The Hybrid Phenomena Theory*

Erling A- Woods
SINTEF Automatic Control
The Norwegian Institute of Technology
Division of Engineering Cybernetics
N 7034 TRONDHEIM
NORWAY

Abstract

The Hybrid Phenomena Theory (HPT) is a framework for formalizing how dynamic state space models of physical systems are built from first principles. The HPT descends from the Qualitative Process Theory (QPT), [Forb us, 1984], from which it inherits basic concepts like views, phenomena and influences. However, the HPT redefines some of these concepts in a more strict manner in order to represent knowledge of physics with the accuracy needed to develop full parametric models. Specifically, influences may specify quantified non-linear functions of several variables. A mechanism denoted subsumption is introduced to ensure consistency in the emerging models when different simplifying assumptions are made. The HPT has been implemented in CLOS.

1 Background and Motivation

Qualitative reasoning, [Bobrow, 1984; Weld and de Kleer, 1990], has provided valuable insight and methods for analyzing physical systems. Yet, several fundamental problems have been identified. The inherent ambiguity described by Kuipers [1986] poses a problem in some potential areas of application. The problems with the qualitative mathematics pointed out by Struss [1990] imposes limitations on what can be accomplished using any form of qualitative simulation.

It is something of a paradox that while Hayes [1985] argued that it was time for AI to move away from the toy problems commonly addressed by AI research at that time and tackle a real world problem, he simultaneously argued for a purely qualitative approach. But a large class of problems involving the physical world depends on quantitative knowledge. By choosing a purely qualitative approach, many researchers have disqualified their methods from a vast number of potential application areas in science and engineering.

For all the merits of qualitative simulation, it is the authors view that the greatest accomplishment of the qualitative reasoning community is the development of methodologies enabling reasoning about how structural relationships between objects comprising a physical system impose constraints between the parameters and variables describing that system. This reasoning captures essential parts of the reasoning performed by physicists and engineers when building mathematical models of physical systems. However, a predominant fraction of those models incorporate parameters and variables intended to have a quantitative interpretation.

The Hybrid Phenomena Theory (HPT) was developed with monitoring and diagnosis applications in the process industry in mind. The HPT builds on the framework of the QPT [Forbus, 1984; Forbus, 1990]. But contrary to the QPT, the HPT derives parametric state space models. The parameters may have a quantitative interpretation. The mathematical models derived provides a basis for quantitative simulation, filtering and estimation schemes developed within such fields as control and chemical engineering. This allows us to utilize quantitative information when available, thus avoiding the problems associated with qualitative simulation.

De Kleer [1990] laments that physicists have a hard time understanding the goals of qualitative reasoning. The physicists typically argue that they know all that the qualitative reasoning approaches do. And since the methods developed in qualitative reasoning fails to produce the quantitative models the physicists normally use, physicists understandably see little merit in the enterprise. By deriving parametric models commonly used for a number of scientific and practical purposes, the HPT may help convince people outside the AI community about the utility in the qualitative reasoning methods. Perhaps we could then initiate a coordinated effort, towards developing a "non-naive physics".

2 Relationship with the QPT

The QPT can be considered to consist of three components as shown in Figure 1. The purely declarative first component provides the framework, or vocabulary, used to represent a description of the physical systems to be analyzed and knowledge on how different physical phenomena affect relationships between objects and variables. The second component combines the description of an actual system, given in the vocabulary in the

*This research was supported by the Royal Norwegian Council for Scientific and Industrial Research under doctoral scholarship grant no ST.10.12.220290.

