A Formal Modeling Scheme for Continuous-valued Systems: Focus on Diagnosis*

Gautam Biswas[1], Xudong Yu[1], and Ken Debelak[2]

[1]*Dept. of Computer Science*
[2]*Dept. of Chemical Engineering*
Vanderbilt University
Nashville, Tennessee 37235

Tel. (615)343-6204
e-mail: biswas, xudongy, debelak @vuse.vanderbilt.edu

July 8, 1992

Abstract

Even with significant advances in model-based diagnosis methodologies, it is recognized that effective modeling is the key to developing efficient diagnosis algorithms for complex continuous-valued systems. In this paper, we develop a formal modeling methodology based on the bond graph modeling language, and then present schemes for focusing the system model to the diagnosis task by converting equations to conflict sets. This representation greatly facilitates the *candidate generation* and the *measurement selection* processes.

1 Introduction

Diagnosis of engineering systems requires finding a component or a set of components that are the primary cause for observed discrepancies between normal (predicted) behavior and observed behavior of the system[1][5]. Model-based diagnosis researchers (e.g., [4, 8, 13]) have been successful in developing effective and efficient device-independent diagnosis algorithms that consist of two primary subtasks: (i) initial *candidate generation*, and (ii) *measurement selection* to help refine the initial candidate set. However, the availability of appropriate

*Biswas and Yu were partially supported by grants from Federal Express Corporation and Office of Naval Research (N00014-91-J-1769).

[1]An alternate approach is parameter-oriented diagnosis: correct system malfunctions by appropriate control actions (e.g., [11]).

system description models that make all the information required for diagnosis explicit is the key to the success of this methodology[5]. Most past efforts (especially *consistency-* or *logic-based* approaches to diagnosis) have focused on digital circuits[7, 13, 19]. Whereas diagnosis of complex circuits provide formidable computational challenges[8], the problem becomes even more complicated when one tries to apply these methodologies to dynamic, continuous-valued systems. The primary reason for these difficulties can be attributed to the lack of formal schemes for defining computationally tractable models that are precise enough to be useful for diagnosis[3, 5].

Reiter [19] presents a general framework that defines the consistency-based diagnosis paradigm: a system is defined as a triple (SD,COMPS,OBS), where SD, the system description, is a set of first order sentences, and COMPS, the components in the system, is a finite set of constants. Given OBS, a set of observations represented as first order sentences, we adopt the *minimal diagnosis* paradigm[7]: generate the set of minimal candidate components that are consistent with the available measurements (OBS). In this framework, the set of diagnoses can be represented by a sentence in the disjunctive normal form (DNF), where each clause is an alternate diagnosis. In general, a diagnosis procedure starts by generating the *conflicts*, a sentence that describes the diagnoses in conjunctive normal form (CNF)[14]. Techniques have been developed that convert the description from CNF to DNF (e.g., the ATMS-based approach used in GDE[7]), or from CNF to a partial DNF (a partial sets of rank-ordered candidates), for matter of efficiency[8].

In this paper, we present a formal modeling scheme that generates appropriate system descriptions of continuous-valued physical systems so that efficient component-oriented, consistency-based algorithms can be applied for diagnosing system failures. In order to achieve this, it is important that: (i) individual components and component behaviors be explicitly represented, and (ii) relations between measurable parameters and individual components should be readily derivable from the models. We accomplish this by adopting the bond graph modeling language[20] from system dynamics. Our modeling scheme first constructs a bond graph model of the system, from which equations that express the relations between measured variables and component behaviors are derived. A list of conflicts are then derived, and causal analysis is employed to adapt the *prediction-constrained tracing* methodology[12] to continuous-valued systems and generate a more precise list of candidates. Results of additional measurements can then be incorporated to refine the candidate set.

Our goal in this work is to create a system description that allows the development of efficient algorithms for candidate generation and measurement selection. Adopting this modeling framework provides a unifying framework for studying diagnosis of both continuous and discrete (digital) systems.

2 Modeling and Diagnosis

Our overall diagnosis framework involves three major steps:

(1) **Build bond-graph model of the system:** The human modeler starts with the physical schematics and a description of the overall functionality of the system, and identifies and characterizes: (i) the primary *mechanisms* that govern system behavior (functionality), (ii) a set of assumptions that characterize the physical setting of the system, and (iii) the set of *parameters* that are important for diagnostic analysis. This set includes a subset called

component parameters that directly represent the components under diagnostic scrutiny. The choice of mechanisms and parameters are based on the bond graph modeling language that we describe in detail in Section 4.

(2) **Generate equations that relate observations to components:** Using the bond graph model of the system, and a current set of observations, a set of output equations are generated. Each output equation represents the relationship between one observed parameter and components of the system. During diagnosis, new output equations are generated dynamically as additional measurements are made. The primary steps in going from the bond-graph model of the system to the system description `(SD,COMPS)`[2] as output equations can be summarized as:

- Derive state equations from the bond graph models of the system.
- Using steady state assumptions, derive output equations that relate measurable parameters to individual component parameters.

