QCMF: a tool for generating qualitative models from compartmental structures

L. Ironi[(1)] and M. Stefanelli[(2)]
[(1)] Istituto di Analisi Numerica del C.N.R., Pavia, Italy
[(2)] Dipartimento di Informatica e Sistemistica dell'Università di Pavia, Pavia, Italy
liliana@supers1.ian.pv.cnr.it

Abstract

This paper describes a framework, called QCMF (Qualitative Compartmental Modeling Framework), which assists the user in formulating models of a physical system and in analyzing their behaviors through the simulation of the effects of a variety of different perturbations of the system. QCMF has adopted the compartmental theory as modeling ontology: a system is represented as a finite set of interacting compartments. The user enters, through an iconic language and menus, the compartmental structure of a physical system and defines the kinds of functional relationships describing the interactions between compartments. Then, QCMF automatically generates a behavior model of the system. Such a model consists of a set of ordinary differential equations, which are currently qualitatively expressed, and is directly coded into the language (QSIM) which is interpreted by the simulation algorithm. The system behavior can be obtained by simulating the model starting from an initial state which describes the perturbations acting on the system. The code defining the initial state is automatically built by QCMF as well. Finally, explanations of the predicted behavior are also automatically generated. Moreover, QCMF is capable to build and to maintain a library of models for a given system.

Keywords: Mathematical modeling, qualitative modeling, qualitative simulation, compartmental theory, graphical interface.

Introduction

System dynamics modeling is a powerful tool for reasoning about physical systems. Model-based reasoning goes through different stages: first, model formulation and generation, then prediction and explanation of the system behavior. Model formulation, that is the definition of the relations between model variables, requires the expert's domain knowledge as well as a large body of domain independent knowledge. This includes mathematical concepts and fundamental principles of physics. Moreover, in order to be computationally tractable, the formulated model needs to be manipulated so that it can be represented through the language required by the adopted simulation algorithm. This part of the modeling process, called model generation, is strictly domain independent and has been proved to represent a heavy and time-consuming activity. In fact, although today available simulation languages are more user-friendly than standard programming languages, they still require precise syntax and semantics. A good knowledge of the simulation algorithm is also necessary for interpreting the simulation results. Hence, the need derives for building tools that have the necessary methodological knowledge for assisting the user in the model formulation phase, and for automatically executing the generation, simulation and explanation steps.

This paper describes a framework, called QCMF (Qualitative Compartmental Modeling Framework), for assisting the user in the process of model building (i.e., model formulation and generation) and simulation. The chosen modeling ontology is the compartmental one: a system is viewed as a finite set of *compartments* which interact by exchanging or transforming material. Flows through the system are represented as transfers from one compartment to another. Compartmental modeling can be properly exploited for the formal description of the dynamics of chemical reactions and material transfer processes. Although its most common application remains bio-medical science, its use has spread to other domains such as ecology, epidemiology, hydrology as well as chemical engineering.

QCMF includes knowledge of the theory of compartmental systems (Atkins, 1974), while the domain ontology, that is the decomposition of a specific system into compartments and the network of interactions between these, is entered by the user through an iconic language. A representation of the compartmental topology in a given system is called *structure model*, whereas the set of representations of the behaviors of each compartment, namely a set of Ordinary Differen-

*A more complete version of this paper will appear in *Computer Methods and Programs in Biomedicine* tial Equations (ODE) based on the mass balance law, defines a *behavior model* of the given structure. Within QCMF, a *system model* is represented by a behavior model associated with an initial state which describes the perturbations acting on a structure model.

Although our long-term goal aims at building a framework capable to execute the tasks of generating and simulating both quantitative and qualitative models, this paper deals only with qualitative models. Nevertheless, a quantitative formulation can be easily derived from the structure model by symbolically manipulating the acquired knowledge in order to simplify the process of writing the set of ODEs, and by mapping the behavior model into its corresponding numerical one according to the selected numerical simulation algorithm. Similar facilities are now provided by the most recent release of SAAM (SAAM, 1993), which could supplement QCMF with quantitative compartmental modelling tools.

QCMF embeds all the needed methodological knowledge, and therefore allows the user to go through the different modeling phases very easily:

1. A structure model is entered through a user-friendly graphical interface;
2. The behavior model is formulated by performing an automatic analysis of the given structure, and by acquiring information about functional relationships between variables;
3. The behavior model is automatically generated in the QSIM code (Kuipers, 1986);
4. The initial state corresponding to either single or multiple perturbations acting on the system is automatically generated in the QSIM code;
5. The system model is simulated in order to predict the effects caused by the considered perturbations;
6. A causal explanation of the simulated results, in addition to the graphical representation given by QSIM, is automatically generated.

The choice of QSIM among the several proposed approaches (Bobrow, 1984; Weld and De Kleer, 1990) to qualitative modeling is due to the fact that it seems to offer the most suitable formalism to represent the dynamics of processes whose behavior can be quantitatively described by ODEs (Kuipers and Kassirer, 1985; Kuipers, 1989).

Let us observe that each system can be described by several models, each of them providing a description at a different level of detail. This means that, according to the available knowledge and the purpose of model building, a system can be decomposed in different ways and different assumptions on the functional relationships between variables can be explored. Hence the need for building a library of models for a given system, where different entities composing models are represented through frames, and the whole set of frames is organized into a hierarchical structure. The problem of automatically selecting within the library the appropriate model for a specific goal is crucial in model-based reasoning (Addanki *et al.*, 1991; Falkenhaier and Forbus, 1991; Iwasaki, 1992; Nayak *et al.*, 1991; Weld, 1990), and will be considered in the future work.

QCMF, which has a practical value as a stand-alone system, could also be viewed as an acquisition tool of knowledge represented through system models. The availability of a library of models allows a Knowledge-Based System (KBS) to reason exploiting deep knowledge, so enhancing its problem solving ability.

This paper is organized as follows: after a brief introduction to compartmental theory, the next sections deal with an overview of QCMF. The user-QCMF interaction in model formulation, generation and simulation phases is illustrated through an example taken from the medical domain. Finally, the major strenghts and limits of this work are discussed.

Compartmental modeling

A compartment is fundamentally an idealized store of a substance, homogeneously distributed: if a substance is present in a physical system in several forms or locations and passes from one form or location to another, then each form or location constitutes a separate compartment for that substance. It should be noted that different compartments can coexist in the same location of a system and that the compartments do not always correspond to physically identifiable components.

