

# Entwicklung von Konfigurierungssystemen

H. Kleine Büning B. Stein

Universität-GH- Paderborn  
FB 17 – Praktische Informatik  
Warburger Str. 100  
4790 Paderborn

## Zusammenfassung

Technischer Fortschritt, Konkurrenzdruck, Verkürzung der Lieferzeiten etc. sind Faktoren, mit denen viele Unternehmen konfrontiert sind. Weil Konfigurierungssysteme die Auftrags erfassung und -prüfung, die Kalkulation etc. unterstützen, nehmen sie im Zusammenhang mit den aufgeführten Faktoren eine wichtige Rolle ein.

Dieses Papier führt anhand von drei verschiedenen Ansätzen in die aktuelle Entwicklung von Konfigurierungstechniken ein. Dabei wird zunächst die allgemeine Idee vorgestellt und im Anschluß daran die Umsetzung dieser Idee an einem konkreten System veranschaulicht. Innerhalb dieses Spektrums werden sowohl Systeme vorgestellt, die im betrieblichen Einsatz sind, als auch Ansätze beschrieben, die sich noch im Forschungsstadium befinden. In allen vorgestellten Konzepten und Systemen nehmen Techniken aus dem Bereich wissensbasierter Systeme eine wichtige Stellung ein.

## 1 Einführung

Die stetige Weiterentwicklung technologischer Produkte führt sowohl im Konsum- als auch im Investitionsgüterbereich neben einem technischen Fortschritt auch zu immer kundenspezifischeren Produkten. Mittlerweile ist für viele Firmen eine starke Orientierung an speziellen Kundenwünschen absolut notwendig, um konkurrenzfähig zu bleiben. Ein bestimmtes Produkt wird nicht mehr in wenigen Standardversionen angeboten, sondern ist in einer großen Anzahl von Varianten erhältlich. Verschärft wird diese Entwicklung noch durch eine kürzere Produktlebensdauer und der Forderung nach kurzen Lieferzeiten [Zimmermann 88]. Diese Problematik stellt eine große Herausforderung in vielen Unternehmensbereichen dar: Neben der technischen Beherrschbarkeit muß vom ersten Angebot bis zum ausgelieferten System die *Konsistenz* eines Auftrags gewährleistet und dessen finanzieller Rahmen kalkulierbar sein.

In diesem Zusammenhang nehmen Konfigurierungssysteme eine zentrale Stellung ein. Sie unterstützen die Auftrags erfassung, die Auftragsprüfung, die Kalkulation, die Arbeitsplanerstellung etc. Wir verstehen hier unter einem Konfigurierungssystem ein Computerprogramm, daß – ausgehend von den Wünschen eines Kunden – ein System zusammenstellt (konfiguriert), so daß die Kundenwünsche erfüllt sind. In bestimmten Fällen reicht anstelle einer automatischen Konfigurierung auch die *Prüfung* eines manuell zusammengestellten Systems aus.

Nach einer kurzen Einführung in die hier interessierende Konfigurierungsproblematik stellt dieses Papier verschiedene Entwicklungen von Konfigurierungssystemen vor. Dabei wird zunächst die zugrundeliegende Idee erläutert und anschließend ein konkretes System, welches diese Idee operationalisiert, vorgestellt.

Auf der einen Seite des Spektrums steht die Idee, ein technisches System als einen hierarchisch aufgebauten Komplex aufzufassen, dessen *Skelett* den Konfigurierungsprozeß steuert. Abschnitt 4 zeigt jedoch, daß man sich nicht unbedingt an dem strukturellen Aufbau eines technischen Systems, sondern auch ausschließlich an dessen *Funktionalität* orientieren kann. Eine aktuelle Entwicklung wird im Abschnitt 5 beschrieben: Mittels einer graphischen Maus & Menü Oberfläche wird von dem Benutzer das interessierende System am Bildschirm zusammengestellt. Konfigurierungsrelevantes Wissen, wie topologische Informationen, Verhaltensbeschreibungen und Nebenbedingungen sind implizit formuliert und werden automatisch aus der Zeichnung gewonnen und abgearbeitet.

## 2 Allgemeine Grundlagen

Unter *Konfigurieren* verstehen wir einen Prozeß, der die Auswahl, Anordnung und Parametrisierung von Komponenten zu einem einer Anforderungsdefinition entsprechenden Gesamtsystem zum Ziel hat [Weiner 91].

Eine *Anforderungsdefinition* besteht aus einer flachen oder strukturierten Auflistung von Funktionen, die ein zu konfigurierendes System erfüllen muß. Sie repräsentiert den formalisierten Kundenwunsch, der als Input für den Konfigurierungsprozeß dient.

Das Resultat der Konfigurierung ist eine modellhafte Beschreibung des Gesamtsystems und wird von uns im folgenden mit dem Begriff *Konfiguration* bezeichnet.

Neben der (Neu-) Konfigurierung unterscheiden wir noch die Konfigurationsprüfung und die Änderungskonfigurierung. Aufgabe der Konfigurationsprüfung ist es, eine manuell erstellte Konfiguration auf ihre Zulässigkeit hin zu überprüfen. Bei einer Änderungskonfigurierung wird die Lösung nicht völlig neu, sondern in Anlehnung an eine gegebene Konfiguration für bekannte Anforderungen bestimmt. Hier tritt das Adaptionproblem auf: Die Leistung der gegebenen Konfiguration erfüllt nicht die aktuellen Anforderungen; sie ist entsprechend durch Hinzufügen, Wegnehmen oder Umordnen von Teilen zu modifizieren.

