Simulation in Fluid*SIM*

Benno Stein Daniel Curatolo Marcus Hoffmann

Department of Mathematics and Computer Science—Knowledge-based Systems
University of Paderborn, D–33095 Paderborn, Germany
email: stein@uni-paderborn.de

Keywords: computer-aided analysis, interactive simulation, structure analysis, design support in fluidics.

Abstract

Based on our research and experiences in fluid engineering (hydraulics, pneumatics), we have realized tools for drawing, simulating, and structure visualization of electro-fluidic circuits. These tools can provide a new quality when analysing or synthesizing fluidic systems since they enable a user to quickly evaluate and modify solutions for hydraulic and pneumatic manipulation problems.

This overview comprises selected concepts of our tools as well as of our current research in this field.

1 Problem Formulation at the Mental Model Level

A technical system can be modeled at very different levels. A *declarative* specification, e. g. in form of a mental model, enables an engineer to describe the system without having the processing of his model in mind. On the other hand, an *algorithmic* model encodes knowledge about model processing and is definitely much closer to the computer. Figure 1 shows the abstraction hierarchy of models [13].

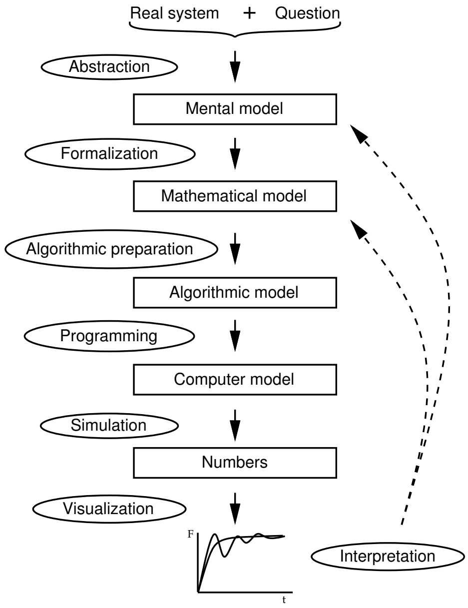

Figure 1: An abstraction hierarchy of models.

In connection with hydraulic and pneumatic systems, the common mental model level is the CAD drawing of the related circuits. Elements on this modeling level are DIN symbols each of which representing a hydraulic component (cf. Figure 2). Put another way, in the fluid domain the mental model level is usually equivalent to the graphical component level.

Figure 3 does also depict a graphical model. However, this model level is far away from being on a mental level. It differs in two major points from the model in Figure 2, which make it to an algorithmic model: (*i*) It falls back on a mathematical level and thus it is less abstract, (*ii*) it prescribes the direction of computation.

Processing an algorithmic model is much easier than processing a model specified at the graphical component level. Fluid*SIM* realizes the latter case. It enables an engineer to formulate his problems directly on the mental level. I. e., Fluid*SIM* derives all information necessary for a simulation from the drawing: Local component descriptions are selected and synthesized towards a global model. Figure 4 shows a screen snapshot.

Details relating model synthesis and simulation in Fluid*SIM* can be found in [8; 9].

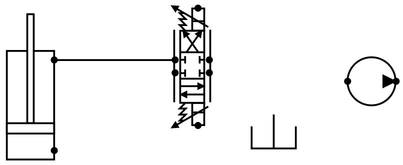

Figure 2: Describing hydraulic concepts at the mental model level. From left to right: cylinder, pipe, valve, tank, pump.

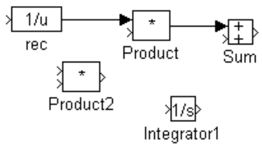

Figure 3: Describing hydraulic concepts at an algorithmic level with SIMULINK [6].

2 Event-based Simulation

Fluid*SIM* distinguishes two classes of events. Events of the first class are triggered either by physical thresholds or by *reactive* state changes of particular components such as relief valves. These events result from the dynamic component behavior and require that $\dot{p} \neq 0$ or $\dot{Q} \neq 0$. Events of the second class are triggered by *active* state changes of components like cylinders (piston reaches stop), directional valves (via a solenoid actuation) and sensor switches, or by an operation of the user himself.

The simulation in Fluid*SIM* is event-driven. During simulation, the state vector and the input vector are checked whether the simulation's underlying model is still correct. In case of an incorrect model, the invalid behavior constraints of the respective components are identified, retracted, and replaced by the actually valid constraints.

To reason this way about the behavior constraints, the model of a component o is organized as a collection of partial models m_{o_i}. Each partial model in turn is equipped with a meta constraint, the so-called model selection constraint. 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 partial 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. We call the problem of identifying and combining the valid partial component models towards a correct global model the *model synthesis problem* [8].