Such modeling technique is applicable when the study concerns the flow of some substance through the system. In such models, the flow through the system is represented as transfers from one compartment to another. The *structure model*, that is the system's decomposition and the network of interactions between compartments, is usually displayed by means of a compartmental diagram. Decomposing a system into compartments leads directly to its *behavior*, namely a set of equations based on the mass balance law. The state variables of the system, which are denoted by $x_i(t)$, represent the concentration or amount of substance in the i–th compartment which exchanges matter with other compartments or the external environment at time t. The rate of change of x_i is obtained by the equation:

$$\dot{x}_i = f_{i0} + \sum_{\substack{j=1 \\ j\neq i}}^{n} f_{ij}(x_j) - \sum_{\substack{j=1 \\ j\neq i}}^{n} f_{ji}(x_i) - f_{0i}(x_i) \quad (1)$$

where $\dot{x}_i$ denotes the time derivative of x_i; f_{ij} and f_{ji} denote flow variables, that is the rates of mass transfer into the $i-th$ compartment from the $j-th$ compartment and into the $j-th$ compartment from the $i-th$ compartment, respectively. (The compartment 0 denotes the external environment). In general, the flow of material depends on the quantity or concentration of material in the source compartment and may also be controlled by the quantity or concentration in some other compartments, that is:

$$f_{ij} = f_{ij}(x_j; x_l, x_m, ...)$$

where x_j, x_l, x_m, .. denote the state variable of the source compartment and the variables which control f_{ij}, respectively. The dependency of a flow variable on state variables can be either linear or non-linear. Both state and flow variables are always non-negative.

In order to formulate the behavior model, the functional dependencies of each flow f_{ij} on x_j in (1) must be expressed. The nature of these functional dependencies will vary according to the system under investigation and will be suggested either by observing the real system behavior or by the available knowledge.

Overview of QCMF

QCMF is a tool which integrates the steps necessary to reason about a system behavior: model building, model simulation, and results explanation. The interest for automating such processes has recently arisen, and a number of methods has been proposed by several researchers (e.g. (Addanki *et al.*, 1991; Falkenhaier and Forbus, 1991; Iwasaki, 1992; Nayak *et al.*, 1991; Weld, 1990; Capelo *et al.*, 1993; Crawford *et al.*, 1992; Low and Iwasaki, 1992)).

Fig. 1 gives an overview of QCMF's modules . It is menu-driven and its major components are the GENERATE, SIMULATE, EXPLAIN and BROWSE modules.

The GENERATE module helps (1) the user in entering both the compartmental structure and the assumptions about the functional relationships between variables; (2) then, it automatically analyzes the structure in order to (3) write down the model equations into the QSIM language.

The so generated qualitative model is given as input to the SIMULATE module. This module (1) manages the information entered by the user about the perturbations acting on the system; (2) it updates, if needed, the behavior model; (3) it automatically generates the initial state in the QSIM language; (4) it predicts, by simulating the model, the possible behaviors of the system which are presented to the user in a graphical form. The analysis of such an outcome can suggest a model revision of either the structure and behavior model or the initial state.

The user is allowed to request a causal explanation of the predicted behaviors. The EXPLAIN module has access to all the information about the system model and to the predicted behavior to be explained in order to generate a causal chain describing the system transition from one state to another. To this end, it exploits: (1) the structure model as a causal graph where the effects of a change propagate following the paths in the model topology, (2) the behavior model to detect the 'primary' causes of the change, and (3) knowledge of the QSIM algorithm, namely the transition rules, to

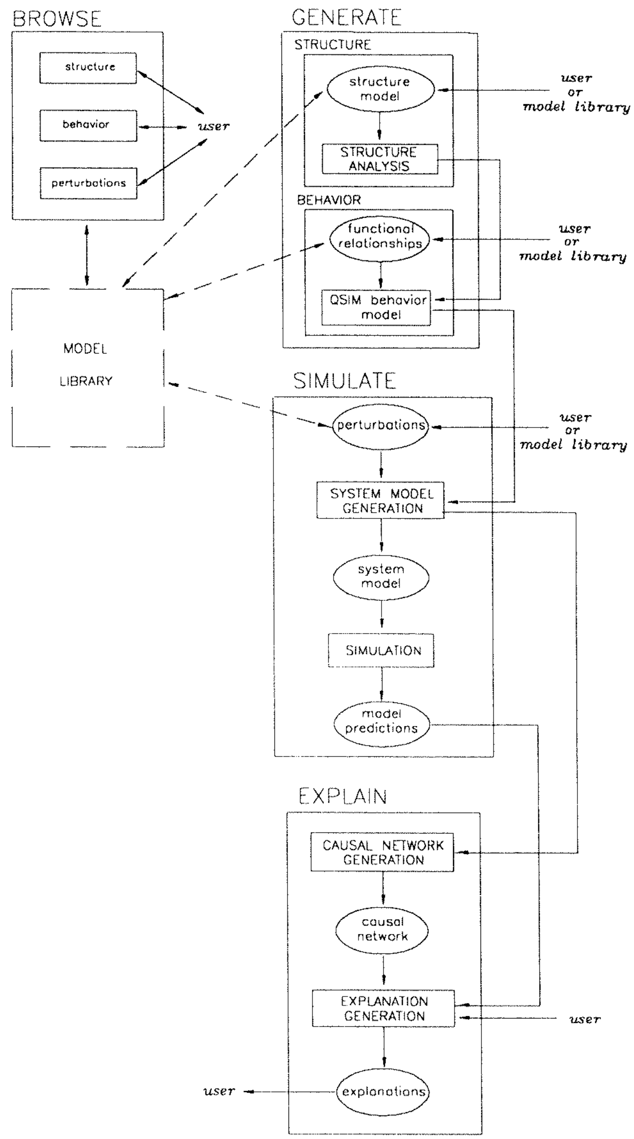

Figure 1: Overview of QCMF's modules. Information entered by the user are explicitely indicated by arrows. Rectangules indicate tasks automatically executed by the system

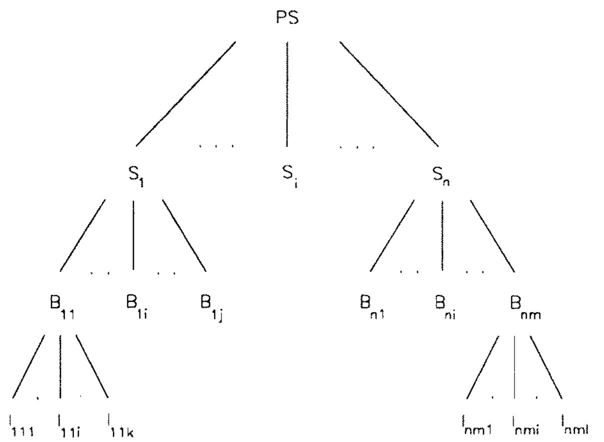

Figure 2: The hierarchical structure of the library of models

identify which changes occur from one system state to its successor.

