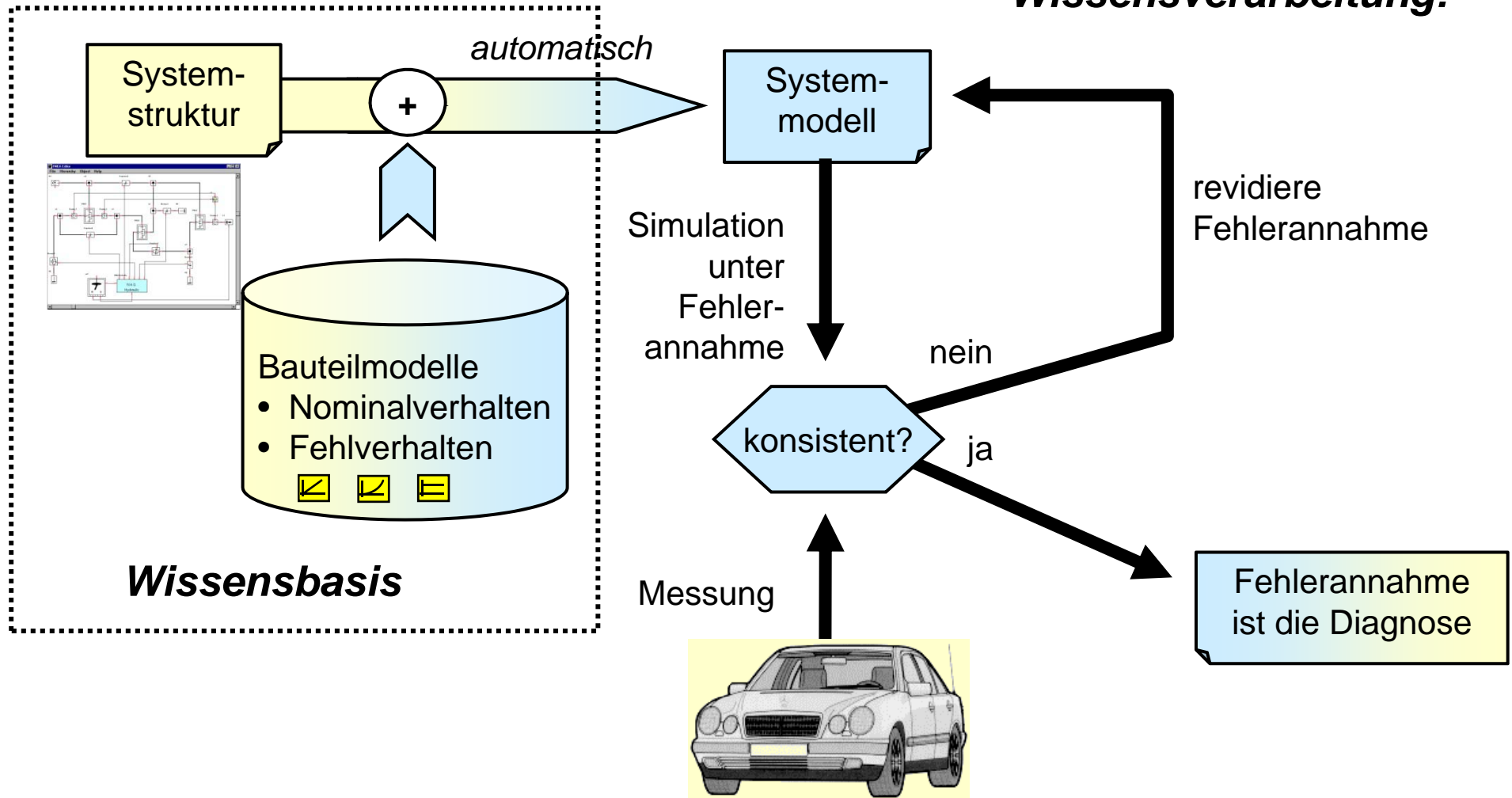
Struktur der Wissensbasis:

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

3. Modellbasiertes Schließen

Basisfunktionalität der modellbasierten Diagnose

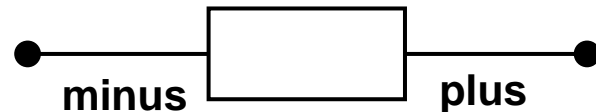
Wissensverarbeitung:



3. Modellbasiertes Schließen

Modellierung der elektrischen Komponenten:

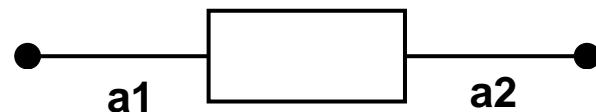
Batterie:



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

Regeln:
 $\text{ok} \Rightarrow (\text{minus} = \text{Masse})$
 $\text{ok} \Rightarrow (\text{plus} = \text{Versorgungsspannung})$
 $\text{defekt} \Rightarrow (\text{minus} = \text{Masse})$

Kabel:

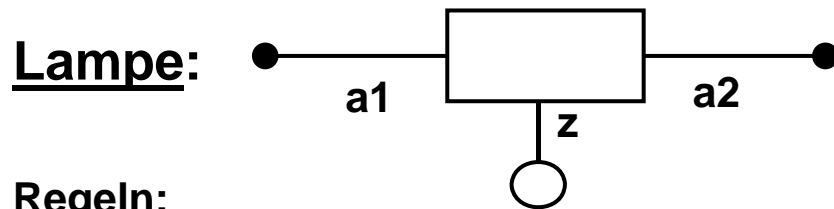


Wertebereiche: $\text{a1, a2} \in \{ \text{Masse, Versorgungsspannung} \}$

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

3. Modellbasiertes Schließen

Modellierung der elektrischen Komponenten:



Wertebereiche:

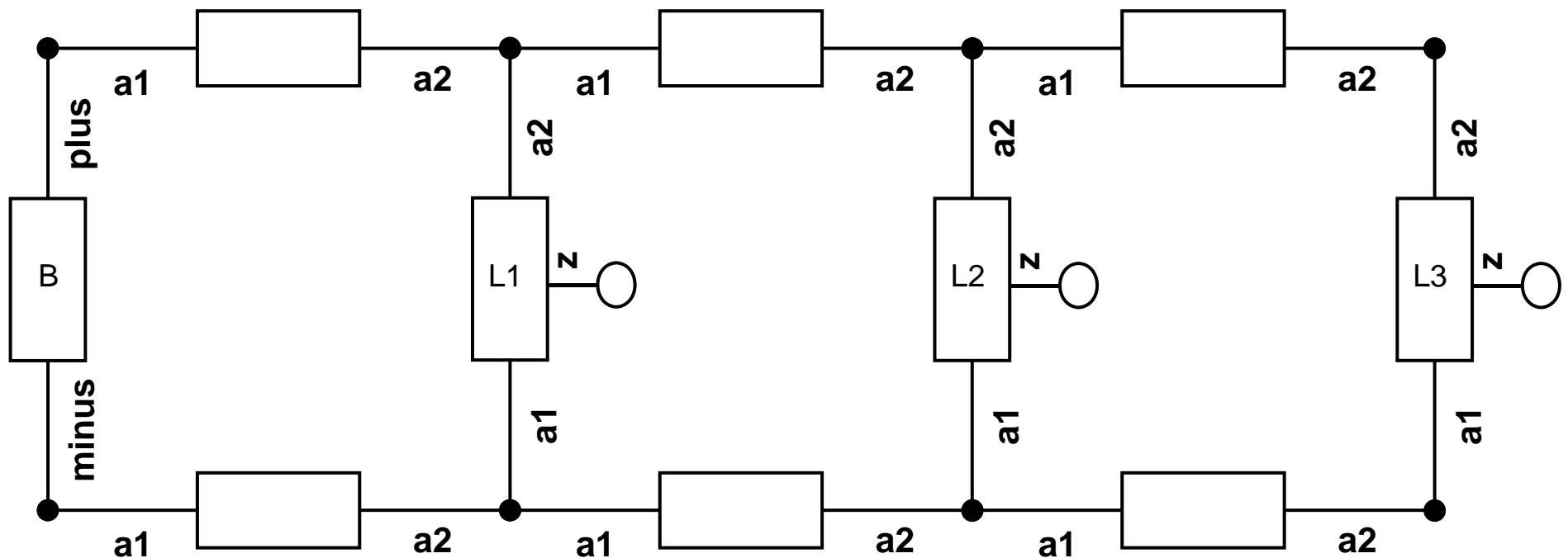
$a1, a2 \in \{ \text{Masse, Versorgungsspannung} \}$
 $z \in \{ \text{hell, dunkel} \}$