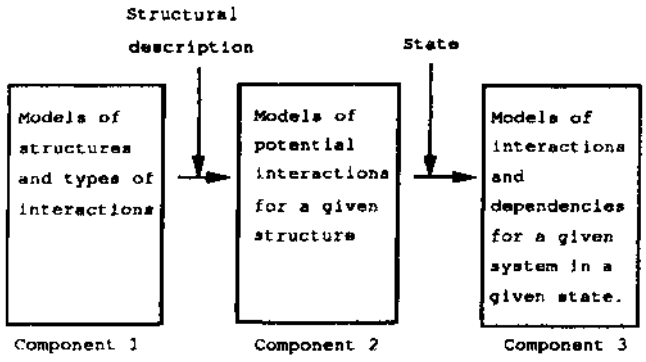

Figure 1: Components in the QPT and HPT.

first component, with information on the current state of the system. It produces a set of constraints describing the system in its current state. The procedural third component uses the constraints to derive possible successor states. For each successor state, the procedure of component two is reactivated to produce a new set of constraints. The QPT then iterates until quiescence occurs or the new state equals a previously encountered state. The final result is an envisionment, a description of all possible sequences of qualitative states which the physical system might evolve through.

The HPT comprises the same three components. The concepts included in the first component of the QPT are also found in the HPT, but some concepts have been renamed and others reinterpreted to allow for the more accurate description of knowledge necessary to derive state space models rather than qualitative constraints. The concepts of instantiation and activity are retained and the procedures employed in the second component of the HPT are similar to those of the QPT. With the HPT, numerical simulation normally replaces envisionment in the third component.

The term *phenomenon* in the HPT replaces the term "process" in the QPT. "Process" is commonly used in a number of contexts. It normally denotes a series of actions or changes invoked to make or transform something. A heat exchanging process is commonly interpreted to include a number of physical phenomena such as fluid flows, heat flows and possible condensation and evaporation. It is the notions of such individual phenomena which must be formalized in the HPT. The word phenomenon is more descriptive in the current context and therefore less apt to create confusion.

Similarly, the use of the terms *direct* and *indirect influences* as employed in the QPT are counter-intuitive. A direct influence describes how the value of one quantity influences the *derivative* of another. This means that a change in the influencing quantity only indirectly affects the value of the influenced quantity. An indirect influence on the other hand describes how the value of one quantity directly relates to the value of another quantity. The notion of influences is retained in the HPT, but the names, syntax and semantics differ from those employed by the QPT.

3 Influences in the HPT

Influences express how the value of a given variable is affected by one or a set of other variables. There are two types of influences, *dynamic influences* and *algebraic influences.* Dynamic influences correspond to the direct influences of the QPT and specify how a term affecting the derivative of the influenced variable is affected by a set of influencing variables. The syntax for a dynamic influence is as follows:

```
(dyn-inf <influenced-variable>
         <list of influencing variables>
         <function-spec>)
```

An example where a variable *X*\ is dynamically influenced by the two variables x_2 and x3 is shown in (1). If this is the only influence affecting x1, the derivative of *x*\ will be computed from (2).

$$(dyn-inf\ x_1\ (x_2 x_3)\ (sqrt(x_2 x_3))) \quad \text{(i)}$$

$$\dot{x}_1 = \sqrt{x_2 x_3} \quad (2)$$

Algebraic influences corresponds to the indirect influences of the QPT, for algebraic influences it is the actual value of the influenced variable which is affected by the influencing variables. The syntax is as follows.

```
(alg-inf <influenced-variable>
         <list of influencing variables>
         <function-spec>)
```

Consider a container with a closed top. The container is partially filled with liquid and a gas is taking up the remaining volume. The liquid has a level h, the gas a pressure *p1.* The pressure at the bottom of the container, p2, are affected by two different influences given in (3) and (4). If no other influence specifies p2 as the influenced variable, P2 will be given by (5).

$$(alg-inf\ p_2\ (p_1)\ p_1) \quad (3)$$

$$(alg-inf\ p_2\ (h)\ (\rho g h)) \quad (4)$$

$$p_2 = p_1 + \rho g h \quad (5)$$

Note that a given variable influencing a another variable may do so in more than one influence. This accounts for the possibility of a variable being involved in several terms in an equation simultaneously.