Obige Definitionen sind vom technischen Verständnis geprägt und auch nur in technischen Gegenstandsbereichen sinnvoll. Ein technischer Gegenstandsbereich grenzt sich von anderen Domänen dadurch ab, daß in der Regel die Konfigurationsobjekte selbst sowie ihre gegenseitigen Abhängigkeiten bekannt sind. Steels [85] umschreibt diesen Sachverhalt wie folgt:

*Technical systems, such as trains, computers, airplanes, power plants, or cars, are artifacts constructed to perform a particular function. In contrast to biological or natural systems which formed the domain of most of the first generation expert systems, they are in principle completely understood. All components are known and the behavior of the whole can theoretically be predicted from the behavior of the parts.*

Die Komplexität eines Konfigurierungssystems wird weniger durch das zugrundegelegte Konzept beeinflusst, sondern hängt vielmehr von den Anforderungen der konkreten Anwendung ab. Dabei werden die nachfolgend skizzierten Punkte als besonders wichtig erachtet (teilweise in [Puppe 90]):

- Die Konfigurationsobjekte sind bereits ausgewählt. Zu bestimmen sind nur die Eigenschaften des Gesamtsystems.
- Falls nur die *Auswahl* der Konfigurationsobjekten und nicht deren Anordnung von Interesse ist, muß räumliches Wissen bei der Problemlösung nicht berücksichtigt werden.
- Gerichtete Beziehungen zwischen Konfigurationsobjekten lassen sich durch Regeln beschreiben. Im Gegensatz hierzu sind zur Formulierung von ungerichteten Beziehungen häufig komplexe Constraint-Mechanismen erforderlich.
- Läßt sich das Wissen einer Anwendungsdomäne so vorstrukturieren, daß während des Konfigurierungsprozesses keine Kontrollentscheidung revidiert werden muß, kann auf aufwendige Mechanismen zur Rücknahme bereits getroffener Entscheidungen verzichtet werden.
- Damit Konfigurierungssysteme keine Insellösung darstellen, besteht die Notwendigkeit, sie an bestehende Systeme für die Auftragseingabe, PPS-Systeme etc. zu koppeln. Hierbei reicht das Spektrum von einer losen Kopplung über Schnittstellen bis hin zu einer festen Kopplung über eine integrierte Wissensbasis.
- Bei der Wartung und Aktualisierung eines Konfigurierungssystems ist entscheidend, ob die Wissensbasis direkt durch einen Anwender oder durch einen KI-Experten gepflegt werden soll.

Einige der hier angesprochenen Punkte werden in den nachfolgenden Kapiteln im Zusammenhang mit einer Realisierung noch einmal aufgegriffen.

## **3 Skelettkonfigurierung**

### **3.1 Konzept**

Die Skelettkonfigurierung ist ein Konfigurierungsprinzip, das sich in erster Linie an der Struktur des zu konfigurierenden Systems orientiert. D.h., Voraussetzung für die Anwend-

barkeit einer solchen Strategie muß eine hierarchische Strukturierbarkeit der Anwendungsdomäne sein. Ein *Strukturmodell* einer Anlage läßt sich dann durch einen hierarchischen Und-Oder-Graphen beschreiben:

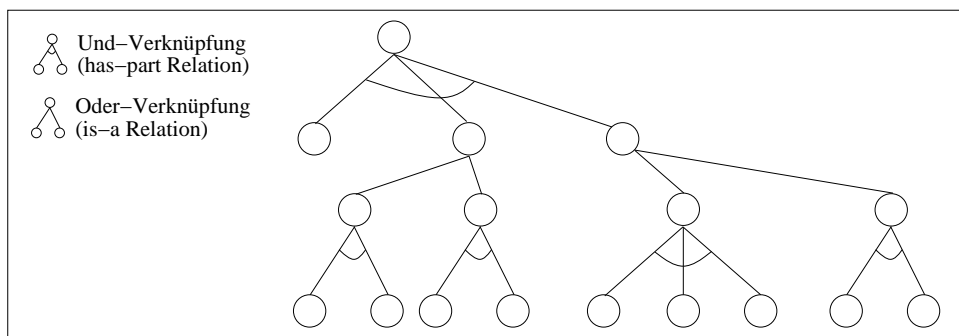


Abbildung 1: Beispiel für einen Und-Oder-Graph

Die Lösungsfindung mit Hilfe eines Und-Oder-Graphen beruht auf der schrittweisen Zerlegung in Teilprobleme (Und-Verknüpfung), wobei ein Teilproblem durch die Auswahl mehrerer Alternativen verfeinert werden kann (Oder-Verknüpfung). Durch diese Repräsentationsform werden grundsätzliche Entscheidungsmöglichkeiten für das Konfigurationssystem expliziter dargestellt, als dies durch einen assoziativen Ansatz mit einer unstrukturierten Menge von Regeln möglich wäre. Auf einem solchen Strukturmodell bauen zwei wesentliche Konfigurierungsstrategien auf:

1. TOP-DOWN-Konfigurierung: Der Konfigurierungsprozeß geht vom Wurzelknoten aus und verfeinert bzw. zerlegt die Objekte.
2. BOTTOM-UP-Konfigurierung: Falls der Benutzer konkrete Komponenten und Parameter spezifiziert, werden diese zunächst instanziiert und entsprechende Schlußfolgerungen gezogen. Anschließend beginnt eine TOP-DOWN-Konfigurierung mit den bereits instanziierten Objekte als zusätzliche Randbedingung.

