

Entwurfsunterstützung in der Hydraulik



Universität-GH Paderborn
Praktische Informatik, FB 17

Wissensbasierte Systeme haben sich bei der Bearbeitung von Konfigurierungsaufgaben bewährt; eine Reihe von Konfigurierungssystemen befinden sich im praktischen Einsatz und helfen bei der Lösung sehr unterschiedlicher Syntheseprobleme.

Dabei ist, aus menschlicher Sicht, die Grenze zwischen Konfigurierung und Entwurf fließend, und es stellt sich die Frage, inwieweit mit Konfigurierungstechnologie auch die Lösung von anspruchsvollen Entwurfsaufgaben unterstützt werden kann.

Entwurfsunterstützung ist in hohem Maße domänenabhängig. Dieser Beitrag beschreibt, wie auf dem Gebiet der Hydraulik eine leistungsfähige Entwurfsunterstützung realisiert werden kann und stellt das System ^{art}deco vor.

Vereinfachend gesagt versteht man unter Konfigurierung die Auswahl, das Zusammensetzen und Parametrisieren von Einzelkomponenten mit dem Ziel, ein neues System C zu schaffen, das eine gewünschte Funktionalität realisiert. Darüberhinaus können noch weitere Bedingungen an C geknüpft sein: Unter Umständen ist gefordert, bestimmte Kosten nicht zu überschreiten, ein vorgegebenes Gewicht einzuhalten, etc.¹

Der Entwurf einer hydraulischen Anlage fällt auch unter diese informelle Definition: Gegeben ist eine Menge von Komponenten (verschiedene Ventile, Zylinder, Pumpen) sowie eine Benutzeranforderung D , z.B. die Realisierung einer Manipulationsaufgabe. Ziel ist die Erstellung einer Anlage C , die diese Manipulationsaufgabe ausführen kann.

Jedoch handelt es sich bei der geschilderten Aufgabe offensichtlich eher um eine Entwurfs- als um eine Konfigurierungsaufgabe. Warum?

Betrachten wir zum Vergleich die Konfigurierung eines anderen technischen Systems, z.B. einer Telekommunikationsanlage²: Gegeben ist eine Menge von Komponenten (verschiedene Steckkarten, Kabel, Gehäuse, Netzteile) und als Anforderung eine Menge gewünschter Vermittlungsdienste. Ziel ist die Erstellung einer Anlage C , die diese Vermittlungsdienste realisiert.

Beide Synthesaufgaben können als ein Suchproblem aufgefaßt werden und besitzen eine Reihe von Gemeinsamkeiten; folgende Unterschiede verdienen jedoch Beachtung:

1. Die Schnittstellen und Funktionen der Komponenten in den beiden Aufgaben besitzen *aus Sicht des Syntheseprozesses* einen unterschiedlichen Abstraktionsgrad: Während die

Komponenten des obigen Konfigurierungsbeispiels komplexe Dienste wie „ISDN-Anschluß“ oder „Bussteckplatz“ bieten, müssen die Schnittstellen von hydraulischen Komponenten auf der wenig abstrakten, physikalischen Ebene charakterisiert werden.

2. Die Überprüfung, ob eine Telekommunikationsanlage technisch korrekt konfiguriert wurde, erfordert in erster Linie den Abgleich von Funktionalitäten und die Verarbeitung einfacher Constraints. Die Überprüfung einer hydraulischen Anlage erfordert die Aufstellung eines globalen Verhaltensmodells sowie dessen Simulation.
3. Für die möglichen Strukturen von Telekommunikationsanlagen existiert ein relativ enger Rahmen; die Struktur einer hydraulischen Anlage hingegen muß praktisch für jede Aufgabenstellung neu entworfen werden.

Die Komplexität des hydraulischen Entwurfsprozesses hat also mehrere Ursachen. Weiterhin ist auch offensichtlich, daß dieses Problem mit keiner universellen Konfigurierungsmethodik gelöst werden kann.

Die folgenden Abschnitte stellen den Entwurfsprozeß genauer vor und zeigen, daß trotz der aufgezeigten Komplexität eine Entwurfsunterstützung möglich ist.

Entwurf hydraulischer Systeme

Hydraulische Komponenten sind die Bausteine des von uns betrachteten Syntheseprozesses. Sie werden in drei Klassen eingeteilt: Arbeitskomponenten (Zylinder), Steuerkomponenten (Ventile) und Versorgungskomponenten (Pumpen, Leitungen, Tanks).

