A Semi-Quantitative Physics Compiler

Adam Farquhar
Knowledge Systems Laboratory
Stanford University
Palo Alto, CA, USA
Ph: +001 415-723-9770
Fax: +001 415-725-5850
Adam_Farquhar@ksl.stanford.edu

Giorgio Brajnik[1]
Dipartimento di Matematica e Informatica
Università di Udine
Udine – Italy
Ph: +39 (432) 272210
Fax: +39 (432) 510755
giorgio@udmi5400.cineca.it

Content Areas:
Qualitative Reasoning, Simulation, Control

A Semi-Quantitative Physics Compiler

Abstract

Incomplete information is present in many engineering domains, hindering traditional and non–traditional simulation techniques. This paper describes SQPC (semi–quantitative physics compiler), an implemented approach to modelling and simulation that can predict the behavior of incompletely specified systems, such as those that arise in the water control domain. SQPC is the first system that unifies compositional modeling techniques with semi–quantitative representations. We describe SQPC's foundations, QSIM and QPC, and how its extends them. We demonstrate SQPC using an example from the water supply domain.

Content Areas:

Qualitative Reasoning, Simulation, Control

1 Introduction

Consider the problem of water supply control. A lake has a dam with floodgates that can be opened or closed to regulate the water flow through power generating turbines, the water level (stage) of the lake, and the downstream flow. The goal of a controller is to provide adequate reservoir capacity for power generation, consumption, industrial use, and recreation, as well as downstream flow. In exceptional circumstances, the controller must also work to minimize or avoid flooding both above and below the dam. This task is both difficult and vitally important to the residents of surrounding areas. The work of controllers could be substantially eased by sound automatic modeling and simulation tools.

There are several forms of incomplete information that appear in this domain. The precise shape and capacity of lakes or reservoirs is rarely known; the outflow from opening a dam's floodgates is only crudely measured; empirical data on the level/flow–rate curve for rivers becomes less and less accurate when flood conditions approach; few quantities are measured (e.g. flow rates of minor tributaries are not measured at all); the amount of runoff to be expected from a given rainfall depends on difficult to measure surface characteristics such as saturation; the amount of rainfall that actually falls on a lake and surrounding areas is difficult to predict and is imprecisely measured. Nonetheless, both mathematical analysis and observations do provide rough bounds on the quantities involved. Often, rough accurate bounds suffice to select appropriate actions.

This domain, however, is challenging for existing approaches to modeling and simulation. Pure qualitative reasoning techniques [Forbus, 1984; Kuipers, 1986] do not exploit the partial information available and consequently provide insufficiently strong predictions. Traditional numeric methods require much more precise information than is available, forcing modelers to make assumptions which may invalidate results and which may be difficult to evaluate.

Fortunately, recent advances in *semi-quantitative* simulation techniques provide a method for predicting the behavior of such systems. This work extends the purely qualitative representation [Kuipers, 1986] with means for representing semi–quantitative information [Berleant and Kuipers, 1988; Kuipers and Berleant, 1992; Kay and Kuipers, 1993; Vescovi *et al.*, 1994]. In this work, semi-quantitative information is represented in two forms: bounds on variable values and functional bounds (envelopes) on otherwise unspecified monotonic functions. This is exactly the kind of information that is available in the water supply and many other engineering domains.

Several systems [Forbus and Falkenhainer, 1990; Iwasaki and Low, 1991; Amador *et al.*, 1993] have been developed that use compositional modelling techniques, and exploit qualitative models to provide explanations of simulations. They are unable, however, to represent or use semi-quantitative information. In order to provide any sort of numeric simulation, they all require complete precise initial conditions and algebraic equations.

This paper describes SQPC (semi–quantitative physics compiler), an implemented approach to modelling and simulation that uses semi-quantitative knowledge. SQPC extends the QPC system [Farquhar, 1993; Crawford *et al.*, 1990] to exploit the recent advances in semi-quantitative simulation. SQPC is the first compositional modeling system to employ semi–quantitative representation and simulation.

