Transformation of Qualitative Dynamic Models – Application in Hydro-Ecology

Ulrich Heller, Peter Struss

Department of Computer Science
Technical University of Munich
Orleansstr. 34, 81667 München, Germany
{heller, struss}@informatik.tu-muenchen.de

Abstract

Hydro-ecological systems comprise complex interaction among physical, chemical, and biological processes. Compositional modeling, i. e. creating a system's behavior model by aggregating models of its constituents, is crucial for making the modeling task feasible. However, the composed model is often too fine-grained for a particular task, for instance, in containing too many irrelevant intermediate variables or obscuring the basic interdependencies. For this reason, the model may have to be transformed and simplified. The paper presents a graph-oriented representation for dynamic systems closely related to existing process languages, and a set of syntactic operators that transform such a model while preserving certain properties of the model. The formalism is motivated and illustrated by an example taken from our work on modeling hydro-ecological systems, but we also demonstrate its utility for technical applications.

Introduction

In our work on modeling complex ecological systems for decision-support systems, a number of important challenges arises. In particular, our efforts to obtain prediction models for algal blooms Rio Guaíba (Southern Brazil), have to address problems of:

- **Compositional modeling,** i. e. generating a behavior model of a complex system through aggregation of models of its elementary constituents taken from a library.
- **Modeling of dynamic systems**, i.e. adequately capturing the evolution of the system and its phenomena over time, which in our application comprise a variety of processes from the flux of the river to chemical reactions.
- **Qualitative modeling** in order to make the essential distinctions only, thus enabling the modeling of **classes** of situations and the exploitation of partial information, since both knowledge about the relevant types of processes and information about their specific instances is inherently imprecise, and available measurements are sparse w.r.t. time and space.

Several existing qualitative reasoning systems, such as QPE ([Forbus 84]) and QPC/QSIM ([Crawford/Farquhar/Kuipers 90]) have been built to satisfy these needs, and we were able to formulate models of relevant processes in our domain using QPC. However, we encountered several difficulties that we consider to be instances of general problems involved in compositional qualitative modeling of dynamic systems. Several of them are related to the **granularity of the model**, seen from different perspectives:

- **Compositional modeling** effects the **structural granularity** and there is the potential of generating overly detailed models. This is because the constituent models in the library have to be stated in terms of local variables and parameters only. The resulting model can be inappropriate both from its cognitive and a technical point of view. Many variables and parameters may be irrelevant from the perspective of the entire system or a particular task and conceal the elementary influence structure of the system. Besides, some reasoning tools exhibit exponential behavior in the number of variables, so it is desirable to keep the model small.
- **Models of dynamic systems** affect the **temporal granularity** of the model. It can be can be too fine-grained, again for both humans and predictive engines, if it captures all aspects of the dynamics. For instance, rapid but minimal fluctuations of the concentrations of some substances that are basically held in chemical equilibrium complicate the long term prediction of the behavior of the entire system.
- **Qualitative modeling** concerns the **granularity of behavioral distinctions**. Qualitative models can be too weak to derive all possible conclusions. This does not only concern the domains of variables and parameters, but also the qualitative description of functional dependencies is limited to monotonic functions, as is the case for many qualitative simulators, the analysis of their counteraction or comparison may lead to spurious results.

Some research has been carried out to address these issues:

- **Structural aggregation**, particularly hierarchical modeling responds to the first problem by eliminating internal variables and parameters.
- **Behavioral approximation** through distinction of time scales (sometimes called time-scale "abstraction") aims

at suppressing the irrelevant details of temporal evolution ([Iwasaki 92], [Kuipers 87]).

- **Hybrid modeling** attempts to introduce more distinctions through (semi-)quantitative information. This is mostly confined to possibly repeated refinement of quantity spaces and algebraic operations (for instance [Williams 88]).

Our general view on the task is as follows. Often the following steps are distinguished and potentially supported by different systems (Figure 1):

- **creation of a model library**, i.e. representing primitive model fragments for a particular domain,
- **model composition**, i.e. aggregating appropriate model fragments from the library to establish the model of a specific system (this is, for instance, the task of QPC),
- **prediction/simulation**, i.e. generating a description of the behavior of the entire system (e.g. QSIM´s job).

We propose to explicitly introduce a step of

- **model transformation** that takes an initial model generated by composition and transforms it into a model appropriate for a particular task (Figure 1).

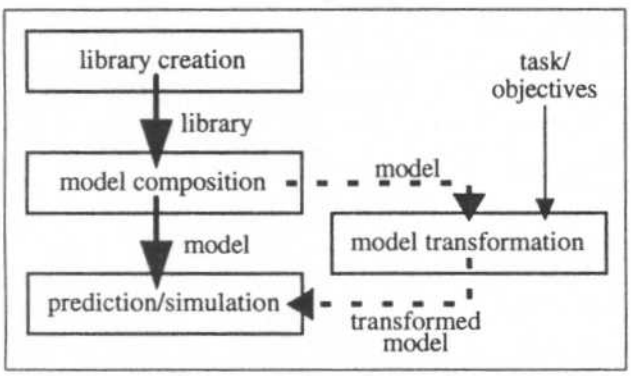


Figure 1: Overview of model-based prediction

This reflects our oppinion that a representation of knowledge relevant to modeling in science and engineeing should distinguish between **basic models** and knowledge how to **modify and use these models**, and that **both** types of knowledge should be represented in a formal and declarative way. This is different form other approaches to compositional modeling which presume in some way that all elements needed for an adequate compositional model are already "pre-manufactured" and simply need to be collected from the library. We do not believe that it is feasible to produce in advance all kinds of combinations of abstractions, approximations, and simplifications of the basic models.