Moreover, the GENERATE and SIMULATE modules organize and maintain a library of models which can be efficiently retrieved and used by a KBS for a given goal. Each model M can be identified with the tuple (PS, S, B, I), where PS denotes a physical system which has been modeled by specifying for each considered structure model (S) different behavior models (B), each one requiring the definition of the considered perturbations (I). All these entities are represented by frames which are hierarchically organized in a tree structure as shown in Fig. 2. Any branch of the tree represents a different model M of PS. It may represent either a different level of abstraction (i.e., different structure model), or a different theory (i.e., different behavior model), or different working conditions (i.e., different perturbation). Thus, each piece of knowledge entered by the user during her interaction with QCMF fills a slot of a suitable frame of the model components hierarchy shown in Fig. 2. This view makes clear which different models have been built for representing the available knowledge. This organization is useful on one side to QCMF to correctly access the model library, and on the other side to the user to easily examine the assumptions made in building the models. The latter of these operations is performed by the BROWSE module, which provides facilities to visualize to the user all the information stored in the library.

Model formulation and generation

Two different sub-modules, which can be selected through a sub-menu associated with the GENERATE button, deal separately with the structure and behavior model.

Structure model

The structure model consists of information about the decomposition of the system into compartments, and about the network of interactions between compartments and external environment.

A user-friendly graphical interface allows the user to enter the definition of the structure model and/or to edit a previously defined structure. Each object of the structure, i.e. compartments, flows, controls, must be defined by the user according to the available knowledge. Each object is represented through an icon, and is entered by the user by clicking the corresponding button. According to the selected object, further information must be entered by the user. Namely, it is required to specify:

- for a compartment, the name of the substance it contains, and the position on the input/output graphical window;
- for a flow, the connections between compartments and the exchanges with the environment, i.e. which compartment the flow is leaving from to reach another compartment or the environment, and which compartment the flow is entering from another compartment or from the environment;
- for a control, the controlling variable and the controlled flow.

Each compartment is automatically labeled with a number, and the flows are automatically labeled taking into account their direction. This facilitates the model formulation process since the direction of each flow can be easily identified, as well as the model variables. Thus, writing the equations in the behavior model generation phase is straightforward. All input operations are guided: they mostly occur through either menu and/or the selection of compartments.

The components of the structure are recorded as values of slots of the *structure frame* that the user can modify at any time during the generation phase. The values associated with the compartment, flow and control attributes/slots are respectively the list of the compartment number labels with their respective dubbed names, the list of the flows, and the list of the controlling compartment number labels with their respective controlled flows. Moreover, information about the system which the structure is related to and the list of its associated behavior models have to be recorded into slots of the structure frame in order to suitably locate the structure within the library of models. Therefore, the structure frame is described by five slots, namely physical-system, behaviors, compartments, flows and controls.

In order to draw a clear and nice view of the compartmental diagram, functions looking for the minimal connection path of two compartments, and avoiding the overlapping of objects have been implemented. Graphical editing facilities are available to the user in order to modify a structure: compartments, flows and controls can be deleted. The system updates the structure frame and all related lists and the compartmental diagram, that is the compartments and flows are relabeled. The structure model can be stored, and then retrieved, through the frame and lists containing graphical information. However, the saving operation is performed only after an automatic debugging of the introduced structure ensures that it is an acceptable compartmental structure, i.e. all compartments are in some way connected among themselves.

Behavior model

The formulation of the behavior model of a given structure requires to specify the variables, the functional relationships between variables, and then to write down the differential equation (1) for each compartment.

The identification of all the relevant variables is automatically performed by QCMF by exploiting the information represented in the structure: the number of compartments allows us to identify the state variables x_i, and their time derivative variables $\dot{x}_i$; the flow slot permits the identification of the flow variables f_{ij}, and for each of them to state if it is a leaving or entering flow, and consequently its dependency on the variable x_j. The possible dependency of f_{ij} on other variables can be derived by reading the content of the control slot. In the case of controlled flows, the multi-valued functional dependencies between flow and state variables are expressed as sum or product of single-valued functions. These functions, which actually are auxiliary variables, are automatically dubbed and introduced into the list of variables. Then, the initial quantity spaces L_i must be specified for each variable. First, the sets L_i associated with the flow and state variables are instantiated to (0 INF) for the non-negativity property of these variables, whereas the sets L_i associated with the variables $\dot{x}_i$ are instantiated to (MINF 0 INF), where MINF and INF in the QSIM notation mean $-\infty$ and $+\infty$, respectively. As soon as further information about the system is provided by the user, the sets L_i are updated by inserting new landmarks values. The order relation of the new landmarks with respect to the previously defined ones is stated by the user, through menus, on the basis of her knowledge.

In addition to update the quantity space associated with each variable, and to translate into the QSIM language the relationships between variables, QCMF derives automatically for each compartment the differential equation which describes the dynamics of the contained substance. First, for each compartment, all entering and leaving flows must be identified so that the corresponding mass balance equation (1) can be written. Moreover, the analysis of the compartmental diagram allows us to identify particular features of the structure, and to write further equations describing global properties of the system (for example, the mass conservation law for closed structure). The translation into the QSIM qualitative constraints is straightforward in most cases. However, since QSIM constraint operators are relations and not functions, the syntax does not allow nesting of constraints to create complex expressions. In such cases, QCMF introduces a variable, which has no physical meaning, to represent intermediate results. In the following, the variables x_i, f_{ij}, $\dot{x}_i$ are denoted by Xi, Fij, DXi, respectively, when they are instantiated by QCMF. The kind of relationship between variables is suggested by the specific domain knowledge: therefore, it is entered by the user who is helped in this operation by a menu. The allowed kinds of relationships allow the user to describe qualitatively a wide set of functional dependencies between variables in terms of their regions of monotonicity.

