Kapitel 10:
Wie passen die Teile zusammen?

- Diagnosesysteme -
- Bau von XPS -
- Schluß -

Einiges Hintergrundmaterial

- Michael M. Richter: Prinzipien der Künstlichen Intelligenz (2. Auflage), Teubner Verlag, Stuttgart [diesem Buch folgt die vorliegende Darstellung zum Diagnosevorgehen]
- Tilo Pfeifer / Michael M. Richter (Hrsg.): Diagnose von technischen Systemen, Deutscher Universitäts-Verlag [hieraus sind die Beispiele aus dem Bereich Diagnose von CNC-Maschinen entnommen]
- Das Buch von Puppe gibt viel Drumherum zur Vorgehensweise und Grundprinzipien der Durchführung von XPS-Projekten.
- Unter http://d3.informatik.uni-wuerzburg.de/d3.shtml wird die XPS-Diagnose-Shell D3 aus der Arbeitsgruppe von Prof. Puppe beschrieben, zu deren Nutzung auch ein Buch im Springer-Verlag erschienen ist

Bau von Expertensystemen heißt ...

... Die Anwendungs**domäne** verstehen

(1) Grundbegriffe, Zusammenhänge, Strukturen

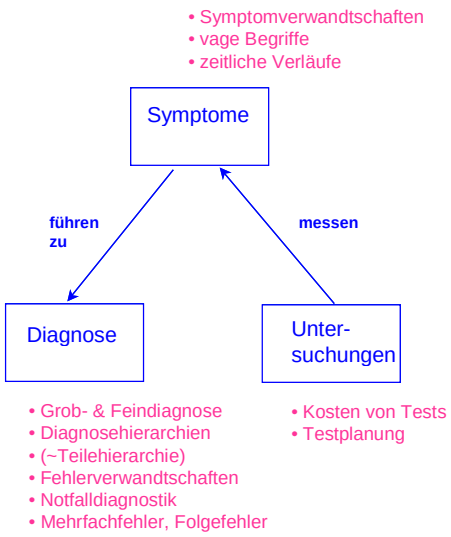
Wissensrepräsentation
(Akquisition, Repräsentation, Wartung)

... Die Anwendungs**problemklasse** verstehen

(2) Methoden, Vorgehensweisen, Strategien

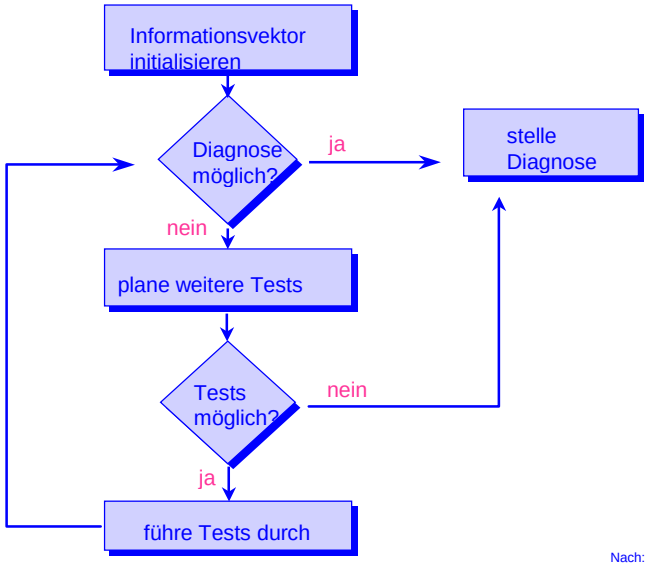
Problemlösemethoden
(Systemaufbau, Basisstruktur der Verarbeitung)