The input to SQPC is a *domain theory* and a *scenario.* The domain theory is composed primarily of *model fragment* definitions the describe both the conditions under which physical phenomenon are active and their consequences. The scenario specifies objects that are known to be of interest, some initial conditions, and some relations that hold throughout the scenario. Both the domain theory and scenario may include bounds on numeric values and monotonic functions. From this, SQPC generates a set of behavioral descriptions, guaranteed to cover any system trajectory consistent with the scenario and domain theory. A behavior may pass through a number of distinct operating regions, each of which is characterized by a distinct mathematical model, or SQDE (semi–quantitative differential equation).

2 SQPC foundations

2.1 Semi–quantitative simulation

SQPC is built on top of the QSIM qualitative simulator [Kuipers, 1986; Kuipers, 1993]. The input to QSIM is a *qualitative differential equation* (QDE) which specifies: (i) a set of variables (continuously differentiable functions of *time*); (ii) a *quantity space* for each of these variables, specified in terms of a totally ordered set of symbolic *landmark* values; (iii) a set of constraints expressing (algebraic, differential or monotonic) relationships between variables. A QDE is an abstract description of a, perhaps infinite, set of ordinary differential equations. The output of QSIM is a set of behaviors. Each behavior is a sequence of states, where a state is a mapping of variables to qualitative values. A *qualitative value* represents the magnitude of the variable, which is either equal to a landmark or in the open interval specified by adjacent landmarks, and its direction of change (the sign of its time derivative: *dec, std, inc*). Each state describes either a time point or an open temporal interval.

In the semi-quantitative framework employed by SQPC, the basic qualitative representation is augmented [Berleant and Kuipers, 1988; Kuipers and Berleant, 1992; Kay and Kuipers, 1993]. Each landmark may be bounded with a precise numeric upper and lower bound. Each monotonic function constraint may be bounded with a precise functional upper and lower bound.

A monotonic function constraint represents an element of an infinite set of real valued functions. The monotonic function constraint has the general form:

$$((M\ s_1 \ldots s_n)\ x_1 \ldots x_n\ y) \quad \equiv \quad \begin{cases} y = f(x_1, \ldots, x_n) \\ \mathrm{sign}(\frac{\partial f}{\partial xi}) = s_i \end{cases} \tag{1}$$

where each s_i is a sign and f is a continuous *reasonable* function [Kuipers, 1986]. An envelope for a constraint is a pair of functions $F = <\underline{f}, \overline{f}>$ that bound the actual function f.

The semi-quantitative simulators augment behavior with the numeric bounds. They are also able to use the semi-quantitative information to rule out qualitatively possible behaviors. The semi-quantitative techniques developed in [Berleant and Kuipers, 1988] propagate bounds throughout each time-point state and then used the mean-value theorem to constrain the values across time. The dynamic envelope technique [Kay and Kuipers, 1993] constructs extremal equations for the derivative of each state variable. These extremal equations are then explicitly integrated to provide bounds on variable values across time intervals. Neither technique strictly dominates the other. As a result, the bounds provided by the two methods may be intersected, yielding sometimes stronger predictions than either alone.

2.2 Modeling language

SQPC is an extension of QPC, whose modeling language descended from the expressive modeling language Forbus developed for his Qualitative Process theory [Forbus, 1984]. The input to QPC is a *domain theory* and *scenario* specified in the QPC modeling language. A domain theory consists of a set of quantified definitions, called *model fragments*, each of which describes some aspect of the domain, such as physical laws (e.g. mass conservation), processes (e.g. liquid flows), devices (e.g. pumps), and objects (e.g. containers). Each definition applies whenever there exists a set of participants for whom the stated conditions are satisfied. The specific system or situation being modeled is partially described by the scenario definition, which lists a set of objects that are of interest, some of their initial conditions, relations that hold throughout the scenario, and boundary conditions.

Influences are compositional relations between variables that are particularly convenient for asserting fragments of information that can be composed into constraints.

