

# Stand der Verfahren zur Lösungsfindung beim Konzipieren

*H. Grabowski, S. Rude und P. Milde*

*Institut für Rechneranwendung in Planung und Konstruktion (RPK)*

*Universität Karlsruhe (TH)*

*Kaiserstraße 12, D-76131 Karlsruhe*

*E-Mail: gr@rpk.mach.uni-karlsruhe.de*

## **Zusammenfassung**

Zur effizienten Unterstützung von Konstrukteuren werden Konstruktionssysteme zunehmend hinsichtlich der Verfahren zur Wissensverarbeitung weiterentwickelt. Es existieren bereits einige Systeme, denen wissensbasierte, ganzheitliche Rahmenkonzepte zugrunde liegen. In diesem Beitrag werden die wesentlichen Grundlagen der Wissensrepräsentation und Wissensverarbeitung aus dem Bereich der Künstlichen Intelligenz (KI) behandelt und bestehende Ansätze von Konstruktionssystemen zur Lösungsfindung, -beschreibung, und -bewertung beim Konzipieren vorgestellt.

## **1. Grundlagen des Problemlösungsvorganges**

Zur Lösung von Problemstellungen wird in der VDI-Richtlinie 2221 der aus der Systemtechnik bekannte Problemlösungszyklus beschrieben. Wird dieser auf das Problem der Lösungsfindung beim Konstruieren angewandt, kann für alle Konstruktionsphasen eine allgemeine Handlungsstrategie abgeleitet werden. Dabei werden in fünf Schritten, von der Problemdefinition bis zur

Entscheidung, Lösungen für die Problemstellung entwickelt und ausgewählt (Bild 1).

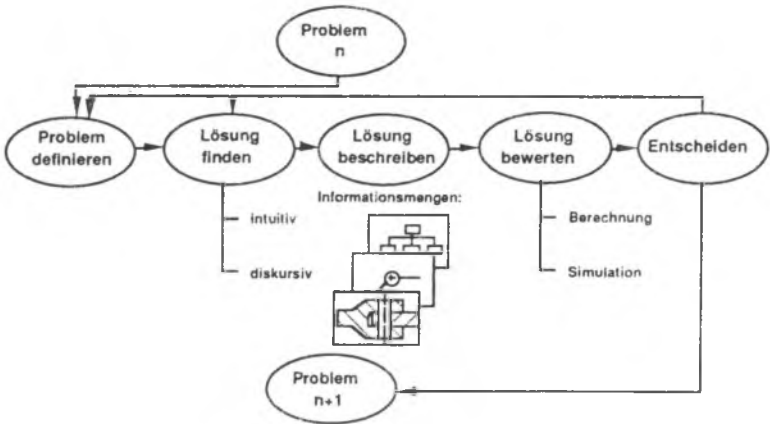


Bild 1. Problemlösungszyklus beim Konzipieren

Der Schritt *Problem definieren* dient dem Erkennen von Anforderungen und der Extraktion der relevanten Informationen zur Bearbeitung des Problems. Diese gesammelten Informationen werden in eine zur Lösungssuche geeignete Form gebracht. *Lösung finden* ist der eigentliche kreative Schritt im Konstruktionsprozess. Hier werden Lösungen, die möglichst alle Anforderungen an das Produkt erfüllen, entwickelt. Der Konstrukteur findet auf der Basis definierter Informationen weitere Informationen des vorliegenden, eines vorangehenden oder eines nachfolgenden Arbeitsschrittes. Dabei ist eine lösungsfindende Tätigkeit immer auch mit einer Modellierung des gefundenen Ergebnisses verbunden. Die Lösungsfindung kann aufgrund vorliegender Lösungssammlungen (z.B. Konstruktionskatalogen) geschehen oder aber auf der Basis von auf dem Rechner verfügbarer Wissensspeicher (z.B. Lösungsmuster). Dabei soll methodisch gesehen nicht zwischen persönlich, unternehmensweit oder unternehmensübergreifend verfügbaren Wissensspeichern unterschieden werden.

Im nächsten Schritt (*Lösung beschreiben*) werden die Lösungen anhand ihrer Beschaffenheitsmerkmale beschrieben /Dyll-91/. Dies erfolgt in einer für die jeweilige Konstruktionsphase geeigneten Darstellungsform (konventionell in

Form von Skizzen oder Zeichnungen; rechnerunterstützt durch Datenmodelle). Beispielsweise kann so von der Modellierung von Anforderungen gesprochen werden, wenn Anforderungen auf dem Papier in eine Anforderungsliste eingetragen werden, aber auch, wenn auf dem Rechner Anforderungen editiert werden.

Der Schritt *Lösung bewerten* beinhaltet eine Vielzahl von Verfahren, die es erlauben, festgelegte Lösungen mit der Aufgabenstellung zu vergleichen. Dabei kann die Bewertung durch Berechnung, durch Simulation und auch durch Austesten einer vorgeschlagenen Lösung erfolgen. Besonders Testverfahren sind eine methodisch gesehen unterschätzte Informationsquelle für das in der Praxis immer noch auf Erfahrungswissen begründete Ingenieurwesen.

Die sich anschließende *Entscheidung* bestimmt anhand der Bewertungen welche Lösung weiter bearbeitet wird und die weitere Vorgehensweise.

Das Konstruieren kann in diesem Kontext also als ein Problemlösungsvorgang mit folgenden Merkmalen angesehen werden:

- Die Zahl prinzipiell möglicher Lösungen (Lösungsraum) ist sehr groß.
- Die Berücksichtigung von Abhängigkeiten zwischen Merkmalsausprägungen einzelner Lösungsobjekte ist erforderlich.
- Die Rücknahme von Entscheidungen, wenn sie sich als unhaltbar erweisen.
- Ein hierarchisch strukturiertes Vorgehen bei der Konstruktion.