The presentation of our work focuses on the transformation step. This means, we assume an existing composed model of a system as an input to our formalism which then generates a new model under preservation of certain properties. There are two contributions in this work:

- a **modeling language** for specifying influences and
- a set of **local syntatic operators** that transform models expressed in this language.

In order to improve both the predictive power of the models and the results of the operators, the confinement of qualitative functional relationships to monotonic dependencies only has to be overcome. A common answer to ambiguity and insufficient distinctions in quantitative models is "Hybrid models by integrating quantitative information!". We believe that this seemingly obvious solution is often inappropriate and obscures the fact that the expressive power of the models can be extended **without having to leave the realm of qualitative descriptions.**

Our modeling language extends the expressive power of languages like QPE/QPC in allowing for

- more distinctions between functional dependencies (than just monotonic and algebraic ones) and
- more general types of influence combinations (than just linear combination).

We illustrate the application of the language using an example from the hydro-ecological domain. We aim at a formal characterization of model relations and transformations is required a) for an automated solution to the problem and b) for determining the impact of the transformation, i. e. the properties gained and the properties preserved by the transformation, which is required.

The set of operators includes

- generation of strict abstractions of a model as well as
- approximation of dynamic relationships through functional dependencies

These transformations are independent of the quantity spaces chosen for the variables and parameters (in contrast to [Williams]). The use of the operators is illustrated by the ecology example and, to demonstrate the versitility of our formalism, by an example taken from a technical domain. We continue by giving a brief introduction to the hydro-ecology background.

The Problem Domain

In an international collaboration between researchers of Brazil, France and Germany, we have been examining a specific ecosystem, namely the Rio Guaíba in Southern Brazil, with the objective of analyzing and predicting undesirable occurrences of algal blooms. The modeling of the complex hydrodynamics and the various chemical and biological processes involved provided us with important challenges for our modeling and reasoning techniques.

Among the elementary conditions for the possibility of algal blooms is the availability of nutrients, which is influenced primarily by distribution and transformation processes. In this paper, we will examine a typical example of an interaction of two such processes.

Advection

Advection is the transport of matter by directed flow of water. The complex hydrodynamics in the bays of the Rio Guaíba prevent us from using a linear water flow model and we had to choose a flexible repesentation of spatial distributions and water transportation. By using compartments, the elements of a topological partitioning of the water body (described below and in more detail in [Heller 95]), and by locating the transport processes between adjacent compartments, we also gain more generality.

The advective effect on the concentration of some specific chemical constituent in two adjacent compartments can be easily determined if the volumes are assumed constant (requiring the net flow for each compartment to be zero). A simple model under this assumption is discussed below.

Ammonia Dissociation

One of the most important constituents is ammonia, appearing both in free (NH_3) and ionized form (NH_4). Both forms can act as nutrient, but free ammonia in high concentrations can also exhibit toxic effects, so we have to study the chemical equilibrium ($NH_4 + OH \leftrightarrow NH_3 + H_2O$) established by the counteracting reactions of ionization and dissociation.

Both reactions are strongly influenced by the pH of the location. To put it more precisely, if the ratio between (the molar concentrations of) NH_3 and NH_4 is below $10^{(pH-9.26)}$, then the dissociation reaction dominates ionization. Above the given reaction constant, the ionization is predominant. Both reactions will be modeled as a single process with a rate that is linearly dependent on the difference between the ratio NH_3/NH_4 and the reaction constant (modeled as positively monotonic in the pH, see Figure 7).

Model Representation

We present a modeling language with a flexible representation of functional and integrative influences and we depict models in this language by using a graphical notation, which will help to illustrate the examples throughout the paper.

A system model consists of a finite set of variables with continuous real-valued functions over time (or any appropriate qualitative abstraction thereof) as domains. There is a set of constraints on these functions, represented by the existence of "influence functions" specifying the dependency of a variable on a set of other variables.

Characterization and Combination of Influences

The basic influence function is a multivariate monotonic function with multiple parameters. In particular, we want to express that a variable, A,depends monotonically on a set of other variables, $\{B_1, B_2, ..., B_n\}$ (possibly with different direction coefficients, $S_1, S_2, ..., S_n \in \{+1, -1\}$). Formally:

$$\exists f \in Mon_{(S_1,S_2,...S_n)}\ \forall t \in \mathbb{R}$$
$$A(t) = f(B_1(t), B_2(t), ..., B_n(t)),$$

$$\text{where } f \in Mon_{(S_1,S_2,...S_n)} \text{ iff } \forall i \in \{1, 2, ..., n\}$$
$$f(x_1, ..., x_i, ..., x_n) > f(x_1, ..., x_i', ..., x_n) \Leftrightarrow (S_i \cdot x_i > S_i \cdot x_i')$$
$$\text{with } S_i \in \{+1, -1\}\ (1 \le i \le n) \quad \text{(simply "+" or "−").}$$

This allows for more general forms of combination than the linear combination assumption implicitly used in other modeling languages (e. g. QPC, see [Crawford/Farquhar/Kuipers 90]).

There are several ways to represent additional information about the influence function. The most important one is a further restriction by a Lipschitz condition or even linearity in one of the arguments. A Lipschitz condition in the i-th argument is given by

$$\exists M \in \mathbb{R}^+ \mid f(x_1, ..., x_i, ..., x_n) - f(x_1, ..., x_i, ..., x_n) \mid \le M \cdot \mid x_i' - x_i \mid$$

and linearity in the i-th argument can be expressed as

$$\forall a, b \in \mathbb{R}\ ((x_i =) \wedge ((j \neq i \Rightarrow x_j'' = x_j' = x_j')) \Rightarrow$$
$$f(x_1, ..., a \cdot x_i + b, ..., x_n) = a \cdot f(x_1, ..., x_i, ..., x_n) + f(x_1, ..., b, ..., x_n).$$