Typischer Vertreter für dieses Konfigurierungskonzept ist die Entwicklungsumgebung PLAKON [Cunis & et al. 91] und das nachfolgend vorgestellte System WIST [Weiner 87; Kleine Büning & Schmitgen 88].

## 3.2 Realisierung

Das System WIST (Wissensbasierte Stücklistenverarbeitung) ist eine Umsetzung der oben beschriebenen Methodik des Skelettkonfigurierens. Wesentliche Merkmale von WIST sind die einfache Integrierbarkeit in das operationale Umfeld einer Unternehmung sowie das Erstellen und Warten einer Wissensbasis durch Nicht-KI-Spezialisten. WIST wurde in unserer Forschungsgruppe in Zusammenarbeit mit einem Anwendungspartner aus dem Maschinenbau entwickelt.

Kernidee ist die Erweiterung eines Stücklistenkonzeptes um Regeln: Ausgehend von Kundenwünschen (Auftragsparametern) werden mittels parametrisierter Stücklisten eindeutige Stücklisten generiert, welche die Erzeugnisstruktur eines Produktes beschreiben. Im folgenden gehen wir konkreter auf die Wissensrepräsentation und Verarbeitung in WIST ein:

## Wissensrepräsentation

Kern der Wissensrepräsentation bilden Modellstücklisten folgender Art:

Stücklistenkopf			Ident-Nr. 2234-123		
Pos	Sach-Nr.	Menge	Beschreibung	Bedingung	Aktion
10	471-277	25	Schraube		
20	119-200	2	Gehäuse, groß	Motor = stark Land aus {Europa, USA}	
20	119-102	2	Gehäuse	Motor = mittel	
20	119-100	2	Gehäuse, einf.	ELSE	
30	225-301		Geräuschdämm.	Land = USA	Menge = Anz_Geh, Dämmung = ja

Abbildung 2: Beispiel einer Modellstückliste

Eine Modellstückliste unterscheidet sich von einer gewöhnlichen Stückliste dadurch, daß zu jeder Stücklistenposition eine Bedingung und mehrere Maßnahmen angegeben werden können. Im Bedingungsteil einer Stücklistenposition kann mit Hilfe der Parameter, die Kundenwünsche und -vorgaben enthalten, eine Bedingung angegeben werden, welche die Auswahl dieser Stücklistenposition steuert. Im Aktionsteil können Maßnahmen – wie etwa Wertzuweisungen an Parameter, Wertzuweisungen an Felder der aktuellen Stücklistenzeile etc. – formuliert werden, welche bei Gültigkeit dieser Stücklistenposition ausgeführt werden. Eine Stücklistenposition kann man somit als Regel auffassen, die folgendes besagt:

*Wenn* die Bedingung einer Stücklistenzeile erfüllt ist  
*Dann* ist die Stücklistenzeile gültig und  
 alle Maßnahmen im Aktionsteil müssen ausgeführt werden

Um komplexere Beziehungen innerhalb eines Produktes beschreiben zu können, wird ferner die Formulierung von Beziehungen zwischen Parametern mittels *stücklistenexterner* Regeln erlaubt. Diese Regeln sind eindeutig einer Modellstückliste zugeordnet (vgl. Abb. 3). Mit ihnen lassen sich Beziehungen folgender Art formulieren:

*Wenn* Land = USA *und* Gewicht  $\leq$  1000 kg  
*Dann* Versandart = Flugzeug

Durch die Zuordnung *stücklistenexterner* Regeln zu einer Modellstückliste und durch die Verweise innerhalb einer Modellstückliste auf untergeordnete Stücklisten ergibt sich der in Abbildung 3 dargestellte Aufbau der Wissensbasis. Die Ermittlung einer eindeutigen Produktvariante orientiert sich an dieser Struktur.

## Wissensverarbeitung

Nach Auswahl einer Einstiegsstückliste wird ein Auftrag wie folgt bearbeitet: Die Stücklistenzeilen einer Modellstückliste werden sequentiell getestet. Falls der Bedingungsteil einer Stücklistenzeile erfüllt ist, wird diese Zeile als gültig markiert und der zugehörige Aktionsteil ausgeführt. Verweist eine gültige Stücklistenposition auf eine weitere Stückliste, so wird

