Supporting Hydraulic Circuit Design
by Efficiently Solving the Model Synthesis Problem

Benno Stein

Department of Mathematics and Computer Science—Knowledge-based Systems
University of Paderborn, D–33100 Paderborn, Germany
email: stein@uni-paderborn.de

Keywords: automated modeling, design, search, simulation

Abstract

The design of hydraulic systems is founded on deep physical connections. It needs creativity and, at the present time, it cannot be automated.

Nevertheless, powerful design support in hydraulics is still possible. This paper presents a new view to the design of hydraulic systems and shows how sophisticated subtasks of the hydraulic design procedure can be tackled.

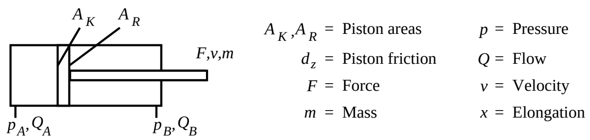
We identify the "model synthesis problem" as a crucial factor in the design procedure and develop methods to solve the problem. Although our approach is tailored to hydraulics, it reveals the nature of model synthesis, and thus it provides insights to tackle model synthesis in other domains as well.

1 Design of Hydraulic Systems

This section replaces an introduction. Its purpose is twofold: To convey a rough idea of hydraulic circuit design, and to explain why such a sophisticated task can be supported at all.

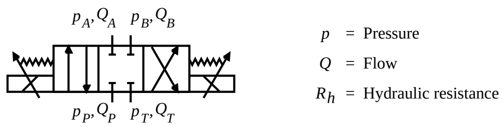
The starting point for a design process is a task that shall be performed by hydraulics. This task might be a lifting problem, the actuation of a press, or the realization of a robot's kinematics. The result of the design process is a system consisting of hydraulic components.

Hydraulic components are the building blocks in this design process; they can be divided into three classes: (*i*) cylinders, which transform hydraulic energy into mechanical energy, (*ii*) various forms of valves, which control flow and pressure of the hydraulic medium, and (*iii*) service components such as pumps, tanks, and pipes, which provide and distribute the necessary pressure p and flow Q. Figure 1 shows the basic structure of a differential cylinder and a proportional valve respectively. Below these figures a small extract of their behavior description is given.



State: "piston_movable"
Behavior:

$$F = p_A A_K - p_B A_R - m\,\dot{v} - d_z v \quad \text{(balance of forces)}$$
$$Q_A = A_K\, v \quad \text{(continuity condition)}$$



State: "crossed_position"
Behavior:

$$p_P = p_B + sign(Q_P)\, R_h\, Q_P^2 \quad \text{(pressure drop)}$$
$$Q_P = -Q_B \quad \text{(continuity condition)}$$

Figure 1: Hydraulic cylinder and valve.

Note that hydraulic components have states that determine which part of their behavior description is actually valid.

1.1 The Generic Design Procedure

The demands D at a hydraulic system result from the task to be performed and are specified by several diagrams. These diagrams indicate the course of the forces and velocities of the cylinders, the switching positions of the valves, and other dependencies. Based on such diagrams, an engineer has to design the system's topology, select the necessary components, and analyze the stationary and the dynamic behavior of the system [Lemmen, 1994].

The most creative part in the design process is the creation of a system's topology [Kleine Büning *et al.*, 1994]. The selection and parameterization of the components are also demanding and need a lot of experience as well as technical and mathematical know-how. Due to the complexity of the design process, one cannot get from the demand set D to the readily designed system C in a single step. Rather, there is a cycle of synthesis, parameterization, analysis, evaluation, and modification of intermediate designs. In accordance with Gero, Figure 2 illustrates this cycle.

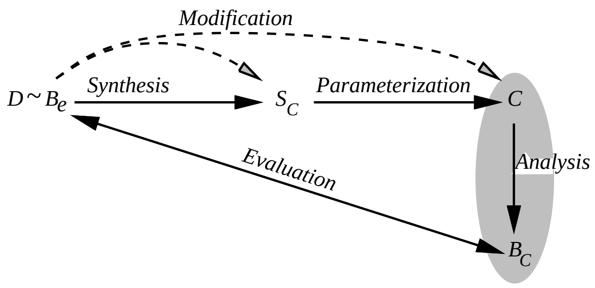


Figure 2: An abstract view of hydraulic system design.