Zylinder sind die Aktuatoren eines hydraulischen Systems; Ventile wie Druckbegrenzungs-, Drossel-, Rückschlag- oder Proportionalventile steuern Druck und Fluß des hydraulischen Mediums. Pumpen stellen die hydraulische Energie zur Verfügung. Abbildung 1 zeigt die Grundstruktur eines Differentialzylinders und eines Proportionalventils sowie einen kleinen Ausschnitt ihrer Verhaltensbeschreibungen.

Die meisten hydraulischen Komponenten besitzen verschiedene Zustände und ein zustandsabhängiges Verhalten.

Der Entwurfszyklus

Die Anforderungen an eine hydraulische Anlage hängen von der Manipulationsaufgabe ab. Sie können informell und mit Hilfe von Diagrammen beschrieben sein. Z.B. machen Funktionsdiagramme Vorgaben für den Verlauf der Kräfte und Geschwindigkeiten von Zylindern, oder sie legen notwendige Schaltzustände von Ventilen fest. Ausgehend hiervon entwirft der Konstrukteur

¹Verschiedene Definitionen und Sichtweisen des Konfigurierungsbegriffes können u.a. in [8], [4] und [1] gefunden werden.

²Für diese Domäne wurden von uns Konfigurierungssysteme entwickelt [5].

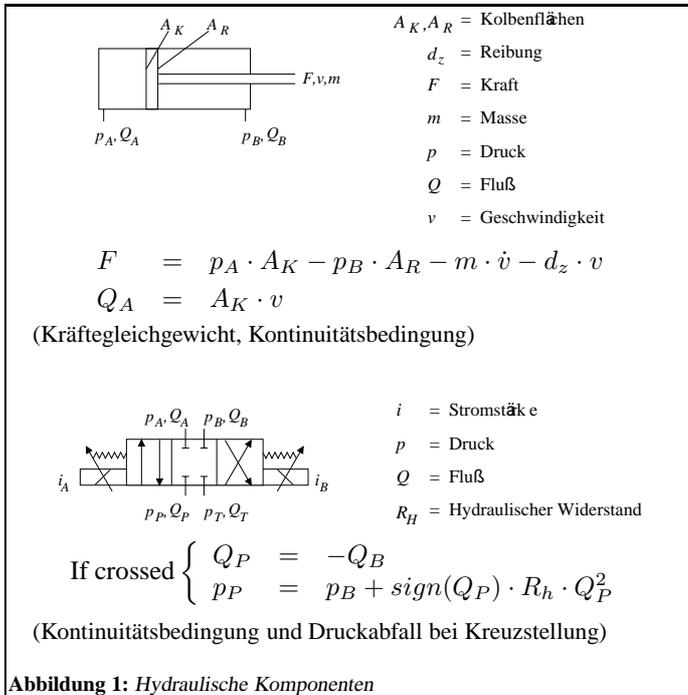


Abbildung 1: Hydraulische Komponenten

die Struktur einer hydraulischen Anlage, wählt die notwendigen Komponenten aus und versucht u.a. das Verhalten der Anlage zu verifizieren [7].

Der kreativste Teil dieses Prozesses ist der Entwurf der Anlagenstruktur; aber auch die Auswahl und Parametrisierung der Komponenten ist anspruchsvoll und erfordert neben Erfahrung fundiertes technisches und mathematisches Know-How. In der Regel ist es nicht möglich, von der Anforderungsmenge D in einem Schritt zu der endgültigen Anlage C zu gelangen. Der Entwurf ist vielmehr ein evolutionärer Prozeß, der aus den Schritten *Synthese*, *Parameterisierung*, *Analyse*, *Evaluierung* und *Modifizierung* verschiedener Konfigurationen C besteht. Abbildung 2 illustriert diesen Zyklus in Anlehnung an Gero [3].

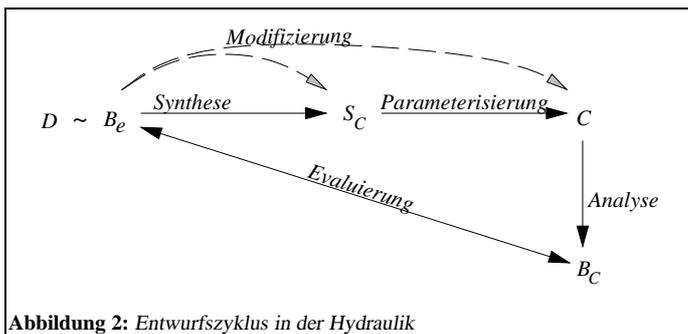


Abbildung 2: Entwurfszyklus in der Hydraulik

B_e bezeichnet das erwartete Verhalten, das in der Hydraulik kanonisch von den Anforderungen D abgeleitet werden kann, S_C bezeichnet die Anlagenstruktur, und B_C bezeichnet das Verhalten, das tatsächlich von C verursacht wird. Der Vergleich von B_e mit B_C soll Antwort auf folgende und andere Fragen geben: Sorgt die Schaltlogik für das gewünschte Verhalten? Sind die Kräfte und Geschwindigkeiten der Zylinder wie geplant? Sind die entstehenden Drücke zulässig?