Three kinds of influences are supported.
An *indirect influence* such as $(Y\ Q^{+}\ X)$ means that in the absence of countervailing influences, an increase in X causes an increase in Y. Furthermore, Y is functionally determined by the set of influencing variables on (through a number of different influences). More precisely $(Y\ Q^{+}\ X)$ means that there exist f, a continuous function, and a family of quantities $\{Q_i\}_{1 \leq i \leq n}$ with $n \geq 0$ such that $Y = f(X, Q_1, \ldots, Q_n)$ and $\frac{\partial f}{\partial X} > 0$. In the case of $(Y\ Q^{-}\ X)$ then $\frac{\partial f}{\partial X} < 0$.
Algebraic influences provide additional information on the form of the function f. QPC's language offers four kinds of algebraic indirect influences, one for each basic arithmetic operation. $(Y\ Q_{add}\ X)$ means that there exists a family of quantities $\{Q_i\}_{1 \leq i \leq n}$, with $n \geq 0$ such that $Y = X + \sum Q_i$. Similarly for Q_{sub}, Q_{mult} and Q_{div}.
Finally, a *direct influences* such as $(Y\ I^{+}\ X))$ expresses the fact that if there are no other countervailing influences, a positive X causes an increase of Y. Direct influences equivalent to algebraic influences on the derivative of the influenced variable (*i.e.*, $(Y\ I^{+}\ X) \equiv Y' = \frac{dY}{dt}$ and $(Y'\ Q_{add}\ X)$).

QPC takes as input a description of a specific situation (the *scenario*) and then: (i) it determines, for each time instant, the set of active model fragments with respect to the configuration specified in the scenario; (ii) for each set of active model fragments QPC assumes a closed-world and resolves influences into appropriate constraints by composing together all variables influencing a same variable; (iii) it defines a model in terms of a QSIM QDE; (iv) it computes an initial state and (v) it performs an attainable envisionment[2] producing a graph of QSIM qualitative behaviors. When the simulation generates events that signal the crossing of the operating region of the model, then QPC resumes the modeling activity, retaining values of all variables from the last state before the region transition.

3 SQPC extends QPC

SQPC extends both the representation that supports the QPC modeling language and the inference methods that QPC employs. This section describes these extensions.

3.1 Representation

SQPC extends the QPC modeling language by adding dimensional information, numeric bounds on magnitudes, bounding envelopes on monotonic functions, and func-

[2]That is, a simulation from a partially known initial state which does not create new landmarks.

tions specified by tables.

Dimensional information Quantity types, variables and magnitudes are partitioned into dimensions. SQPC defines the seven International System (SI) dimensions as well as a *null-dimension* is also provided to represent "pure number" quantities such as the efficiency of a turbine. A domain theory may also introduce derived dimensions specified by a list of dimension names with integer exponents (for example, the `power-dimension` is ml^2/t^3).

Representing dimensions explicitly allows SQPC to:

1. perform dimensional analysis and verify that equations and order relations are well formed. Dimensional errors are common when writing equations and can be easily detected;

2. constrain inference about order relations. It is not meaningful to compare variables that do not have the same dimension.

Numeric values SQPC represents numeric and qualitative quantities in a single framework. A numeric magnitude represents a precise, but unknown, real number that may be constrained to lie within a numeric range. Note that two magnitudes constrained by the same range are, in general, not equal. This extension is orthogonal to dimensions.

Bounding envelopes An *envelope* is defined by a form similar to that for defining model fragments. It states a set of conditions under which a specific form of monotonic function over a tuple of variables is bounded by a functional envelope. The envelope is specified by a pair of algebraic functions and perhaps their inverses.

Tabular functions Tabular functions provide an important practical extension to the modeling language. A large portion of empirically collected knowledge about time-varying systems is represented and summarized in tabular form. The SQPC language permits numeric functions (used to specify envelopes) to be defined by data in a multi-dimensional table. SQPC assumes that these tables are coarse descriptions of the continuous reasonable functions that satisfy monotonic function constraints.

3.2 Reasoning

QPC employs a hybrid architecture in which the model building portion is separated from the simulator. This allowed it to exploit semi-quantitative information without changing the overall algorithm. There are, however, several extensions that need to be made in order to accommodate the representational extensions described above.

Dimensional information allows the inferences that compute order relations between variables and magnitudes to be focused. SQPC never compares two quantities with incompatible dimensions. The compatibility test is simplified by reducing all dimension specifications to a canonical form, represented by a vector of exponents (each position in the vector corresponds to a basic dimension).

The numeric bounds on magnitudes require an orthogonal change in the computation of order relations. Except for the simple cases, in which the bounding intervals do not overlap, SQPC leaves the computation of numeric order relations to the semi-quantitative QSIM extensions. QSIM does a good job of propagating the bounds through the constraints in the SQDE. Recreating this in the SQPC knowledge base would be unnecessary, redundant, and inefficient.