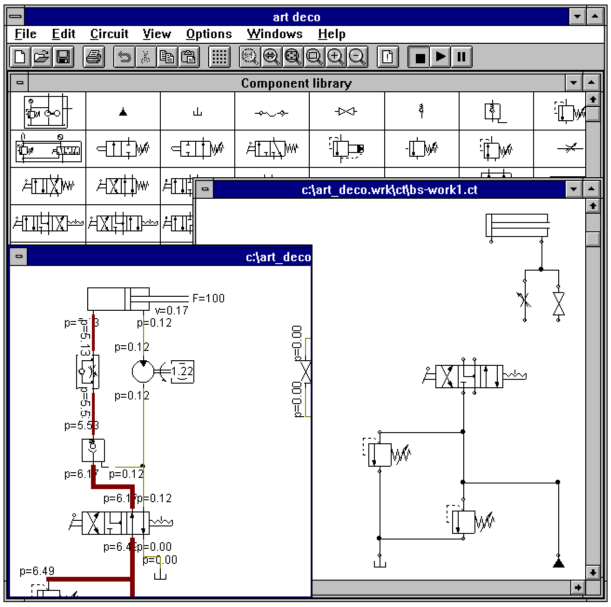

Figure 4: Working at the mental model level.

3 Topological Circuit Analysis

At present, we are integrating algorithms for a topological analysis into Fluid*SIM*.

A topology or structure analysis shall help to focus on the relevant parts of a complex system. A natural level at which a topological analysis of a system can be oriented is its functional level. Within fluidic systems the functional level is reflected by so-called hydraulic or pneumatic axes: Each axis is responsible to fulfill a particular function. Figure 5 shows a few examples for hydraulic axes that have been cut free from a large circuit.

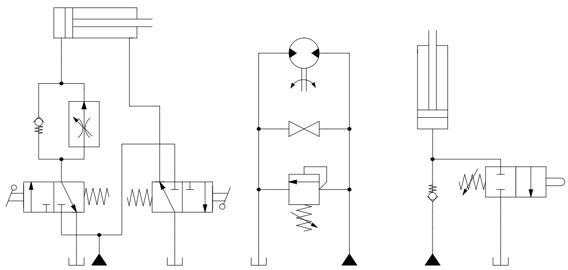

Figure 5: Examples for hydraulic axes.

Focusing onto a single axis is of a great value for many analysis and synthesis tasks: complex simulations can be restricted to a single axis, faults must not be searched in the entire system [4; 2], demands at particular functions can be matched against their respective axes, the structure of large systems can be unraveled, etc.

From an engineering standpoint, a hydraulic axis A both represents and fulfills a subfunction f of an entire hydraulic plant. A defines the connections and the interplay among those working, control, and supply elements that realize f. This definition leaves a scope of interpretation—e.g., regarding the components which actually must be count to an axis and which not. However, it conveys a useful idea of what we are looking for within the topological analysis process.

The analysis procedure that we have developed to identify the axes in a circuit is comprised of the following steps:

1. *Graph-theoretical Formulation.* Starting point is an abstraction from a circuit C onto a simplified graph data structure $G_h(C)$, see Figure 6. As described in [10], this data structure also forms the basis for a precise definition of the couplings between axes.

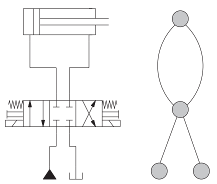

Figure 6: Abstraction of a circuit diagram towards a hydraulic graph.

2. *Preprocessing.* To reduce G_h's complexity—but, in first place, to make axes identification possible at all, G_h is simplified by means of merging, deletion, and condensation rules, see Figure 7.

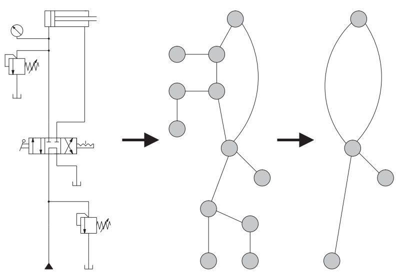

Figure 7: Preprocessing of the hydraulic graph.

3. *Axes Identification.* Identifying a hydraulic axis means to search for a set of nodes in the hydraulic graph whose counterpart in the circuit realizes a particular function. Among others, each such set must contain at least one working element and one supply element. Axes identification is realized by the combination of several shortest path runs.

4. *Coupling Type Determination.* Coupling type determination requires the comparison of supply paths between the axes' working elements.

5. *Node Expansion.* Finally, the nodes that were eliminated during step 2 are re-introduced. Their assignment to the detected axes is realized by means of attraction rules, which define the strength of component's linkage to other components.

A detailed description of the above analysis approach can be found in [10].

4 Design Support

Our present and future research concentrates on a design support within hydraulics and pneumatics. Note, however, that fluidic manipulation jobs vary from simple lifting problems up to the realization of complex robot kinematics, and, given a demand description D for such a manipulation job, the design of an appropriate drive is a truly creative job [3; 1]. Thus an inherent property of our approach is not to start circuit design from scratch: Our working hypothesis is that we still *have* a preliminary design C' of a circuit which, roughly speaking, incorporates the potential to fulfill D.