Graphically, we represent variables by boxes and the influence functions by labeled arrows and a combination information box containing the direction indicators $S_1, S_2, ..., S_n$. The basic elements of this notation, called *influence diagram*, are shown in Figure 2. The arrow labels denote the function restriction of the dependency ("Mon" for monotonicity, "Lin" for additional linearity and "Lip" for the Lipschitz condition). For a function with a single parameter, we use also strict identity ("Id") and the following (proper) inclusions are valid:

$$Id \subset Lin \subset Lip \subset Mon.$$

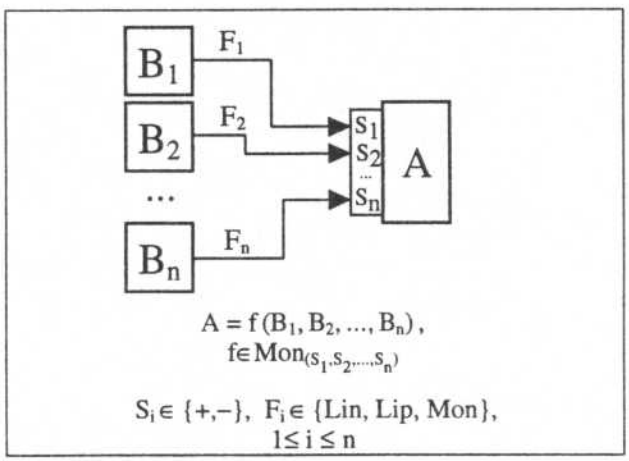


Figure 2: Basic elements of influence diagrams

Decomposable Influences

Furthermore, certain influence functions can be decomposed in the sense that they are known to consist of groups of influences combined additively or multiplicatively. More precisely, a function $f \in Mon_{(S_1,S_2,...S_n)}$ is said to be decomposable additively, iff

$$\exists i \in \{1, ..., n-1\}\ \exists f_1 \in Mon_{(S_1,...S_i)}\ \exists f_2 \in Mon_{(S_{i+1},...S_n)}$$
$$\forall x_1, ..., x_n \in \mathbb{R}\ f(x_1, x_2, ..., x_n) = f(x_1, ..., x_i) + f(x_{i+1}, ..., x_n)$$

A special case is the complete decomposition into single influences, which corresponds to the assumption of linear combination.

We depict decomposed influences by separating the combination information. Compare Figure 3 for the notation for completely additively decomposed influences. The restrictions given at the arrows refer only to the respective group of influences, thus making e. g. linearity a weaker condition in the decomposed case.

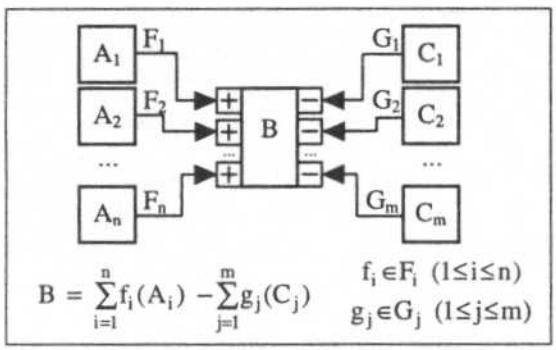


Figure 3: (Completely) additively decomposed influences

Analogously we can decompose (single) influences multiplicatively. For more sophisticated constructions, intermediate variables have to be used. The graphical notation uses two new combination symbols ("×" and "/", see Figure 4).

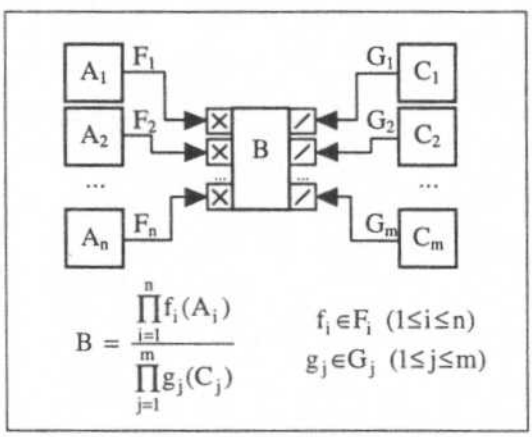


Figure 4: (Completely) multiplicatively decomposed influences

Integrative Influences

The discussed influences correspond to the qualitative proportionalities used in QPT ([Forbus 84]). To represent the so-called "direct influences" of QPT, we need an integrative influence, expressing that the derivative of a variable A is (monotonically) dependent on a group of other variables $B_1, B_2, ..., B_n$:

$$\frac{dA(t)}{dt} = f(B_1(t), ..., B_n(t)), \quad f \in Mon_{(S_1,S_2,...S_n)}$$

We use all of the constructions of function restrictions and decomposition discussed above. In graphical display we enclose the combination information in a circle or a rectangle with rounded edges, as is shown in Figure 5.

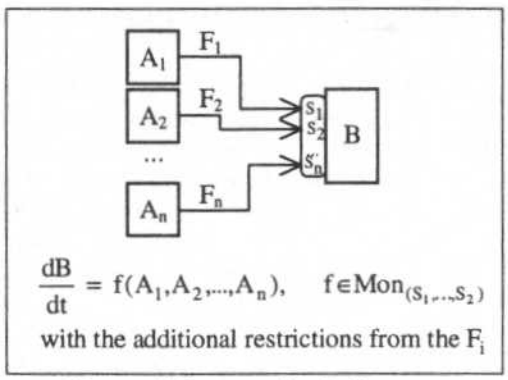


Figure 5: Integrative influences

In this way we can represent a qualitative abstraction (with respect to the functions involved) of an ordinary differential equation in our modeling language and in turn extract a partially specified differential equation from a diagram. Together with a mechanism for instantiating and composing model fragments, we can also visualize models written in QPE/QPC notation.