SQPC needs to determine which envelopes to include in the SQDE for each model. This is non-trivial because there are several ways to describe a monotonic relationship among a set of quantities. Because each envelope that can be included is likely to strengthen the predictions, it is important to include all of the relevant ones. For instance, suppose that the model contains the constraint $(M\ (+\ -)\ X\ Y\ Z)$, and there is an envelope defined for the constraint $(M\ (+\ +)\ Y\ Z\ X)$. This second constraint should be added to the SQDE with its envelope. This is because the two constraint forms are simply permutations of each other. SQPC adds any constraint and envelope into the SQDE that is a permutation of a constraint in the SQDE.

In order to produce qualitative simulations of large systems, QPC extended QSIM to generate attainable envisionments. An attainable envisionment is a tree of behaviors with cycles, cross and back edges (i.e. a rooted directed graph). The QSIM semi-quantitative simulation methods [Berleant and Kuipers, 1988; Kuipers and Berleant, 1992; Kay and Kuipers, 1993], however, work from the more standard behavior tree. Thus, SQPC uses QSIM to generate a behavior tree. Another difference is in the automatic generation of landmarks. In order to guarantee that the attainable envisionment is finite (and thus guarantee termination), new landmarks are not automatically generated under QPC. Because the semi-quantitative representation can automatically generate new landmarks and provide them with bounds, SQPC allows automatic generation of new landmarks by the simulator.

4 An Example

We demonstrate SQPC on a problem from the domain of water supply control. We consider a portion of the system of lakes and rivers to be found in the scenic hill country surrounding Austin, Texas. The Colorado river flows into Lake Travis. The Mansfield Dam on Lake Travis produces hydroelectric power, and controls the level of the lake and the flow into the downstream leg of the Colorado.

The problem is to evaluate a "what if" scenario (figure 1). We are given an initial level for Lake Travis (a typical value between 690.2 and 690.3 feet) and a rough projected inflow from the Colorado river (between 791 and 950 cft/s). The task is to determine what happens to the lake level and evaluate how long the hydroelectric plant can deliver power at the requested rate of 10Mw.

In addition to the numeric bounds variables in the scenario, there are several other sources of semi-quantitative information in this problem. All of the turbines in the dam are similar. An envelope definition (see figure 4) establishes bounds on the relation between the *head* of water above a turbine, the desired *power* output, and the *discharge rate* of water downstream. This envelope definition applies whenever the conditions and constraint portions of the envelope form are satisfied. Lake Travis is a unique object. The relation between the *stage* (level) and *capacity* (volume) for Lake Travis is provided by an envelope specified in the scenario definition. All of the semi-quantitative information in this domain theory is specified in the form of tables.[3] The tables reflect both observations and engineering estimates about the relationships between important variables.

Solving this problem is made slightly more complex because of the behavior of the turbines. The turbines are controlled by a servo-mechanism designed to generated the desired amount of power regardless of the hydraulic pressure, which is determined by the *head* at the turbine. This is possible as long as there is sufficient *head*. When the *head* drops below the minimum threshold for a given power output, then less power is released. The domain theory captures this accurately (see figure 3). The domain theory also includes model fragments for conservation laws (e.g. of mass and energy), basic hydraulic principles (e.g. flow is proportional to head), and instantiations of these basic laws for this domain.

Figure 2 shows the SQPC output for this scenario. Under the specified conditions, the desired power level can be maintained until time `t1`, at least 45 days ($3.94 * 10^6$ seconds) after the start time. After `t1`, there will be insufficient hydraulic pressure to