(1) Grundbegriffe der Diagnose



Nach: M.M. Richter

(2) Top-Level Diagramm der Diagnose



Nach: M.M. Richter

Wissensrepräsentation: Beschreibung der Grundbegriffe

- Im folgenden beschreiben wir Symptome, Untersuchungen (Tests) und Diagnosen (Krankheiten, Fehler) etwas näher.
- Dazu geben wir zu jedem dieser Grundobjektttypen die Attribute an, die zur Beschreibung einer Instanz erforderlich sind.
- In einem Framesystem wären dies die einzelnen Slots.
- Wenn wir damit ein Framesystem initialisieren würden, wäre dies eine „leere“ Domänenwissensbasis, die dann mit Instanzen der spezifischen Anwendungsproblemstellung (Diagnose von Blutkrankheiten oder Atemwegserkrankungen, von Automotor-schäden oder Maschinenfehlern, ...) gefüllt werden müßte.
- Für jeden Slot geben wir außer Name und Erläuterung zwei zusätzliche Daten an:
  - stat / dyn gibt an, ob es sich um ein statisches (zur Systemdefinitionszeit einmalig zu erfassen und dann stabil) oder dynamisches (Ausprägung des Attributs hängt vom konkreten Problemfall und wird zur Systemlaufzeit gefüllt bzw noch verändert) Attribut handelt.
  - i / m gibt an, ob das Attribut immer oder nur in medizinischen Diagnose-fragestellungen relevant ist

Beschreibung von Symptomen

Value (s)	Wert von s	dyn	i
Range (s)	Wertebereich	stat	i
Parity (s)	rechts-links	dyn	m

Forschungsbereich Wissensmanagement

Nach: M.M. Richter

Andreas Abecker

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Beschreibung von Tests / Untersuchungen

Parity (t)	rechts-links	dyn	m
Risk (t)	Risiko	stat	i
Time (t)	benötigte Zeit	stat	i
Pain (t)	Schmerz	stat	m
Cost (t)	Aufwand, Kosten	stat	i
Access (t)	Zugang	stat	i
Done (t)	Test ausgeführt	dyn	i

Forschungsbereich Wissensmanagement

Nach: M.M. Richter

Andreas Abecker

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Beschreibung von Krankheiten / Fehlern (1)

Definition (K)	**zwei Formeln K+ und K- für Akzeptanz bzw Zurückweisung von K**	**stat**	**i**
Formula (K+)	**bisher partiell ausgewertete Formel für Akzeptanz**	**dyn**	**i**
Formula (K-)	**analog für Zurückweisen**	**dyn**	**i**
State (K)	**Beweisstatus für K**	**dyn**	**i**
Plus (K)	**Anzahl der bisher zu true ausgewerteten Teilformeln von K+**	**dyn**	**i**
Minus (K)	**analog für K-**	**dyn**	**i**

Nach: M.M. Richter

Einschub: Diagnoseformeln

- Beschreiben Bedingungen zum Zeigen oder Widerlegen einer Diagnose.
- Bilden Formeln aus Attributwertbedingungen zu Symptomen, z.B.
  Fieber µ 39.5 . Blutdruck µ 160 . Geschlecht = männlich
- Sind zu einem bestimmten Zeitpunkt des Systemlaufs i.a. noch nicht vollständig ausgewertet und können daher Beweisstatus wie z.B. *wahr, falsch, möglich* haben.
- Im Fall einer noch unvollständigen Wissensbasis ist nicht notwendigerweise (K+ – K-) immer wahr, ebenso im Falle inkonsistenter Information (K+ . K-) nicht notwendigerweise falsch.
- Für den Beweisstatus einer Diagnose K sind Zustände wie *möglich, verdächtig, bestätigt, widerlegt, unbeweisbar* etc. möglich.

Beschreibung von Krankheiten / Fehlern (2)

Uncertainty (K)	Anzahl der Teilformeln von K+, die zu 'ungefähr wahr' ausgewertet wurden	dyn	i
Parity (K)	rechts-links	stat	m
Urgency (K)	Dringlichkeit der Therapie, falls K vorliegt	stat	m
Effect (K)	Gefahren von K	stat	i
Probability (K)	a-priori Wahrscheinlichkeit von K	stat	i
Cost (K)	Therapiekosten für K	stat	i

Nach: M.M. Richter

Wissensrepräsentation: Beschreibung von Beziehungen

- Im folgenden führen wir zweistellige Beziehungen zwischen Symptomen (s), Tests (t) und Krankheiten (K) auf.
- Diese sind
  - teilweise noch vollständig Bestandteil der Modellierung des Domänenwissens (z.B. Aufbau einer Verfeinerungshierarchie von Symptomen),
  - teilweise sind sie relevant für den Ablauf der Diagnose (z.B. Kontraindikationen für bestimmte Untersuchungen),
  - teilweise ist die explizite Repräsentation eine reine Implementierungsfrage (z.B. ein Symptom taucht in der definierenden Formel für eine Diagnose auf).
- Die Darstellung solcher zweistelliger Relationen könnte beispielweise in einem objektzentrierten Logiksystem durch Fakten geschehen, in einer Framedarstellung könnte man sie einfach durch Attribute repräsentieren, oder selber zu Frames machen, die die beiden betroffenen Objekte verbinden.