Process Models for the Domain Problem

Processes are described by partial influence diagrams (possibly with parameters) and additional information about how to compose them with other processes acting on common variables. A process is instantiated by giving the parameters defined values that can be obtained from the system description and aggregating the partial diagrams into the system model. The formal semantics described in the last section depend on the closedness of the model. In the cases discussed here (transport and chemical transformation), we have apparently additive combination of influences.

We developed models for prediction tasks for both the short and the long term behavior of hydro-ecological systems. Here we will present only two simple ones to study their interaction.

As a basic modeling decision, we divided the water body of the river under consideration into compartments, i. e.

regions with similar flow characteristics, that are assumed to have homogenous parameter values. The partitioning is a spatial abstraction that preserves only topological information (basically the neighborhood relation) and some individual properties of the compartments (e. g. volume). Variables are associated with single compartments (e. g. ammonia concentration) or a set of compartments (e. g. the directed water flow between adjacent compartments).

A simple generic process description for advection of some constituent (e. g. ammonia) between two adjacent compartments (*src* and *dest*) is shown in the influence diagram in Figure 6.

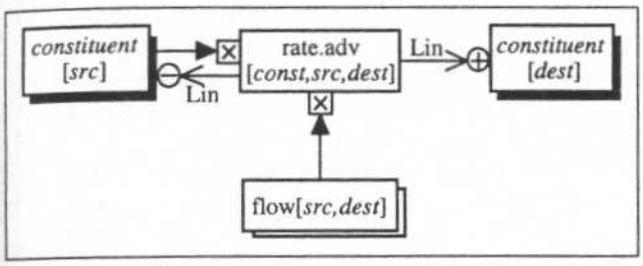


Figure 6: The advection process (simple version)

Unlabeled arrows are to be read as bearing the identity label "Id". The boxes with a black shadow denote important state variables. They represent concentrations. Thus, the transported amount of matter is obtained by multiplication of the source compartment concentration with the (absolute) flow between the compartments. The loss respectively gain in concentration is then calculated as a linear function (the linearity factor being in either case the reciprocal of the volume of the respective compartment, which is assumed constant).

It will instantiated for various chemical constituents, *const(ituent)*, and locations, *src* and *dest*. Note that the semantics in the strict sense given in the last sections will be valid for the complete (composed) model only.

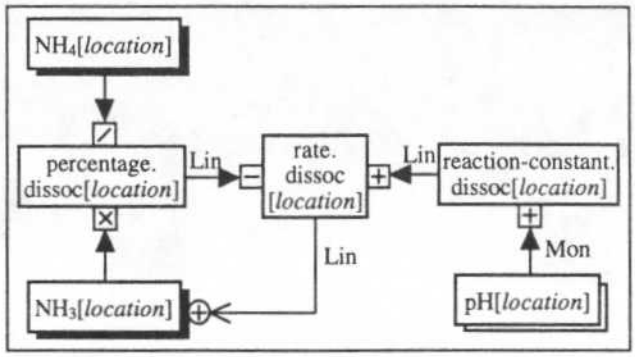


Figure 7: Dissociation process without feedback

However, the combination of the influences on the concentration in the destination compartment are assumed to be additively decomposable from other influences.

Furthermore, a version of the dissociation process without feedback will be used. The concentration of NH_4 will be treated as equaling the total ammonia concentration. Thus, we can neglect the loss of NH_4 by the transformation. The resulting influence diagram is presented in Figure 7.

Model Composition

We compose the advective transport of ionized ammonia (which we treat as total ammonia, so that NH_3 is assumed not to be subject to advection) from compartment "In" into a specific compartment, X, and from X to compartment "Out", with the dissociation taking place inside compartment X, we obtain the diagram in Figure 8.

So we benefit from being able to compose the system model from a simple structure description and a library of generic process descriptions (both described in detail in [Heller 95]), but the simulation of the resulting model is unnecessarily complicated by the large difference in the

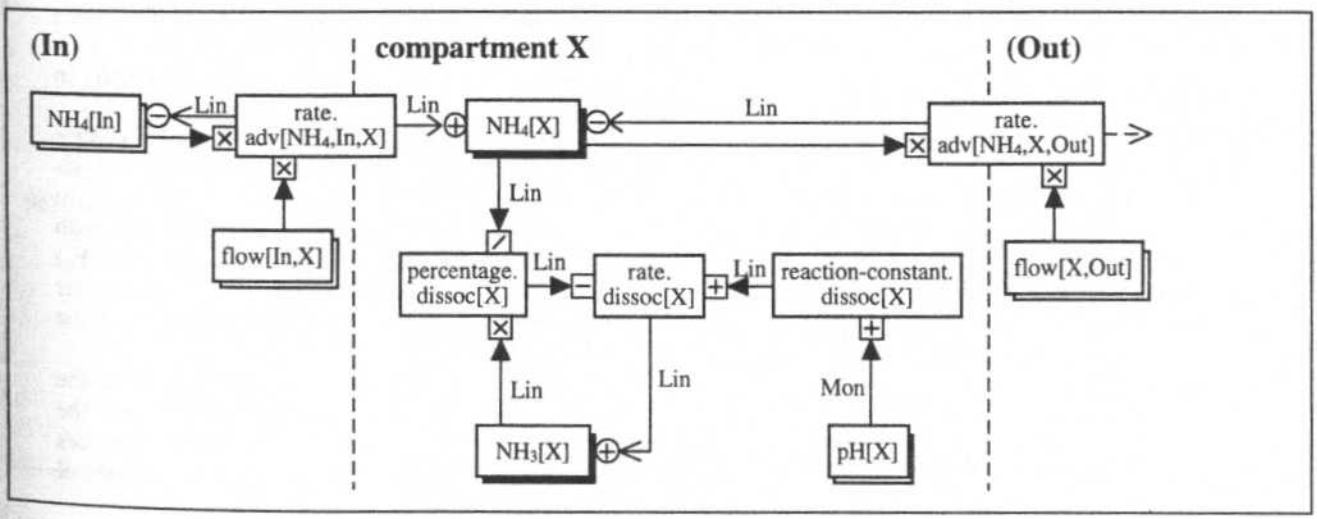


Figure 8: Interaction of advection and ammonia dissociation

strength of the integrative influences. Some qualitative simulation frameworks lack a way to express the different orders of magnitude and therefore even produce spurious solutions by erroneous assessment of the counteraction.