[3]The Lower Colorado River Authority (LCRA) has contributed actual tables of empirical data to the Qualitative Reasoning Group of the University of Texas for evaluation.

```
(defScenario Travis-1-turbine
    "Turbine from controlled to uncontrolled regime."
  :entities  ...
  :landmarks ((top-of-dam :variables ((stage Travis)) :range 714))
  :relations ((= (flow-rate Colorado-up) (791 950))  ; cft/s
              (= (top Mansfield) top-of-dam)
              (is-open turb))
  :initial-conditions ((= (power turb) 10)                     ; Mw
                       (= (stage Travis) (690.2 690.3)))       ; ft
  :envelopes ((stage-capacity
               :constraint (M+ (stage Travis) (capacity Travis))
               :upper-envelope (hi (lake-travis-stage-capacity
                                    '(st cap) (list (stage Travis))))
               :lower-envelope (lo (lake-travis-stage-capacity
                                    '(st cap) (list (stage Travis))))
               :upper-inverse (lo (lake-travis-stage-capacity
                                   '(cap st) (list (capacity Travis))))
               :lower-inverse (hi (lake-travis-stage-capacity
                                   '(cap st) (list (capacity Travis)))))))
```

Figure 1: Part of the scenario description. The initial conditions specify the desired power output of the turbine and the stage (level) of Lake Travis. The flow-rate of the upstream leg of the Colorado is set for the duration of the scenario. The `:envelopes` clause specifies the state-capacity table for Lake Travis. Any model in the scenario which includes the $M+$ between the stage and capacity of lake Travis will include this envelope.

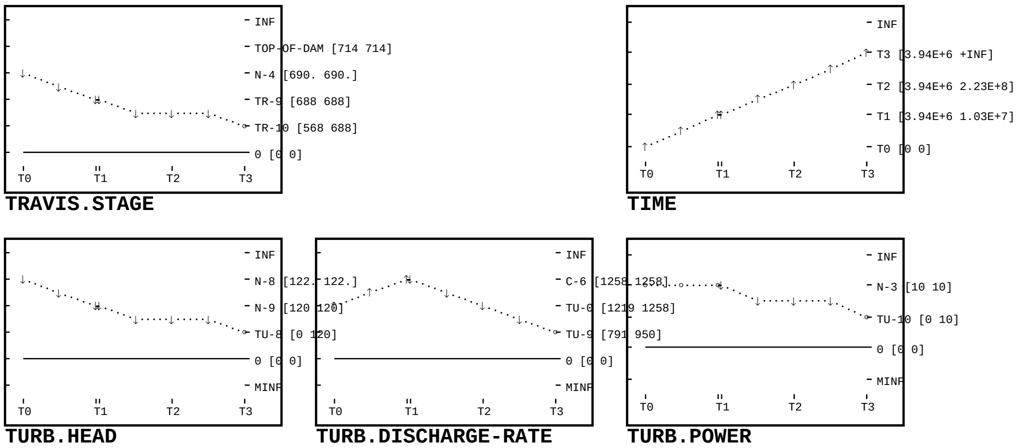

Figure 2: Behavior plot for several variables in the scenario. The power output by the turbine after `t1` is below the desired level.

```
(defmodelfragment NORMAL-TURBINE-MF
    :participants ((t :type turbines)
                   (dam :type dams :conditions ((has-valve dam t)
                                                (is-open t)))
                   (lake :type lakes :conditions ((connects dam lake river)))
                   (river :type rivers))
    :operating-conditions ((> (stage lake) (base t))
                           (<= (min-head t) (head t))
                           (<= (head t) (max-head t)))
    :consequences ((I- (capacity lake) (discharge-rate t))
                   (Q- (discharge-rate t) (head t))
                   (Q+0 (discharge-rate t) (power t))
                   (Q-add (flow-rate river) (discharge-rate t))
                   (Q+0 (min-head t)(power t))
                   (Q+0 (max-head t)(power t))))
```

Figure 3: A model fragment definition describes normally operating turbines. An instance exists if there is an open turbine on a dam. It is active when the head above the turbine is between the variables `min-head` and `max-head` and the lake level is above the base of the turbine. The consequences specify equations that hold when an instance is active. The qualitative influences `Q-` and `Q+0` partially specify monotonic functions on the `discharge-rate` of the turbine (which is causally influenced by both `head` and `power`). The `Q-add` specifies that the `discharge-rate` sums into the river's `flow-rate`. The final influences specify that both `min-head` and `max-head` are affected by the desired `power` output.

```
(defEnvelope MANSFIELD-TURBINE-DR-E
      :comment "HEAD x POWER --> DISCHARGE-RATE of a Mansfield turbine."
      :participants  ((dam :type dams)
                      (lake :type lakes
                            :conditions ((connects dam lake river)))
                      (river :type rivers)
                      (t :type Mansfield-turbines
                            :conditions ((has-valve dam t))))
      :constraint ((M - +) (head t) (power t) (discharge-rate t))
      :upper-envelope
        (hi (lake-travis-turbine-discharge-rate
                  '(hd pw dr) (list (head t) (power t))))
      :lower-envelope
        (lo (lake-travis-turbine-discharge-rate
                  '(hd pw dr) (list (head t) (power t)))))
```