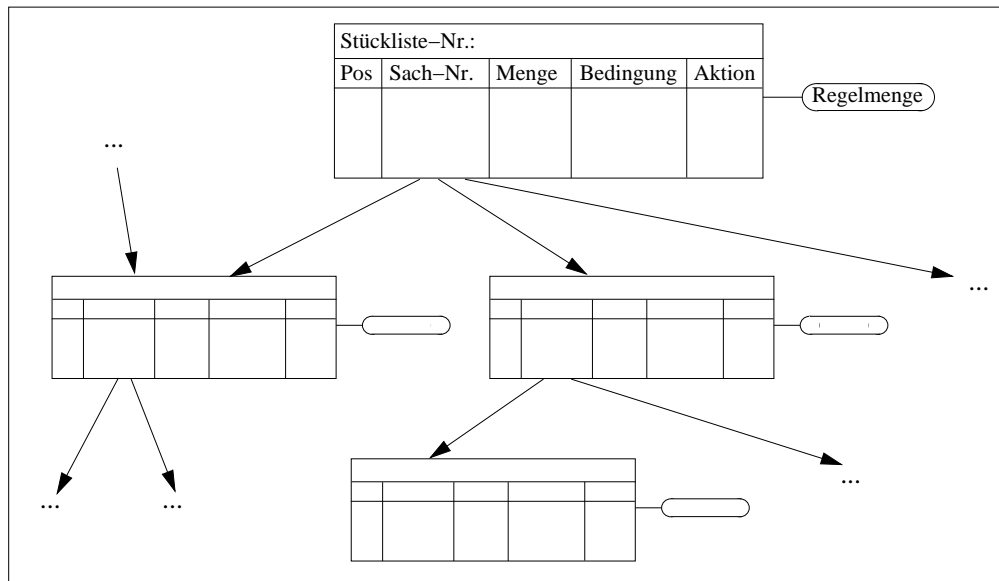


Abbildung 3: Aufbau der Wissensbasis

diese nach der gleichen Vorgehensweise bearbeitet. Die stücklisteninternen Regeln (Stücklistenzeilen) werden also sequentiell abgearbeitet. Die stücklistenexternen Regeln werden hingegen “bei Bedarf” herangezogen. Die stücklistenexternen Regeln werden gleichzeitig datengetrieben und zielgetrieben verarbeitet, d. h. eine stücklistenexterne Regel wird zum einen angewendet, wenn ihr Bedingungsteil erfüllt ist. Damit können von den stücklistenexternen Regeln automatisch Parameter für die weitere Verarbeitung bestimmt werden. Weiterhin wird eine stücklistenexterne Regel rückwärtsverkettend verarbeitet, falls ein Parameterwert benötigt wird und für diesen Parameter eine entsprechende Regel zur Wertermittlung existiert.

Die Erfahrung mit WIST hat gezeigt, daß durch die Strukturierung des Konfigurierungswissens mittels Modellstücklisten und zugeordneter Regeln eine Wissensrepräsentationsform vorliegt, die im besonderen Maße die Anforderungen der betrieblichen Praxis erfüllt. Die Erweiterung der Stücklisten um Regeln hat den Vorteil, daß die Gestalt der Stückliste weitgehend erhalten bleibt und damit die Konstrukteure, die selbständig ohne Einbeziehung eines KI-Spezialisten die Wissensbasis erstellen sollen, ein gewohntes Format vorfinden. Durch den Verzicht auf hybride Wissensrepräsentationsformen und Verarbeitungsmechanismen kann das System durch eine herkömmliche Datenbankanwendung, die um eine Regelverarbeitungs-komponente zu erweitern ist, realisiert werden. Einen Erfahrungsbericht über den erfolgreichen Einsatz dieses Systems in der betrieblichen Praxis findet sich in [Kleine Büning et al. 90].

## 4 Ressourcenorientierte Konfigurierung

### 4.1 Konzept

Ressourcenorientierte Konfigurierung basiert auf der Idee, die Konfigurierungsobjekte als Anbieter und Verbraucher von Ressourcen zu betrachten [Heinrich 91; Stein & Weiner 90].

In der Regel handelt es sich bei diesen Ressourcen um Funktionalitäten, die in der technischen Verwendung dieser Objekte begründet liegen. Abgrenzend zur Skelettkonfigurierung steht nicht die Struktur, sondern ein *funktionales Modell* der Anwendungsdomäne im Vordergrund.

Basis eines solchen funktionalen Modells ist die Beschreibung der Domäne durch eine Menge von Komponenten  $K$ , Funktionalitäten  $F$  und Prädikate  $P$ . Hierbei stellen die Funktionalitäten die Eigenschaften dar, die von den Konfigurationsobjekten angeboten oder gefordert werden. Jeder Funktionalität aus  $F$  ist ein Wertebereich, zwei Abbildungen sowie ein Prädikat zugeordnet. Mit den Abbildungen ist die Verarbeitung der Eigenschaften zwischen den Konfigurationsobjekten definiert. Das Prädikat dient zur Festlegung, unter welcher Bedingung der Angebotswert einer Funktionalität die gestellte Forderung erfüllt. Weiterhin gehören zu jedem Konfigurationsobjekt aus  $K$  eine Angebotsmenge und eine Forderungsmenge, die Elemente der Form (*Funktionalität Wert*) enthalten. Auch die Anforderungsdefinition eines Kunden wird mit Tupeln dieser Form beschrieben.

Das Konfigurierungsproblem besteht nun darin, Konfigurationsobjekte aus  $K$  derart auszuwählen, daß die angebotene Gesamtfunktionalität dieser Objekte die geforderten Funktionalitäten gemäß den zugehörigen Prädikaten erfüllen. Eine wesentlich genauere formale Beschreibung dieses Konfigurierungsproblems findet sich in [Najmann & Stein 91]. U.a. werden hier auf Grundlage des formalen Gerüsts theoretische Zusammenhänge zum Skelettkonfigurieren hergestellt.

## 4.2 Realisierung

MOKON (Modellbasierte Konfigurierung) ist ein Konfigurierungssystem, das die Idee der ressourcenorientierten Konfigurierung operationalisiert.