Betrachtet man die Lösungsfindung detaillierter, so ergeben sich, von einem gegebenen Lösungszustand ausgehend, verschiedene Fortschreitungsrichtungen um eine Problemlösung zu entwickeln. Lösungszustände unterscheiden sich hierbei in Konkretisierungs-, Detaillierungs- und Variationsgrad. Diese Fortschreitungsrichtungen des Konstruierens, als Achsen in einem dreidimensionalen Raum aufgetragen, erlauben es einen Modellraum des Konstruierens aufzuspannen /Rude-91/ (siehe [Bild 2](#)). Innerhalb dieses Modellraumes bewegen sich alle grundsätzlich möglichen Konstruktionstätigkeiten, die durch Iterationszyklen und Rücksprünge sowohl innerhalb der Konstruktionsphasen als auch phasenübergreifend gekennzeichnet sind. Das Voranschreiten in eine Richtung des Modellraumes bedingt eine Änderung der

Informationen in den Datenmodellen, die als neue Problemdarstellungen interpretiert werden können.

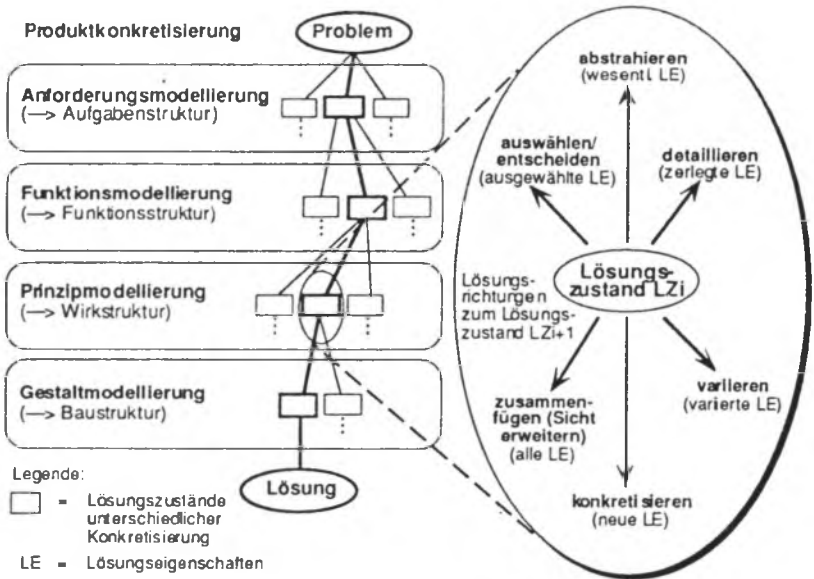


Bild 2. Elementare Lösungsrichtungen im Modellraum des Konzipierens

Durch *Konkretisierung* wird einer Problemstellung ein Lösungsbaustein einer konkreteren Konstruktionsphase zugeordnet. Beispielsweise wird einer Funktion ein Wirkprinzip zugeordnet. Die entgegengesetzte Fortschrittsrichtung wird als *Abstraktion* bezeichnet.

Durch *Detaillierung* kann eine Problemstellung innerhalb einer Konstruktionsphase verfeinert werden. Dies wird durch Zerlegen des Gesamtproblems in Teilprobleme (Dekomposition) erreicht. Als *Sicht erweitern* wird das Zusammenfügen von Teilproblemen bezeichnet.

Durch *Variation* können alternative Lösungen für eine Problemstellung innerhalb einer Konstruktionsphase entwickelt werden. Durch vergleichende Beurteilung wird eine der möglichen Lösungen ausgewählt.

## 2. Wissensrepräsentation als Voraussetzungen für die Rechnerunterstützung

Die Anwendung des Rechners setzt eine strenge Formalisierung der im betrachteten Teilgebiet des Konstruierens zu erzeugenden Informationen (Wissen) voraus. Dies stellt sich beim gegenwärtigen Stand der Technik derzeit noch als schwieriges Unterfangen dar, da eine klare und allgemein anerkannte Begriffsbildung für den Konstruktionsprozeß eigentlich noch aussteht /Fran-92/.

Als Möglichkeiten zur Formalisierung von Wissen stehen vor allem Wissensrepräsentationstechniken aus der KI zur Verfügung. Wissen läßt sich in diesem Zusammenhang zunächst in theoretisches und praktisches Wissen unterscheiden. In der KI wird theoretisches Wissen auch als Tiefenwissen und praktisches Wissen als Oberflächenwissen bezeichnet /NCSE-89/.

- Theoretisches Wissen (Tiefenwissen)

Theoretisches Wissen formalisiert grundsätzliche Beziehungen, die jederzeit logisch nachvollziehbar und verifizierbar sind. Es umfaßt die Begriffsdefinitionen, Ursache-Wirkungs-Beziehungen zwischen Objekten und mathematische Gesetze. Dieses Wissen kann theoretisch erworben werden, beispielsweise in der Schule, auf der Universität oder durch Lesen von Fachbüchern.

- Praktisches Wissen (Oberflächenwissen)

Praktisches Wissen wird verwendet, wenn die Problemstellungen nicht ausreichend strukturiert sind und die Theorie nicht weiter hilft. Es umfaßt Erfahrungen, Heuristiken, "Daumenregeln" und vages bzw. unsicheres Wissen.

Jeder dieser beiden Wissensklassen können wiederum die Wissensarten Allgemeines Fachwissen (deklaratives Wissen), Problemspezifisches Wissen (prozedurales Wissen) und Allgemeines Problemlösungswissen (Kontrollwissen, Meta-Wissen) zugeordnet werden /CuGS-90/ (Bild 3).

- Allgemeines Fachwissen (auch Domänenwissen, deklaratives Wissen oder statisches Wissen)

Allgemeines Fachwissen umfaßt Fakten, Objekteigenschaften und Beziehungen, die für beliebige Objekte innerhalb eines bestimmten

Anwendungsbereichs problem- bzw. aufgabenunabhängig gültig sind. Dieses Wissen ist hierarchisch oder netzwerkartig strukturiert. Für den Konstruktionsbereich wird das allgemeine Fachwissen durch hochgradig vernetzte Produktmodelle repräsentiert.

- Problemspezifisches Wissen (auch prozedurales Wissen oder dynamisches Wissen)

Problemspezifische Wissen ist methoden- bzw. ablauforientiert und bezieht sich auf eine konkrete Problemstellung. Neben der Aufgabenstellung umfaßt das problemspezifische Wissen alle Teillösungen und Lösungsvarianten. Das für den Konstruktionsbereich maßgebliche Wissen ist durch Konstruktionsmethoden und Evaluierungs- bzw. Berechnungsverfahren repräsentiert.