Figure 4: An envelope that is applicable to any turbine of the kind found on Mansfield dam. It bounds the function that relates the head above the turbine, the desired power output, and the discharge rate. The bounds are provided by a table.

provide the full power output, the discharge rate from the turbine will decrease until it reaches equilibrium with the inflow at a rate between 791 and 950cf/s, and the lake level will stabilize between 568' and 688'. These predictions are strong enough to be useful to a system controller. Note how imprecise the problem description really is: the flow rate was very coarse; there are no semi-quantitative bounds for the relationship between *power* and *head* in the low-head situation after `t1`; the table relating *stage* and *capacity* becomes extremely weak below 600'. More precise information in the domain theory or scenario will result in more precise predictions. This is the strength of the semi-quantitative inference methods. The precision of the predictions is proportional to the precision of the model.

This is illustrated by running the same scenario under differing conditions. If the upper bound on the inflow rate is reduced from 950cf/s to 800cf/s, then the upper bound on the `t1`, the time that power generation drops below the desired rate, is reduced to 76days, a 58% improvement. The envelopes can be tightened by using a linear interpolation for the stage-capacity curve instead of a step function. This tightens the range for `t1` to 50-58days, an improvement of 89% from the original. Increased precision in the input or model leads to increased precision in the output.

4.1 Implementation status

SQPC is fully implemented in Lucid Common Lisp as an extension to QPC, which in turn uses the Algernon knowledge representation system [Crawford, 1991] and QSIM. We are currently experimenting with SQPC in the water flow control domain; SQPC has been run on over a dozen examples comparable to the one shown in this paper. The runtime for this example is around 4 minutes on Sun Sparc4/75. The bulk of this time is spent computing order relations with interpreted rules. Using standard rule compilation techniques or a special purpose inequality reasoner will result in a substantial (orders of magnitude) speedup.

5 Related work

In recent years, several research efforts have worked towards the development of self–explanatory simulators that construct numerical simulations and use the qualitative representation to help explain the results. Unlike SQPC, they do not use semi-quantitative information. Their predictions are either precise numeric ones, or purely qualitative.

SIMGEN [Forbus and Falkenhainer, 1990] computes a total envisionment of the scenario and then, for each envisionment state[4] it builds a numerical simulator, monitors the simulation and, at the end of the analysis, interprets numerical results in terms of the envisionment graph. Furthermore, SIMGEN requires precise and complete numerical equations, initial and boundary conditions for the simulation. SIMGEN must be capable of building a numerical model for each envisionment state touched during the simulation; to this end it is supplied with a library of numeric procedures for every possible combination of influences.

DME (the Device Modeling Environment) [Iwasaki and Low, 1991] is an incremental compositional modeling system capable of generating self-explanatory simulations. DME can work in two exclusive modes: qualitative or numeric. In the former case DME constructs qualitative states, and uses QSIM to generate successors; in the latter case, DME builds numerical models for simulation. In both modes, crossing an operating region triggers remodelling. DME is highly interactive and provides sophisticated explanation capabilities [Gautier and Gruber, 1993; Gruber and Gautier, 1993]. DME requires precise numerical equations, initial and boundary conditions.

Pika [Amador *et al.*, 1993] builds a numerical model for each operating region of the system as soon as this is needed. Pika monitors the numerical simulation and, at the end of the analysis, is capable of engaging in a simple question/answering dialogue. Pika requires precise equations, complete initial conditions (unlike the other systems), and complete specification of boundary conditions (in particular inequalities are not allowed).

6 Conclusion

We have presented SQPC, the first system to unify compositional modeling techniques with semi-quantitative simulation. SQPC automatically constructs semi-quantitative models and produces useful predictions with imprecise knowledge. We have shown how QPC was extended to accomplish this. We argued that semi-quantitative knowledge is crucial to many applied engineering domains the one chosen for demonstrating SQPC, water supply control.