(3) **Perform Diagnosis:** Given the set of measurements made on the system (`OBS`) and the system description `(SD,COMPS)`, the diagnosis algorithm can be summarized as:

- Generate the *conflict set* by performing qualitative causal analysis on the set of output equations. This involves a number of steps that are discussed in Section 6. Note that measurement values are reported as: `above-normal`, `normal`, and `below-normal`.
- Generate the *candidate set* based on the current set of conflicts.
- Perform measurement selection based on the established relationship between measurable parameters and individual components (the set of output equations) using an information-theoretic method, such as the one used in GDE[7].

Qualitative causal analysis links individual component malfunctions expressed as directions of change in their parameter values with deviations in measurement values. To refine the candidate set generated by the diagnosis algorithm, we link deviations in abnormal measurements (i.e., above or below normal) to corresponding directions of change in the values of component parameters. By forcing consistent directions of change in component parameters across multiple measurements the candidate set is further reduced. In this paper, we focus on the modeling task; details of the candidate generation and measurement selection steps of the diagnosis algorithm are discussed elsewhere[24].

3 The Pneumatic System

Our modeling and diagnosis tasks focus on the part of the pneumatic system (see Fig. 1) that regulates air pressure and temperature drawn from one of three engines before it is delivered through the manifold system to different subsystems of the aircraft that constitute loads (e.g., the wing de-icing system).

[2]This representation was first proposed by Reiter[19].

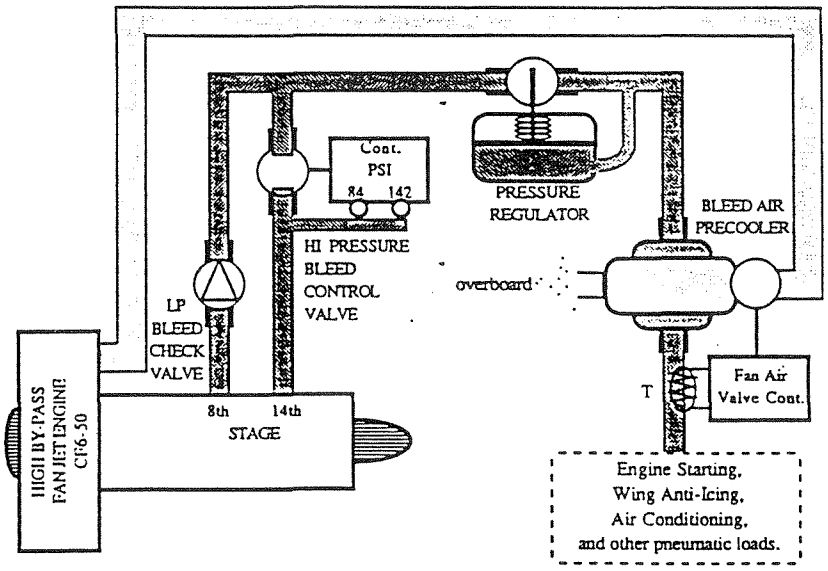

Figure 1: **The Pneumatic System**

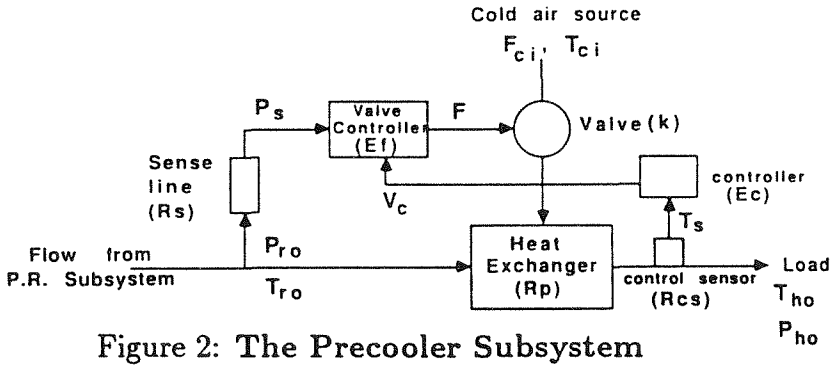

Figure 2: **The Precooler Subsystem**

The pneumatic pressure is regulated by a *pressure regulator subsystem.* The pressure regulator valve is modeled as a first-order system, where the opening of the regulating valve is determined by the changes in pressure at the regulator output. The temperature is controlled by a *precooler subsystem*, whose primary component, a heat exchanger, draws cool air from a second source to cool the bleed air from the engine. Feedback mechanisms sense the temperature at the precooler output. This information is fed back to the valve controller that fixes the opening of the valve to control the amount of cold air input to the heat exchanger, using the power obtained from the hot air transmitted through the sense line. For the diagnosis model, both the pressure regulator and precooler subsystem are modeled in more detail in terms of primitive components. For example, the precooler subsystem is modeled in terms of six primitive components (Fig. 2): (i) the heat exchanger, (ii) the feedback controller, (iii) the valve, (iv) the valve controller, (v) the temperature control sensor, and (vi) the sense line.