- Allgemeines Problemlösungswissen (auch Kontrollwissen, Handlungswissen oder Meta-Wissen)

Die beiden zuvorgenannten Wissensarten beschreiben ausschließlich Wissen, welches den Lösungsraum einschränkt. Über mögliche Wege zum Ziel bzw. zur Lösung wird jedoch nichts ausgesagt. Allgemeines Problemlösungswissen beschreibt die Art und Weise, mit der durch Kombination der beiden erstgenannten Wissensarten ein Problem gelöst werden soll.

Wissens- klassen \ Wissens- arten	Allgemeines Fachwissen (deklaratives Wissen)	Problemspezifisches Wissen (prozedurales Wissen)	Allgemeines Problem- lösungswissen (Kontrollwissen)
Theoretisches Wissen (Tiefenwissen)	Fakten- oder Tabellenwissen	Formel- oder Regelwissen	Wissen um die Vorgehensweise
Praktisches Wissen (Oberflächenwissen)	Vages oder unvoll- ständiges Wissen	Heuristiken und Daumenregel	Strategie bei der Lösung einer Aufgabe Auswahl der "richtigen" Maßnahme oder Faustregel

Bild 3. Wissensklassen und Wissensarten

Alle Wissensklassen bzw.-arten müssen im Hinblick auf eine Rechnerunterstützung rechnerintern repräsentiert werden. Für eine solche formale Abbildung stellt die KI geeignete Abbildungsansätze und Programmier-Paradigmen bereit. Im folgenden werden die wichtigsten Ansätze vorgestellt.

## **2.1. Regel-basierter Ansatz**

Der regel-basierte Ansatz zeichnet sich durch eine der menschlichen Ausdrucksweise ähnlichen Form, die Modularität und die Anpaßbarkeit der Regeln an konkrete Problemstellungen aus /Pupp-88/. Das Domänenwissen wird als sog. Produktionsregeln in Form von WENN-DANN-Regeln, die an den Paradigmen der Prädikatenlogik orientiert sind, dargestellt. Fakten und Annahmen, auf denen die Regeln operieren, werden als Objekt-Attribut-Wert-Tripel oder als Listenstruktur repräsentiert (Datenbasis). Zur Problemlösung werden mustergesteuerte Inferenzmechanismen sowie vorwärts- und rückwärtsgerichtete Folgerungsstrategien verwendet, die Regeln verarbeiten. Bei der Lösung komplexer Problemstellungen ist der Schlußfolgerungsprozeß nur sehr schwer nachvollziehbar. Mit wachsender Größe der Regelbasen sinkt der Nutzen regelbasierter Systeme /Lu-91/, da die Leistungsfähigkeit sinkt und der Aufwand für Wartung und Pflege steigt /Pupp-88/.

## **2.2. Frame-basierter Ansatz**

Frames (Objekte) können als spezialisiertes "object oriented programming" betrachtet werden. Daten und Prozeduren sind in einem Objekt untrennbar zusammengefaßt. Objekte können Eigenschaften an andere Objekte vererben und über sog. "Messages" miteinander kommunizieren /Pupp-88/.

Bei einfachen regel-basierten Systemen steckt das eigentliche Wissen in den Regeln. Die Datenbasis ist eine unstrukturierte und passive Ansammlung von Fakten. Ein frame-basierter objektorientierter Ansatz hingegen ermöglicht eine wohlstrukturierte Darstellung von Objekten und Objektklassen /Kläg-93/. Frames repräsentieren sowohl deklaratives Objektwissen als auch prozedurales Objektverwendungswissen.

Eine Frame-Klasse ist eine formularähnliche Objektbeschreibung (Datenstruktur). Objekteigenschaften werden durch sog. Slot Attributes repräsentiert. Von jeder Frame-Klasse können beliebig viele Instanzen gebildet werden, die eine konkrete mit Werten belegte Objektausprägung darstellen. Durch Zuordnung von Frame-Klassen können hierarchische Objektstrukturen abgebildet werden. Eigenschaften können von sog. "Parent"-Frame-Klassen an "Child"-Frame-Klassen weitervererbt werden. Diese hierarchischen Strukturen können graphisch als ein gerichtetes semantisches Netz dargestellt werden.

### **2.3. Constraint-basierter Ansatz**

Technische Probleme zwingen den Konstrukteur eine Vielzahl von Zwangsbedingungen zu berücksichtigen, die nicht verletzt werden dürfen. Das Ziel von Constraint-basierten Systemen ist das Finden einer Lösung unter Beachtung dieser Zwangsbedingungen.

Constraints (Zwänge, Beschränkungen, Restriktionen) dienen der Abbildung und Auswertung von Abhängigkeiten zwischen den Objekten, welche erfüllt werden müssen, ohne jedoch eine konkrete Lösung vorzugeben. Wird ein Constraint mit anderen Constraints verknüpft entsteht ein Constraint-Netz. Ein Constraint kann man oft als Gleichung, die ungerichtete Zusammenhänge zwischen Variablen ausdrückt, betrachten. Das Constraint-Netz als das zugehörige Gleichungssystem. Das Ausrechnen des mit Variablenwerten belegten Gleichungssystems wird als "constraint propagation" bezeichnet.

Das Constraint-Konzept ist ein mächtiger Ansatz zur Abbildung konstruktionsimmanenter Abhängigkeiten. Constraint-basierte Ansätze werden zur Zeit in Verbindung mit frame- bzw. regel-basierten Ansätzen bei wissensbasierten Systemen verwendet. Ein Anwendungsschwerpunkt liegt hierbei im Bereich der geometrischen Constraint-Modellierung.