The current version of SQPC relies primarily on the Q2 static envelope [Berleant and Kuipers, 1988] method which is based on the mean-value theorem. The NSIM [Kay and Kuipers, 1993] dynamic envelope method, which performs explicit numeric integration of extremal bounding equations, often provides tighter bounds. Reliance on Q2 is due partly to software development schedules and partly to the extensive use

[4] corresponding to an operating region.

of tabular information in the water supply control domain. Because NSIM performs explicit numeric integration, it requires smooth continuous functional bounds. This requires an extension to our current tabular data tools to generate smooth monotonic functional bounds for the tables. We are currently developing this extension.

The ability to use semi-quantitative information in a compositional modeler is tremendously exciting. We look forward to extending existing domain theories, such as the chemical engineering theory constructed by Catino [Catino, 1993] to include semi-quantitative information and exploring the construction of substantial engineering quality models that are tractable due to greater available precision.

Coupled with compositional modeling, the semi-quantitative techniques have the promise of achieving one of the major goals of qualitative reasoning: to make predictions as strong as possible, given whatever information is available.

References

[Amador *et al.*, 1993] Amador, F.; Finkelstein, A.; and Weld, D. 1993. Real–time self–explanatory simulation. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

[Berleant and Kuipers, 1988] Berleant, D. and Kuipers, B. 1988. Using incomplete quantitative knowledge in qualitative reasoning. In *Proc. of the Sixth National Conference on Artificial Intelligence*. 324–329.

[Catino, 1993] Catino, Cathy 1993. *Automated Modeling of Chemical Plants with Application to Hazard and Operability Studies*. Ph.D. Dissertation, Department of Chemical Engeneering, University of Pennsylvania, Philadelphia, PA.

[Crawford *et al.*, 1990] Crawford, J.; Kuipers, B.; and Farquhar, A. 1990. Qpc: a compiler from physical models into qualitative differential equations. In *Proc. of the Eight National Conference on Artificial Intelligence*. AAAI Press / The MIT Press.

[Crawford, 1991] Crawford, J. 1991. Algernon — a tractable system for knowledge representation. *SIGART Bulletin* 2(3).

[Farquhar, 1993] Farquhar, Adam 1993. *Automated Modeling of Physical Systems in the Presence of Incomplete Knowledge*. Ph.D. Dissertation, Department of Computer Sciences, the University of Texas at Austin. Available as technical report UT-AI-93-207.

[Forbus and Falkenhainer, 1990] Forbus, K. and Falkenhainer, B. 1990. Self–explanatory simulations: an integration of qualitative and quantitative knowledge. In *Proc. of the Eight National Conference on Artificial Intelligence*. AAAI Press / The MIT Press. 380–387.

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

[Gautier and Gruber, 1993] Gautier, P. and Gruber, T. 1993. Generating explanations of device behavior using compositional modeling and causal ordering. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

[Gruber and Gautier, 1993] Gruber, T. and Gautier, P. 1993. Machine–generated explanations of engineering models: a compositional modeling approach. In *Proc. International Joint Conference on Artificial Intelligence*, Chambery, F.

[Iwasaki and Low, 1991] Iwasaki, Y. and Low, C. M. 1991. Model generation and simulation of device behavior with continuous and discrete changes. Technical Report KSL 91–69, Knowledge Systems Laboratory — Stanford University.

[Kay and Kuipers, 1993] Kay, H. and Kuipers, B. 1993. Numerical behavior envelopes for qualitative models. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

[Kuipers and Berleant, 1992] Kuipers, Benjamin and Berleant, Daniel 1992. Combined qualitative and numerical simulation with Q3. In Faltings, B. and Struss, P., editors 1992, *Recent advances in qualitative physics*. MIT Press. 3–16.

[Kuipers, 1986] Kuipers, B. 1986. Qualitative simulation. *Artificial Intelligence* 29:289–338.

[Kuipers, 1993] Kuipers, B. 1993. Reasoning with qualitative models. *Artificial Intelligence* 59:125–132.

[Vescovi *et al.*, 1994] Vescovi, M.; Farquhar, A.; and Iwasaki, Y. 1994. Numerical interval simulation: Bounding behaviors of non-monotonic systems. Submitted to QP94.