Information about the behavior model is stored into a frame, which is represented by five slots, namely physical-system, structure, perturbations, flows and controlled-flows. The first three slots store all the information needed for identifying, in the hierarchical tree structure of the model library, the branch which the behavior model belongs to, and its possible associated branches. Their values correspond, respectively, to the names of the physical system and the structure which the behavior is related to, and to the list of its associated perturbations. The flows and controlled-flows slots allow us to characterize a behavior model of an underlying structure, that is they give information about the assumptions made on the functional relationships between flows and state variables. The values given to these slots are the list of flows, or auxiliary variables defining flows, with their respective information about the assumed type of relationship, and the list of controlled flows with their associated kind of algebraic operator which allows us to define the multi-valued functional dependency of the flow on the state variables.

Model simulation

The main task of the SIMULATE module deals with the automatic generation of the initial state of the perturbed system directly in the QSIM code, i.e. the automatic definition for each variable of its qualitative value (*qval*) and its direction of change (*qdir*). QCMF tries to build, starting from the values of the perturbed variables, the initial state of the subset of system variables which guarantees the generation of a single complete system state. QCMF builds the initial state of the variables x_i and $\dot{x}_i$. Knowing the state of such variables is sufficient when the flow variables are not controlled, otherwise the user may be asked to give further information, if any, about the state of the controlled flows which, being the result of algebraic operations between single valued functions, could be not univocally determined. Such a partial initial state together with the behavior model are given as input to QSIM, which determines its consistent completion by propagating the state values of the specified variables through the constraints. The basic steps in the design and implementation of the algorithm generating the initial state are the following:

1. the user enters, through menus, the qualitative value of one or more perturbed variables;
2. QCMF automatically identifies, by analyzing the behavior model, the variables which are linked to the perturbed ones by functional relationships, and then indirectly interested by the perturbation, and determines their *qvals*;
3. the sign of $\dot{x}_i$ is calculated by exploiting the *qval* of the variables f_{ij} appearing in the equation which defines $\dot{x}_i$. This allows us to determine the *qdir* of x_i, and of all the flow variables linked by functional relationships to x_i. Multiple perturbations are allowed. Whenever, due to multiple perturbations, there is ambiguity in defining the sign of $\dot{x}_i$, the user is asked to state, according to the available knowledge, which is the prevailing perturbation.
4. the *qdirs* of the variables $\dot{x}_i$ are determined by considering the *qdirs* of the related flow variables. When a *qdir* can not be univocally determined, the user is asked to select one on the basis of her knowledge.

Once the initial state has been defined, the user can run a simulation by selecting the related option in the SIMULATE menu or simply save the generated code. Another task performed by the SAVE option is the updating of the behavior frame and its associated perturbation frame within the library of models. The *perturbation frame* is described by four slots, three of them identify the physical system, the structure and the behavior models the system perturbations refer to, and the last one records the perturbed variables and their associated perturbation.

The user is allowed to consider two different kinds of scenario in defining the initial state. In one case, the effects of perturbing a reference steady state of the system can be simulated; in the other one, a tracer kinetics can be analyzed. Although there is no conceptual difference in building the initial state, two sub-modules, which are selected through a sub-menu of the SIMULATE button, deal separately with the different scenarios. The main difference between the two sub-modules concerns the interaction between the user and QCMF.

Perturbation of the steady state

System models should describe how one or multiple disturbances on a system cause the transition from its normal state to an abnormal one. Then, such models are obtained by perturbing the normal state supposed to be steady, i.e. by changing the qualitative state of some variables, and possibly by introducing either structural or behavioral changes in the model describing the physical process during the GENERATION phase. On the other hand, it is equally interesting the simulation of the time evolution of the system from an abnormal steady state to a normal one, for example , in a medical context, caused by a therapy.

In both cases, the initial state is defined by both the reference steady state and the perturbed one. Therefore, the first task executed by this sub-module is the definition of the steady value of each variable. This operation is performed automatically by QCMF whenever the quantity space of the variable defined in the behavior model is $(0\ \infty)$; otherwise the user is asked to specify, through a menu, in which interval the steady value of a variable lies. New landmarks representing the steady values are added to their respective quantity spaces in the behavior model by exploiting the functional dependencies between variables, and new corresponding values are added to the corresponding constraints. Then, the user must choose, always through a menu, which variables are perturbed, whether their value is greater or lower than the steady one, and possibly the ordinal relation with the other landmarks. The options offered in the menus represent either the variables that can be perturbed, i.e. state variables, constant flows and fractional transfer rate coefficients which define linear flows, or the intervals where the steady and the perturbed values can lie. As the system is initially supposed to be in equilibrium, this means that the *qval* of $\dot{x}_i$ must be equal to zero, that is the flows entering a compartment perfectly balance the flows leaving the compartment. QCMF checks the consistency of the specified steady values with the structure model.

Tracer kinetics

Compartmental modeling plays an important role in describing both the distribution and metabolism of either tracer labeled compounds or drugs. An ideal tracer experiment consists in the application of a test signal having negligible mass and behaving metabolically exactly as the original substance to which is added as a marker. A number of useful properties emerge (Carson *et al.*, 1983) from such hypotheses. First, the tracer exhibits linear dynamics if the system is in a constant steady state and irrespective of whether or not the functional dependencies defining the intercompartmental flows are linear. Furthermore, these linear dynamics are time invariant. Qualitative models of tracer kinetics can easily allow the user to investigate the consequences of removing the hypothesis of "small-signal perturbation" by considering non linear functional dependencies.

In this scenario, the generation of the initial state could be seen as a particular case of the perturbation of a steady state where the steady value of each variable is assumed to be equal to zero (this means absence of tracer in all compartments), and there is a single perturbation which corresponds to the introduction of a tracer labeled compound into one compartment. However a specific sub-module has been implemented mainly to manage in a suitable way the dialogue with the user. Within this context, the variables represent the concentration and transfer of the tracer substance.

The user must choose, through menus, both the compartment where the tracer is introduced and the kind of input signal for the tracer. The input signal can be either an impulse (injection experiment) or a step signal (infusion experiment). QCMF verifies if the structure model is consistent with the chosen input signal, that is the absence or presence of a flow from the external world to the compartment, respectively.