### **2.4. Fall-basierter Ansatz**

Konstrukteure arbeiten bei der Lösung von Konstruktionsaufgaben nicht ausschließlich mit allgemeinem Konstruktionswissen. Sie orientieren sich



häufig an Wissen über existierende Lösungen für ähnliche Aufgabenstellungen, die ihnen als Heuristik dienen. Dieses auf Analogien beruhendes Schließen ist für den Menschen eine der wichtigsten Methoden zur Lösungsfindung. Es werden hierbei sowohl allgemeine Erfahrungen aus der Vergangenheit, als auch konkrete Einzelfälle bei der jeweils vorliegenden Problemstellung herangezogen. Daher werden in der KI auch Begriffe wie "experience-based" oder "analogical reasoning system" verwendet.

Voraussetzung für ein fall-basiertes Vorgehen ist das in Wissensbasen (case bases) in Form von Bibliothekslösungen (Lösungsmuster) /Suhm-93/ verwaltete Wissen über Lösungen bzw. Teillösungen früherer Konstruktionsprobleme. In einer solchen Wissensbasis können Fallbeschreibungen in Form von Frame- oder Constraint-Datenstrukturen abgelegt sein. Eine Fallbeschreibung muß Aussagen über die Aufgaben- bzw. Problembeschreibung, die Lösung und den Lösungsvorgang enthalten /GCSB-89/. Mit Hilfe von Mustererkennungsverfahren (pattern matching) wird versucht, eine möglichst ähnliche Lösung zu finden. Vergleichsmaßstab für nahezu alle Mustererkennungsverfahren ist die Ähnlichkeit der gegebenen Problemstellung mit einer bereits gelösten. Hierbei wird vorausgesetzt, daß für ähnliche Problemstellungen auch ähnliche Lösungen erwartet werden. Ein grundlegendes Problem des Fall-basierten Ansatzes ist das Erkennen von ähnlichen Lösungen bzw. die Festlegung von Ähnlichkeitsmerkmalen.

### **3. Bestehende Ansätze für die Rechnerunterstützung beim Konzipieren**

Die folgenden Ausführungen beziehen sich ausschließlich auf die frühen Konstruktionsphasen Funktionsfindung und Prinzipiarbeitung. Grundsätzlich sind Hilfsmittel zur Unterstützung des Konstrukteurs in Abhängigkeit von den dem Konstrukteur bereits bekannten Informationen sowie in Abhängigkeit von den zur Verfügung stehenden Wissensspeichern einzuteilen. Für die Einteilung von Hilfsmitteln zur methodischen Entwicklung von Lösungsprinzipien wird weiterhin unterschieden in Verfahren zur (Bild 4)

- o Lösungsfindung,
- o Lösungsbeschreibung und
- o Bewertung.

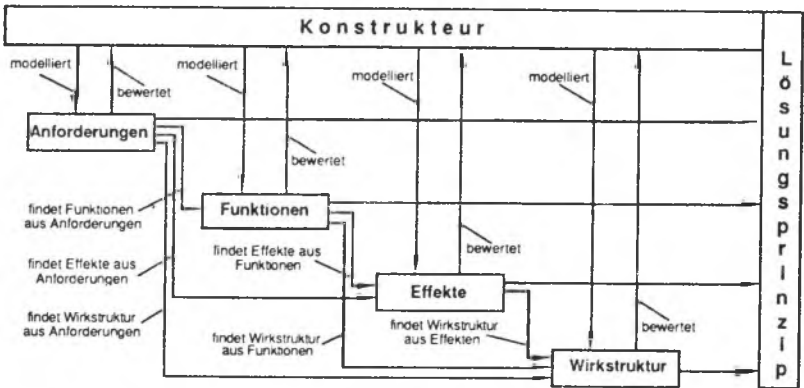


Bild 4. Ermitteln von Lösungsprinzipien

Ausgehend von den Arbeitsschritten des Konstruierens sowie anhand der vorgestellten Einteilung werden im folgenden der Stand der Technik bezüglich der verfügbaren Hilfsmittel zur methodischen Erarbeitung von Lösungsprinzipien vorgestellt. Diese werden ausschließlich unter dem Aspekt der Rechnerunterstützung dargestellt.

### 3.1. Lösungsbeschreibung

In diesem Abschnitt werden verschiedene Ansätze zur Modellierung von Anforderungen, Funktionen, Effekten und Wirkstrukturen vorgestellt.

#### Modellieren von Anforderungen

Für das Modellieren von Anforderungen wurden vor allem im Rahmen von Forschungsprojekten Ansätze entwickelt.

Das Projekt RA-IQSE /Irge-91/ (Revision Advisor - An Integrated Quality Support Environment) entwickelte hierbei Ansätze zur Modellierung von Anforderungen (Requirements Specifications RS) zur Beschreibung von Funktions-, Material- und Geometrieangaben. Dieser Ansatz bezieht sich vorrangig auf die Variantenkonstruktion. Für diese liegt bereits Wissen über die Produktstruktur, die Produktfunktionalität und über qualitätsrelevante

Daten vor. Damit wird der Konstruktionsprozeß auf ein Konfigurationsproblem, bei dem bereits existierende Lösungen durch regelbasierte Schlußfolgerungsmechanismen zu einer Gesamtlösung zusammengesetzt werden, eingeschränkt. Geometrische Dimensionierungsgrößen (Konstruktionsparameter) einschließlich Toleranzen, Materialarten und -eigenschaften sowie wesentliche Leistungsdaten (funktionsbezogene Anforderungen) können als Anforderungen in einem formularorientierten Editor eingegeben werden. Die Suchfunktion wird über Parameterabfrage der Einzelkomponenten anhand von Check- bzw. Anforderungslisten durchgeführt. Zusätzlich wird die Vollständigkeit der Anforderungsspezifikation durch Vergleich mit den Parametern der Einzelkomponenten überprüft.

Im Rahmen des IIICAD-Projektes /XTKT-91/ wird ausgehend von funktionalen Anforderungen durch Schlußfolgerungen eine initiale Lösungsidee entwickelt. Diese wird so lange verfeinert, bis eine vollständige Lösungsbeschreibung erzielt wird. Hierbei wird der Konstruktionsprozeß als ein evolutionärer Prozeß angesehen, der ein Konstruktionsobjekt von einem Zustand  $M(i)$  in eine detaillierteren Zustand  $M(i+1)$  überführt. Das IIICAD-System orientiert sich an einer eigens entwickelten Konstruktionstheorie. Diese basiert auf drei verschiedenen Modellen, die jeweils ein wesentliches Charakteristikum des Konstruktionsprozesses widerspiegeln /TVTY-90/. Das deskriptive Konstruktionsprozeßmodell erklärt wie konstruiert wird, das kognitive Konstruktionsprozeßmodell bildet das menschliche Problemlösungsverhalten und das ausführbare Konstruktionsprozeßmodell, das die Representation derjenigen Problemlösungs- und Schlußfolgerungsmechanismen umfaßt, die vom Rechner ausgeführt werden können.