In this cycle B_e denotes the expected behavior that can be derived canonically from D, S_C denotes the hydraulic system's structure, and B_C denotes the behavior that is produced by the designed system C. A comparison of B_e and B_C answers questions of the following type: Does the switching logic realize the desired behavior? Will the piston velocities and forces be as prescribed? Are the maximum pressure values permissible?

Let us again consider the design procedure shown in Figure 2. A far-reaching design support seems to be hardly possible because of the creativity that is needed within the synthesis step $B_e \longrightarrow S_C$. On second thought, however, the situation is not hopeless: Neither the creative synthesis step nor the experienced-based modification steps are time-consuming for a *human expert*. Put another way, a reduction of just the analysis step's complexity would lead to a noticeable simplification of the entire design procedure.

This observation will guide our philosophy of a design support—rather than automating the entire design process we will concentrate on those tasks that ground on the *analysis* of hydraulic systems.

2 Automated Behavior Analysis of Hydraulic Systems

As argued in the previous section, an automated analysis of hydraulic systems is the key factor for a design support in hydraulics. This section investigates the analysis step in greater detail and shows that the automation of this step requires the solution of a "model synthesis problem".

2.1 From Local to Global Behavior

Loosely speaking, hydraulic systems analysis takes a circuit diagram as input and produces a behavior description of the entire circuit. For this job, aside from the simulation problem, a model synthesis problem has to be tackled as well.[1]

Model synthesis consists of all steps that are necessary to set up a model which is both correct in a physical sense and locally unique.

Note that even though a circuit diagram has a useful physical interpretation, its mathematical description cannot be derived in an ad-hoc manner: Each component of the circuit is defined by a *set* of behavior constraints from which the relevant ones must be selected. Verifying the correctness of a local behavior description needs an expensive simulation of the *global* system in most cases.

The indeterminacy of local behavior descriptions originates from the following causes:

1. *Component States.* Most components have different physical states, each coupled with a particular behavior description. The actual validity of a state depends on the entire system and the actual input parameters. Example: A pressure relief valve may be in the state "opened" or "closed".
2. *Topology.* A hydraulic system's topology can change with a component's state. Example: Depending on its switching position a proportional valve connects different parts of a hydraulic network.
3. *Physical Thresholds.* Even for a fixed state the direction or the absolute value of a physical quantity, which is a-priori unknown, may cause different behaviors of a component. Example: A turbulent flow is described by another pressure drop law than a laminar flow.

[1] There is particular research in connection with model composition problems (cf. [Nayak, 1992; Falkenhainer and Forbus, 1991; Iwasaki and Levy, 1993]). Note that the mentioned as well as related work focuses on the construction or selection of adequate models with respect to different tasks (simulation, diagnosis) or different levels of granularity. This is not the case here: Although both the task and the level of granularity are given, there is a synthesis problem, which results from the indeterminacy of local behavior descriptions in the hydraulic domain.

The above points reveal that the analysis step $C \rightarrow B_C$ also contains a selection step $C \rightarrow M_C$, where M_C denotes a set of behavior descriptions that are chosen respecting the actual situation of the hydraulic system. To solve this selection problem a cycle of model selection and model simulation is necessary. Figure 3 illustrates this cycle.

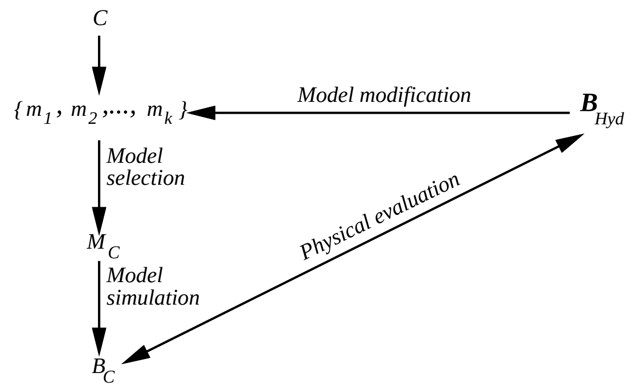


Figure 3: What happens in the hydraulic analysis step.