Betrachtet man den Entwurfszyklus in Abbildung 2, so scheint – schon allein wegen der im Schritt $B_e \rightarrow S_C$ notwendigen Kreativität – eine effiziente Unterstützung des Konstrukteurs aus-

geschlossen. Bei genauerer Untersuchung zeigt sich jedoch, daß der kreative Syntheseschritt und auch der sehr erfahrungsabhängige Modifizierungsschritt für einen *menschlichen* Experten relativ schnell durchführbar sind. D.h., eine nicht unerhebliche Zeit beim Anlagenentwurf steckt im Analyseschritt.

Der Analyseschritt

Ausgangspunkt der Analyse ist ein Hydraulikschaltplan. Dieser Schaltplan muß bezüglich verschiedener Aspekte untersucht werden; von besonderer Wichtigkeit in diesem Zusammenhang ist die Verhaltensanalyse. Hierfür muß neben der Simulation ein *Modellsyntheseproblem* gelöst werden. Mit dem Begriff „Modellsynthese“ seien hier die Schritte zusammengefaßt, die notwendig sind, um ein Modell aufzustellen, das sowohl physikalisch korrekt als auch *lokal* eindeutig ist.

Auch wenn ein Hydraulikschaltplan eine physikalisch sinnvolle Interpretation besitzt, so ist das zugehörige (mathematische) Verhaltensmodell oft nicht lokal eindeutig: Jede Komponente des Schaltplans ist durch eine Menge von Verhaltensconstraints beschrieben, von denen die aktuell gültigen ausgewählt werden müssen. Die Überprüfung der Korrektheit der ausgewählten lokalen Modelle kann in den meisten Fällen nur durch deren Synthese mit anschließender Simulation des globalen Modells geschehen.

Die Mehrdeutigkeit lokaler Verhaltensbeschreibungen hat folgende Ursachen:

1. *Modellierungstiefe*. Hydraulische Komponenten können mit unterschiedlicher Modellierungsgenauigkeit beschrieben werden. Um überflüssigen Simulationsaufwand zu vermeiden, gilt es, abgestimmt auf den restlichen Schaltplan und die Simulationsintention, einen adäquaten Detaillierungsgrad zu finden.
2. *Komponentenverhalten*. Die meisten hydraulischen Komponenten besitzen verschiedene physikalische Zustände, an die ein bestimmtes Verhalten gekoppelt ist. Der tatsächliche Zustand einer Komponente hängt von ihren Wechselwirkungen mit dem Rest des Systems als auch von den aktuellen Parametern ab. Beispiel: Ein Druckbegrenzungsventil kann offen oder geschlossen sein.
3. *Topologie*. Die Topologie eines hydraulischen Systems kann sich mit einem Komponentenzustand ändern. Beispiel: Abhängig von der Schaltposition eines Wegeventils sind unterschiedliche Teile des hydraulischen Netzwerks miteinander verbunden.
4. *Physikalische Grenzwerte*. Selbst für einen festen Zustand kann der Wert einer physikalischen Größe zu unterschiedlichem Komponentenverhalten führen. Beispiel: Ein turbulenter Fluß sorgt bei gleichem Widerstand für einen anderen Druckabfall als ein laminarer Fluß.

Bei Punkt 1 ist in erster Linie die Erfahrung eines Ingenieurs und heuristisches Wissen gefragt [11]. Die Punkte 2, 3 und 4 zeigen, daß der Analyseschritt $C \rightarrow B_C$ auch einen Selektionsschritt $C \rightarrow M_C$ beinhaltet; M_C bezeichnet dabei die Menge der lokalen Verhaltensbeschreibungen, die für die aktuelle Situation des hydraulischen Systems gültig sind. Oft ist ein zusätzlicher Zyklus von Modellselektion und Simulation notwendig, um dieses

- *Wertepropagierung*. Gegeben ist eine Menge von symbolischen und numerischen Relationen sowie ausgezeichnete Parameter, die mit initialen symbolischen oder numerischen Werten belegt sind; Lösung ist eine Parameterbelegung, die eine Obermenge der initialen Belegung ist und alle Constraints erfüllt.
- *Regelverarbeitung*. Regelverarbeitung in der Form von Forward-Chaining ist notwendig, um u.a. die Modellauswahlconstraints zu verarbeiten. Das bedeutet, daß durch Regeln nicht nur Beziehungen zwischen Parametern, sondern auch zwischen Mengen von Verhaltensbeschreibungen abzubilden sind.
- *Algebraische Umformung*. Algebraische Umformungen sind notwendig, um implizit definierte numerische Constraints umzuformen, und um Normalformen für die numerischen Verfahren zu generieren.