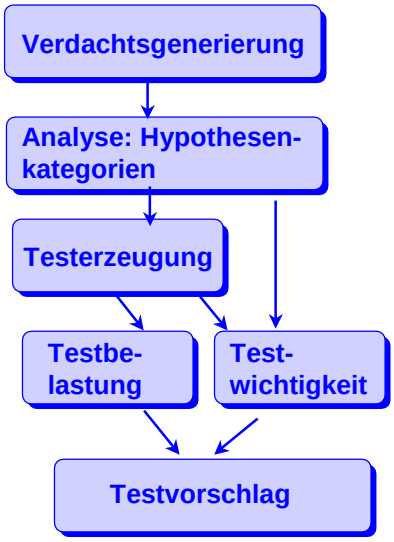
Zweistellige Beziehungen / Relationen (1)

Refines (s1,s2)	Symptom s1 verfeinert s2	stat	i
Neighbour (s1,s2)	s1 und s2 inhaltlich benachbart	stat	i
Eval (s, t)	Test t mißt Symptom s	stat	i
Contra (s, t)	wenn s bestimmte Werte hat, darf Test t nicht ausgeführt werden	dyn	m
Occurs (s, K)	s kommt in Definition von K vor	stat	i
Relevant (s, K)	s kommt in Formula(K+) oder Formula(K-) vor und sein Wert ist noch von Bedeutung	dyn	i

Zweistellige Beziehungen / Relationen (2)

Pointer (s, K)	Symptom s weist deutlich auf K hin	stat	i
Key (s, K)	S ist Schlüsselsymptom für K	stat	i
Covers (s, K)	Liegt K vor, so ist Ausprägung von s hinreichend erklärt	stat	i
Diff (K1, K2)	K2 ist Differentialdiagnose für K1: läßt sich K1 nicht beweisen, so ist K2 neue Hypothese	stat	i
Compl (K1, K2)	K2 ist Komplikation von K1: ist K1 bewiesen, könnte die schlimmere Ausprägung K2 auch noch vorliegen	stat	i
Needed (t, K)	Test t wird zum Nachweis von K+ oder K- noch benötigt	dyn	i

Testplanung als zentrale Systemkomponente



Initialphase:

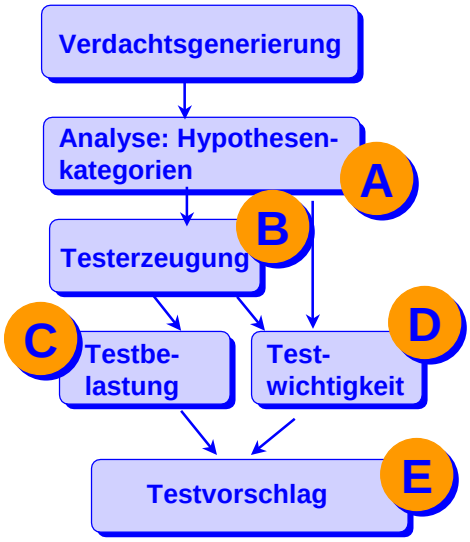
- datengetrieben, Vorwärtsregeln erzeugen neue Tests
- unabhängig vom genauen Wissensstand und Hypothesen
- es werden ohne genaues Ziel neue Symptomwerte erfragt
- Medizin: Anamnesephase

Finalphase:

- Hypothesen werden erzeugt und analysiert
- Hypothesen erzeugen neue Tests

Nach: M.M. Richter

Die folgenden Folien erläutern die Einzelschritte



(A) Erzeugung von Hypothesenkategorien: Hinweisstärke

Einflußfaktoren für Hypothesenkategorien:

- %-Index: [#erfüllte Teilformeln - 1/2 #unerfüllte Teilformeln] / [# Teilformeln]
- #Hinweise: wieviele Pointersymptome wurden beobachtet?
- #Diff: wie oft war eine Krankheit K1 mit Differentialdiagnose K verdächtig, ist jetzt aber unbeweisbar?
- #Komplik: wieviele Krankheiten K1 mit Komplikation K wurden bestätigt?
- #Schlüssel: wieviele Schlüsselsymptome wurden beobachtet?