The set $\{m_1, \ldots, m_k\}$ in Figure 3 comprises the behavior descriptions, say, models of all components. From this set a subset is selected ($= M_C$), simulated ($\Rightarrow B_C$), and compared to $\boldsymbol{B}_{Hyd}$ which stands for the universal behavior laws of hydraulics. In the case that the simulated behavior B_C is physically contradictory or undetermined, M_C must be modified.

This cycle of selection, simulation, evaluation, and modification is necessary to get a grip on the model synthesis problem. It is solved when the physically correct behavior descriptions according to the points 1, 2, 3 and $\boldsymbol{B}_{Hyd}$ are determined.

Remarks. Model synthesis constitutes an inherently combinatorial problem in the hydraulic domain. Also note that this problem is not treated explicitly in the literature. Existing simulation tools leave the problem to the engineer who has to set up the correct equations and conditions respectively. In this connection we investigated the following special-purpose and standard tools that support the design of hydraulic systems: MOSIHS [Piechnick and Feuser, 1994], OHCS [Nakashima and Baba, 1989], MOBILE [Kecskeméthy, 1993], or SIMULINK [Math Works Inc., 1992].

2.2 Automated Model Synthesis: The Basis of an Automated Behavior Analysis

As argued before model synthesis is not a deterministic procedure here. There exist choice points where the valid component model must be selected, depending on the actual input values, parameter alternatives, or physical regularities.

For each component o in a circuit C let $M_o = \{m_{o_1}, m_{o_2}, \ldots, m_{o_k}\}$ be comprised of the k behavior alternatives of o. If a component o has a locally unique model, say, a pipe for instance, $|M_o| = 1$. Let $\mathcal{M}_C$ be the Cartesian product of the M_o. Obviously, $\mathcal{M}_C$ comprises the possible global models of the hydraulic system C, and thus, $\mathcal{M}_C$ defines the total *synthesis search space*.

Before all physical parameters of a hydraulic system C can be computed, the physically consistent model $M_C \in \mathcal{M}_C$ has to be determined. Conversely, whether a behavior model M_C is physically consistent can solely be verified via simulation.

To reason about the behavior constraints m_{o_i} of a component o, some kind of meta constraints, the so-called model selection constraints, are required. Example:

IF x is of type relief_valve
  AND x is in state *open*
  THEN $m_{rv_1} := \{Q_A = Q_B\}$ is valid

The IF-clause constitutes a model selection constraint, m_{rv_1} is one local behavior model of the relief valve, and "$Q_A = Q_B$" is the only behavior constraint that belongs to m_{rv_1}. A model selection constraint is called "active" if its conditions are fulfilled.

Given the concept of model selection constraints, the search for a physically consistent model M_C can be realized as a cycle containing the following steps:

1. *Component Selection.* Select a component that possesses several states, that is to say, behavior alternatives.
2. *State Selection.* Choose a definite state for this component.
3. *Synthesis.* Identify and evaluate active model selection constraints[2]. Synthesize the local behavior models into one global model.
4. *Simulation.* Simulate the synthesized behavior model by evaluating the behavior constraints.
5. *Modification.* In case of physical inconsistencies or unfulfilled demands, trace back to a choice point and formulate additional synthesis restrictions (see the next section).

Figure 4 depicts the search process graphically.

The search comes to an end if either a consistent global behavior model is found or if no further choice point exists.

Remarks. Different components constrain the model synthesis process in a different manner. Hence, the order by which undetermined components are processed plays a crucial role.