Behavior explanation

We consider explanations as presentation of information that offer a meaningful interpretation of simulation results, which describes, in a language that is comprehensible by the user, how and why a behavior occurs. Thefore, the core of the EXPLAIN module, which has been fully designed, but still under development, deals with a method for generating a causal interpretation of behaviors; more precisely, given a succession of qualitative states describing a simulated behavior, QCMF identifies the cause-effect links between variables in successive states. One of the basic distinctions between the proposed different approaches to the generation of causal explanations of the system behaviors (De Kleer and Brown, 1986; Iwasaki and Simon, 1986a; Iwasaki and Simon, 1986b; Gautier and Gruber, 1993) is whether the relations or constraints used in the model are directional or not. In the former case, a causal account is simple to be obtained because the values are propagated through the directional constraints supplied by the model builder. Alternatively, bidirectional constraints require causal ordering techniques because they lack an explicit representation of causality. The context in which the EXPLAIN module generates a causal account is limited to compartmental modeling methodology in which relationships are directional. This allows us, given a bidirectional QSIM model, to interpret the system's behavior by combining simulation results and the knowledge of the directional relationships expressed through the compartmental diagram. In the transition from a state (QS_i) to its successor (QS_{i+1}) one or more variables will change qualitative state. This transition from state QS_i to QS_{i+1} will be necessarily due to the *qdirs* in state QS_i describing the evolution occurring in the system at time t_i. This principle is the basis of the algorithm underlying the EXPLAIN module because it allows QCMF to causally concatenate the changes from a state to its successor: the algorithm when comparing two successive states will single out those changes that are a direct consequence of the directions of change in QS_i and will propagate the effects of these changes through the directional constraints to account for the remaining changes. Applied to a behavior chosen by the user, this procedure is repeated beginning from the initial state to the whole sequence of qualitative states describing the behavior. The causal account we are aiming at arises from chaining the separate accounts obtained for the successive transitions. The user will be allowed to ask for causal explanations of either a specific behavior of the system or the differences the whole set of possible system behaviors exhibits. The presentation of explanations are provided at different level of details: first, the relevant events are presented, and then, on the user request, more and more detailed causal explanations for any change in the system variables are produced.

An example of application of QCMF to the medical domain

The lack of methodological knowledge necessary for building and simulating models has been a serious deterrent for physicians, much more than for expert in other domains, from using system models in spite of their great utility for reasoning about pathophysiological systems. As shown in (Ramoni *et al.*, 1992), medical reasoning may be broken down into two different phases: first, initial information is exploited to select candidate problem's solutions (*hypothesis selection phase*), and then these solutions are used as starting conditions to forecast expected consequences that should derive if adopted as final solutions (*hypotheses testing phase*). Diseases represent the solutions of diagnostic reasoning, i.e. the search for the best explanation of the current situation of a patient, while treatments represent the solutions of a therapy planning problem, i.e. the search for the best action to ameliorate a patient's conditions. Structure and behavior models provide an essential knowledge representation formalism for exploiting genuine pathophysiological theories, when available, in the testing phase of medical reasoning (Ironi *et al.*, 1990): the consequences of a selected solution can be obtained by simulating a model of the considered pathophysiological process. A disease can be modeled by changing the values of some model variables, and possibly either the functional relationships between variables or the structure model with respect to the one describing the physiological behavior. Thus, the model simulation provides the new steady state, which is abnormal, as well as the transient to achieve it. In case of therapy planning, also the action of the hypothesized treatment needs to be modeled: for example a drug's farmacokinetics is essential to describe how it affects the system's behavior. Thus, a KBS should contain different structure and behavior models which could be used whenever the solution's hypotheses they represent are selected as candidate solutions. In the medical context, QCMF can be viewed as a powerful tool for the acquisition of pathophysiological knowledge.

Let us consider, for example, the Glucose-Insulin Regulatory system (G-I-R) in order to describe QCMF at work. After the selection of the system, the user is then allowed to enter one of the model-reasoning phases. The structure model shown in Fig. 3 corre-

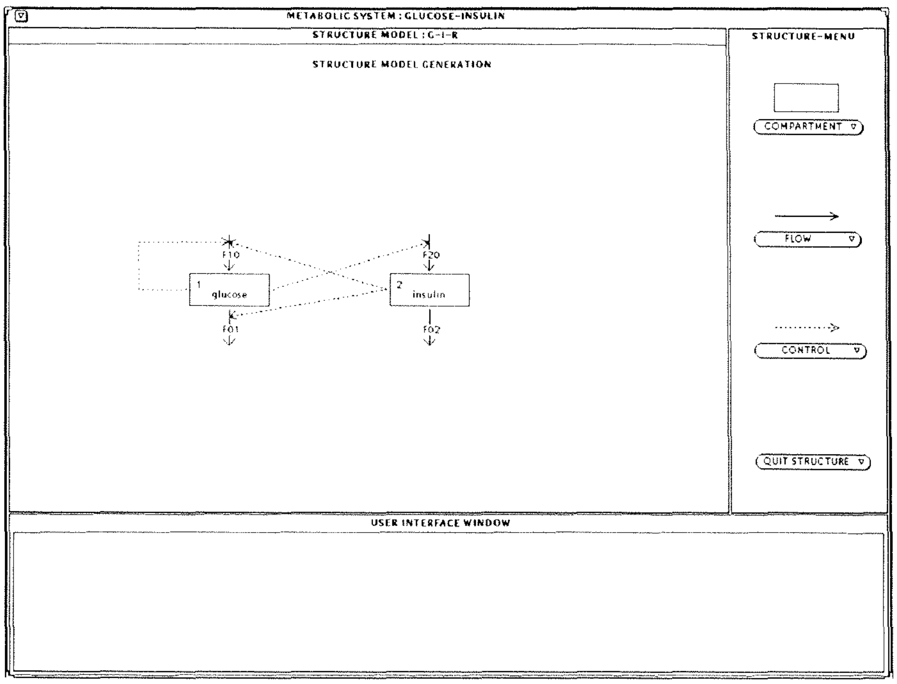

Figure 3: A structure model of the glucose-insulin regulatory system

sponds to the simplest decomposition for G-I-R, and includes two compartments: glucose and insulin.

The glucose is described by a single compartment representing extracellular fluids. The processes that have been explicitly considered are the liver glucose production f_{10} and the glucose elimination f_{01}. The liver glucose production depends on both extracellular glucose and plasma insulin; while the glucose elimination is due to renal excretion, as well as to the insulin-dependent glucose utilization by muscles and adipose tissues and the insulin independent glucose utilization by the central nervous system and red blood cells. The insulin subsystem is also described by a single compartment representing plasma insulin. The insulin production derives from the pancreatic response to glucose stimulation. There is no exchange of flow between the two compartments, but they interact to control the glucose production and elimination and the insulin production.