For testing purposes, we transformed the obtained influence diagram into the modeling language of QSIM ([Kuipers 86]), like QPC would do. Unfortunately, we loose the causal information represented in our models, which is partially responsible for some problems of efficiency. A part of the QSIM algorithm exhibits exponential behavior in the number of involved variables.

From the misjudgement of the relative orders of magnitude of the effects of transport and transformation also impossible behavior branches resulted. Even for slowly rising NH4 values, the ratio of the concentrations is hypothesized to be significantly out of equilibrium. For the illustration of this effect, an extended example is given in [Heller 95].

We propose a solution that will will both reduce the number of variables and make use of the information about the different orders of magnitude in the effects of the interacting processes to rule out spurious solutions. This will be achieved by local syntactical operators transforming a given influence diagram.

Model Transformations

We developed a set of transformation operators to simplify influence diagrams in order to identify the basic influence structure in more complex interactions. The goal is to examine in a formal way the applicability of the so-called time-scale abstraction. Time-scale abstraction, as introduced by Benjamin Kuipers ([Kuipers 87]), will formally be treated as an approximation. In the formal framework of model relations developed in [Struss 92], abstraction transforms a model into a strictly weaker version, whereas approximation replaces one model by another one that may violate validity.

Abstraction Operators

An overview of abstraction operators is shown in Figure 9 (on the next page).

If a variable specified in a model fragment is assumed constant in the context of the complete model, we can eliminate it, because it unnecessarily complicates the reasoning task. The **elimination of constants** is achieved by the operator (9a). The class of functions on the right hand side is obtained by taking the maximum with respect to set inclusion (remember that Id $\subset$ Lin $\subset$ Lip $\subset$ Mon). The resulting model transformation is an abstraction (even more precisely, a "view" as defined in [Struss 92]). The proof for this operator and for the following ones can be found in [Heller 95]).

Some variables might be irrelevant, e. g. because they are not observable. The **elimination of intermediate variables** for multiple influences is shown as (9b).

Analogous operators exist for integrative influences, on some variable C_i. For an integrative influence on B there is a restriction (at least in the semactics used): B can only be eliminated, if all of the influence originating from B are linear (see 9c).

Partial decomposition of influences can be preserved, if the relating function is linear. For completely decomposable influences, the operator has the form shown in (9d).

Various cases with additional influences on C, not originating from the intermediate variable B, are considered in [Heller 95], but will not be discussed here.

All of the operators above reduce the number of variables, which is an advantage in itself.

Another class of operators achieves the **subsumption of parallel influences**, i. e. of influences with the same source and destination and the same combination symbol (either "+" or "−"). Figure 9e shows the decomposed case, which is the simplest one.

Time-Scale Abstraction as Approximation

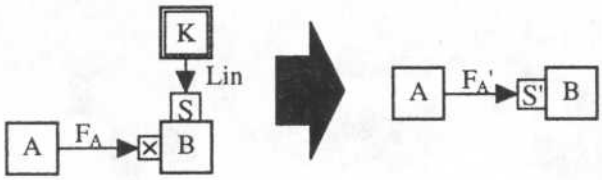
To cope with widely separated time-scales and to make the reasoning task feasible in cases where "fast" and "slow" processes interact, we intend to identify subsystems (by employing the operators introduced above) that can – under certain conditions – be substituted by functional dependencies, while committing only a neglectable approximation error. This corresponds to the technique of "abstraction by time-scale" as defined in [Kuipers 87]. If the elementary influence structure has one of the following forms, we use the solution of the equilibrium equation as substitute.

In Figure 10 we show two operators acting on closely related structures, namely on direct linear self-stabilization (10a) and on multiplicatively mediated linear self-stabilization (10b). Both are discussed in detail in [Heller 95].

In the first case it is even possible to derive precise bounds on the approximation error committed, by analysis of the underlying ordinary differential equation. In general, the quality of the approximation increases with the linearity factor of the stabilizing function (class F_B) and decreases with the Lipschitz coefficient of the transfer function (class F_A) and the maximum variation of the derivative of A.

So we profit from preserving the information about the function class restrictions (namely linearity and the Lipschitz condition) while using the abstraction operators shown in Figure 9. At this point the additional information represented pays off.

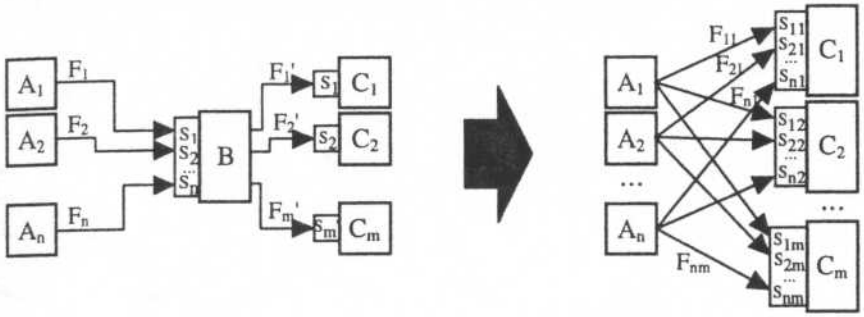
9a) Elimination of (multiplicative) constants:



$F_A \in \{Id, Lin, Lip, Mon\}$
$S \in \{\times, /\}$,
K constant

$F_A' = \max(Lin, F_A)$
$S' = sign(K) \in \{+,-\}$

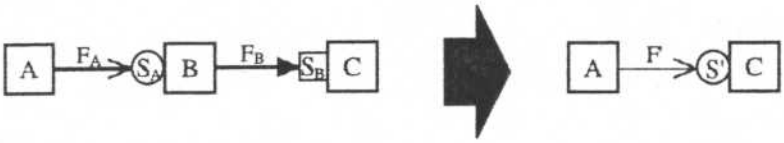
9b) Elimination of intermediate variables (with multiple influences):



$F_i, F_j' \in \{Lin, Lip, Mon\}$,
$S_i, S_j' \in \{+,-\}$
$(1 \le i \le n,\ 1 \le j \le m)$

$F_{ij} = \max(F_i, F_j')$,
$S_{ij} = S_i \cdot S_j'$
$(1 \le i \le n,\ 1 \le j \le m)$

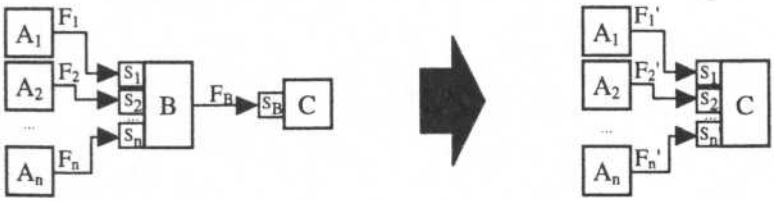
9c) Elimination of intermediate variables (shown for a single integrative influence):



$F_A \in \{Id, Lin, Lip, Mon\}$
$F_B \in \{Id, Lin\}$
$S_A, S_B \in \{+, -\}$

$F' = \max(F_A, F_B)$
$S' = S_A \cdot S_B$

9d) Variant for completely decomposed influences:



$F_i \in \{Lin, Lip, Mon\}$ $(1 \le i \le n)$
$F_B \in \{Id, Lin\}$
$S_B, S_i \in \{+,-\}$ $(1 \le i \le n)$

$F_i' = \max(F_B, F_i)$ $(1 \le i \le n)$
$S_i' = S_B \cdot S_i$ $(1 \le i \le n)$

9e) Subsumption of parallel influences (decomposed case):

$F_1, F_2 \in \{Id, Lin, Lip, Mon\}$
$S_1 = S_2 \in \{+,-\}$

$F' = \max(F_1, F_2, Lin)$
$S' = S_1 (= S_2)$

Figure 9: A selection of basic abstraction operators for influence diagrams

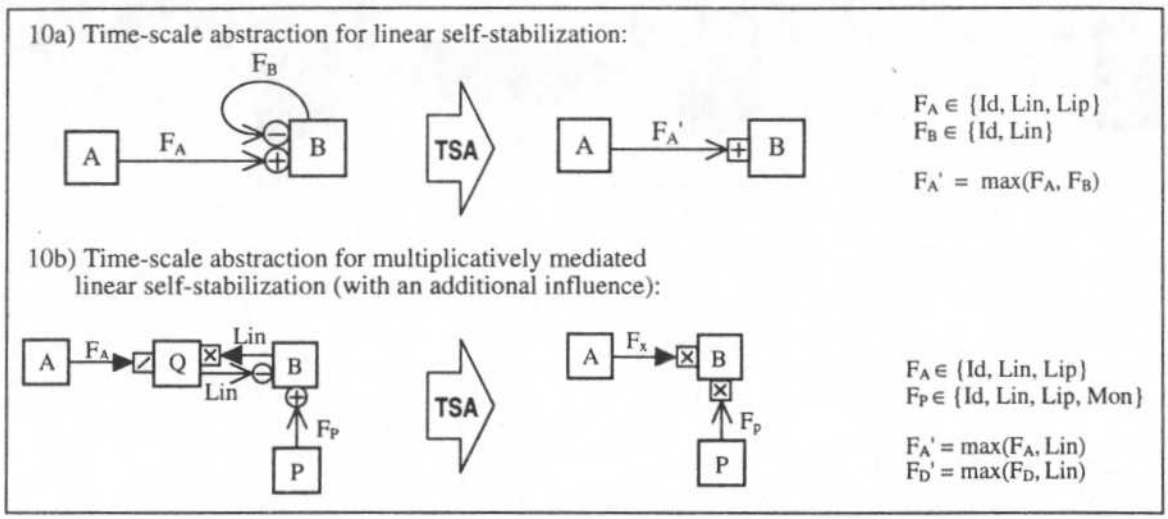


Figure 10: Two time-scale abstraction operators for influence diagrams

Application to the Example Model

To use this kind of approximation for the model given in Figure 8, we have to identify the elementary influence structure of the faster subsystem. Therefore, the variables reaction-constant.dissoc[A] and then rate.dissoc[A] are leiminated using the the operator from Figure 9b, which yields the influence diagram on the left hand side of Figure 11.

The influence structure that appears now in the lower part of the figure is a case of a multiplicatively mediated linear self-stabilization. It will be approximated by using the operator shown in Figure 10b, which is justified by the strong stabilization by the chemical reaction and the comparatively slow changes in NH_4 (the effects differ by a factor of about 10^7) . The background knowledge about the orders of magnitude of the influences can be attached to the model fragments by the modeler (and propagated consistently through all abstraction operations), so the decision about the application of the approximation operator can be taken by formal reasoning about local information.

So the simpler model on the right hand side of Figure 11 can be used for purposes of middle and long term prediction with a substantial increase in efficiency. In our test runs with QSIM, we obtained much more focused predictions (usually a single one instead of more than 10 behavior branches) and all truly spurious solutions were ruled out.