Auswertung:

%-Index	[50	[60			[75
#Hinweise		μ 1		μ 2	
#Diff			μ 1		
#Komplik				μ 1	
#Schlüssel					1
Hinweisstärke	schwach	deutlich		stark	

Forschungsbereich Wissensmanagement

Nach: M.M. Richter

Andreas Abecker

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

(A) Erzeugung von Hypothesenkategorien: Gesamtbewertung

Hinweisstärke		deutlich	deutlich	stark	deutlich	deutlich
Probability		häufig				selten
Urgency					jetzt	keine
Effect			Lebens-gefahr		Lebens-gefahr	Lebens-gefahr
Cost						
Hypothesen-kategorie	tertiär	sekundär	sekundär	sekundär	primär	wider-legen

A

Forschungsbereich Wissensmanagement

Nach: M.M. Richter

Andreas Abecker

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Im Schnelldurchgang: (B) Testerzeugung, (C) Testbelastung und (D) Testwichtigkeit

B

- Die **Testerzeugung** benutzt die aktuell betrachteten Hypothesen und das statische Wissen darüber, welche Symptome auf welche Diagnosen deuten, sowie welche Tests diese Symptome messen.
- Die relevanten Informationen sind in den Datenstrukturen für eval, occurs, relevant gespeichert. Dabei liefert das Attribut contra Randbedingungen.
- Der momentane Stand dieser Auswertungen wird dynamisch in needed gehalten.

C

- Die Bewertung der **Testbelastung** abstrahiert wiederum in einer Entscheidungstabelle Wertekombinationen der Attribute risk, pain, time und cost der verschiedenen Tests.

D

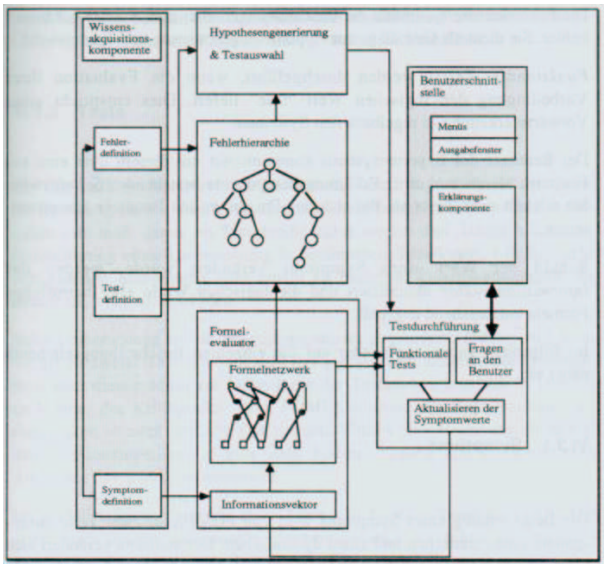
- Die **Testwichtigkeit** verrechnet die Anzahl der Hypothesen, die einen gewissen Test noch benötigen und die jeweiligen Hypothesen-kategorien.