At first, the automatic analysis of the structure in Fig. 3 identifies the following variables: X1, X2, DX1, DX2, F10, F01, F20, F02. The information stored in the structure frame (Fig. 4) allows QCMF to identify which flow and state variables are related to. The flow F01 depends on X1 and on the controlling variable X2, F10 depends on both X1 and X2, while F20 and F02 are respectively function of X1 (controlling variable), and X2.

As far as the liver glucose production (F10) is concerned, it is reasonable to assume that F10(X1,X2) is expressed by summing two single valued functions of

name: *G-I-R*

pathophysiological-system: *Glucose-Insulin*

behaviors: *normal beh-1 beh-2*

compartments: *1-glucose 2-insulin*

flows: *F10 F01 F20 F02*

controls: *1-F10 2-F10 2-F01 1-F20*

Figure 4: The frame representing the structure model of the Glucose Insulin Regulation system shown in Fig. 3. Slot names are indicated in bold while slot values in italic

X1 and X2, whose dubbed names X1F10 and X2F10 are automatically built by joining the names of the controlling variable and the controlled flow. Both X1F10 and X2F10 can reasonably be assumed to behave as a monotonic decreasing function of X1 in the interval (0 X1*) and of X2 in the interval (0 X2*), respectively, and constant elsewhere. Analogously, the function F01(X1,X2), describing the glucose elimination, can be defined by the sum of X1F01 and X2F01, where X1F01(X1) can be assumed to be monotonic increasing, and X2F01(X2) monotonic increasing in the interval (0 X2**), and constant elsewhere. The production of insulin (F20) can be reasonably assumed to increase monotonically with the concentration of glucose (X1) reaching a saturation value (F20**) for

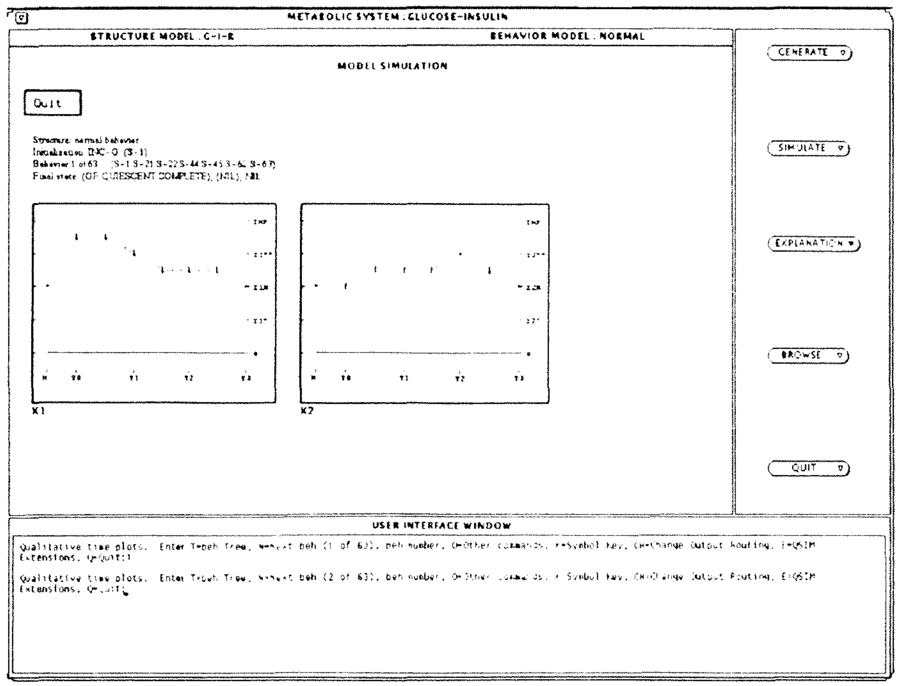

Figure 5: A possible behavior of the G-I-R system in response to an increased glucose concentration with respect to the normal value, denoted in correspondence to N

X1=X1**. Such a choice is supported by available experimental knowledge. All the above mentioned landmarks are automatically generated and included in the quantity spaces of their respective variables. Let us notice that a landmark can also be removed from its quantity space when new information allows us to narrow either the domain or the codomain of the corresponding variable. For example, the quantity space of F20, firstly instantiated to $(0\ \infty)$, is restricted to (0 F20**) as F20** represents the maximum value F20 can take on. The insulin elimination F02 can be assumed to be a monotonic increasing function of X2. As far as the liver glucose production (F10) is concerned, it is reasonable to assume that F10(X1,X2) is expressed by summing two single valued functions of X1 and X2, whose dubbed names X1F10 and X2F10 are automatically built by joining the names of the controlling variable and the controlled flow. Both X1F10 and X2F10 can reasonably be assumed to behave as a monotonic decreasing function of X1 in the interval (0 X1*) and of X2 in the interval (0 X2*), respectively, and constant elsewhere. Analogously, the function F01(X1,X2), describing the glucose elimination, can be defined by the sum of X1F01 and X2F01, where X1F01(X1) can be assumed to be monotonic increasing, and X2F01(X2) monotonic increasing in the interval (0 X2**), and constant elsewhere. Of course, other kinds of functional dependencies between variables could be hypothesized, and, therefore, the assumed behavior model is not the only possible one for the structure in Fig. 3.

Let us now suppose that the perturbed condition of G-I-R consists of an increased glucose concentration with respect to the normal value (reference value). The initial state generated by QCMF results in a LISP function describing both the normal state and the perturbed one. By the analysis of the behavior model, the variables identified as possibly interested by the perturbation of X1 are F20, X1F10, X1F01, and consequently F10 and F01. As the normal value of X1 is supposed to lie in the interval (X1*, X1**), as well as X2 does in (X2*, X2**), F20, X1F01 and F01 take on a value greater than the normal one, whereas X1F10, and then F10, do not change since the perturbed value of X1 still lies in the saturation interval. Therefore, it is DX1 < 0 (DX1= F10 - F01), and DX2 > 0 (DX2= F20 - F02). This means that *qdir*(X1) is decreasing and *qdir*(X2) increasing, and consequently, by the analysis of the constraints, it is *qdir*(DX1) increasing and *qdir*(DX2) decreasing.