4 From Physical Schematics to Bond Graph Models

Effective problem solving using model-based approaches requires the ability to dynamically construct models of the system under consideration that are both parsimonious and adequate for the specific task[9, 17]. In this research our focus is on developing a formal modeling method for effective diagnosis of component failures in continuous-valued systems. The overall modeling method is more elaborately described as a four step process:

1. decompose the system into subsystems based on schematics and functionality, i.e., expected behaviors of interest,
2. construct bond graph models of the system,
3. generate equations that explicitly relate observations to individual components of the system, and
4. focus the model for diagnosis by generating conflict sets for observed measurements using causal and qualitative sign analysis on the generated equations.

In this section, we discuss the first two steps, i.e., the *model construction* problem. Section 5 discusses *equation generation*, and Section 6 focuses on the method for *generating conflict sets*.

Formally, the model construction problem is defined as follows:
Given:

- A *schematic description* S that includes a description of the components of the system and their interconnections, and a *functional description* F that defines the expected behaviors of the system as a whole.
- A *domain theory* Th consisting of a set of model fragments represented as bond graphs or bond graph components, and a set of rules that determine their use. Each model fragment representes a mechanism which defines the behavior of a subsystem and the role it plays in determining overall system behavior.
- A *task description* T that defines the task to be accomplished (e.g., diagnosis) and the level of detail in which this task needs to be performed. For our diagnosis task this specifies the list of components that can be considered as possible diagnostic candidates.

Produce:

- A bond graph model of the overall system from which the behavior of the system can be derived. Since our focus is on diagnosis, it is also important that given adequate measurements any fault among components (specified by T) should be identifiable.

Note that the bond graph model of a system may be made up of a number of individual bond graphs. In the modeling philosophy that we adopt, individual bond graphs usually correspond to different domains, such as thermodynamics and fluid mechanics, that the system behavior covers. Therefore, the above steps are carried out to construct each of the required bond graphs. The rest of this section focuses on: (i) system decomposition, (ii) the bond graph modeling language, and (iii) building system models using bond graphs. The decomposition process is best understood in terms of the bond graph modeling framework that is discussed next.

4.1 Bond Graphs as a Modeling Language

The effectiveness of any form of reasoning about a system is strongly dependent on the characteristics of the modeling method used. The bond graph methodology[20] provides a formal and systematic language for modeling dynamic systems that helps make a number of assumptions and issues about system functionality explicit. Bond graphs are highly organized domain independent structures that are based on a small number of primitive elements: resistances or dissipators (R), energy storage elements (capacitors C and inertial elements I), ideal sources of effort S_e and flow S_f, and distribution elements (transformer TF and gyrator GY). These elements take on different forms in different domains, but interactions between them is again expressed in a domain independent way: as energy transfers which are represented as *directed bonds*. Each bond has an associated *effort* and a *flow variable*, where $effort \times flow = power$, the rate of energy transfer. Connections between multiple elements are established by *junctions*, which can be of two types: *common flow* (i.e., series) or 1-junctions, and *common effort* (i.e., parallel) or 0-junctions.

Though the exact procedure for building bond graph models of physical systems differs slightly from domain to domain, a human modeler can, in general, follow the basic steps summarized below to build system models[20]:

- Identify the dominant variables in the domain. In mechanics these are the flow (velocity) variables, in the fluid, pneumatic, and electrical domains it is effort (the fluid pressure and voltage, respectively).
- Establish a junction for each instance of that variable, i.e., a 0-junction for each instance of dominant effort variables, and 1-junction for each instance of dominant flow variables.
- Establish bonds from these junctions to storage elements (i.e., inertial elements to 1-junctions and capacitance elements to 0-junctions).
- Connect these junctions to each other using the complement junctions, and attach dissipative elements (R) whenever necessary.
- Identify the sources of effort and flow (exogenous variables), and connect them to the proper junctions.
- Assign directions to bonds. They establish reference directions for power flow.
- Simplify the graph wherever possible, e.g., in some cases 0- and 1-junctions can be replaced with simple bonds.

The bond graph model for a simple heat exchange mechanism is illustrated in Fig. 3. The heat exchanger model depicts heat flow through a resistive junction R between two materials represented as capacitances C_1 and C_2. As discussed above, the bond graph model for this mechanism is created by identifying the dominant effort variables, which, in this case, are the temperatures T_1 and T_2. A 0-junction is established for each of these variables, which are then linked to the storage elements. The two 0-junctions are connected to each other via a 1-junction, and the dissipative element R is connected to this junction. $\dot{Q}$, the flow variable, represents the amount of heat flow that occurs across the resistive junction. Note that this representation is not unlike the view-process structure that forms the fundamental basis for

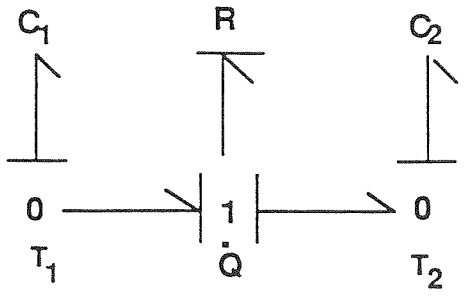

Figure 3: **Heat Exchange Mechanism: Bond Graph Library**

modeling in the QPT framework[10]. The R element models a resistive heat flow junction, and represents the relationship between heat flow rate $\dot{Q}$ and temperature difference $T_1 - T_2$. If $T_1 - T_2$ is not large, the relation can be linear, i.e., $R \cdot \dot{Q} = T_1 - T_2$. The C elements are thermal capacitances which represent the thermal energy stored in an amount of material as a function of the temperature of the material, i.e., $\Delta Q \cdot C = \Delta T$.