Das *art deco* System

*art deco*³ ist ein System, das den Analyseschritt im hydraulischen Entwurfsprozeß automatisiert. Zentrale Konzepte dieses Systems sind

1. die effiziente Suche in \mathcal{M}_C , dem Raum der globalen Verhaltensmodelle, und
2. der direkte Problemzugang durch die graphische Formulierung hydraulischer Analyseprobleme.

D.h., vom Prinzip bleibt die Umsetzung einer Anforderungsspezifikation D in eine Konfiguration C Aufgabe des Konstrukteurs – jedoch wird diese Aufgabe durch *art deco* stark vereinfacht. Dieser Abschnitt stellt die wichtigsten Konzepte vor.

Problemformulierung

Mit dem Begriff „Problemformulierung“ fassen wir die Schritte zusammen, die notwendig sind, um eine Instanz des Analyseproblems $C \rightarrow B_C$ zu formulieren. In erster Linie gilt es, folgende Frage zu beantworten: Wie kann ein Anwender sein Analyseproblem in akzeptabler Zeit spezifizieren?

Im klassischen Entwurfsprozeß stellen Anlagenzeichnungen die wichtigste Arbeitsunterlage des Ingenieurs dar. Um möglichst nahe an diese mächtige und für Ingenieure vertraute Abstraktionsebene heranzukommen, wurde in *art deco* eine sehr weitgehende graphische Problemformulierung realisiert: Eine Anlage wird konstruiert, indem Komponenten aus einer Bibliothek ausgewählt, auf der Zeichenfläche platziert und miteinander durch Leitungen verknüpft werden. U.a. wird während des Zeichenprozesses die Kompatibilität von Komponenten und Leitungsanschlüssen geprüft. Gleichzeitig entsteht neben der Zeichnung ein Topologiemodell der Anlage, das die Grundlage für den Analyseschritt bildet.

Abbildung 5 zeigt die Arbeitsoberfläche von *art deco* während ein Schaltplan gezeichnet wird; im Hintergrund befindet sich die

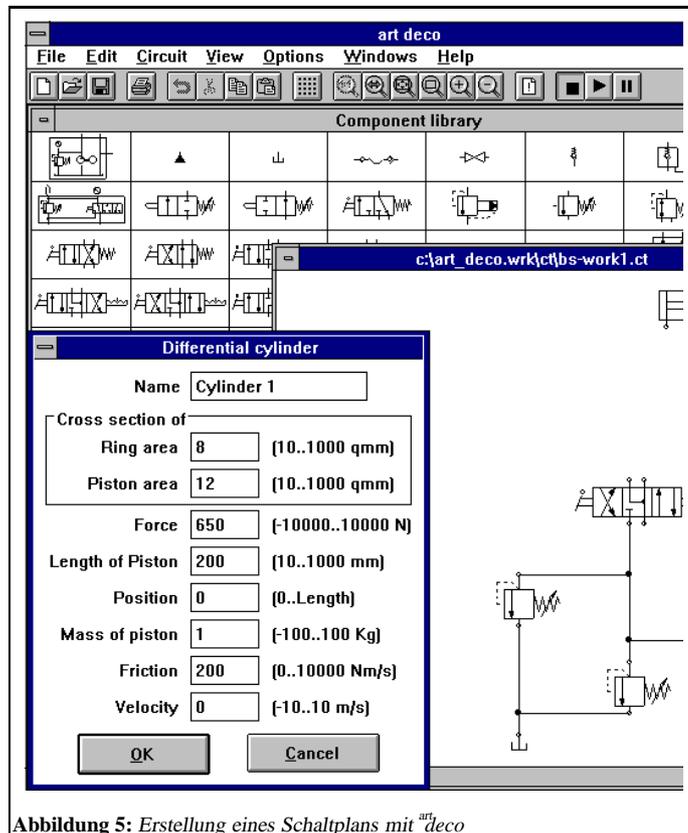


Abbildung 5: Erstellung eines Schaltplans mit *art deco*

Komponentenbibliothek, im Vordergrund ist die Dialogbox eines Differentialzylinders zu sehen.