Regeln:

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

3. Modellbasiertes Schließen

Zusammenbau des Systemmodells aus den Komponentenmodellen:



Werte an den Verbindungsknoten müssen gleich sein

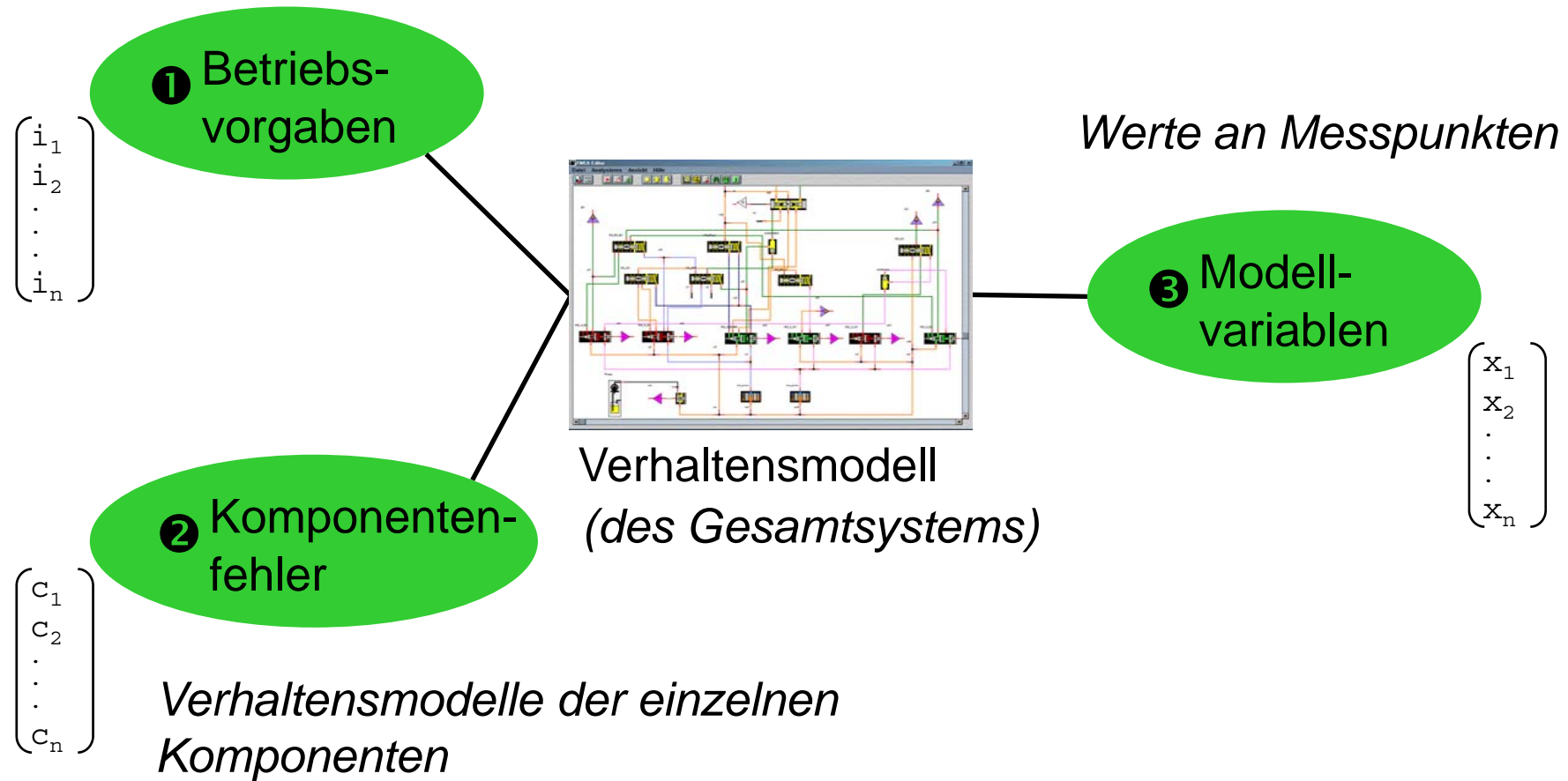
Bei Widerspruch: Konflikt der den Werten zugrunde liegenden Verhaltensmodelle