Any influenced variable, whether directly or indirectly influenced, is computed as the sum of all the influences affecting it. There is an historical evolution here. The notion of a process was first introduced by Hendrix [1975]. But as is pointed out in the overview and motivation of chapter 1 in [Weld and de Kleer, 1990], Hendrix approach suffered from poor modularity because variables are calculated inside a process, different processes are not allowed to influence a given variable. The QPT is modular with respect to dynamic variables, it computes the value of the derivative of all directly influenced variables as the sum of all influences affecting it. However, the QPT is not fully modular for non-dynamic variables. Although several views and processes may specify indirect influences affecting a given variable, the QPT fails to specify a general mechanism for deriving the value of an indirectly influenced variable whenever two or more influences are pushing it in different directions. The HPT is fully modular in both kind of variables.

4 Different types of Quantities

The HPT distinguish between three types of quantities; constants, parameters and variables. Constants are universal and invariable, Avagadros number, and the specific heat capacity of water at atmospheric pressure are two examples.

Parameters are quantities which are associated with a specific object. The value of a parameter may change with time, but the value can always be computed from a prescribed function of other parameters and variables associated with that very object. The heat capacity of an amount of water found in a specific container is an example. This parameter may, somewhat simplified, be described as a function of the mass of the water, no variables associated with other objects will affect it.

Variables are quantities whose value may depend on other variables as expressed in algebraic and/or differential equations. Influences are specified between variables only. Parameters and constants, which are within the scope of the definition which the influence belongs to, may be used in the function specification! part of the influence to quantify the effect of the influence. Parameters and constants are never influenced.

5 Views and Phenomena

Definitions of individual views and phenomena describe *types* of interactions in the physical world. The definitions prescribe a set of conditions which must be satisfied for the definition to be relevant in a specific situation. They also specify consequences when an instance of the definition is actively participating in a situation. Examples of definitions of views and phenomena in the HPT are shown in Figure 2. The purpose of the different parts of the definitions are broadly consistent with the corresponding parts in the QPT, a brief explanation follows below.

The individuals are listed in the first line of the definition, *src, dst* and *hbr* are the individuals in the definition of *heat-flow* in Figure 2. The individual conditions govern instantiation of views and phenomena definitions. Once a set of objects which fulfill the conditions specified in the individuals field have been identified, an instance of the definition is created. The objects fulfilling the conditions are bound to the corresponding individuals. Such an instance represent a *potential* interaction between the specific objects bound to individuals in that instance.

For the view *container-with-liquid* in Figure 2, it does not suffice that an object A is a *container-with-open-top* and that the object B is a *liquid.* The liquid B must in addition be situated in the container A. When all these conditions are satisfied, an instance of *contatner-with-liquid* where A is bound to *c* and B is bound to / is created. In general, a system may consist of many tanks with liquids. Each set of objects which satisfy all of the conditions for the individuals in a definition gives rise to an instance of that definition. Each definition is thus applicable to many different objects.