Another Example: Motor with Control Circuit

We give a short example for the use of the modeling language and the transformation operators in the technical domain. We have modeled a direct current motor with control circuit (described in more detail in [Malik/Struss 96]). The influence diagrams of the components were derived directly from the following differential equations (for the parameter descriptions refer to the table below):

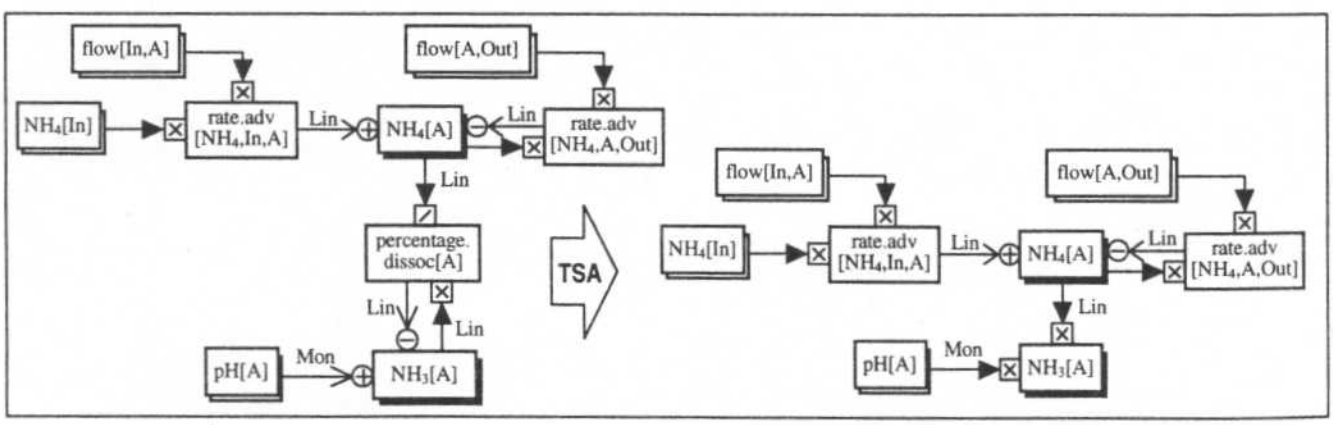


Figure 11: The example model. Left side: after the elimination of intermediate variables, Right side: after applying time-scale abstraction

$$v = \frac{2d - \omega_m}{c_C} \qquad \text{(controller)}$$

$$\omega_m = (1 - S) \cdot \omega \qquad \text{(sensor)}$$

$$\frac{d\omega}{dt} = \frac{c_M \cdot v - \omega}{T} \qquad \text{(motor behavior)}$$

v	driving voltage
ω	rotational speed (of the motor axis)
ω_m	measured rotational speed
d	desired rotational speed
c_C	controller constant
c_M	motor constant
T	motor inertia
S	slip (of the measuring pulse wheel)

From a simple structure description, the following influence diagram will be derived (Figure 12):

Application of the developed operators eliminates the constants (inclusively the derived constant 1-S) from the model (operator from Figure 9a) and also successively the intermediate variables 2d, $2d-\omega_m$, $c_M v$, $d\omega/dt$, and $c_M v-\omega$ (operators shown in Figure 9, b through d) yielding the simpler model in Figure 13.

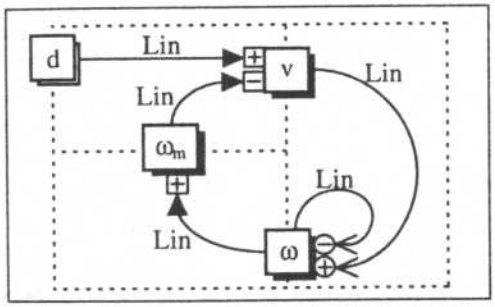


Figure 13: Motor model after the first simplification

In Figure 14 the result of further elimination (of ω_m and then v) is shown:

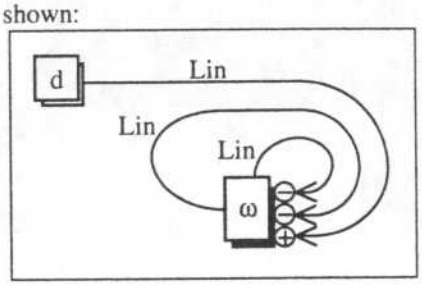


Figure 14: The motor model after the second simplification

Finally, subsumption of the resulting parallel influences (operator 9e) identifies the elementary influence structure as a (direct) linear self-stabilization that can be approximated by a functional dependency (see Figure 15, TSA-operator from Figure 10a):

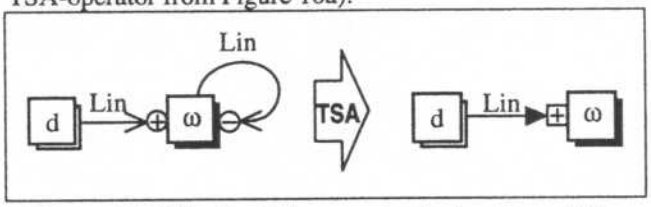


Figure 15: The final time-scale abstraction of the motor model

In the desired case (no slip: $S = 0$, controller constant equals motor constant: $c_C = c_M$), the approximation error can be bounded by $\frac{1}{2} \cdot T \cdot c$, c being a bound on the derivative of d, thus showing the response of the motor being dependent solely on the inertia. For details refer to [Heller 95].