(E) Erzeugen eines Testvorschlags

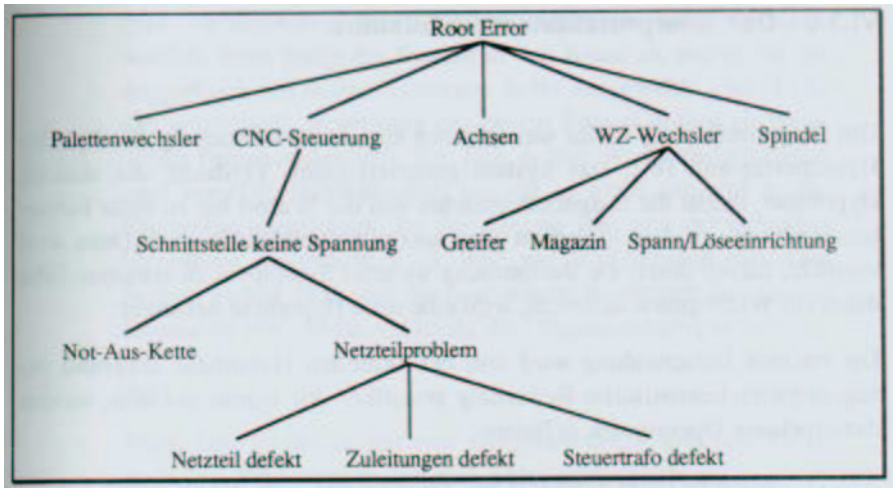
- Im letzten Schritt werden wiederum in einer Entscheidungstabelle Wertekombinationen für die Testwichtigkeit (D) und die Testbelastung (C) zusammen mit Werten des Attributs für die Zugänglichkeit access aggregiert.
- Insgesamt kann man für die Erzeugung eines **Testvorschlags** schließlich als Vorschlagskategorien die Klassen tertiär, sekundär, primär und dringend aufbauen.

E

Beispiel: Systemarchitektur zur Umsetzung der beschriebenen Vorgehensweise



Beispiel: Diagnosehierarchie einer CNC-Maschine

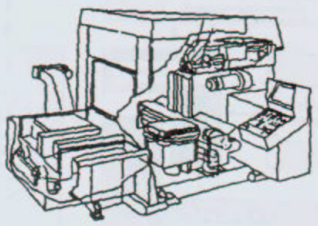


Beispiel: Symptome des Werkzeugwechslers

Symptom	Wertebereich
Antriebsmotor dreht sich	ja/nein
Aufnahmekegel Verschmutzung	ja/ nein
Diodenwiderstand	Kurzschluß/ o.k.
Widerstand elektrischer Leitungen	brüchig/ o.k.
Endschaltersignal	1-Signal/ 0-Signal
Greiferstellung	geschlossen/ Zwischenstellung/einseitig offen/ offen
Hauptgetriebestellung	Mittelstellung/ sonstige Stellungen
IO-Status	logisch 0/ logisch 1
Verlauf	spezielle Impulsverläufe
Kontaktpunktspannung	ja/ nein
Magazin dreht sich	ja/ nein
Position	undefiniert/ inkorrekt
Dichtigkeit mechanischer Leitungen	dicht/ undicht
Spannung Potentialpunkt	24V/ sonstige
Potential	ja/ nein
Relaisstellung	angezogen/ in Ruhe
Ventilstellung	geschaltet/ in Ruhe
Werkzeugnut verschmutzt	ja/ nein
WZ-Wechsler Bewegungsstörung VOR-Bewegung	ja/ nein
WZ-Wechsler Bewegungsstörung RÜCK-Bewegung	ja/ nein

Beispiel: Benutzerschnittstelle eines Diagnosesystems für CNC-Maschinen (MOLTKE)

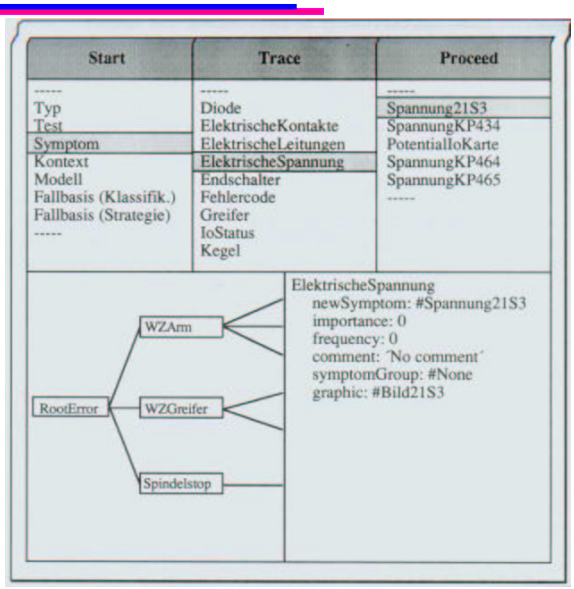
Start	Edit
Start der Diagnose Kontext CNCMaschineDefekt Code: Bitte geben Sie den Wert ein	----- I01 I05 I27 I30 I31 I32 I34 I34 I35 I36