Also note that instances of views and phenomena are themselves objects and thus candidates for binding to individuals in other definitions of views or phenomena.

```
(defview container-with-liquid (c 1)
  (inst-name view-inst-with-heat-capacity)
  (individuals
    (c (is-a container-with-open-top)
       (can-contain-liquid))
    (1 (is-a liquid))
    (placed-in 1 c))
  (quantityconditions (> mass(l) 0))
  (relations
    (define-variable temp
       (rvalue 0 :unit "K"))
    (define-parameter hcp
       (:what? "heat capacity" runit "??"
        :compfun
         (+ hcp(c) (* shcp(l) mass(l)))))
    (connect-quantity area area (c))
    (if (rests-on(c x))
        then (rests-on(self x)))
    (il (has-heat-leading-bottom(c))
        then (has-heat-leading-bottom(self)))
    (alg-infl h (mass(l))
       (/ mass(l) (* density(l) area(c))))
    (alg-infl p (h)
       (* h density(1) g(global)))))

(defview heat-bridge (ob1 ob2)
  (individuals
    (obi (is-a object-with-heat-capacity))
    (ob2 (is-a object-with-heat-capacity))
    (heat-connected ob1 ob2))
  (relations
    (define-parameter alfa
      (rvalue 120 :what? "heat transfer coeff"
       runit "J s-1 K-l m-2"))
    (define-parameter salf
      (:what? "heat transfer pr degree Kelvin"
       runit "J s-1 K-l"
       r compfun
        (* alfa (MIN area(obl) area(ob2)))))))

(defphenomenon heat-flow (src dst hbr)
  (individuals
    (hbr (instance-of heat-bridge(src dst)))
    (src (is-a object-with-heat-capacity))
    (dst (is-a object-with-heat-capacity)))
  (quantityconditions
    (> temp(src) temp(dst)))
  (relations
    (define-variable hstr
      (rwhat? "heatflow from src to dst"
       runit "J s-1"))
    (alg-infl hstr (temp(src) temp(dst))
       (* salf(hbr) (- temp(src) temp(dst)))))
  (dynamics
    (dyn-infl temp(dst)
       (hstr)(/ hstr hcp(dst)))
    (dyn-infl temp(src)
       (hstr)(/ hstr hcp(src)))))
```

Figure 2: Definitions of views and phenomena.

To satisfy the conditions for the individual *hbr* in the definition of *heat-flow* in Figure 2, the object must be an instance of the view definition *heat-brtdge.*

Preconditions describe conditions which may depend on external actions. Turning on or off the power supply for a hot-plate is an example. No heat will be generated in the hot-plate unless the power is turned on, but the power does not come on or off depending on the state in the system. At least not until we introduce a feedback loop and starts to control the power. It is beyond the scope of any computer program to deduce when a human will turn the power on or off.

Quantityconditions concern the values of variables and thus depend on the state in the system. A quantity-condition either specifies absolute limits on one variable, or limits relative to two variables. An example of the first is that there is no acceleration unless the sum of the forces differs from zero. An example of the second is that there is no direct heat flow from object A to object B unless the temperature of object A is greater than that of B. The truth value for a quantity condition may change in the course of a simulation or operation of a process plant, but it is always derivable by a computer program having access to the values of the variables.

Preconditions and quantityconditions together form both the necessary and sufficient conditions for an instance of a view or phenomenon to become active. Before introducing the fields describing the effects of instantiation and activity, the HPT concept *scope* must be explained. A variable, parameter or constant is said to be within the scope of an object if it is defined by that object, or if it is within the scope of an object bound to one of the individuals in the object. In addition, there exists a set of global constants which are inside the scope of all objects.

The relations field specifies properties to be associated with each instance of the definition as well as consequences of activity. A statement in the relations field performs one of the following tasks: defines a new variable or parameter to be associated with each instance of the definition, establishes a logical relation involving the instance, specifies creation of a new object which comes into being as a consequence of the instance becoming active, or defines an algebraic influence between variables within the scope of the instance.

All fields mentioned thus far are common for both views and phenomena definitions. In addition, phenomena incorporate a dynamics field which specifies dynamic influences between variables within the scope of the phenomenon instance. A view may thus be considered a special kind of phenomenon where no dynamics are involved.

6 Assumptions and Subsumption

6.1 Assumptions are the Essence of Modeling

It is tempting to assume that it is possible to formalize a mapping from a description of a physical system to a mathematical model of that system. This would be a big mistake. There is no "correct" mathematical model of a physical system, any given model is more or less adequate for the intended application. Obtaining a, suitable model requires that the correct simplifying assumptions are made. The effects of a specific assumption propagates through the modeling procedure and will typically influence a number of later decisions. In recognition of this, the HPT incorporates a mechanism denoted *subsumption.* The purpose of the subsumption mechanism is to disable instances of views and phenomena which become invalid because of an assumption.