Bei TNO in Holland /Tolm-91/ wird unter dem Begriff des „Hamburgers“ ein Lösungsmuster (Product Definition Unit, PDU) zerlegt in eine Aufgabe (Functional Unit, FU) und deren Lösung (Technical Solution, TS). Diese Arbeiten entstammen der Bauingenierbranche, lassen sich jedoch gleichermaßen auf den Maschinenbau übertragen.

Auf ähnlichen Ansätzen basiert die von Suh /Suh-90/ entwickelte Konstruktionstheorie. Allerdings steht hierbei die auf der Basis definierter Anforderungen (Functional Requirements FRs) vorgenommene Bewertung von Lösungen (Design Solutions DSs) im Vordergrund.

Das System ALLTOOL /BaDS-89/ wurde zur konstruktionsbegleitenden Erstellung und Bearbeitung von Anforderungslisten in Braunschweig ent-

wickelt. Basierend auf einer am Produktlebenslauf orientierten Suchmatrix zur Anforderungsermittlung /Fran-75/ sowie aspektbezogener Checklisten /RoDr-89/ wird das Editieren von Anforderungen auf dem Rechner mittels eines formularorientierten Eingabedialoges ermöglicht. In der vorliegenden Form sind sowohl Suchmatrix als auch die aspektbezogenen Checklisten starre Schablonen, die den Konstrukteuren lediglich das "Durchblättern" erlauben. Diekhöner /Diek-90/ erarbeitet hierauf aufbauend Konzepte zur Umsetzung dieser Ansätze in Industrieunternehmen.

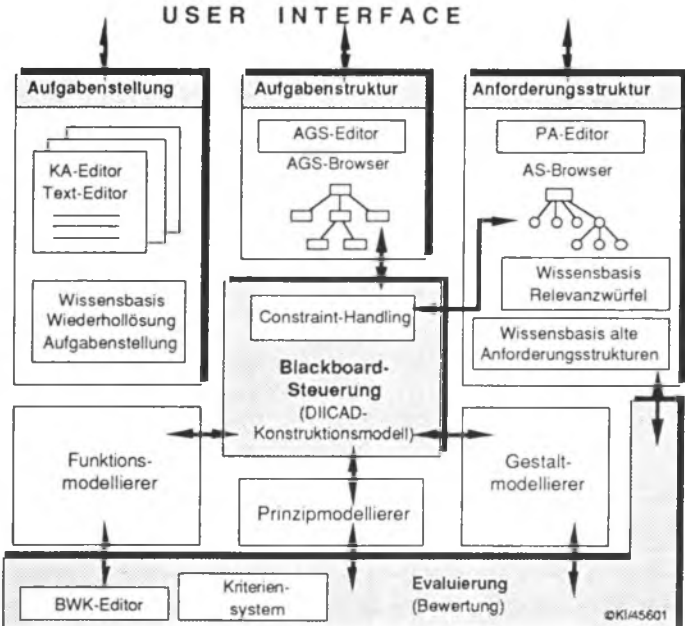
Innerhalb des Systemes KALEIT /Groe-90/ (Konstruktions-Analyse- und Leitsystem) wurde ebenfalls ein Werkzeug zur Modellierung von Anforderungen entwickelt. Konzeptionelle Grundlage bildet die Leitlinie mit Hauptmerkmalen zum Erstellen einer Anforderungsliste nach Pahl/Beitz /PaBe-86/. Die Abbildung erfolgt in Form von hierarchischen Baumstrukturen, in denen die Hauptmerkmale weiter detailliert werden. Zur Modellierung von Aufgabenstellungen steht ein Textformular-Editor zur Verfügung. Hierzu können einzelne Zweige des Strukturbaumes ausgewählt und Anforderungen formalisiert in einer Datenbank abgelegt werden. Die Suche nach ähnlichen Aufgabenstellungen erfolgt auf der Basis der formal editierten Anforderungen.

In Karlsruhe wurde im Rahmen der DIICAD-Entwicklung /Kläg-93/ (Dialogorientiertes Integriertes Intelligentes CAD-System) an der Entwicklung eines Systemes zur Modellierung von Anforderungen gearbeitet. Im Vordergrund steht hierbei die Verwendung der modellierten Anforderungen für das Finden und das Bewerten gefundener Lösungen. Für die Modellierung von Anforderungen ergeben sich die Problemschwerpunkte Akquisition der relevanten Anforderungen, Definition, Strukturierung, Zuordnung und Evaluierung.

Jede Konstruktionsaufgabe beinhaltet in der Regel neben explizit vorgegebenen Anforderungen, die die durchzuführende Konstruktionsaufgabe eindeutig bestimmen, weitere implizit vorhandene Anforderungen und Randbedingungen. Implizite Anforderungen können während des gesamten Produktlebenszyklus im Zuge des Aufgabenklärungsprozesses aus expliziten Anforderungen abgeleitet werden. Zur Definition und Strukturierung von Anforderungen wurden Beschreibungs- und Eigenschaftskenngrößen zusammengestellt und als operationale Definitionsgrößen postuliert. Anforderungen werden dazu nach Forderungscharakter (Muß-/ Wunschforderung), Erfüllungscharakter (nur Erfüllung/Grad der Erfüllung), Forderungstyp (quantitativ/qualitativ), Akquisitionstyp (explizit/implizit), Herkunftsbereich

(ursachenbezogen/quellenbezogen) und Konkretisierungsgrad (zusammengesetzt/elementar) unterschieden. Zur Bewältigung des Zuordnungsproblems müssen Anforderungen zunächst konsolidiert und anschließend in bezug auf lösungsfindende Prozesse zielgerichtet aufbereitet und zugeordnet werden. Bei der Konsolidierung werden die Anforderungen inhaltlich nach den Gesichtspunkten Widerspruchsfreiheit (Anforderungen sind unabhängig), Überdeckungsfreiheit (Anforderungen sind redundanzfrei), Genauigkeit und Vollständigkeit überprüft. Grundsätzlich können die Abhängigkeiten Anforderungsneutralität, Anforderungsgleichsinnigkeit, Anforderungskonflikt und Anforderungsgegensatz auftreten.