To extend bond graph modeling for *component-oriented diagnosis*, individual components and their relations with measurable parameters (OBS) need to be represented explicitly. In the bond graph framework, primitive elements such as resistors and capacitors represent mechanisms[21], and, therefore, may or may not be in 1-1 correspondence with individual system components. To deal with this problem, we extended parameter definitions used in the bond graph framework.

Typically, bond graphs have effort and flow variables associated with bonds, and parameters associated with primitive elements (e.g., R, C, etc). For our diagnosis framework, we divide the element parameters into two sets: *component parameters* and *co-component parameters*. Component parameters directly relate to the functionality of components under diagnostic scrutiny, e.g., the parameter R (resistance) directly relates to a primary functionality of a heat exchanger component. It models the junction at which heat transfer occurs because of temperature differences. Note that a component definition may include multiple component parameters, where each parameter represents an aspect of the functionality of the component. As part of a component parameter definition, its Possible Directions of Change (PDC) is also recorded. This is a characteristic of a specific component, e.g., some resistance values can only increase as a system degrades. In general, the PDC of a component parameter can take on one of three values: $+$, $-$, and ?, which implies that the parameter values can only increase, decrease, or deviate in either direction, respectively. It is important to note that the PDC of a parameter often determines how the parameter (and hence the corresponding component) can affect an output parameter. Using this information helps narrow down the set of conflicts, and hence, the set of candidates.

Co-component parameters are not directly associated with primitive component functionality, but they represent bond graph elements that are introduced to complete system functionality description. For example, the thermal capacitances of the heat exchanger (Fig. 3) in the precooler system (Fig. 2) do not represent individual components of the system but model the air masses that exchange heat. The air masses are in turn related to flow rates of the incoming air streams. Following this relation helps define these co-component parameters in terms of other component parameters.

Effort and flow variables are also characterized as *input*, *output*, and *state* variables. Input variables are associated with source elements of the bond graph, and, therefore, are exogenous to the system being analyzed. Output variables represent values which can be measured as part of the observation set, and state variables represent the minimum set of energy-related

variables (e.g., heat flow rate and temperature in thermodynamics, velocity and force in mechanics) that uniquely describe the state of a dynamic system[3].

Characterization of parameters depend on the viewpoint in which we analyze a system. Parameters that are known to be insignificant, or unchanged for a specific diagnostic task can be considered as constant. Parameters that express interactions between the system and other subsystems that are not modeled are also considered constant (because their effects are considered to be exogenous to the diagnostic situation).

For the pneumatic system, jet engine pressure and temperature represent input variables, because the engine is not included as part of the diagnosis task. Possible output or measurable parameters are P_{ho} and T_{ho}, the output pressure and temperature of the bleed air at the load, and P_{ro} the pressure at the pressure regulator output. Resistance at the heat exchanger junction R_p and the resistance of the sense line to liquid flow R_s, are examples of component parameters for the precooler subsystem.

Given `S`, `F`, `Th`, and `T`, model building with bond graphs can be described as a two step process:

1. decomposing the system by domain (i.e., thermodynamic, electrical, etc.) and selecting a set of primary mechanisms that define system behavior in that domain, and

2. refining the primary mechanisms based on *assumption classes* so that can be replaced by specific model fragments and composing the model fragments to generate the overall bond graph model of the system.

These steps are discussed in detail below.

4.2 System Decomposition

In general, system decomposition is task- and viewpoint dependent, and, therefore, hard to automate. For example, consider the *jet engine* as part of the pneumatic system. For the diagnosis task, if it is sufficient to determine that the cause of a problem is engine failure then the engine can be modeled as an effort source. However, if the diagnosis task requires that the cause of the problem within the engine be determined, then it is important to model the *pistons* and *valves* within the engine explicitly, and the mechanisms that determine engine functionality need to be represented in more detail. We make the assumption that the modeler performs the system decomposition task. As discussed earlier, this involves decomposing the systems functionality by domain, and selecting a set of *primary mechanisms* that define the systems functionality in that domain.

In the bond graph framework, primary mechanisms specify how a subsystem affects system behavior by controlling energy transfers between components. More formally, the primary mechanisms in any domain can be classified as: (i) *energy sources*, (ii) *energy flow and storage mechanisms* (those that transfer energy from one location to another or store energy at a location), and (iv) *energy transformation mechanisms* (those that convert energy from one form to another). In addition, we define a special class of mechanisms called *feedback mechanisms* to facilitate the modeling process. Examples of primary mechanisms in the fluid domain are sources of flow (e.g., pumps), fluid storage mechanisms (e.g., tanks and pipes), and flow or transport mechanisms (e.g., pipes).

[3]State variables may also be measurable.