Consider an example with a pan containing an amount of water placed on a hotplate. We want to model the effect of the heatflow from the hotplate to the water. A common simplifying assumption would be to consider the pan and water as one object with respect to the heatflow. The definition of *container-with-liquid* in Figure 2 implements this assumption, the view defines a parameter equaling the combined heat capacity of the two objects. An Instance of this view binding the *pan* and *water* will satisfy the individual conditions in the definitions describing the heat-flow from the hotplate. But the *pan* and *water* still exist as objects, we can not and will not remove them merely because they are bound to individuals in some instance of a view or phenomenon. Consequently, the pan is going to satisfy ail the conditions for heat exchange with the hotplate and thus give rise to instantiations of the same view and phenomena definitions describing exchange of heat with the hotplate as the instance representing the combined pan and water object.

6.2 Subsumption affects one kind of assumptions

The subsumption mechanism ensures that conflicts between several instances of the *same view or phenomenon definition* do not arise. This will only happen when an assumption involving more than one object is required. The subsumption mechanism works from the principle that the most specific alternative prevails. In the example, the choice is between looking at a heat flow from the hotplate to the pan, or a heat flow from the hotplate to a view instance involving the pan. Since the view instance is considered more specific than the pan as such, the alternative involving the view instance is selected. The subsumption mechanism merely marks one of the instances as subsumed, it does not remove objects. Subsumed objects may still be bound to individuals in other instances, but they will not be tested for activity and hence can never contribute to a mathematical model. Any instance where one or more of the objects bound to the individuals are subsumed, is said to be indirectly subsumed and treated as any other subsumed object.

The subsumption mechanism is defined as follows: For any two instances *INSl* and *INS2* of a given view or phenomenon definition with individuals *D1 D2 .. Dm, INSl* subsumes *INS2* if and only if, for any individual Di (i = 1 .. m) the object bound to *Di* in *INSl* is an instance *IN SB* of some view or phenomenon definition such that the object bound to *Di* of INS2 is also bound to an individual in *IN SB,* and the object bound to each of the other individuals in *INSl* is identical to the object bound to the same individual in *INS2.*

6.3 Other kinds of Assumptions

Phenomenona may appear in different manners. As an example, think of fluid flows which may be laminar or turbulent. Which mathematical description to apply depends on the actual appearance of the flow. With the HPT, we would define at least two different phenomena, one describing a laminar flow and another one describing a turbulent flow. These definitions would incorporate different functional specifications in the influence part. They should be mutually exclusive in the sense that their pre and quanticonditions should prohibit both definitions to become active at the same time. Selecting one of the definitions will amount to making an assumption. For this kind of assumptions, the subsumption mechanism is irrelevant.

As was pointed out earlier, different assumptions are required for different applications. The present implementation of the HPT presumes that the conditions controlling the validity of an assumption is incorporated in the view or phenomenon definition. This amounts to incorporating heuristics, an approach which is necessary if we want to incorporate the HPT in systems performing automatic monitoring and diagnosis.

If the HPT is to be used as a basis for an intelligent modeling assistant, the user of the program will want to control which assumptions are made, check out the consequences of the assumptions, and possibly modify these. The present HPT implementation explicitly represent all dependencies between instances. Retracting a view or phenomenon instance automatically causes all other instances depending on it to be retracted, and removes possible subsumptions caused by the instance. By incorporating some kind of "query the user" precondition, the HPT implementation will be able to support the functions needed in an intelligent modeling assistant. Investigating diagnosis of electrical circuits, [Davies, 1984] made the observation that making assumptions was vital and that the important thing was to explicitly represent and carefully manage those assumptions. This is certainly true for building mathematical models of physical systems as well. The HPT provides a basis for doing just that.