### Wissensrepräsentation

In MOKON wird explizit zwischen Meta- und Objektwissen unterschieden. Wichtige Wissensarten sind:

- **Priorität der Eigenschaften (Metawissen)**  
Den einzelnen Eigenschaften der Komponenten sind bestimmte Prioritäten zugeordnet. Sie legen fest, in welcher Reihenfolge offene Forderungen befriedigt werden sollen. Mit diesen Prioritätskennzahlen wird von Anfang an ein möglichst hoher Grad der Linearisierung des Konfigurationsprozesses erreicht, d. h. das Rücksetzen auf weit zurückliegende Entscheidungen und die damit verbundenen Neuberechnungen sollen vermieden werden.
- **Attributives Wissen einer Komponente (Objektwissen)**  
Dieses Wissen besteht zum einen aus funktionalem Wissen, wie den Angeboten und Forderungen, die sich aus der Funktionalität der Komponente ableiten lassen, zum anderen aus sonstigen Attributen wie Preis, Lagerbestand etc.:
- **Attributives Wissen einer Eigenschaft (Objektwissen)**  
Hierbei handelt es sich um Wissen, das eine dem kausalen Modell angemessene Be-

Bezeichnung	HD-1	Gehäuse-7
Angebot	Plattenkapazität: (Anzahl * 60) Zugriffsgeschwindigkeit: 20	Sicherheitsnorm: USA Analoge Steckplätze: (Anzahl * 10) Digitale Steckplätze: (Anzahl * 14)
Forderung	Stromwert: (Anzahl * 30) Steckplatz: (Anzahl * 1)	...
Preis	500	1100
Lagerbestand	35	20

Abbildung 4: Komponentenbeschreibung in MOKON

handlung der Eigenschaften ermöglicht. Die Behandlung einer Eigenschaft legt fest, wie aus einer Liste von Forderungen bzw. Angeboten eine Gesamtforderung bzw. ein Gesamtangebot für diese Eigenschaft generiert wird und in welcher Weise diese ermittelten Werte verglichen werden.

## Wissensverarbeitung

MOKON verarbeitet Wissensbasen mit Konfigurationsobjekten, die im wesentlichen durch Angebots- und Forderungsfunktionalitäten beschrieben sind. Durch die Angebots- und Forderungssemantik ist ein funktionales Abhängigkeitsnetz implizit vorgegeben, das den Konfigurierungsprozeß auf folgende Art steuert:

Ausgangspunkt des Konfigurierungsprozesses ist eine Anforderungsdefinition, die aus einer Auflistung von Funktionalitäten besteht, welche das zu konfigurierende System erfüllen soll. Mit diesen Funktionalitäten wird eine *Bilanz* initialisiert, in der Angebote und Forderungen eingetragen werden. Für jede Funktionalitätsforderung innerhalb der Bilanz, der kein geeignetes Angebot gegenübersteht, wird eine Alternativenermittlung angestoßen. Sie versucht eine Alternative aus der Komponentenbibliothek zusammenzustellen, welche die offene Forderung erfüllt. Meistens existieren mehrere Alternativen, die eine offene Forderung befriedigen können. Abhängig von der aktuellen Strategie wird eine Alternative ausgewählt und die Bilanz aktualisiert mit der Konsequenz, daß Angebote und Forderungen, die sich aus den neuen Komponenten ergeben auf der Bilanz nachvollzogen werden müssen. Kann eine Forderung nicht erfüllt werden, so muß im Suchraum zurückgesprungen, d.h. andere Alternativen verwendet werden. Neben der Hill-Climbing-Strategie – lokale Auswahl z.B. der preisgünstigsten Alternative – kann der Benutzer den Suchraum interaktiv verwalten. Der Prozeß des Überprüfens und Aktualisierens der Bilanz wird solange fortgesetzt, bis alle offenen Forderungen erfüllt sind oder festgestellt wird, daß keine Lösung existiert.

In der bisherigen Beschreibung der Wissensverarbeitung wurde offengelassen, in welcher Reihenfolge offene Forderungen abgearbeitet werden. Hierzu zeigt Abbildung 5 ein Abhängigkeitsnetz mit Komponenten (Rechtecke) und Funktionen (Rauten), aus dem sich eine Abarbeitungsreihenfolge ableiten läßt.

Z.B. bietet hier die Komponente des Typs  $K_e$  die Funktionalität  $F_3$ . Bevor nun die Anzahl von  $K_E$  festgelegt wird, sollten die Komponenten, welche die Funktionalität  $F_3$  fordern, bestimmt sein (hier:  $K_c, K_g, K_h$ ).

In MOKON wird aus den Komponentenbeschreibungen automatisch dieses Abhängigkeitsnetz aufgebaut. Innerhalb dieses Netzes werden die Funktionalitäten topologisch sortiert und ihnen eine dieser Sortierung entsprechende Priorität zugeordnet. Werden zyklische