From a procedural viewpoint, the modeler begins the decomposition process by first identifying the different domains that describe system behaviors of interest. For example, in the pneumatic system, the domains of interest are: (i) thermal, (ii) fluid, and (iii) mechanical. Simultaneously the modeler studies the specification task T and the system schematics that describe the set of components that are of diagnostic interest. This leads to the selection of one or more component parameters that govern behaviors of interest for each component. The next step involves selection of the primary mechanisms in the bond graph framework. In our system, the jet engine is not of diagnostic interest, therefore, it is modeled as an ideal effort source. Similarly, based on the schematics of the precooler system (Fig. 2) and its functional description, the subsystem is represented as a composition of three mechanisms: (i) heat transfer between materials through a resistive junction, (ii) feedback mechanism for temperature control, and (iii) resistive fluid flow through a valve.

By adopting bond graphs as the modeling language, we have developed a modeling framework that is formal and easy to interpret. The general principles that govern model building are based on energy transfer processes, and, therefore, are largely uniform across different domains.

4.3 Modeling with the Bond Graph Library

Once primary mechanisms have been identified, model construction takes on a compositional modeling flavor[9, 17]. In our framework this requires the modeler to go through two steps: (i) selection of bond graph fragments that are derived from primary mechanisms and additional information about the system, and (ii) composition of the fragments to build bond graph models of the system under consideration. To facilitate the modeler's task in the first model building step, we have developed *bond graph libraries* that represent collections of mechanisms in various domains. An example of an element in the bond graph library is the bond graph fragment for the heat exchange mechanism (Fig. 3). Note that the system decomposition step produces primary mechanisms and their list of associated components. For each primary mechanism, the modeler's task is to index into the library and pick the appropriate bond graph fragment(s) that corresponds to this mechanism, and then to map physical system components into this generic structure. For example, in the heat exchanger, if heat transfer occurs uniformly across a thin slab of material, the thermal resistance R is a function of the thermal conductivity, the cross-sectional area, and the thickness of the material. If the heat exchange occurs between two blocks of metal, the capacitance C of each block is a function of its mass and specific heat. On the other hand, if the heat exchange occurs between two fluids flowing through pipes, the capacitance value computations are more complex, and computed in a manner shown in Section 5.

Given a description of a primary mechanism and its components, how does the modeler index into the bond graph library and pick the appropriate fragment ? In this work, we take the approach proposed by Falkenhainer and Forbus[9] and Nayak et al.[17] and describe the indexing mechanisms in terms of *assumption classes*. An assumption class represents a *consistent* combination of the physical setting of the system, its operating conditions, and the conditions that influence the behavior of its components. Turbulent incompressible flow through a resistive pipe represents an assumption class, however, viscous flow through a pipe with no resistance is inconsistent, and cannot be called an assumption class. Assumptions can be described in further detail by classifying them as: (i) *characteristic assumptions*, and (ii)

component assumptions.

Characteristic assumptions pertain to behavioral constraints and assumptions that are *global* in nature. For example, fluid flow in a system can be characterized in the following manner:

- compressibility: compressible versus incompressible flow,
- dimensionality: one versus two versus three dimensional flow,
- velocity: subsonic versus supersonic flow,
- viscosity: inviscid versus viscous flow, and
- turbulence: laminar versus turbulent flow.

As discussed earlier, these assumptions are not all independent, and some combinations (e.g., turbulent inviscid flow) may not be relevant.

Component assumptions define, for each subsystem, the properties of its components that need to be *explicitly represented* in the model. Each of these properties can be represented by a primitive mechanism. Here, we define a primitive mechanism as a specific instance of a primary mechanism. For example, consider a *pipe* that is linked to a *fluid flow* mechanism. If its storage capacity is of importance in the analysis, this can be captured in the model by a primitive bond graph element the *capacitor*, and the pipe may then be modeled as a combination of R and C elements. On the other hand, if its storage properties are not relevant to the task at hand, the pipe can be modeled as a simple resistive element R.

Collections of characteristic and component assumptions represent alternative ways to model the same aspect or phenomenon (see Nayak et al.[17] for details). These are then organized into mutually exclusive assumption classes, which form the basis for indexing into the bond graph library and retrieving appropriate model fragments. A model fragment is a bond graph segment that contains one or more primitive bond graph elements and a set of equations, that define relations between effort and flow variables for individual bond-graph elements[20], e.g., $\dot{Q} \cdot R = T_1 - T_2$ for a resistive heat junction.

A bond-graph segment often has multiple sets of equations that define its behaviors. Each set corresponds to a specific *assumption class*. For example, pressure drop in a pipe in the case of incompressible laminar flow is represented by the equation: $\Delta P = \frac{128 l \mu}{d^4 \pi} Q$, where μ is the viscosity, l is the length, and d is the inside diameter of the pipe. For turbulent flow, the relation becomes: $\Delta P = a_t Q |Q|^{\frac{3}{4}}$, where a_t is a constant that is often determined experimentally.