7 An Example

The HPT has been implemented in CLOS on a TI Explorer lisp machine. The paper does not discuss the subtleties of the implementation, but to illustrate how the HPT works, some results from running the program on the physical system in Figure 3 will be presented. The views and phenomena definitions given in Figure 2 is used in the example. Two additional definitions are involved. *Electric-heat-hotplate* defines how the temperature of a hotplate is affected when the power is turned on. *Boiling* takes two individuals, a container with liquid and a heatflow and defines how the boiling phenomenon will remove heat and evaporate liquid when the temperature exceeds the boiling point of the liquid. The input description to the HPT program is shown in Figure 4. The *defobject* statement takes two required arguments, the name identifying the object and the name of the

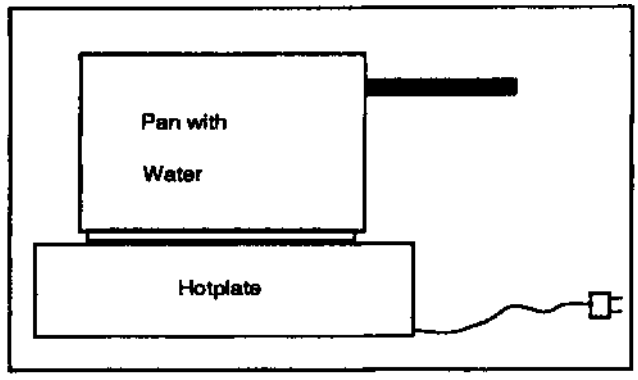

Figure 3: A physical system to be modeled.

```
(defobject pan 'container-with-open-top
   (bottom-area 0.018) ....
   (has-heat-leading-bottom))

(defobject hotplate 'electric-hotplate
   (power-consumption 1500) ....
   (has-heat-leading-top))

(defobject water 'water-liquid
   (vaporization-heat 2.4E6) ....)

(defrelation placed-in (water pan))
(defrelation rests-on  (pan hotplate))
```

Figure 4: Input description to the HPT program.

CLOS class to be instantiated to describe the object. The rest of the arguments are optional and specify initial values for variables and parameters defined for the class in question. A series of dashes indicates that not all the parameter values normally included are shown. The input also defines a set of relations between the objects.

7.1 Instances and Activity

Identifying all possible instantiations of views and phenomena is an iterative task. Whenever a new instantiation has been made, this new object may potentially satisfy the conditions posed for some individual in some definition. This may give rise to a new instantiation. Figure 5 shows the instances created by the HPT program. Instances are described by the name of the definition they arise from with an extra number added to uniquely identify each instances. This is followed by a list of the objects bound to the corresponding individuals in the definition.

Note that symmetric versions of several instances exist. This is natural since when heat can flow from object A to object B, heat will also be able to flow from object B to object A. Two such symmetric instances will never be active simultaneously, the quantityconditions will see to that. Depending on the state in the system, various combinations of instances may now be activated, and different models of the system will emerge. Space does not permit the exemplification of this.

```
Non-subsumed HPT objects:
 CONTAINER-WITH-LIQUID-1 (WATER PAN)
 ELECTRIC-HEAT-HOTPLATE-1 (HOTPLATE)
 HEAT-BRIDGE-5  (HOTPLATE
                 COHTAINER-WITH-LIQUID-1)
 HEAT-BRIDGE-6  (CONTAINER-WITH-LIQUID-1
                 HOTPLATE )
 HEAT-FLOW-5    (HEAT-BRIDGE-6
                 CONTAINER-WITH-LIQUID-1
                 HOTPLATE )
 HEAT-FLOW-6    (HEAT-BRIDGE-5 HOTPLATE
                 CONTAINER-WITH-LIQUID-1)
 BOILING-1      (HEAT-FLOW-B
                 CONTAINER-WITH-LIQUID-1 )

Subsumed objects:
 HEAT-BRIDGE-1 (PAN WATER)
 HEAT-BRIDGE-2 (WATER PAN)
 HEAT-BRIDGE-3 (HOTPLATE PAN)
 HEAT-BRIDGE-4 (PAN HOTPLATE)
 HEAT-FLOW-1    (HEAT-BRIDGE-4 PAN HOTPLATE)
 HEAT-FLOW-2    (HEAT-BRIDGE-3 HOTPLATE PAN)
 HEAT-FLOW-3    (HEAT-BRIDGE-2 WATER PAN)
 HEAT-FLOW-4    (HEAT-BRIDGE-1 PAN WATER)
```

Figure 5: Instances created by the HPT program.