Diagnosen sind Mengen von Verhaltensmodellen, die keinen Konflikt enthalten

3. Modellbasiertes Schließen

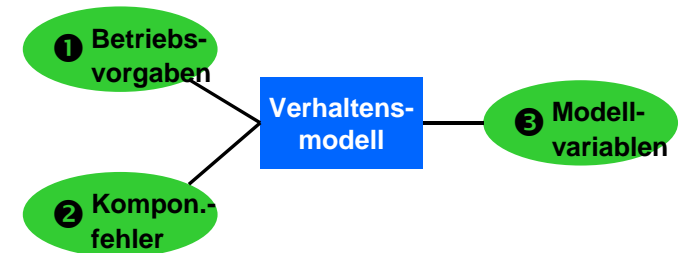
MBR in der Prozesskette

Situationen (Werte an Control Inputs)



3. Modellbasiertes Schließen

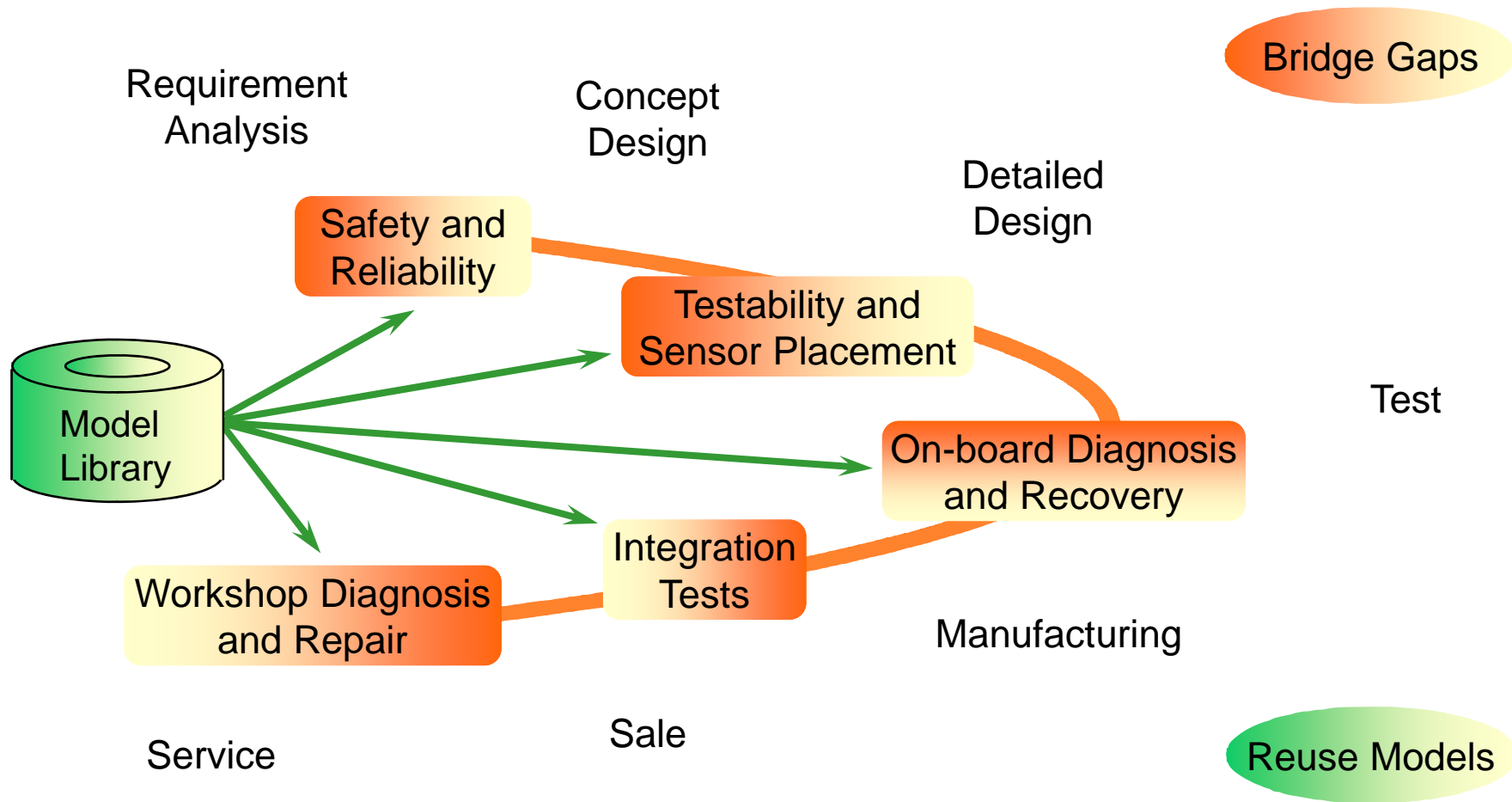
MBR in der Prozesskette



A		FMEA	1 vorgeben, 2 vorgeben, 3 = sicherheitsrelevante Eigenschaften klassifizieren (FMEA Events)
B		FTA	3 sicherheitsrelevante Eigenschaften vorgeben, nach 1 und 2 suchen
C		Testbarkeitsanalyse	1 vorgeben, 2 vorgeben, 3 = vorhandene Sensoren bzw. Meßpunkte (Entscheidungsbaum generieren - eigenst. Fehlerkl.)