Nach dem Starten des Inferenzprozesses wird von *art deco* eine konsistente Belegung für alle unbekannt Zustände und Parameter gesucht. Abbildung 6 zeigt einen Kreislauf, bei dem eine solche Belegung bestimmt werden konnte.

Inferenz

Mit einem Schaltplan als Eingabe untersucht *art deco* das System hinsichtlich folgender Fehler:

1. syntaktische Fehler (z.B. offene Leitungen)
2. geometrische Fehler (z.B. falsche Anschlüsse)
3. logische Fehler (z.B. Kolbenbewegungen, die im Widerspruch zu Ventilschaltstellungen stehen)
4. Dimensionierungsfehler (z.B. Pumpen, deren Leistungsbereich überschritten wird)

Falls keiner dieser Fehler vorliegt, werden von *art deco* die Zustände der Komponenten bestimmt und – zusammen mit unbekannt Geschwindigkeiten, Kräften oder Komponentenparametern – eine Druck- und Flußverteilung berechnet.

Aus Sicht der Problemspezifikation und der Wissensverarbeitung wird in *art deco* zwischen folgenden Constraintklassen unterschieden:

- *Anschlußconstraints*. Anschlußconstraints legen fest, ob und wie zwei Komponenten miteinander verbunden werden können. Die Verarbeitung dieser Constraints umfaßt die Prüfung aller Anschlußtypen, mechanischen Kopplungen

³Die Entwicklung von *art deco* hat seinen Ursprung in einem gemeinsamen DFG-Projekt des Instituts „MSRT“, Gerhard-Mercator-Universität - GH Duisburg (Prof. Dr. H. Schwarz) und der Arbeitsgruppe „Wissensbasierte Systeme“, Universität - GH Paderborn (Prof. Dr. H. Kleine Büning) [6].

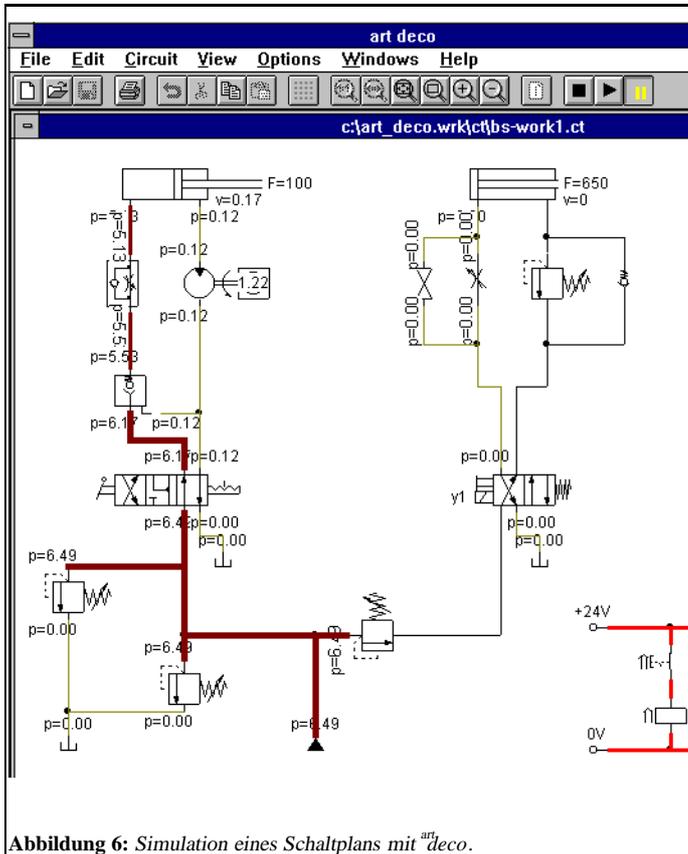


Abbildung 6: Simulation eines Schaltplans mit *art deco*.

und offenen Leitungen. Diese Prüfung ist einfach und kann losgelöst von der übrigen Constraintverarbeitung durchgeführt werden.

- **Topologische Constraints.** Topologische Constraints resultieren aus der Anlagenstruktur und definieren gemeinsame Variablen der lokalen Komponentenbeschreibungen.
- **Verhaltensconstraints.** Verhaltensconstraints definieren das Verhalten einer einzelnen Komponente und können vom Anwender verändert werden. Sie bestehen aus Relationen, die über numerische und symbolische Parameter definiert sind.
- **Modellauswahlconstraints.** Modellauswahlconstraints legen fest, welche Verhaltensalternative mit einem bestimmten Komponentenzustand verbunden ist. Eine Verhaltensalternative faßt eine Menge von Verhaltensconstraints zusammen.
- **Anforderungsconstraints.** Anforderungsconstraints umfassen interne und externe Restriktionen. Interne Anforderungsconstraints prüfen, ob eine Beschreibung physikalisch widerspruchsvoll ist; mit externen Anforderungsconstraints können Benutzeranforderungen in der Form einfacher Regeln und Relationen modelliert werden. Im Gegensatz zu Verhaltensconstraints wird diese Klasse von Constraints destruktiv propagiert.