7.2 Reasoning about Relations

The input description of Figure 4 specifies a relation "rests-on" between the pan and hotplate. The definition of the heat-bridge requires that the two objects to be bound to its individuals must be "heat-connected". The present HPT implementation employs a backward chaining approach from the conditions specified in a definition. The rules accounting for the instances in Figure 5 specify that:

```
(IF rest-on(A,B) THEN heat-connected(A,B))
(IF rest-on(A,B) THEN heat-connected(B,A))
```

7.3 Subsumption revisited

The example illustrates the utility of the subsumption mechanism. All views and phenomena instances which describes some aspect of the potential heat transfers between HOTPLATE and PAN as well as between PAN and WATER has been disabled by subsumption. Note that an instance may be passified in more than one way simultaneously. HEAT-BRIDGE-1 is subsumed by both HEAT-BRIDGE-5 and HEAT-BRIDGE-6 while HEAT-FLOW-1 is subsumed by HEAT-FLOW-5 and indirectly subsumed by HEAT-BRIDGE-4.

8 Future Work

An important aspect of the future development of the HPT is to define a set of relations which is consistent and suffices to express all relevant relations between physical objects in the context of building models. Redundancy should preferably be avoided as this will only complicate the reasoning which must be performed by the HPT. The solution to this problem should be common for both the QPT and the HPT.

The HPT runs into problems whenever the required numeric values for parameters or variables are missing. In this case, qualitative simulation techniques must be employed. The mathematical models produced by the HPT maps easily onto the qualitative constraint models employed by the QPT or QSIM [Kuipers, 1986], qualitative analysis techniques may therefore be integrated with the HPT. To convert to a QPT representation, we start with the instances of views and phenomena derived by the HPT. All elements of information in these instances may be considered specializations of the concepts introduced by the QPT, and the mapping onto a QPT representation is thus straight forward. Also note that since a state space model is merely a special kind of ordinary differential equation (ODE), the procedure for mapping from ODE's onto a set of QSIM constraints described by Kuipers [1986] is directly applicable.

The real challenge ahead lies in developing a set of views and phenomena definitions which are as universally applicable as possible. Hopefully, the connection to parametric models now established will help convince physicists and engineers about the importance of this task. After all, these are the people who, given the proper tools, should be best placed to formalize our knowledge about the physical world.

References

[Forbus, 1984] Kenneth D. Forbus. A Qualitative Process Theory. *Artificial Intelligence* 24, 1984.

[Bobrow, 1984] Daniel G. Bobrow, editor. *Qualitative Reasoning about Physical Systems.* North Holland, 1984.

[Weld and de Kleer, 1990] Daniel S. Weld and Johan de Kleer, editors. *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufman, 1990.

[Kuipers, 1986] Benjamin Kuipers. Qualitative Simulation. *Artificial Intelligence*'29, 1986.

[Struss, 1990] Peter Struss. Problems of Intervall-Based Qualitative Reasoning. In *[Weld and de Kleer, 1990]*

[Hayes, 1985] Patrick Hayes. The second naive physics manifesto. In *Jerry R. Hobbs and Robert C. Moore, editors, Formal Theories of the Commonsense World. Ablex Publishing Corporation, 1985*

[Forbus, 1990] Kenneth D. Forbus. The Qualitative Process Engine. In *[Weld and de Kleer, 1990]*

[de Kleer, 1990] Johan de Kleer Qualitative Physics: A Personal View. In *[Weld and de Kleer, 1990]*

[Hendrix, 1975] Gary G. Hend rix Modeling Simultaneous Actions and Continuous Proceses. *Artificial Intelligence* 4, 1973.

[Davies, 1984] Randal Davies. Diagnostic Reasoning Based on Structure and Behavior. *Artificial Intelligence* 24, 1984.