[2]This type of inference is sometimes called "constraint inference", as opposed to a "value inference" process that is performed within step 4, simulation, [Davis, 1987].

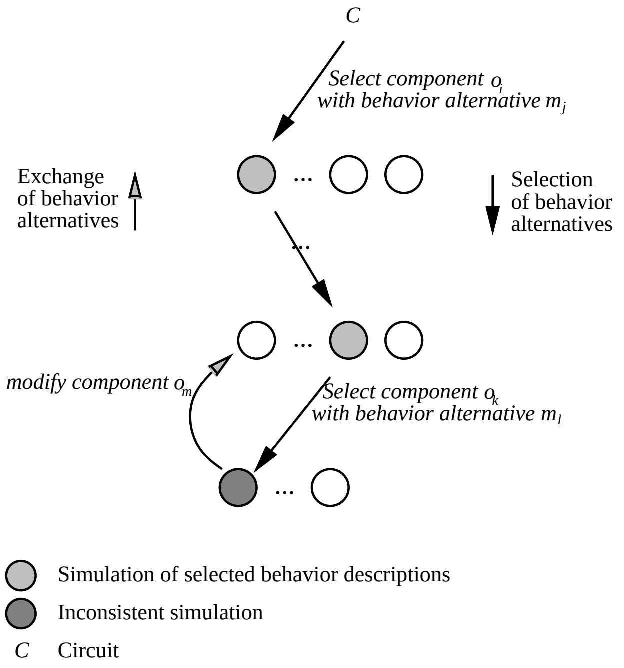


Figure 4: Exploring the synthesis search space $\mathcal{M}_C$.

In this text we cannot engage in constraint processing details; we merely list the subtasks performed within step 3, synthesis, and step 4, simulation. The following are typical synthesis tasks: parsing of behavior constraints, unification of constraint variables according to the topology of a circuit C, separation of symbolic relations from numerical equations, and setting up equation systems in adequate normal forms. The Simulation is realized by the combined application of the following inference methods: local value propagation, rule processing, solving a linear or non-linear equation system, or solving an initial value problem.

3 The art*deco* System[3]

art*deco* is a system that solves the model synthesis problem for a hydraulic system, and hence, it can automatically perform the analysis step within the hydraulic design procedure [Stein, 1995; Kleine Büning and Stein, 1993]. The key features of art*deco* are the following:

1. Efficient search in $\mathcal{M}_C$, the space of possible global behavior models.
2. Direct problem specification, which enables a user to simply formulate his design ideas by means of circuit drawings.

[3] art*deco* originates from a DFG research project where the institute MSRT, University of Duisburg and the working group Knowledge-based Systems, University of Paderborn were involved.

After a short introduction to art*deco* from the user's viewpoint, we focus on those parts of art*deco*'s inference process which tackle the model synthesis problem.

3.1 Problem Specification

By the term "problem specification" we denote the procedure necessary to formulate an instance of an analysis problem $C \longrightarrow B_C$. In first place, the following question has to be answered: "How can a user define his problem in an acceptable time?"

The working document in the "classical" design process is the circuit diagram. Consequently, it would be fair to specify hydraulic analysis problems at the same level of abstraction: Graphic symbols should be selected and connected to a circuit—but in contrast to a CAD system, aside from the drawing, a *functional model* of the hydraulic system should be generated as well.

art*deco* realizes such a graphic problem specification. While the circuit diagram of a system is drawn, a knowledge base containing the topological and the physical connections is created. This knowledge base forms the basis for the analysis step, which can be invoked by pushing a button. I.e., the model synthesis process as well as complex physical dependencies are made transparent: Nearly all information obligatory for the analysis is derived from the technical drawing.