Nach der Auswertung der Anschlußconstraints und der topologischen Constraints wird auf Basis der Schaltplanspezifikation C in einem Zyklus bestehend aus Modellsynthese und Simulation eine konsistente Verhaltensbeschreibung gesucht. Bei der

Simulation kommen die im vorigen Abschnitt aufgeführten Inferenzmethoden zum Einsatz. Die Steuerung der Inferenz – genauer: die effiziente Abarbeitung des Synthesesuchraums \mathcal{M}_C ist jedoch entscheidend dafür, ob eine Analyse überhaupt in akzeptabler Zeit durchgeführt werden kann. Es ist offensichtlich, daß \mathcal{M}_C neben den Modellauswahlconstraints durch zusätzliche Syntheserestriktionen massiv eingeschränkt werden muß.

Syntheserestriktionen können von Komponenten abgeleitet werden, deren Verhaltensbeschreibung physikalisch widerspruchsvoll sind. Beispiel: Eine Pumpe kann keinen Fluß erzeugen, wenn ein in Reihe geschaltetes Druckbegrenzungsventil geschlossen ist. Hier wird folgendes Problem deutlich: Syntheserestriktionen hängen von den physikalischen Verhaltensbeschreibungen und der Art des Einbaus der Komponenten ab; ein Vorausdenken von solchen Restriktionen ist also nur im eingeschränkten Maße möglich.

Ein Ausweg hieraus ist die *automatische Erzeugung von Syntheserestriktionen* unter Berücksichtigung der Anlagenstruktur und der eingesetzten Komponenten. In *art deco* existieren dafür zwei Mechanismen: Abhängigkeitsverwaltung und Strukturanalyse. Die Abhängigkeitsverwaltung führt für alle Zustandsgrößen und Parameter einer Anlage Buch, aufgrund welcher Verhaltensannahmen der aktuelle Wert ermittelt wurde. Tritt bei der Auswertung eines Constraints ein Konflikt auf, so wird aus diesen Annahmen ein Nogood gebildet. Im Rahmen der Strukturanalyse werden ebenfalls Nogoods generiert; dies geschieht jedoch vorab, d.h. vor der Auswertung der Verhaltensconstraints z.B. durch eine Flußrichtungsanalyse.

Die in *art deco* realisierte Abhängigkeitsverwaltung findet also die Ursachen eines Konfliktes und ermöglicht so eine gezielte Korrektur des globalen Verhaltensmodells: Diejenigen lokalen Verhaltensmodelle, die nicht mehr gültig sind, werden zurückgezogen. Hierfür werden während des Inferenzprozesses Abhängigkeiten sowohl zwischen einzelnen Constraintparametern, aber auch zwischen verschiedenen Constraintmengen generiert⁴.

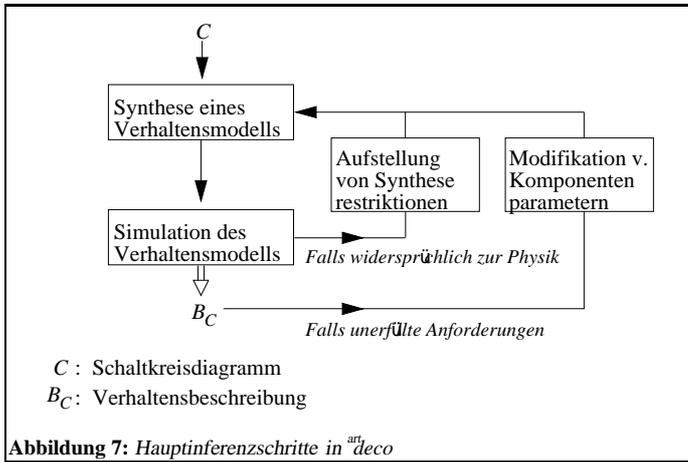
Ein weiterer wesentlicher Punkt hinsichtlich der Effizienz bei der Suche ist das Wiederaufsetzen nach einem Konflikt. Die Annahmen, die zu dem Konflikt führten, bilden einen Nogood – es stellt sich jedoch die Frage, mit welcher Annahmenmenge die Suche fortgesetzt werden soll. Daß mit der Nogoodinformation im Sinne eines abhängigkeitgesteuerten Backtracking (dependency-directed backtracking [10]) im Lösungsraum zurückgesprungen werden kann ist klar. Weil oft jedoch mehrere Annahmen gleichzeitig in einen Konflikt verwickelt sind und weil pro Annahme mehrere ungetestete Alternativen vorliegen können, sollte auch hier noch zusätzliches Auswahl- bzw. Korrekturwissen vorhanden sein. In *art deco* kommen an dieser Stelle Domänenwissen und selbstentwickelte Heuristiken zum Einsatz.