In response to the assumed perturbation, two classes of behaviors have been obtained: transient courses showing damped oscillations or not. An example of behavior taken from the latter class is shown in Fig. 5. Let us notice that all the behaviors generated by the simulation are featured by a final steady state where all the variables assume again the normal value. This is the expected result as the considered model represents the physiological behavior of G-I-R . The simulated behaviors show all the possible ways the system in a perturbed condition reacts to bring itself back to the

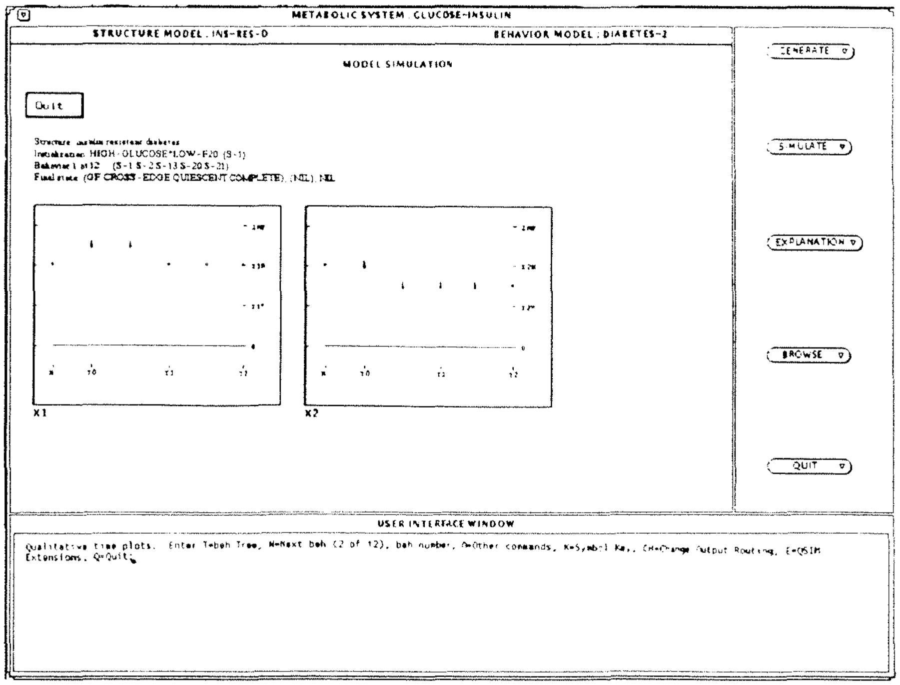

Figure 6: A typical behavior of the type II diabetes model in response to an increased value of the glucose concentration with respect to the normal one

normality. The behaviors grouped into the two classes mainly differ for the different temporal orders in which different events occur. More interesting scenarios describing G-I-R abnormal behaviors, such as type I and type II diabetes, cannot be obtained by simply perturbing the variables of the previous defined model, but structural and behavioral changes are needed. In type I diabetes there is almost no insulin secretion from pancreatic cells, that is the insulin production is not controlled any more by the glucose concentration. In type II diabetes, in addition to a reduced production of insulin, the glucose uptake by muscles and adipose tissues does not depend on insulin. As far as the behavior models are concerned, in the first case the variable F20 does not appear any more in the list of system variables and constraints. In the second case the variable F01 can be supposed monotonically increasing with X1, and F20 a constant function whose value is lower than the normal one.

Fig. 6 shows a typical response of the type II diabetes model in response to an increased value of the glucose concentration with respect to the normal one. As expected, as the glucose elimination is insulin resistent, the glucose concentration can reach again the normal value, whereas the insulin, for its reduced production, decreases and achieves a steady value lower than the normal one. Nevertheless, another class of predicted behaviors, where the glucose concentration achieves a steady value lower than the starting value but still higher than the normal one, is equally acceptable from a pathophysiological point of view.

Conclusion

Compartmental modeling is a well-known modeling technique for the representation of the behavior of complex systems, and its use by researchers in physics, biology and medicine dates back to the 1920's. In the compartmental approach, the behavior of a complex system is represented through an idealized decomposition of the system into interacting compartments, whose behavior is basically governed by the mass balance law. The compartments do not necessarily correspond to actual distinguishable physical components unlike the device-centered approach in which components have a direct correspondence with the different physical parts an artifact is made up of.

This paper describes the architecture of QCMF, a computer-based framework which integrates facilities for generating a model of a compartmental system, simulating its behavior and providing a causal explanation of the obtained behavior.

In other works (Addanki *et al.*, 1991; Falkenhaier and Forbus, 1991; Low and Iwasaki, 1992) dealing with qualitative model construction, the modeling problem is approached by selecting, within a large predefined model library, model fragments, and possibly by composing them. In particular, the selection is performed according to either a set of assumptions or a user's query about the system's behavior. In these works, the model formulation procedure is based on QPT (Forbus, 1984). QCMF, which is based on QSIM, allows the modeller to specify in a graphical way, the compartmental structure of any new system, and acquires the needed knowledge to generate a system model. Then, a new model is stored into a library from which it can be easily retrieved. The basic model components, i.e. compartments, are not worth to be stored separately since they are not physiologically meaningful outside the context where they are placed, i.e. the overall model. This represents the major difference between the two approaches: in the former one components can be fully defined independently from their combination into a specific model, in the latter one models can be built instantiating few conceptual entities, i.e. compartments, flows, control signals, and so on.

In this paper the main emphasis has been given to the description of QCMF's ability in model formulation, generation and simulation. These facilities are now fully operational. The main problems we faced deal with the simulation process. QCMF inherits from QSIM all of its strenghts and limitations (Fouchè and Kuipers, 1992; Kuipers, 1987; Kuipers and Bearlant, 1988; Kuipers *et al.*, 1991; Lee and Kuipers, 1988; Shen and Leitch, 1991; Struss, 1988): it is capable to predict all possible system behaviors but can lead to an intractable branching of the behavioral tree. In order to control the simulation, QCMF automatically invokes filters for reducing the proliferation of behaviors, namely the Higher Order Derivative (HOD) constraint (Kuipers *et al.*, 1991) or, whenever this is not applicable, the chatter box elimination technique (Clancy and Kuipers, 1993). Whenever it is possible, QCMF also exploits filters expressing the mass conservation law by introducing into the behavior model the suitable variables and constraints. Another way to further reduce the behavior tree consists in performing an attainable envisionment, i.e. no new landmarks are introduced. As new landmarks are related to a specific behavior, this makes easier the comparison of behaviors allowing us to aggregate the similar ones. The aggregation of behaviors aims at taking the significant distinctions for the user needs out of the behavior tree. At a first instance, the user could be interested only in the distinctions either in a subset of variables or in the final equilibrium state. For example, in a therapeutic context, knowing if a system in an abnormal steady state, when perturbed by a therapy, is capable to achieve a normal state is the essential information. In fact, if the set of produced alternatives did not contain the expected behaviors, the perturbed model should be revised.