D		Sensorplazierung	1 vorgeben, 2 vorgeben, 3 = alle potenziell mögliche Sensoren / Meßpunkte (Entscheidungsbaum generieren - Sensorplazierung)
E		Diagnose	1 vorgeben, 3 messen, nach 2 suchen = Komponentenfehler
F		Recovery	2 vorgeben, 3 Ziele vorgeben, nach 1 suchen = mögliche Kompensationssteuerung

3. Modellbasiertes Schließen

MBR in der Prozesskette



3. Modellbasiertes Schließen

Zusammenfassung: Was kann MDS ?

System Design

Model-based Failure Analysis

Analysis Task

▶ *Result*

Reliability/Safety

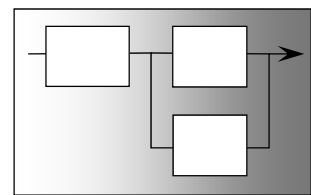
▶ *FMEA/FTA*

Testability

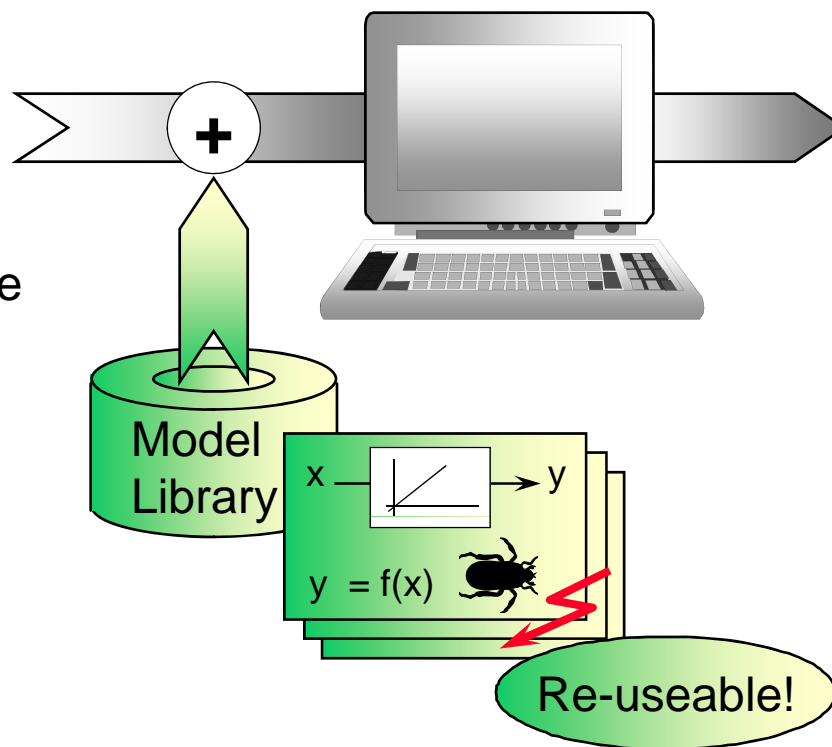
- ▶ *Sensor Placement*
- ▶ *Fault Classes*