Once bond-graph fragments are selected for individual mechanisms, they need to be composed to form the bond graph model of the system. In previous work, automation of model composition has been considered to be a difficult task[9, 17, 1]. However, our use of the bond graph modeling language makes the task much easier. As discussed earlier, interactions between bond graph components are expressed in a domain independent way: as energy transfers which are represented as *directed bonds*, and links between segments in the same domain are established by *junctions*, i.e., *common flow* (series) or 1-junctions, and *common effort* (parallel) or 0-junctions. Connections between bond graph segments in different domains are established if there is energy transfer between the subsystems modeled by these segments. In

this case, the connections are established through energy transform mechanisms: *transformers* and *gyrators*.

The bond graph modeler has been implemented in X window and C with a simple graphics- and menu-based interface. The menu enables the user to retrieve basic bond graph fragments in a particular domain, or access previously created mechanisms by specifying the primary mechanisms and corresponding assumptions. The graphics editor which is mouse-driven enables the user to create and edit bond graph models. Models created can be stored as subsystems (e.g., the precooler or the pneumatic system) or as mechanisms. Mechanisms require the specification of a name, the domain (or domains) to which they apply, and the set of assumptions under which they are valid. The user is also provided with facilities to express the effort-flow parameter relations between components as equations. The modeler can use this facility to modify the linear heat exchange junction and create a non linear resistive heat junction. As part of the modeling system, we have developed adequate bond graph libraries in the thermodynamic and fluid domains. The library of components in the thermodynamics domain contains basic bond graph elements, such as the resistive heat exchange junction, different models of heat capacitance, the temperature source S_e and the flow source S_f, and 0- and 1- junctions for building composed systems. In addition, we include descriptions of standard components, such as the heat exchanger (Fig. 3) and models of basic feedback mechanisms that generate signals whose strengths are proportional to temperature differences. We now discuss the construction of the bond graph model of the precooler system to illustrate the modeling methodology.

4.4 Bond Graph Model – Precooler Subsystem

As we discussed earlier, the temperature regulating part of the pneumatic system is composed of three subsystems: (i) the heat exchange subsystem, (ii) the pneumatic flow valve subsystem, and (iii) the feedback control subsystem. For each subsystem, one or more primary mechanisms are identified. For example, three primary mechanisms are identified for the heat exchange subsystem: the heat exchange mechanism, and two source mechanisms. The bond graph fragment corresponding to the particular heat exchange mechanism (Fig. 4a) is selected from the library based on the assumption that the heat exchange between the two heat masses occurs uniformly through a resistive junction R_p. Each mass is represented as a thermal capacitor. Note that this fragment is an instantiation of the fragment for heat exchange mechanism in Fig 3. Bond graph fragments for the two sources (hot and cold) (Fig. 4b and Fig. 4c) are selected based on the assumption that each source provides heat at constant temperature (T_h and T_c) along resistive pathways, i.e., there is heat loss during transport of the air mass. The bond graph fragments are then connected using bond 1 and 2 to form the model for the heat exchanger subsystem (bond graph 1 of Fig. 5). Note that the arrow on the bonds 1 and 2 indicate that the direction of energy transfer is from the hot source (S_h) to the hot capacitor (C_h) and from the cold capacitor (C_c) to the cold source (S_c).

The bond graph model for the valve subsystem (bond graph 2 of Fig. 5) is built by connecting the bond graph fragment for the resistive flow (model of the valve) with the fragment for an ideal source (model of the source of air). This bond graph indicates that the amount of pneumatic flow is determined by the resistance in the system (i.e., the valve). The resistance, in turn, is determined by the *opening* of the valve, which is controlled by the feedback system.