**Bild 5** zeigt die logische Systemarchitektur des Konstruktionssystem zur Anforderungsmodellierung.



Legende: AGS = Aufgabenstruktur KA = Konstruktionsauftrag PA = Produktanforderung  
 AS = Anforderungsstruktur BWK = Bewertungskriterium

*Bild 5. Logische Systemarchitektur*

### Modellieren von Funktionen

Die Modellierung von Funktionsstrukturen ist gleichermaßen ein Gebiet, auf dem erst jüngere Forschungsansätze entsprechende Hilfsmittel zur Verfügung stellen. Ausgangspunkt ist eine Vielzahl benutzter Terme für Funktionsbegriffe (siehe Bild 6). Dabei ist festzustellen, daß eine weitgehende Übereinstimmung der Autoren Roth /Roth-82/, Pahl/Beitz /PaBe-86/, Beitz /Beit-90/ und Koller /Koll-85/ vorliegt. Allerdings zeigt Bild 6 auch, daß gewisse Unterschiede in den Sichten (beispielsweise die Verben "Sperrern" nach Beitz sowie "Richtung ändern" nach Koller bis heute noch nicht ausdiskutiert sind.

		Roth (1982, S. 279)	Pahl / Beitz (1986, S. 86)	Beitz (1990, S. 22)	Koller (1985, S. 32/37)
		Speichern	Speichern	Speichern	Speichern Entspeichern
Übertragen		Leiten	Leiten	Leiten	Leiten/Isolieren
		Umformen	Ändern	Vergrößern/ Verkleinern	Vergrößern/ Verkleinern
		Wandeln	Wandeln	Wandeln	Wandeln/ Rückwandeln
Verknüpfen	summa	gleiche	Verknüpfen	Verknüpfen	Teilen
		ungleiche			Verbinden
	distrib	gleiche		Verzweigen	Teilen
		ungleiche			Trennen
				Sperrern	
					Richtung ändern

Bild 6. Vergleich der Funktionsverben unterschiedlicher Konstruktionsmethodiker

Im internationalen Rahmen ist zur Darstellung funktionaler Zusammenhänge ausgehend von den Methoden SA (Structured Analysis) und SD (Structured Design) die Methode SADT /Ross-85/ entwickelt worden. Dabei werden die zu bearbeitenden Aufgaben durch Rechtecke (Kästchen) dargestellt. Zwei Aufgabenkästchen können durch Pfeile miteinander verbunden werden, die jeweils Ausgangs- bzw. Eingangsinformationen darstellen. Eine hierarchische Zerlegung von Gesamtaufgaben in Teilaufgabe wird ermöglicht.

Die rechnerunterstützte Erstellung solcher Diagramme wird auch unter dem Namen IDEF0 angeboten, z.B. /NN-89/.

Im Rahmen des Systems KALEIT wird ebenfalls ein Modul zur Funktionsmodellierung entwickelt. Schwerpunkt liegt auf der Betrachtung von Wechselwirkungen der Funktionsbeschreibung mit der Wirkstruktur /Kutt-92/. Funktionsstrukturen können eine hierarchische oder netzwerkartige Ordnung aufweisen. Für die Anordnung von Wirkelementen sind Regeln, die auf der Funktionsbeschreibung beruhen, definiert. Der Einfluß auf die verschiedenen Strukturmodelle der einzelnen Konstruktionsphasen ist zwar erkannt, eine Integration mit diesen Strukturen hat jedoch nicht stattgefunden /Hube-94/.

Auch im Rahmen von IICAD existieren umfangreiche Möglichkeiten zu Funktionsmodellierung /Hube-94/. Die Modellierung der Funktionsweise eines Produktes steht hier im Vordergrund, die Bedeutung der Funktionsbeschreibung für den Lösungsfindungsvorgang wird jedoch nur eingeschränkt berücksichtigt.

In Karlsruhe wird bei Prof. Grabowski /LaMR-92/ zunehmend die Notwendigkeit der Modellierung nicht nur statischer Zusammenhänge, sondern vermehrt dynamischer und funktionaler Zusammenhänge erkannt. Dabei entstehen auch Funktionsmodelle, die den Zeitaspekt stärker als bisher berücksichtigen. Es wurden Sichten auf das Funktionsmodell festgelegt /Hube-94/, um den Anforderungen nach der Repräsentation verschiedener Abstraktionsgrade der funktionalen Produktbeschreibung und der Integration mit anderen Strukturmodellen gerecht zu werden. Dies sind die formal abstrakte (Allgemeine Funktionsstruktur), die physikalische (Spezielle Funktionsstruktur), die zeitlich flußbezogene (Funktionsflußstruktur), die wirkbezogene (Wirkfunktionsstruktur) und die logisch topologische (Funktionale Systemstruktur) Sicht auf das Funktionsmodell.

Bild 7 zeigt die logische Systemarchitektur des Konstruktionssystems zur Funktionsmodellierung.

### **Modellieren von Effekten und Wirkstrukturen**

Das Modellieren von Effekten und Wirkstrukturen wird in erster Linie durch FEM-Preprozessoren ermöglicht. Dabei steht die Bereitstellung der für die Bewertung modellierter Strukturen durch Berechnungsprogramme notwendiger Daten im Vordergrund.