Diagnosis

- ▶ *Decision Trees &*
- ▶ *Interactive Troubleshooting*



System Structure



Klassifizierung der Wissensverarbeitungstechniken

Klassifizierung wissensbasierten Schließens nach Richtung

- **heuristisch:**

wenn <Merkmalskonstellation> dann <Lösung>

- **kausal:**

- überdeckende Klassifikation:

wenn <Lösung> dann <Merkmale>

- funktionale Klassifikation:

komponentenbasiertes Verhaltensmodell =>
Systemfunktion

(suche das beste Verhaltensmodell, das mit der beobachteten
Systemfunktion konsistent ist)

Klassifizierung der Wissensverarbeitungstechniken

Klassifizierung wissensbasierten Schließens nach Richtung

- **fallbasiert:**

Gegeben Fälle von **Merkmalskonstellation mit Lösung**

Wende dann Regressionstechnik an (**Interpolation**)

- mit Ähnlichkeitsmaß:

Beliebige Regression

- mit neuronalen Netzen

Verteilte lineare Regression

- mit Data Mining:

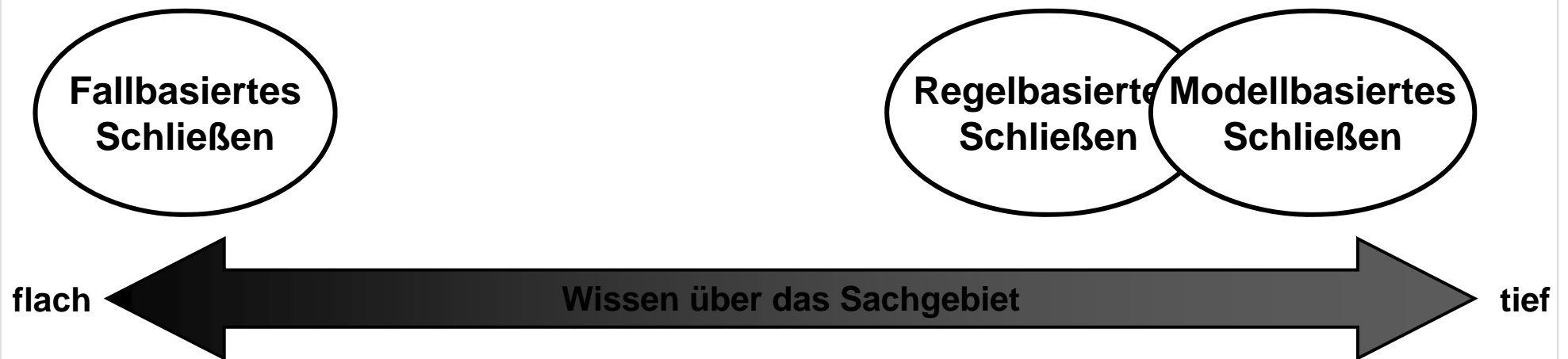
Merkmale aus Fallbasis => neue Zusammenhänge

Wende zusätzlich eine der anderen Methoden an (heuristisch oder kausal)

Klassifizierung der Wissensverarbeitungstechniken

Klassifizierung wissensbasierten Schließens nach Tiefe

- heuristisch *für relativ flaches Wissen*
- kausal *für flaches oder tiefes Wissen*
- fallbasiert (Ähnlichkeitsmaß, Neuronale Netze, Data Mining) *für sehr flaches Wissen*



Das ist im Prinzip beliebig kombinierbar mit den anderen Dimensionen von Wissensqualität:

- sicher vs. unsicher (betrachte die Wahrscheinlichkeit einer Aussage)
- exakt vs. unscharf (betrachte die Genauigkeit einer Aussage)

Abschließender Vergleich für die Praxis

	regelbasiert	fallbasiert	modellbasiert
schnell zur Laufzeit	++	++	o
schneller Wissenserwerb	o	++	+
geeignet für komplex verbundene Systeme	--	++	++
geeignet für komplizierte Komponenten	+	++	--
Wiederverwendbarkeit des Wissens	o	--	++
geeignet zur Entdeckung bisher unbekannter Zusammenhänge	--	a) -- b) o	+
steht zur Produkteinführung zur Verfügung	o	a) -- b) -	++
Aussagekraft der Lösungen	+	a) o b) --	++