MOLTKE-Beispieldialog

Start interpretation
Context CNCMaschineDefekt
Code : Please enter the value. *I34*
Context WerkzeugLaesstSichNichtSpannen
Ventil5Y1 : Please enter the value. *nichtGeschaltet*
Ventil5Y2 : Please enter the value. *nichtGeschaltet*
IoStatusOUT24 : Please enter the value. *logisch1*
Context Ventil5Y2Geschaltet
SpannungKP459 : Please enter the value. *SpannungLiegtAn*
Relais21K7 : Please enter the value. *Geschaltet*
LeitungenSpannLoeseEinrichtung : Please enter the value. *gebrochen*
Es liegt ein Leitungsdefekt vor. Bitte tauschen Sie die gebrochenen Leitungen aus.
End of diagnosis

Der MOLTKE-Browser



Einige wichtige Punkte, die im Rahmen dieser Vorlesung nicht behandelt werden konnten

- Projektmanagement und betriebliche Einbettung von XPS-Vorhaben
  - siehe z.B. Publikationen von Puppe und / oder Peter Mertens
- Methodisches Vorgehen zur Erstellung von Expertensystemen
  - betrachte z.B. die CommonKADS-Methodologie
- Unterstützung der Wissensakquisition und Wartung von Wissensbasen
  - siehe z.B. die Publikationen bei KAW oder EKAW Workshops
- Erklärungsgenerierung
- technische Infrastrukturen zum XPS-Bau
  - KI-Programmiersprachen (Prolog, OPS5, CLIPS, Lisp)
  - allgemeine XPS-Shells: abgestimmte Auswahl kooperierender Wissensrepräsentations- und Inferenzmechanismen (KEE, NexpertObject)
  - aufgabenspezifische XPS-Shells: „leeres XPS“, mit domänenspezifischem Wissen zu füllen, Basisverarbeitung inclusive spezifischer Editoren usw ist vorgegeben (für Diagnose und Konfiguration)

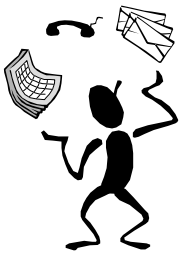
Einige resümierende Worte zur Bedeutung des behandelten Stoffes

- Expertensysteme im engeren Sinn:
  - **Diagnosesysteme** (auch eingebettet) und **Konfiguratoren** werden zu Standardanwendungen. Die Techniken dazu werden zunehmend reifer.
  - Die weittragende industrielle Nutzung anderer Problemklassen ist zur Zeit noch nicht absehbar.
  - Die Bereiche **Umwelt** (datenintensiv, unsicherheitsbehaftet, komplex, wichtig, viele Modellbildungen selber Forschungsgegenstand), **Prozeßkontrolle** komplexer technischer Vorgänge (z.B. Chemie, Kraftwerke) und Intelligente Datenanalyse zur **Entscheidungsunterstützung** (z.B. Kreditwürdigkeitsprüfung) bergen interessantes Anwendungspotential.
- Isolierte Methoden aus den vorgestellten Bereichen:
  - **Constraints** und **CLP-Sprachen** zum Scheduling und (kombiniert mit traditionellen OR-Methoden) für kombinatorische Optimierungsprobleme haben eine gewisse eigenständige Bedeutung.
  - Alle Methoden zur **Wissensrepräsentation** und **Unsicherheitsverarbeitung** sind Basistechniken für die im Rahmen der Internet-Informationsflut boomenden Arbeiten zur Informationsrecherche, Sprachverarbeitung etc. („Semantic Web“)
  - Methoden der **Wissensrepräsentation** und der **Planung** sind wichtige Bausteine für Informationsagenten im E-Commerce.

Was wir gar nicht gemacht haben ...

- Maschinelles Lernen und Data Mining
- Neuronale Netze und Methoden der „Computational Intelligence“
- Fallbasiertes Schließen
- Intelligente Agenten
- Semantic Web
- ...

!! → Es wird nicht langweilig ...



Viel Spaß in den nächsten Jahren!