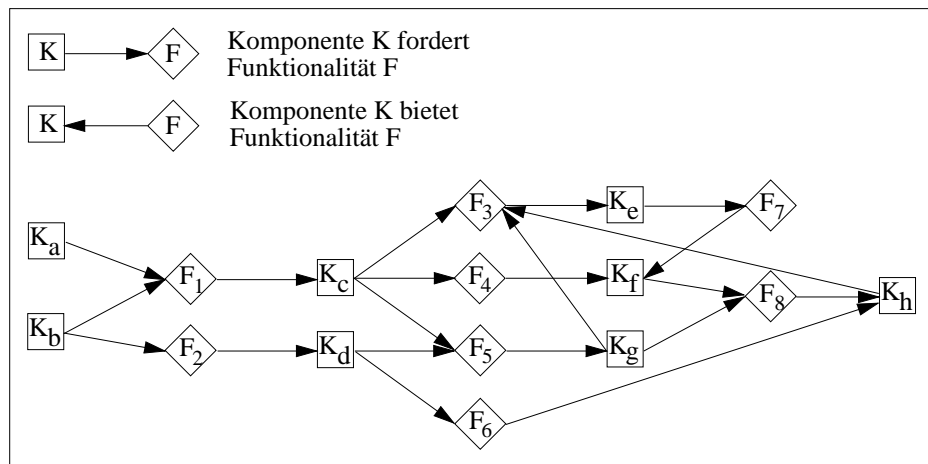


Abbildung 5: Funktionales Abhängigkeitsnetz

Abhängigkeiten festgestellt (starke Zusammenhangskomponenten im Graphen), muß der Benutzer festlegen, mit welcher Funktionalität innerhalb des Zyklus fortgefahren werden soll. Eine Bestimmung der Funktionalitäten, die sich in eine linearisierte Reihenfolge einordnen lassen, die Berechnung dieser Reihenfolge sowie die Identifizierung von zirkulären Abhängigkeiten ist in linearer Zeit möglich [Tarjan 72].

Eine funktionsorientierte Betrachtungsweise zielt auf Domänen, in denen die zu konfigurierenden Komponenten nicht über ein Strukturmodell, sondern über ihre Funktionalität in Beziehung gesetzt werden. Als Anwendungsdomäne dienen u.a. Produkte einer Firma aus dem Telekommunikationssektor [Schmitgen, Stein & Weiner 91]. Ein Vorteil des ressourcenorientierten Ansatzes gegenüber der Skelettkonfigurierung ist, daß die Struktur der interessierenden Anlage nicht berücksichtigt werden muß. Hieraus resultieren insbesondere Vorteile bei der Erweiterung der Wissensbasis: Neue Objekte können einfach hinzugenommen werden – wann und wie sie bei der Konfigurierung Verwendung finden wird vom System automatisch auf Grundlage ihrer Funktionalitäten bestimmt.

## 5 Graphische Konfigurierung

### 5.1 Konzept

Mit Hilfe der graphischen Konfigurierung wird versucht, dem Anwender ein Problembeschreibungsinstrument an die Hand zu geben, das ihn in die Lage versetzt, technische Sachverhalte zu formulieren, die sich nicht mit den bisher vorgestellten Ansätzen beschreiben lassen. D. h., die graphische Konfigurierung basiert – im Gegensatz zu den beiden vorigen Konzepten – nicht unmittelbar auf einer einfachen Idee.

Die Komplexität bei der Konfigurierung eines technischen Systems erreicht eine neue Dimension, wenn neben der Formulierung von Funktionalitäten (vgl. ressourcenorientierte Konfigurierung) auch das *Verhalten* des technischen Systems eine wichtige Rolle spielt. Deshalb könnte man in diesem Zusammenhang auch von "verhaltensorientierter Konfigurierung" oder von "Konfigurierung auf Basis tiefer Modelle" sprechen. Die zu beschrei-

benden technischen Zusammenhänge sind jedoch so komplex, daß für einen Benutzer die Problemformulierung, Wissensakquisition, Wartung etc. praktisch nur noch auf der Ebene einer technischen Zeichnung vollzogen werden kann.

Die Elemente, die dem Benutzer nur als graphische Symbole erscheinen, repräsentieren komplexe Datenobjekte. Sie besitzen Ein- und Ausgänge, um miteinander zu kommunizieren, Funktionalitäten, lokale Verhaltensbeschreibungen etc. Die Charakterisierung dieser Objekte hinsichtlich des lokalen Verhaltens orientiert sich wesentlich an den Konzepten, wie sie z.B. von DeKleer & Brown [84], Struss [88] oder Kuipers [84] vorgestellt und entwickelt wurden. In diesem Zusammenhang ist einer der wichtigsten Punkte die Realisierung des *Lokalitätsprinzips*. Vereinfachend gesagt fordert dieses Prinzip, daß sich das Gesamtverhalten eines technischen Systems aus den Einzelverhalten seiner Komponenten ableiten läßt. Diese Forderung ist deshalb so wichtig, weil bei Konfigurationsproblemstellungen ausgehend von Einzelkomponenten ein zusammenhängendes System erst gebaut wird.

Die genauen Ansätze bei der Wissensrepräsentation und -verarbeitung hängen hier besonders von der Anwendungsdomäne ab. Im nachfolgenden Unterabschnitt wird anhand einer technischen Problemstellung ein konkreter Ansatz vorgestellt.

## **5.2 Realisierung**

### **Anwendungsdomäne**

Unsere Anwendungsdomäne ist der Einsatz von hydraulischen Anlagen. Vereinfachend gesagt, versteht man unter einer hydraulischen Anlage ein System, das Verfah- und Steuerprobleme mithilfe von Flüssigkeiten löst. Ein typisches Beispiel hierfür ist eine hydraulische Hebebühne.