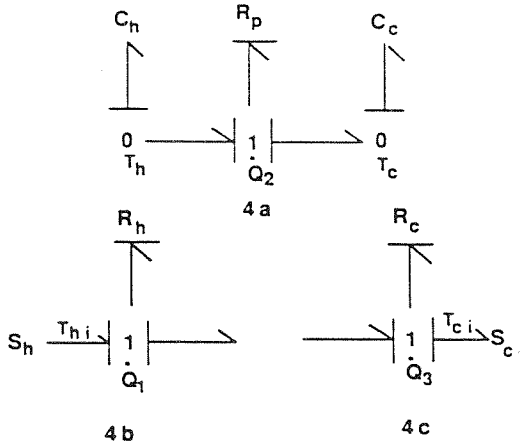

Figure 4: **Bond Graph Fragments: Precooler Subsystem**

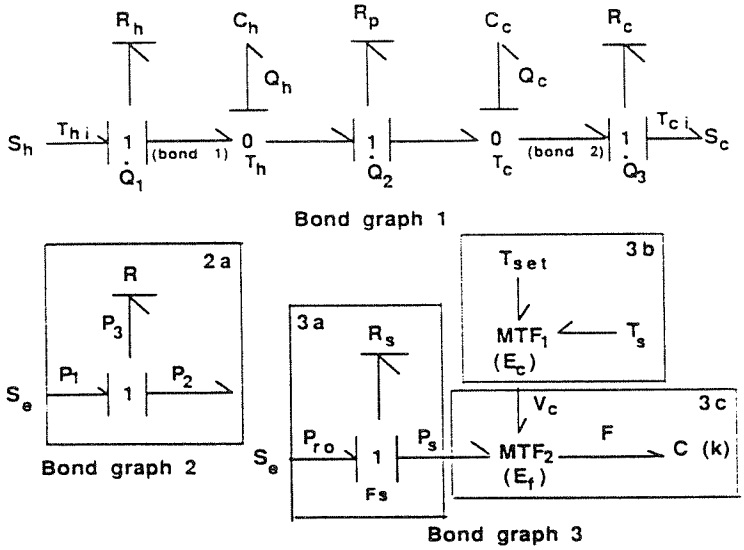

Figure 5: **Bond Graph Model: Precooler Subsystem**

The feedback subsystem consists of four primary mechanisms: (i) the pneumatic source (modeled by S_e), (ii) resistive flow that represents the function of the *sense line* (modeled by fragment 3a in Fig. 5), (iii) energy transformation from temperature difference $T_s - T_{set}$ (T_s represents the sensed temperature, and T_{set} represents the desired temperature) to a voltage signal V_c (modeled by fragment 3b, Fig. 5), and (iv) energy transfer (modeled by fragment 3c, Fig. 5). This bond graph models the physical situation where the valve controller (MTF_2) transfers a fraction of the pneumatic power P_s obtained through a *sense line* (a pipe) into a mechanical force F that acts against the valve spring to determine the opening of the valve. The amount of power transferred is determined by a voltage signal V_c from the controller(MTF_1).

5 Equation Generation

The task of equation generation is to relate output measurements to component parameters. Output equation generation is a three step process: (i) assign causal strokes to bonds in the bond graph, (ii) generate state equations, and (iii) generate output equations and manipulate

them algebraically to convert them to the desired form. The first step is discussed in detail [15, 21] and not repeated here. The algorithms we have developed for the second and third steps are presented below. Their implementation makes calls to the Mathematica package, which performs some of the required symbolic and algebraic manipulations.

5.1 State Equation Generation

The bond graph framework adopts the state space approach to modeling dynamic systems. An *nth* order system is modeled as a set of n first order differential equations. In our methodology, a state equation takes the following form:

$$\dot{x}_i = g_i(z_1, \ldots, z_n, x_1, \ldots, x_l, u_1, \ldots, u_r, c_1, \ldots, c_m),$$

where x_i's are state parameters, $\dot{x}_i$'s are their derivatives, z_i's are component parameters, u_j's are input parameters, g_i's are algebraic functions, and c_k's are co-component parameters.

The method for generating state equations from a bond graph[20] is summarized below:

- Identify the key parameters.
  For bond graph 1 (Fig. 2), the key parameters are: input – T_{hi} and T_{ci}, state – Q_h and Q_c, component – R_p, and co-component – C_c. (The input, component, and co-component parameters are prespecified by the modeler.)

- Formulate initial equations associated with the I, R, and C elements.
  Continuing the example, five initial equations are generated from bond graph 1[4]:

$$C_h: \quad T_h = T_{hi} + \frac{Q_h}{C_h} \qquad (1)$$

$$C_c: \quad T_c = T_{ci} + \frac{Q_c}{C_c} \qquad (2)$$

$$R_p: \quad \dot{Q}_2 = \frac{T_h - T_c}{R_p} \qquad (3)$$

$$R_h: \quad \dot{Q}_1 = \frac{T_{hi} - T_h}{R_h} \qquad (4)$$

$$R_c: \quad \dot{Q}_3 = \frac{T_c - T_{ci}}{R_c} \qquad (5)$$

- Formulate first order differential equations for each state variable, in terms of other variables linked to it through the same 0- or 1- junction. For variables in these equations that are not input variables or component and co-component parameters, follow causal strokes to other junctions, and substitute them with other parameters. This process continues until the equation contains only input and state variables, and component and co-component parameters.

  The $\dot{Q}_h$ variable in bond graph 1 can be expressed as: $\dot{Q}_h = \dot{Q}_1 - \dot{Q}_2$. Making substitutions using equations (1)-(4) we get:

$$\dot{Q}_h = -\frac{Q_h}{C_h R_h} - \frac{T_{hi} - T_{ci}}{R_p} - \frac{Q_h}{C_h R_p} + \frac{Q_c}{C_c R_p}.$$

  Following the same process, the equation for $\dot{Q}_c$ is derived as:

$$\dot{Q}_c = -\frac{Q_c}{C_c R_c} + \frac{T_{hi} - T_{ci}}{R_p} - \frac{Q_c}{C_c R_p} + \frac{Q_h}{C_h R_p}.$$

[4]The two air masses are modeled as lumped systems. More detailed piecewise models can also be created.

5.2 Output Equation Generation

The general form of an output equation is:

$$y_k = f(z_1, \ldots, z_n, u_1, \ldots, u_r),$$

where y_k is an output parameter. Equation generation involves symbolic manipulations, and is based on the following assumptions: (i) the physical systems we deal with are linear or are modeled by linear approximations, therefore, the algebraic functions derived for the state equations are also linear, and (ii) the systems was operating normally in a steady state, and diagnosis is initiated when system parameters deviate from their steady state values (i.e., we perform steady state diagnosis).