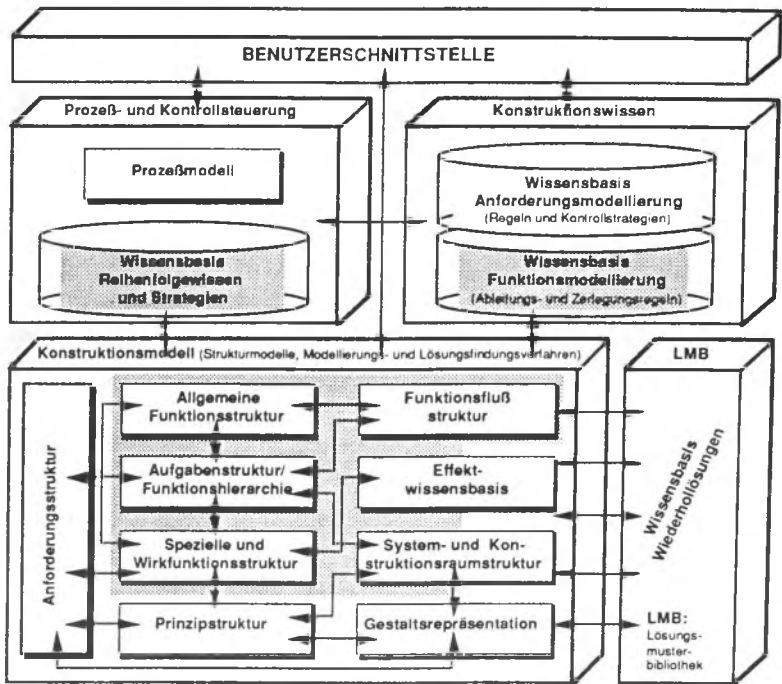


Bild 7. Systemarchitektur

### 3.2. Lösungsfindung

Für das Finden von Lösungen für eine Problemstellung existieren sowohl Ansätze zur Bereitstellung von Wissen aus Katalogen auf dem Rechner als auch weitergehende Ansätze zur wissensbasierten Lösungsfindung.

#### Finden von Funktionen, Effekten und Wirkstrukturen aus Anforderungen

Für das Finden von Funktionen, Effekten und Wirkstrukturen aus Anforderungen werden derzeit erste Ansätze entwickelt /Klåg-93, Suhm-93/. Zielsetzung ist hierbei - zumindest in einem eingegrenzten Bereich - die selbständige, auf Lösungsmuster basierende Suche nach Lösungen und die Entwicklung von Lösungsvorschlägen.



## Finden von Effekten und Wirkstrukturen aus Funktionen

Für die Konstruktion aufbereitete Sammlungen physikalischer Effekte, die jeweils auch durch Prinzipskizzen beschrieben sind, sind die Funktionsgrößen-Matrix nach Roth und die Prinzipkataloge nach Koller.

Bei der Funktionsgrößen-Matrix nach Roth /Roth-82/ wird aufgrund einer gegebenen Kombination physikalischer Größen (einer Eingangs- und einer Ausgangsgröße) der Zugriff auf physikalische Gesetze ermöglicht. Dabei wird der geometrische Gehalt eines Gesetzes gleich anhand einer Prinzipskizze erläutert. Steht beispielsweise als Eingangsgröße eine Kraft zur Verfügung und ist als Ausgangsgröße eine von der Eingangsgröße verschiedene Kraft gefordert, so verweist das entsprechende Matrixfeld auf eine Menge physikalischer Gesetze, die die entsprechende Funktion zu erfüllen in der Lage sind.

Bei Koller /Koll-85/ finden sich Prinzipkataloge, die physikalische Effekte geordnet nach Ursache-Wirkungs-Zusammenhängen anbieten.

Für Roth wie auch für Koller gilt jedoch als Nachteil, daß der gleiche physikalische Effekt im Katalog mehrfach auftaucht. Dies liegt daran, daß der durch ein Gesetz mit mehreren physikalischen Größen beschriebene Effekt nach unterschiedlichen Größen aufgelöst, verschiedene Lösungsprinzipien ergibt und somit in einem manuellen Nachschlagewerk auch mehrfach auftauchen muß.

Dieser Nachteil wird durch die Übertragung des Wissens um physikalische Effekte in einen Wissensspeicher vermieden. Ein Effektspeicher /Rude-91/ kann beispielsweise wie in Bild 8 dargestellt aufgebaut sein und die Suche nach einer geforderten Kombination von Eingangs- und Ausgangsgrößen zulassen.

Die Verarbeitung der solcherart gefundenen Prinzipielemente auf dem Rechner erlaubt ferner die Kombination dieser Elemente zu Prinzipstrukturen /XTKT-91/.

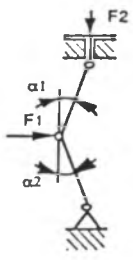
Effektspeicher		
Gesetz		$F_2 = F_1 / (\tan \alpha_1 + \tan \alpha_2)$
Effektname Kniehebel	Prinzipiskizze 	Größen Kraft F1 Kraft F2 Winkel $\alpha_1$ Winkel $\alpha_2$
Quellen Roth 1982 S. 308		

Bild 8. Beispiel eines Wissensspeichers für physikalische Effekte nach Rude 1991 (Auszug)

### 3.3. Bewertung

Fragestellungen der Bewertung werden im VDI-Ausschuß "Berechnungsverfahren" derzeit behandelt. Informationen, die für Rechen- oder Simulationsverfahren dabei aus dem Bereich der Konzipierung benötigt werden, betreffen Anforderungen (z.B. geforderte Lebensdauer), Effekte und Wirkstrukturen (z.B. Kräfte und deren Zusammenhang).

## 4. Vor- und Nachteile der Rechnerunterstützung

Eine effiziente wissensbasierte Rechnerunterstützung sollte den Konstrukteur von Routinearbeiten während des Konstruktionsprozesses entlasten und benötigte Informationen bereitstellen. Dabei muß sich die Art der Unterstützung nach den Bedürfnissen des Menschen orientieren. Um eine Rechnerunterstützung zu ermöglichen muß das Wissen jedoch formalisiert in einem rechnerinternen Modell hinterlegt werden. Diese Formalisierung birgt sowohl Vorteile als auch Nachteile in sich. Als Vorteil kann die einheitliche Vorgehensweise beim Konstruieren, die durch die strenge Formalisierung bedingt wird, angesehen werden. Dadurch werden schon entwickelte Konstruktionslösungen effizienter austauschbar. Die Entwicklungszeiten von

Produkten können reduziert, die Fehlerhäufigkeit verringert und die Qualität der Konstruktionen erhöht werden.