Eine hydraulische Anlage wird gemäß von Kundenwünschen entworfen. Bevor diese Anlage installiert und in Betrieb genommen wird, muß getestet werden, ob sie Fehler enthält und den Anforderungen des Kunden entspricht. Hierbei handelt es sich um ein typisches Konfigurierungsproblem: Gegeben sind bestimmte Anforderungen und es ist zu prüfen, ob die Anlage diese erfüllt. Diese Aufgabe ist sehr komplex und wird manuell von einem qualifizierten Experten durchgeführt [Lemmen 91]. Zur Zeit wird das System "Art Deco" mit dem Ziel entwickelt, wesentliche Teile dieser Konfigurationsprüfung zu automatisieren<sup>1</sup>. Wichtige, d.h. auf andere Domänen übertragbare Konzepte von "Art Deco" sind nachfolgend skizziert.

### **Wissensrepräsentation**

Die Wissensrepräsentation setzt sich zum einen aus verschiedenen Objektwissensbasen, zum anderen aus mehreren, auf den Objekten operierenden, Funktionsmodulen zusammen. U.a. existiert eine Wissensbasis, in der das gesamte graphische Wissen einer Komponente untergebracht ist. Parallel hierzu liegt in einer "technischen" Wissensbasis das hydraulische Wissen der Komponenten. Ein Objekt der technischen Wissensbasis "erbt" u.a. seine

---

<sup>1</sup>Diese Arbeit wird von der DFG gefördert.

zugehörige graphische Information. Darüberhinaus enthält es Wissen über Funktionalitäten (vergleichbar wie in MOKON), Wissen über verschiedene Anschlüsse und Wissen über Verhalten. Dieses Wissen ist bei jedem Objekt in sogenannten “Slots” organisiert. Z.B. enthält derjenige Slot, der für die Beschreibung des Verhaltens notwendig ist, mindestens eine Übergangsfunktion, welche physikalische Eigenschaften der Komponente bezogen auf ihre Anschlüsse beschreibt.

Hierfür wurde ein Konzept entwickelt, das die Formulierung verschiedener Typen physikalischer Constraints sehr einfach macht [Stein, Hoffmann & Lemmen, 1991]. Abbildung 6 zeigt an einem hydraulischen Kreislaufausschnitt, von welcher Art die Verhaltensconstraints sein können.

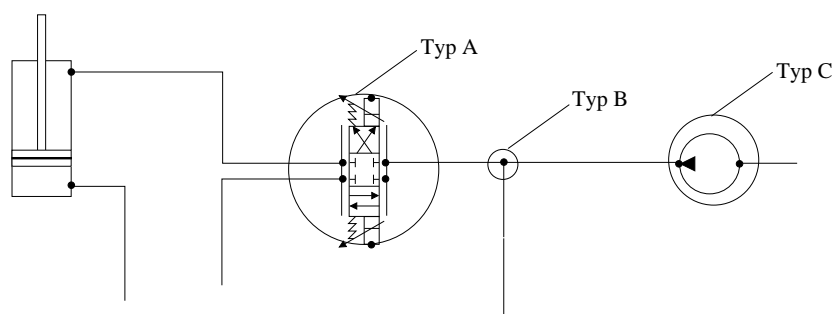


Abbildung 6: Beispiele für verschiedene Constraints

Z.B. wird ein Ventilconstraint (Typ A) durch mehrere Gleichungen der Form  $\Delta P = \frac{2}{5}Q^2$  beschrieben ( $\Delta P$  ist der Druckverlust, der bei einem Fluß von  $Q$  entsteht). Die Kontinuitätsbedingung (Typ B), die an jeder T-Verbindung eingehalten werden muß, wird mit der Übergangsfunktion  $Fluß.1 + Fluß.2 + Fluß.3 = 0$  formuliert. Auch die Festlegung von Konsistenzbedingungen wie  $Durchfluß \leq 120$  für eine Pumpe (Typ C) sind möglich.

## Wissensverarbeitung

Wissensverarbeitung in “Art Deco” bedeutet im wesentlichen die automatische Generierung von Wissensbasen und Constraintverarbeitung.

Dabei läuft die Prüfung eines hydraulischen Kreislaufs zur Zeit folgendermaßen ab: Maus & Menü-gesteuert wählt der Benutzer aus einer Bibliothek hydraulische Komponenten aus, ordnet diese auf dem Bildschirm an und spezifiziert sie mit den Kundenanforderungen. Er verbindet diese Objekte gemäß seiner Vorstellung über die hydraulische Anlage. Schon zu diesem Zeitpunkt werden unzulässige Anschlüsse zurückgewiesen. Weiterhin wird automatisch eine Wissensbasis kreiert, die den gerade entworfenen Schaltkreis mit allen graphischen und technischen Informationen enthält. Auf Grundlage dieser Wissensbasis werden auch alle topologische Informationen, die zur Auswertung der Constraints nötig sind, generiert. Nach der Eingabe durch den Benutzer liegt die vollständige Information vor, um die aktuelle Anlage hinsichtlich ihres Gesamtverhaltens zu überprüfen. Nun startet die Constraintverarbeitung. Die folgende Abbildung zeigt die verschiedenen Phasen:

Innerhalb der ersten Phase werden die gültigen Übergangsfunktionen aus dem Kreislauf bestimmt. Diese werden einem Modul zur algebraischen Propagierung übergeben. Hier werden unter Zuhilfenahme der Kundenparameter alle Druck- und Flußwerte des Gesamt-