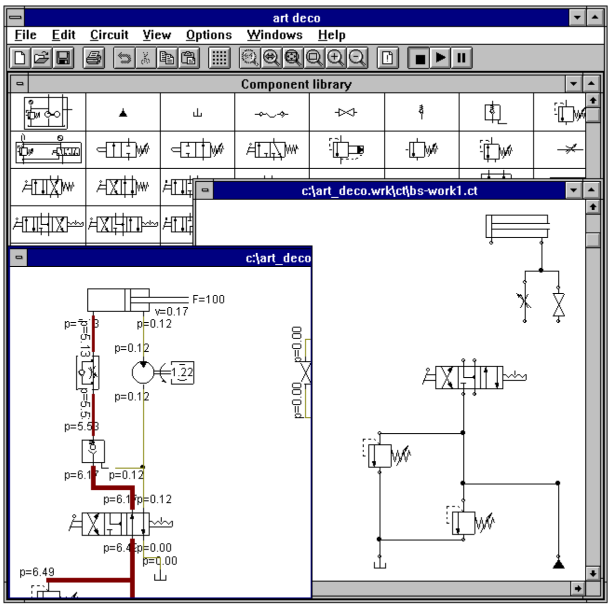


Figure 5: Snapshot of an art*deco* screen.

Figure 5 depicts an art*deco* screen, showing the component library (upper window), a circuit currently edited (right window) and a circuit currently simulated (left window).

3.2 Efficient Model Synthesis

Section 2.2 introduced the idea of the synthesis search space $\mathcal{M}_C$, which contains the possible global behavior models for a given circuit C. Actually we left open how the valid behavior model can be searched efficiently in $\mathcal{M}_C$.

Note that even for a rather small circuit, $\mathcal{M}_C$ might contain several thousand elements. And, checking the consistency of an element $M_C \in \mathcal{M}_C$ requires an expensive simulation of M_C. Thus, an intelligent exploration of the synthesis search space $\mathcal{M}_C$ is the key factor which decides if a circuit analysis can be performed at all in an acceptable time. It is obvious that $\mathcal{M}_C$ must be constrained by some kind of *synthesis restrictions*.

Let $M := \{m_1, \ldots, m_k\}$ comprise the components' behavior models for a circuit C. A synthesis restriction ρ with respect to C is a subset of M with the following property:

$$\forall M_C \in \mathcal{M}_C\text{: If } \rho \subseteq M_C \Rightarrow B_C \text{ is contradictory}$$

I. e., each synthesis restriction ρ defines a set of component models that lead to a physical inconsistency, if all models in ρ are used at the same time. Example: Given is a circuit C where a pump x and a relief valve y are connected in series. Clearly, the pump x cannot output some flow if the relief valve y is shut. Stated another way, the model m_{x_1}, "Pump x produces flow", and the model m_{y_1}, "relief valve y is shut" form a synthesis restriction in the context of C.

This example reveals a central problem: Synthesis restrictions depend on both a component's behavior models m_{o_i} and C, the context of usage of the component; hence an anticipation of generic restrictions, which hold independently from a circuit C, is only possible to a very small extent.

A way out is the automatic generation of tailored synthesis restrictions, which consider a circuits structure and the employed components.

Automatic Generation of Synthesis Restrictions

We developed and realized two concepts for the automatic generation synthesis restrictions in $^{\text{art}}$*deco*:

1. *Dependency Recording.* $^{\text{art}}$*deco*'s dependency recording mechanism records the inference steps performed during a simulation of M_C and tags the inferred data with the underlying assumptions. The assumptions correspond to chosen behavior alternatives of the components; the inferred data consists of both behavior constraints and values. The former result from the evaluation of model selection constraints, the latter result from the evaluation of behavior constraints by means of some inference method.

   If the evaluation of a behavior constraint produces a contradiction, all assumptions necessary to select and to instantiate this constraint are comprised into a new synthesis restriction.

2. *Topological Analysis.* The topological analysis is based on a flow direction investigation and can be performed on the structure graph of C without carrying out a sophisticated simulation. E. g., such an investigation detects directional valves connected in series and derives synthesis restrictions that enforce these valves to be always in the same state.

Figure 6 shows the basic steps during the automated model synthesis.