Put another way, there exists a sequence of modifications $m_1 \ldots m_l$ of C' that transforms C' towards the desired circuit C. In this connection, we focus on the following modification types to repair a circuit that does not fulfill the desired demands [5; 11]: (*i*) Component parameter modifications, (*ii*) component characteristics modifications, (*iii*) topology modifications.

Starting point of a design problem is a preliminary design in form of a circuit C' with unfulfilled demands. The design search space is comprised of all circuits that can be derived from C' by applying a given set of modification measures. A path from the root C' down to a solution defines a sequence of modifications that repairs all unfulfilled demands in C' (cf. Figure 8).

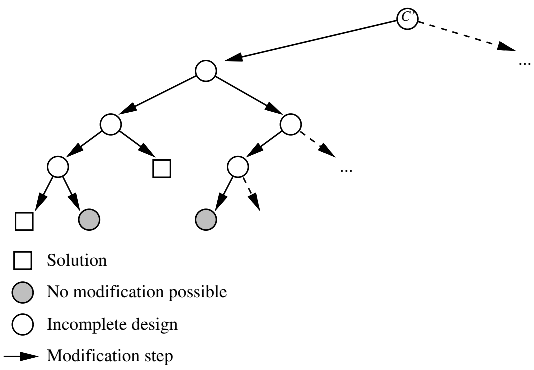

Figure 8: Exploring the design search space.

To operationalize this approach we have been developing a design language, that provides miscellaneous concepts to formulate modification measures [7]:

- *Domain-adequateness.* Features for the formulation of topological relations in circuits are provided; an example:

```
select_component(in_series pump tank_2
                   after cylinder_2)
```

- *Extendibility and Abstraction.* Based upon a set of core functions (`increase_parameter`, `select_component`, etc.) a user is allowed to define new and more complex modification routines.
- *Handling of Alternatives.* Related to an unfulfilled demand, alternative measures can be specified and tagged with a priority.

An expert's design knowledge is organized within so-called modification schemes, which in turn are collected in a knowledge base. A modification scheme belongs to a particular component type; it defines the component's structure, its parameters, possible symptoms indicating unfulfilled demands, and a list of related modification measures formulated in our language. [12] and a forthcoming paper describe the concepts in detail.

References

[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] R. Davis. Diagnostic Reasoning Based on Structure and Behavior. *Artificial Intelligence*, 24:347–410, 1984.

[3] J. S. Gero. Design Prototypes: A Knowledge Representation Scheme for Design. *AI Magazine*, 11:26–36, 1990.

[4] T. Hesse and B. Stein. Hybrid Diagnosis in the Fluidic Domain. *Proc. EIS 98, International ICSC Symposium on Engineering of Intelligent Systems, University of La Laguna, Tenerife, Spain*, Feb. 1998.

[5] C. Krafthöfer. Untersuchung konstruktiver Maßnahmen zur Beinflussung des dynamischen Verhaltens hydraulischer Antriebe. Study work, Gerhard-Mercator-Universität - GH Duisburg, MSRT, Mar. 1997.

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

[7] T. Schlotmann. Formulierung und Verarbeitung von Ingenieurwissen zur Verbesserung hydraulischer Systeme. Diploma thesis, Universität-GH Paderborn, FB 17 Mathematik / Informatik, 1998.

[8] B. Stein. *Functional Models in Configuration Systems*. Dissertation, University of Paderborn, Department of Mathematics and Computer Science, 1995.

[9] B. Stein, D. Curatolo, and M. Hoffmann. Enriching Engineering Education in Fluidics. *Proc. WMC 98, Western Multiconference featuring ICSEE '98, International Conference on Simulation and Multimedia in Engineering Education, San Diego, California*, Jan. 1998.

[10] B. Stein and A. Schulz. Topological Analysis of Hydraulic Systems. Technical Report tr-ri-98-197, University of Paderborn, Department of Mathematics and Computer Science, July 1998.

[11] S. Uecker. Statische Auslegung hydraulischer Translationsantriebe bei der Neu- und Änderungskonstruktion. Study work, Gerhard-Mercator-Universität - GH Duisburg, MSRT, Mar. 1997.

[12] E. Vier and B. Stein. Modeling of Design Strategies for Hydraulic Control Systems. In *MIC 98, IASTED International Conference on Modelling, Identification and Control, Grindelwald, Switzerland*, Feb. 1998.

[13] J. Wallaschek. Modellierung und Simulation als Beitrag zur Verkürzung der Entwicklungszeiten mechatronischer Produkte. *VDI Berichte Nr. 1215*, pages 35–50, 1995.