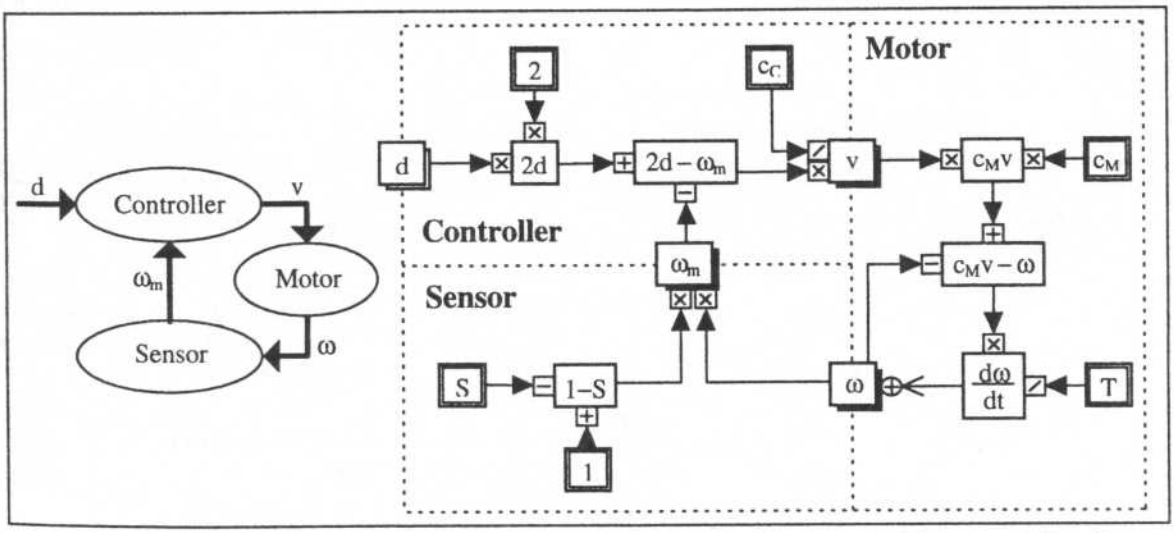


Figure 12: Structure description and influence diagram of a motor with control circuit

Future Work

What we have achieved at the present time, is an initial theory of transforming models of dynamic systems. The sets of operators developed and proved so far is certainly not complete. We also intend to introduce more function classes, for instance, in order to determine dominant influences in a combination of counteracting ones (e. g. overlinear versus linear growth).

Another theoretical issue is to analyze fixed points of the application of the set of operators. A goal would be a guarantee for deriving some normal form of a model, at least w. r. t. information about a specific task and objective of analysis. This raises the issue how to represent such information and how to control the application of the operators (as indicated in Figure 1).

An implementation will be done probably based on a hypergraph grammar approach. Such a model transformation module interacts to some degree with its neighboring tasks. Although the transformed model could, in principle, be fed to a system like QSIM, the increased power of our modeling language in terms of characterization of functional dependencies and the preservation of causal information will presumably also lead to stronger predictors. On the other hand, although a model stated in QPE/QPC could be an input, this would lack the distinctions necessary to achieve the strongest possible results. We consider to the graphical notation introduced in this paper for interactive definition of model fragments and composed models.

Some fundamental issues about the specification of model fragments have to be examined further. So far, the semantics are defined only for the completely composed (and closed) model. However, compositional modeling requires a formulation of isolated fragments in the first place, and it is not obvious what such a "context-free" model fragment "knows" about the appropriate way of combining with other fragments.

For some class of processes, like transportation and transformation processes, it is evident, that the effects combine additively with other processes. Possibly, the ontology has to be extended by a classification of processes which, based on their physical nature, uniquely determines the correct type of combination. Studies in other domains and with different examples will shed a light on different mechanisms for combining processes influencing shared variables.

Acknowledgements

We would like to thank for the valuable collaboration with François Guerrin and Waldir Roque. This work was partially supported by the Brazilian Research Council (CNPq) and the German Ministery of Education and Research (BMBF). We benefitted from discussions with Paulo Marcos Amaral Alves, Elenara Correa Lersch and Maria Mercedes Bendati from the research department of DMAE. Finally we would like to thank the reviewers for their helpful remarks.

References

[Crawford/Farquhar/Kuipers 90] James Crawford, Adam Farquhar, Benjamin Kuipers: QPC: A Compiler from Physical Models into Qualitative Differential Equations. AAAI 1990.

[Forbus 84] Kenneth Forbus: Qualitative Process Theory. In: International Journal of Artificial Intelligence 24 (1-3), 1984.

[Heller 95] Ulrich Heller: Temporale Verhaltensabstraktion am Beispiel von Regelkreisen und hydroökologischen Systemen. Master thesis, Department of Computer Science, Technical University of Munich, Germany, 1995. (in German)

[Heller/Struss 96] Ulrich Heller, Peter Struss: Qualitative Modeling for Environmental Decision Support. Symposium Informatik für den Umweltschutz, Hannover, Germany, 1996.

[Iwasaki 92] Yumi Iwasaki: Reasoning with Multiple Abstraction Models. In: Boi Faltings, Peter Struss (eds.): Recent Advances in Qualitatives Physics, MIT Press, 1992.

[Kuipers 86] Benjamin Kuipers: Qualitative Simulation. In: International Journal of Artificial Intelligence 29(3), 1986.

[Kuipers 87] Benjamin Kuipers: Abstraction by Time-Scale in Qualitative Simulation. AAAI 1987.

[Malik/Struss 96] Andreas Malik, Peter Struss: Diagnosis of Dynamic Systems Does Not Necessarily Require Simulation. QR 1996.

[Struss 92] Peter Struss: What's in SD? - Towards a Theory of Modeling for Diagnosis. In: Hamscher, Console, de Kleer (eds.), Readings in Model-based Diagnosis, Morgan Kaufmann, 1992.

[Struss 93] Peter Struss: On Temporal Abstraction in Qualitative Physics - A preliminary report, QR 1993.

[Williams 88] Brian Williams: MINIMA: A Symbolic Approach to Qualitative Reasoning, AAAI 1988.