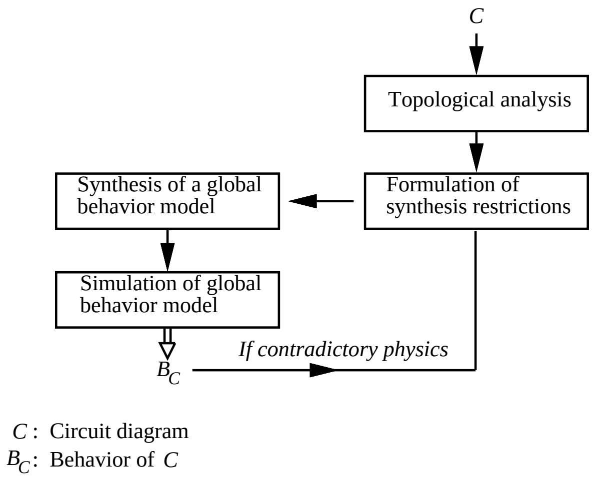


Figure 6: Simplified view to automated model synthesis.

All synthesis restrictions are evaluated during the exploration of $\mathcal{M}_C$ (cf. Figure 4). More precisely: Let the behavior of some component o be still undetermined, and let M'_C be the set of component models selected up to that point. Then a behavior alternative m_{o_i} of o can only be chosen, if for each synthesis restriction ρ the following condition holds:

$$\rho \not\subseteq M'_C \cup \{m_{o_i}\}$$

If such a behavior alternative can be found, the appropriate model selection constraint will be activated and the global behavior model will be updated with this behavior alternative.

More on Dependency Recording

The constraint network constructed during the exploration of $\mathcal{M}_C$ consists of two kinds of nodes: nodes referring to physical parameters and nodes referring to behavior constraints. A constraint node b and a parameter node p are linked, if p stands in the relation defined by b. Based on this network we distinguish between three types of dependency links in $^{\text{art}}$*deco*:

1. *Constraint Dependency.* Constraints can directly depend on other constraints, the so-called model selection constraints. If such a constraint is fulfilled,

the dependent behavior constraints are "active"; otherwise, they are "inactive". Throughout constraint processing additional links are introduced in the constraint network that indicate both the inference and the retraction direction.

2. *Local Value Dependency.* The constraints processed during local propagation define a cause-effect chain between the nodes of the network. These relationships are recorded by the introduction of both support links and node labels. Therefore, the root nodes of an inconsistency can be determined immediately, and their consequences can be traced for disbelief propagation purposes [Martins and Shapiro, 1988].

3. *Global Value Dependency.* Constraints that cannot be treated by local propagation are called global. Global constraints like equation systems establish cyclic dependencies between the parameters. Retracting one node of such a strongly connected component results in the retraction of all nodes involved. In order to avoid labeling effort in $O(n \cdot m)$, with n, m specifying the number of parameters and constraints respectively, only those dependency links are installed in the constraint network that are necessary to instantiate a strongly connected component.

This dependency recording mechanism is triggered by each inference method in $^{\text{art}}$*deco*. Aside from the generation of new synthesis restrictions, it also forms the backbone of a dependency-directed and a knowledge-based backtracking that are performed during the exploration of $\mathcal{M}_C$ [Stallman and Sussman, 1977]. In either strategy the setting back to arbitrary choice points of the model synthesis process is necessary. The former strategy sets back to the root nodes of a contradiction, where a new value assignment (= new alternative) is chosen chronologically. The latter also sets back to the root nodes of a contradiction, but additionally exploits information about which of the alternatives of the root nodes should be modified.

Remarks. The dependency management as realized in $^{\text{art}}$*deco* adopts concepts from Doyle's justification-based truth maintenance system (JTMS) [Doyle, 1979] as well as from deKleer's assumption-based truth maintenance system (ATMS) [de Kleer, 1986]. Employing the classical ATMS-based dependency management would not be useful for performance reasons here: (*i*) Label-inferencing and updating all combinations of assumptions is not necessary, and (*ii*) maintaining ATMS-data structures poses an overhead as compared to recording the cause-effect dependencies during local propagation.