Abbildung 7 beschreibt das prinzipielle Vorgehen bei der Lösungssuche.

Neben den oben beschriebenen Techniken zum schnellen Auffinden einer Lösung sind in *art deco* verschiedene Konzepte zur Wiederverwendung berechneter Ergebnisse realisiert:

1. Instanzen numerischer Teilprobleme einschließlich Lösung werden in einem Hash-Array gespeichert.

⁴Eine parallele Verwaltung von Annahmenmengen wie bei einem ATMS [2] wurde nicht für sinnvoll erachtet.



2. Vollständig spezifizierte Instanzen eines Analyseproblems einschließlich Simulationsergebnis werden ebenfalls in einem Hash-Array gespeichert.
3. Konnte für eine Anlage ein global konsistentes Verhaltensmodell gefunden werden, so wird dieses Verhaltensmodell mit der Anlagenstruktur als Schlüssel für eine spätere Wiederverwendung in einer Datei abgelegt.⁵

Module von ^{art}deco

Kernmodule von ^{art}deco sind die Benutzerschnittstelle, das Objektsystem, sowie das Inferenzsystem.

Die Benutzerschnittstelle besteht aus drei Teilen: Einer Zeichenfläche, auf der ein Konstrukteur seine Entwürfe durch Manipulation graphischer Objekte erstellt, einem Aktions-Interpreter, der die graphischen Aktionen auswertet und einem CAD-Modul, das speziell für die Hydraulik entwickelte Graphik-Routinen bereitstellt.

Im Objektsystem sind die ^{art}deco-Wissensbasen realisiert. Die graphische Wissensbasis legt u.a. das Aussehen einer Komponente und die auf ihr anwendbaren Aktionen fest. Die technische Wissensbasis definiert mit Hilfe einer Beschreibungssprache für jede Komponente ihr physikalisches Verhalten.

Das Inferenzsystem von ^{art}deco enthält alle Inferenzmechanismen, die notwendig sind, um die ^{art}deco-Constraints zu verarbeiten. Eine übergeordnete Kontrollinstanz steuert das Zusammenspiel der einzelnen Mechanismen und realisiert den Prozeß der Modellsynthese.

Realisierung

Bei der Realisierung von ^{art}deco wurde nicht die Strategie verfolgt, existierende Shells, Mathematikpakete und ein CAD-System miteinander zu koppeln, sondern es wurde ein programmiersprachenbasierter Weg gewählt. Folgende Gründe waren ausschlaggebend für diese Entscheidung:

- Ein Hydraulikschaltplan in einem CAD-System muß die Semantik der Komponenten berücksichtigen, um tatsächlich die Anlagenstruktur im Sinne der Hydraulik abzubilden. Für

⁵Die Wiederverwendung von gespeicherten Verhaltensmodellen bei Gleichheit der Anlagenstruktur ist eine Heuristik; die Abbildung von der Anlagenstruktur auf das globale Verhaltensmodell ist nicht eindeutig.

die Hydraulikkomponenten im CAD-System müssen technische Eigenschaften und Verhalten spezifizierbar sein.

- Die Integration von Domänenwissen in ein CAD-System ist schwierig – doch genau das ist notwendig: Unerlaubte Benutzerentscheidungen sollten so früh wie möglich erkannt werden, um überflüssigen Simulationsaufwand zu vermeiden.
- Die Constraintformulierung, die Constraintverarbeitung wie auch das Truth-Maintenance-System sollten aus Akquisitions- und Effizienzgründen genau auf die Constraints in der Domäne zugeschnitten sein.
- Die Integration von Domänenwissen in fertige Numerikbibliotheken ist kaum möglich.

Weil diese Probleme nicht in einem Schritt lösbar sind und weil Anwender so früh wie möglich in den Entwicklungsprozeß integriert werden müssen, verlief die bisherige Entwicklung von ^{art}deco in zwei Phasen.