As regards its possible use, many issues can be considered. First, QCMF can work as a stand-alone system resulting in a powerful didactic tool for reasoning about the pathophysiological behaviors of a system. Then, QCMF can be fully integrated within larger knowledge-based systems that use different formalisms. In fact, it may generate knowledge sources that can be properly exploited in the deductive inference of medical reasoning in the execution of both diagnostic and therapeutic tasks.

Acknowledgements

We especially thank B. Kuipers who provided the Common LISP implementation of QSIM. We are also indebted to A. Cattaneo who implemented part of QCMF in his Master Thesis, to D. Clancy for providing his last extensions to QSIM, and to S. Tentoni for technical support. This work is part of the AIM project A2034, a General Architecture for Medical Expert Systems II (GAMES II), which is supported by the European Community.

References

Addanki, S.R.; Cremonini, R.; and Penberthy, J.S. 1991. Graphs of models. *Artificial Intelligence* 51:145–177.

Atkins, G.L. 1974. *Multicompartmental Models in Biological Systems.* Chapman and Hall, London.

Bobrow, D.G.(ed) 1984. *Qualitative Reasoning about Physical Systems.* North Holland, Amsterdam.

Capelo, A.C.; Ironi, L.; and Tentoni, S. 1993. A model-based system for the classification and analysis of materials. *Intelligent Systems Engineering* 2(3):145–158.

Carson, E.R.; Cobelli, C.; and Finkenstein, L. 1983. *The Mathematical Modeling of Metabolic and Endocrine Systems.* Wiley, New York.

Clancy, D.J. and Kuipers, B.J. 1993. Behavior abstraction for tractable simulation. In *Proc. 7th International Workshop on Qualitative Reasoning*, Seattle. 57–64.

Crawford, J.; Farquhar, A.; and Kuipers, B.J. 1992. QPC: A compiler from physical models into qualitative differential equations. In B. Faltings, P. Struss, editor 1992, *Recent Advances in Qualitative Physics.* MIT Press. 17–32.

De Kleer, J. and Brown, J.S. 1986. Theories of causal ordering. *Artificial Intelligence* 29:33–61.

Falkenhaier, B. and Forbus, K.D. 1991. Compositional modeling: finding the right model for the job. *Artificial Intelligence* 51:95–143.

Forbus, K.D. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Fouchè, P. and Kuipers, B.J. 1992. Reasoning about energy in qualitative simulation. *IEEE Transactions on Systems, Man and Cybernetics* 22:47–63.

Gautier, P.O. and Gruber, T.R. 1993. Generating explanations of device behavior using compositional modeling and causal ordering. In *Proc. 7th International Workshop on Qualitative Reasoning*, Seattle. 89–97.

Ironi, L.; Stefanelli, M.; and Lanzola, G. 1990. Qualitative models in medical diagnosis. *Artificial Intelligence in Medicine* 2:85–101.

Iwasaki, Y. and Simon, H.A. 1986a. Causality in device behavior. *Artificial Intelligence* 29:3–32.

Iwasaki, Y. and Simon, H.A. 1986b. Theories of causal ordering: reply to de kleer and brown. *Artificial Intelligence* 29:63–72.

Iwasaki, Y. 1992. Reasoning with multiple abstraction models. In Faltings, B. and Struss, P., editors 1992, *Recent Advances in Qualitative Physics.* MIT Press. 67–82.

Kuipers, B.J. and Bearlant, D. 1988. Using incomplete quantitative knowledge in qualitative reasoning. In *Proc. AAAI-88*, Los Altos. Morgan Kaufmann. 324–329.

Kuipers, B.J. and Kassirer, J.P. 1985. Qualitative simulation in medical physiology: a progress report. Technical Report MIT/LCS/TM-280, MIT.

Kuipers, B.J.; Chiu, C.; Dalle Molle, D.T.; and Throop, D.R. 1991. Higher-order derivative constraints in qualitative simulation. *Artificial Intelligence* 51:343–380.

Kuipers, B.J. 1986. Qualitative simulation. *Artificial Intelligence* 29:280–338.

Kuipers, B.J. 1987. Abstraction by time-scale in qualitative simulation. In *Proc. AAAI-87*, Los Altos. Morgan Kaufmann. 621–625.

Kuipers, B.J. 1989. Qualitative reasoning with causal models in diagnosis of complex systems. In Widman, L.E.; Loparo, K.A.; and Nielsen, N.R., editors 1989, *Artificial Intelligence, Simulation and Modeling.* Wiley, New York. 257–274.

Lee, W.W. and Kuipers, B.J. 1988. Non-intersection of trajectories in qualitative phase space: a global constraint for qualitative simulation. In *Proc. AAAI-88*, Los Altos. Morgan Kaufmann. 286–290.

Low, C.M. and Iwasaki, Y. 1992. Device modelling environment: an interactive environment for modelling device behaviour. *Intelligent Systems Engineering* 1(2):115–145.

Nayak, P.; Joskowicz, L.; and Addanki, S. 1991. Automated model selection using context-dependent behaviors. In *Proc. 5th International Workshop on Qualitative Reasoning*, Austin. Austin Univ. 10–24.

Ramoni, M.; Stefanelli, M.; Magnani, L.; and Barosi, G. 1992. An epistemological framework for medical knowledge based systems. *IEEE Transactions on Systems Man and Cybernetics* 22:1361–1374.

SAAM, 1993. User's guide manual. Resource Facility for Kinetics Analysis - Univ. Washington.

Shen, Q. and Leitch, R.R. 1991. Combining qualitative simulation and fuzzy sets. In Faltings, and Struss, P., editors 1991, *Recent Advances in Qualitative Physics.* MIT Press. 83–100.

Struss, P. 1988. Global filters for qualitative behaviors. In *Proc. AAAI-88*, Los Altos. Morgan Kaufmann. 275–279.

Weld, D.S. and De Kleer, J.(eds.) 1990. *Readings in Qualitative Reasoning About Physical Systems.* Morgan Kaufmann, Los Altos.

Weld, D.S. 1990. Approximation reformulations. In *Proc. AAAI-90*, Los Altos. Morgan Kaufmann. 407–412.