4 Conclusion

Can the design of hydraulic systems be simplified?

We introduced hydraulic design as a problem that is founded on deep physical connections and that can be mastered by domain experts only. However, a closer look to the design process shows that most of the time is spent on analyzing a hydraulic system rather than synthesizing hydraulic building blocks towards a new circuit. I. e., the main engine of a hydraulic design support is the automation of the analysis task.

Automating hydraulic systems analysis is difficult. A main problem is the derivation of a circuit's global behavior model from the local behavior models of its components. This problem is called model synthesis here.

The paper in hand shows how the model synthesis problem can be tackled. In particular, it presents methods to efficiently explore the synthesis search space of a given hydraulic circuit. A key concept in this connection are synthesis restrictions, which help in cutting down a large synthesis search space. These synthesis restrictions can be *automatically* generated by a combination of intelligent backtracking, truth maintenance concepts, and topological investigations.

The development of $^{\text{art}}$*deco*, a system which operationalizes these concepts answers the question above: Yes, it can.

References

[Davis, 1987] Ernest Davis. Constraint Propagation with Interval Labels. *Artificial Intelligence*, 32:281–331, 1987.

[de Kleer, 1986] Johan de Kleer. Problem Solving with the ATMS. *Artificial Intelligence*, 28:197–224, 1986.

[Doyle, 1979] Jon Doyle. A Truth Maintenance System. *Artificial Intelligence*, 12:231–272, 1979.

[Falkenhainer and Forbus, 1991] Brain Falkenhainer and Kenneth D. Forbus. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence*, 51:95–143, 1991.

[Iwasaki and Levy, 1993] Yumi Iwasaki and Alon Y. Levy. Automated Model Selection for Simulation. Knowledge Systems Laboratory, Stanford University (QS93), 1993.

[Kecskeméthy, 1993] A. Kecskeméthy. MOBILE—An Objectoriented Tool-Set for the Efficient Modeling of Mechatronic Systems. In M. Hiller and B. Fink, editors, *Second Conference on Mechatronics and Robotics*. IMECH, Institut für Mechatronic, Moers, IMECH, 1993.

[Kleine Büning and Stein, 1993] Hans Kleine Büning and Benno 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.

[Kleine Büning *et al.*, 1994] Hans Kleine Büning, Daniel Curatolo, and Benno Stein. Knowledge-Based Support within Configuration and Design Tasks. In *Proc. ESDA '94, London,* pages 435–441, 1994.

[Lemmen, 1994] Ralf Lemmen. Checking the Static and Dynamic Behaviour of a Hydraulic System. In *Proceedings of the 1th Asian Control Conference, Tokyo*, 1994.

[Martins and Shapiro, 1988] Joao P. Martins and Stuart C. Shapiro. A Model for Belief Revision. *Artificial Intelligence*, 35:25–79, 1988.

[Math Works Inc., 1992] Math Works Inc. SIMULINK *User's Guide*. Math Works Inc., Nattik, Massachusetts, 1992.

[Nakashima and Baba, 1989] Yusei Nakashima and Tomio Baba. OHCS: Hydraulic Circuit Design Assistant. In *First Annual Conference on Innovative Applications of Artificial Intelligence*, pages 225–236, Stanford, 1989.

[Nayak, 1992] P. Pandurang Nayak. *Automated Model Selection*. Dissertation, Stanford University, 1992.

[Piechnick and Feuser, 1994] Martin Piechnick and Alfred Feuser. MOSIHS – Programmsystem zur Simulation komplexer elektrohydraulischer Systeme. In *AFK, Aachener Fluidtechnisches Kolloquium*. Mannesmann Rexroth GmbH, Lohr, Germany, 1994.

[Stallman and Sussman, 1977] Richard M. Stallman and Gerald J. Sussman. Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis. *Artificial Intelligence,* 9:135–196, 1977.

[Stein, 1995] Benno Stein. *Functional Models in Configuration Systems*. Dissertation, University of Paderborn, Institute of Computer Science, 1995.