In einer Prototypphase wurde untersucht, inwieweit der Entwurfsprozeß in der Hydraulik überhaupt unterstützt werden kann. Parallel hierzu verliefen Forschungen bzgl. der Problemformulierung, den notwendigen Inferenzverfahren, adäquaten Datenstrukturen, einer Komponentenbeschreibungssprache sowie auch dem Zusammenspiel der verschiedenen Mechanismen. Während dieser Phase wurde das Expertensystemwerkzeug KEE eingesetzt. Die Algorithmen waren in Common Lisp geschrieben, die Wissensbasen setzten auf das KEE-Object-System auf, und die Benutzerschnittstelle wurde mit Hilfe des KEE-Picture-Systems realisiert. Neben der Entwicklung und dem Test der Konzepte stand mit der KEE/Lisp-Version von ^{art}deco eine realistische Diskussionsbasis zwischen Anwendern und Entwicklern zur Verfügung.

In der Reimplementierungsphase wurde aus den oben genannten Gründen – nicht zuletzt aber auch wegen der Portabilität – auf den Einsatz von Shells verzichtet. Die Benutzerschnittstelle von ^{art}deco besteht aus einem kleinen, plattformabhängigen Graphikkern, auf den eine plattformunabhängige „semantische“ Graphikschnittstelle aufsetzt. Diese Graphikschnittstelle stellt leistungsfähige Funktionen zur Verfügung, die auf den Anwendungsbereich abgestimmt sind und die die Benutzeraktionen interpretieren sowie die technische und die graphische Wissensbasis manipulieren. Die Wissensbasen sind in einem selbstentwickelten Objektsystem abgelegt. Für die Beschreibung und die Verarbeitung der erforderlichen Wissensarten wurde eine Constraintrepräsentation einschließlich speziell zugeschnittener Verarbeitungsmechanismen entwickelt.

Die aktuelle Version von ^{art}deco ist in C, C++ und Common Lisp realisiert. Die Benutzerschnittstelle ist in C++ und die numerischen Routinen sowie das Objektsystem sind in C geschrieben; die Algorithmen zur symbolischen Constraintverarbeitung (Abhängigkeitsverwaltung, Inferenzsteuerung, algebraische Routinen, Propagierungsmethoden) sind in Common Lisp geschrieben.

Seit Mai diesen Jahres wird von der FESTO DIDACTIC eine Schulungsversion von ^{art}deco vertrieben. Eine Erweiterung auf die Pneumatik sowie eine Aufbauversion für Hydraulik befinden sich in Vorbereitung.

Literatur

- [1] D. Brown and B. Chandrasekaran. An Approach to Expert Systems for Mechanical Design. In *Trends and Applications '83*. IEEE Computer Society, NBS, Gaithersburg, MD, 1983.
- [2] J. de Kleer. Problem Solving with the ATMS. *Artificial Intelligence*, 28:197–224, 1986.
- [3] J. S. Gero. Design Prototypes: A Knowledge Representation Scheme for Design. *AI Magazine*, 11:26–36, 1990.
- [4] A. Günter, editor. *Workshop "Modelle beim Konfigurieren"*, LKI -M-3/93. Universität Hamburg, Labor für Künstliche Intelligenz, 1993.
- [5] H. Kleine Büning, D. Curatolo, and B. Stein. Knowledge-Based Support within Configuration and Design Tasks. In *Proc. ESDA '94, London*, pages 435–441, 1994.
- [6] H. Kleine Büning and B. Stein. Supporting the Configuration of Technical Systems. In M. Hiller and B. Fink, editors, *Second Conference on Mechatronics and Robotics*. IMECH, Institut für Mechatronic, Moers, IMECH, 1993.
- [7] R. Lemmen. *Zur automatisierten Modellerstellung, Konfigurationsprüfung und Diagnose hydraulischer Anlagen mit dem Beispiel tankdruckangehobener Differentialzylinderantriebe*. Number 503 in Fortschrittsberichte VDI, Reihe 8: Mess-, Steuer- und Regelungstechnik. VDI-Verlag, Düsseldorf, 1995.
- [8] F. Puppe. *Problemlösungsmethoden für Expertensysteme*. Springer-Verlag, 1990.
- [9] H. R. Schwarz. *Numerische Mathematik*. B. G. Teubner, Stuttgart, 1986.
- [10] R. M. Stallman and G. J. Sussman. Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis. *Artificial Intelligence*, 9:135–196, 1977.
- [11] M. Suermann. *Wissensbasierte Modellbildung und Simulation von hydraulischen Schaltkreisen*. Diploma thesis, Universität-GH Paderborn, FB 17 Mathematik / Informatik, 1994.