Nachteilig macht sich der Aufwand bei der Aufbereitung des Wissens sowie bei der Pflege und Wartung des Wissens bemerkbar. Bei Expertensystemen sind z.B. sog. Wissensingenieure notwendig um das Wissen von Experten strukturiert in einer Wissensbasis zu hinterlegen und diese zu Pflegen. Da das Vorgehen bei der Konstruktion sehr stark individuell geprägt ist, kann das streng formale Vorgehen den einzelnen Konstrukteur in seiner Arbeit behindern. Bei der Anwendung von neuen Technologien ist in der Anlaufphase mit Einarbeitungsschwierigkeiten zu rechnen, die zu einer erhöhten Fehlerquote, einer Reduzierung der Qualität der Arbeitsergebnisse und zu einer Erhöhung der Durchlaufzeit führen.

## **5. Ausblick**

Die Lösungsansätze zur Wissensverarbeitung für die Weiterentwicklung von Konstruktionssystemen haben das Ziel, den Konstrukteur in seiner Arbeitsweise bestmöglich zu unterstützen. Ein Kernproblem ist hierbei nicht zuletzt aufgrund der ständig zunehmenden Datenflut die Bereitstellung von relevantem und jederzeit aktuellem Konstruktionswissen. Weitere Forschungsaktivitäten ergeben sich auch aus der Zielsetzung, Konstruktionssysteme mit "Intelligenz", d.h. mit der Fähigkeit, aus Benutzerentscheidungen zu lernen, zu entwickeln.

## Literatur

- /BaDS-89/ Barrenscheen, Jörg; Drebing, Uwe ; Sieverding, Hubert:  
Rechnerunterstützte Erstellung von Anforderungslisten. In: VDI-Z 131, Nr. 4, April 1989.
- /Beit-90/ Beitz, Wolfgang: Konstruktionsleitsystem als Integrationshilfe. In: VDI-Berichte Nr. 812, VDI-Verlag, 1990.
- /CuGS-90/ Cunis, R.; Günter, A.; Strecker, H.: Das PLAKON-Buch - Ein Expertensystemkern für Planungs- und Konfigurationsaufgaben in technischen Domänen. Informatik Fachberichte 266, Subreihe Künstliche Intelligenz, Springer Verlag, 1990.
- /Diek-90/ Diekhöner, G.W.: Konstruktions-Know How rechnerunterstützt bereitstellen. In: VDI-Berichte Nr. 812, VDI-Verlag, 1990.
- /Dyll-91/ Dylla, Norbert: Denk- und Handlungsabläufe beim Konstruieren. Konstruktionstechnik München, Band 5, Carl Hanser Verlag, München 1991.
- /Fran-75/ Franke, H.-J.: Methodische Schritte beim Klären konstruktiver Aufgabenstellungen. In: Konstruktion 27, 1975.
- /Fran-92/ Franke, H.-J.: Gibt es eine Logik der Produktgestaltung? In: VDI-Berichte, Nr. 953, VDI-Verlag, Düsseldorf 1992.
- /GCSB-89/ Günter, A.; Cunis, R.; Syska, I.; Bode, H.; Peters, H.: Kontrolle in Konstruktionsexpertensystemen - Einführung und Lösungsvorschlag. Beiträge zum 3. Workshop Planen und Konfigurieren, Arbeitspapiere der GMD 388, St. Augustin, 1989.
- /Groe-90/ Groeger, B.: Ein System zur rechnerunterstützten und wissensbasierten Bearbeitung des Konstruktionsprozesses. In: Konstruktion 42, 1990.
- /Hube-94/ Huber, Ralf: Wissensbasierte Funktionsmodellierung als Grundlage zur Gestaltsfindung in Konstruktionssystemen.. Dissertation. Universität Karlsruhe 1994.

- /Irge-91/ Irgens: Unterstützung der qualitätsgerechten Konstruktion durch ein wissensbasiertes System auf der Basis funktionaler Formelemente. In: VDI-Berichte Nr. 903, Düsseldorf 1991.
- /Kläg-93/ Kläger, Roland: Modellierung von Produktanforderungen als Basis für Problemlösungsprozesse intelligenter Konstruktionssysteme. Dissertation. Universität Karlsruhe 1993.
- /Koll-85/ Koller, Rudolf: Konstruktionslehre für den Maschinenbau. 2. Aufl., Springer-Verlag, 1985.
- /Kutt-92/ Kuttig, D.: Die Funktionsstruktur als integraler Bestandteil des rechnerunterstützten Konstruktionsprozesses. In: Konstruktion 44, 1992.
- /LaMR-92/ Langlotz, Gerd; Meis, Eike; Rude, Stefan: Methodische Beschreibung integrierter Produkt- und Produktionsmodelle - Voraussetzung zur Beherrschung der Komplexität realer Informationszusammenhänge im Unternehmen. In: Kolloquiumsbericht des SFB 346 "Rechnerintegrierte Konstruktion und Fertigung von Bauteilen", Universität Karlsruhe 1992.
- /Lu-91/ Lu, S. C.-Y.: Die Generation nach regelbasierten Expertensystemen - Stand der Forschung in technischen Anwendungen der Künstlichen Intelligenz in der USA. VDI-Berichte 903 - Erfolgreiche Anwendung wissensbasierter Systeme in Entwicklung und Konstruktion. S. 61-90, VDI-Verlag, 1991.
- /NCSE-89/ Nguyen, Huu; Chau, Thuy; Schnupp, Peter; Endres, Albert: Wissensverarbeitung und Expertensysteme, Muenchen, Wien, Oldenbourg 1989.
- /NN-89/ N.N.: Design/IDEF. C.I.T. GmbH, Berlin 1989.
- /PaBe-86/ Pahl, Gerhard; Beitz, Wolfgang: Konstruktionslehre. 2. Aufl. Springer-Verlag, Berlin 1986.
- /Pupp-88/ Puppe, F.: Einführung in Expertensysteme. Studienreihe Informatik, Springer Verlag, 1988.

Gutachter: Wojciech Cholewa