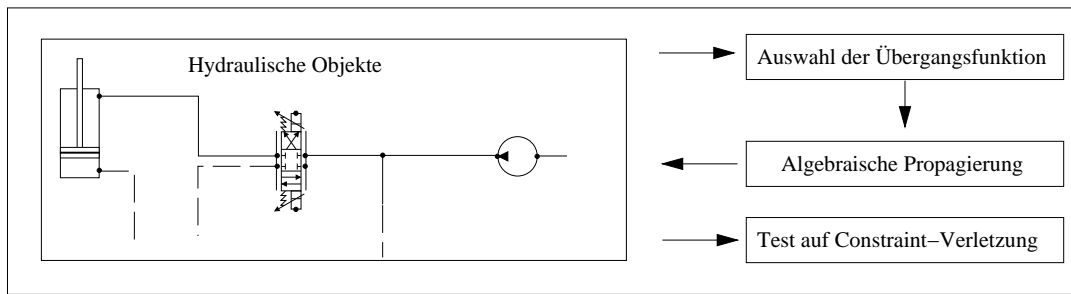


Abbildung 7: Phasen innerhalb der Constraintverarbeitung

systems berechnet. Eine Konfigurationsprüfung ist erfolgreich, wenn sich aufgrund der berechneten Werte keine Widersprüche ergeben.

## 6 Fazit

Nach einer Einführung in die Konfigurierungsproblematik wurden verschiedene Konzepte und Systeme zur Bewältigung von Aufgabenstellungen innerhalb des Problembereichs präsentiert.

Dabei erkennt man, daß die Auswahl und Realisierung eines bestimmten Ansatzes sehr stark von der konkreten Problemstellung beeinflusst wird. Zum einen muß die Aufgabenstellung aus technischer Sicht gelöst werden, zum anderen hat eine Realisierung hinsichtlich der Wissensaquisition und -pflege den betrieblichen Rahmenbedingungen zu genügen: Ein Konfigurierungssystem muß ohne Einbeziehung von KI-Spezialisten wart- und bedienbar sein. Wie eine solche betriebsgerechte Umsetzung aussehen kann, wurde vor allem mit dem System WIST gezeigt.

Losgelöst vom unmittelbaren Einsatz in das operative Umfeld einer Unternehmung wird das System "Art Deco" entwickelt. "Art Deco" ist ein Beispiel dafür, daß mittlerweile sehr komfortable Konfigurierungssysteme realisierbar sind, die auch weniger qualifiziertes Personal in die Lage versetzen, komplexe Konfigurierungsaufgaben selbständig zu bearbeiten.

## 7 Literatur

- [Cunis et al. 91] *Das PLAKON-Buch – Ein Expertensystemkern für Planungs- und Konfigurationaufgaben in technischen Domänen*. Springer Verlag, Berlin, 1991.
- [De Kleer, Brown 84] *A Qualitative Physics Based on Confluences*. Artificial Intelligence 24, 1984.
- [Heinrich 91] *Ressourcenorientierte Modellierung als Basis modularer technischer Systeme*, in: Beiträge zum 5. Workshop “Planen und Konfigurieren”, LKI-M-1/91.
- [Kleine Büning, Schmitgen 88] *Konzept zur Lösung des Variantenproblems in der Stücklistenverarbeitung*. CIM Management 2/1988, Oldenbourg Verlag.
- [Kleine Büning et al. 90] *Wissensbasierte Produktkonfiguration*. VDI-Z 132, Nr. 10, 1990.
- [Kuipers 84] *Commonsense Reasoning about Causality: Deriving Behaviour from Structure*. Artificial Intelligence 24, 1984.
- [Lemmen 91] *Akquisition und Analyse von Wissen zur Inbetriebnahme von hydraulischen translatorischen Anlagen*. Diplomarbeit, Universität Duisburg, Inst. f. Meß-, Steuer-, und Regelungstechnik, 1991.
- [Najmann & Stein 91] *A Theoretical Framework for Configuration*. Erscheint in Proc. der IEAAIE'92, Paderborn, 1992.
- [Puppe 90] *Problemlösungsmethoden für Expertensysteme*. Springer, Berlin, Heidelberg, 1990.
- [Schmitgen, Stein & Weiner 91] *Anlagenkonfiguration – Analyse des Erweiterungsgeschäfts*. Interne Studie, TELENORMA, Frankfurt, 1991
- [Steels 85] *Second Generation Expert Systems*. Future Generation Computer Systems, Vol.1, North-Holland, 1985.
- [Stein & Weiner 90] *MOKON – Eine modellbasierte Entwicklungsplattform zur Konfiguration technischer Anlagen*. Schriftenreihe des Fachbereichs Mathematik 178, Uni -GH- Duisburg, 1990.
- [Stein, Hoffmann & Lemmen 91] *Art Deco: A System which Assists the Checking of Hydraulic Circuits*. ECAI'92-Workshop on Model-based Reasoning, Wien, 1992.
- [Struss 88] *Assumption based reasoning about device models*, in H.W. Früchtenicht: Wissensrepräsentation und Schlußfolgerungsverfahren, Oldenbourg Verlag, München, 1988.
- [Tarjan 72] *Depth First Search and Linear Graph Algorithms*. Society of Industrial and Applied Mathematicians, J. Comp. 1, pp. 146-160, 1972.
- [Weiner 87] *Anforderungen an die Verarbeitung variantenreicher Unterlagen*. Diplomarbeit Universität Karlsruhe, 1987.
- [Weiner 91] *Aspekte der Konfiguration technischer Anlagen*. Dissertation, Universität Duisburg, Institut für praktische Informatik, 1991.
- [Zimmermann 88] *Produktionsplanung variantenreicher Erzeugnisse mit EDV*. Springer Verlag, 1988.