The algorithm for generating output equations is summarized below:

1. Assumption (ii) implies that for a state variable x_i, $\dot{x}_i$ is either 0 (no change in x_i) or a constant (x_i changes at a constant rate). Therefore, $x_i - x_{i0} = \dot{x}_i \Delta t$ is a reasonable approximation for x_i over a small time interval Δt. Computation of the Δt is situation specific, and is often dependent on the modeler's viewpoint and understanding of the system. By resolving the Δt, $\dot{x}_i$'s can be replaced by $\frac{x_i}{\Delta t}$, and the state equations assume the form:
$$0 = g_i(z_1, \ldots, z_n, x_1, \ldots, x_l, u_1, \ldots, u_r, c_1, \ldots, c_m).$$

2. Solve this set of n linear equations in n unknowns to produce equations of the form:
$$x_i = h_i(u_1, \ldots, u_r, c_1, \ldots, c_m).$$

3. Transform the equations with output parameters to the form:
$$y_i = f_i(z_1, \ldots, z_n, x_1, \ldots, x_l, u_1, \ldots, u_r, c_1, \ldots, c_m),$$
and eliminate all state variables to produce: $y_i = f_i'(z_1, \ldots, z_i, u_1, \ldots, u_r, c_1, \ldots, c_m)$.

4. Generate equations that relate co-component parameters to component parameters. This process often involves deriving output equations from bond graphs where the co-component parameters (or their related parameters) are treated as output parameters. Repeat the process until all equations for co-component parameters are of the form:
$$c_i = f_i''(z_1, \ldots, z_i, u_1, \ldots, u_r).$$

The algebraic manipulations and solution of linear equations in steps 2-4 are executed using the Mathematica package. This method applied to the output temperature variable of the pneumatic system T_{ho} is illustrated below. Since we make the lumped mass assumption, this corresponds to stating that heat transfer from hot to cold air in the precooler occurs at a fairly steady rate during the time the bleed air is in the heat exchanger. In this case, we approximate $\Delta t = \frac{l}{v}$, where l is the length of the path the air masses traverse in the precooler, and v is the velocity of the air flow, approximated as $\frac{v_h + v_c}{2}$[5]. Therefore, $\dot{Q} = \frac{Qv}{l}$ for both $\dot{Q}_h$

[5]A more exact solution would assign different velocities to the two air masses.

and $\dot{Q}_c$. Substituting for $\dot{Q}_h$ and $\dot{Q}_c$, and solving these equations produces:

$$Q_h = -\frac{T_{hi} - T_{ci}}{\frac{vR_p}{l} + \frac{R_p}{C_h R_h} + \frac{1}{C_h} + \frac{1}{C_c}} \quad (6) \qquad Q_c = \frac{T_{hi} - T_{ci}}{\frac{vR_p}{l} + \frac{R_p}{C_c R_c} + \frac{1}{C_h} + \frac{1}{C_c}} \quad (7)$$

The output parameter T_{ho} is equal to T_h after time period Δt, and, therefore, using equation (1) and (6) we get:

$$T_{ho} = T_{hi} - \frac{T_{hi} - T_{ci}}{C_h(\frac{R_p v}{l} + \frac{R_p}{C_h R_h} + \frac{1}{C_h} + \frac{1}{C_c})} \quad (8)$$

The next step is to solve for the co-component parameters. For the pneumatic system, equation for co-component parameter C_c is generated from domain knowledge,

$$C_c = Vc\rho = \Delta t F_c c\rho = \frac{l}{v} F_c c\rho, \quad (9)$$

where V is the volume of the cold air in the heat exchanger unit, ρ is the density, c is the unit thermal capacitor of the air, and F_c is the flow rate of the cold air. In this case, both ρ and c are constant. The output equation for parameter F_c is derived from bond graph 2:

$$F_c = \sqrt{\frac{P_3}{C_3}}(X_{max} - X), \quad (10)$$

where P_3 is the pressure difference over the valve, C_3 is the valve constant, and X_{max} is the max length of the opening of the valve. Note that while P_3 and C_3 are both constants, X is still a co-component parameter, and it is determined by the force exerted on the spring. Bond graph 3 is now analyzed with X considered as an output parameter to produce:

$$X = \frac{A}{k}(P_{ro} - R_s F_s)E, \quad (11)$$

where P_{ro} is the incoming pressure of the flow from the pressure regulator subsystem, A is the area of the opening of the sense line, E is the percentage of power that is applied to the valve spring based on readings from the temperature sensor. A is a constant, while E is a co-component parameter, which is represented by:

$$E = E_f + E_c(T_{set} - R_{cs} q_r), \quad (12)$$

where F_s is the flow rate through the sense line, and T_{set} is the desired temperature. T_{set} and F_s are all constants. Note that P_{ro} is considered to be an input parameter when the diagnosis focuses on the precooler subsystem. However, when the whole system is under scrutiny, P_{ro} become co-component parameter, and equations also need to be derived that relates it to components of the pressure regulator subsystem. The following equations are generated using the same techniques described earlier:

$$P_{ro} = P_{in}(1 - \frac{R_p}{R_p + \frac{C_2}{X_h}}) \quad (13)$$

where P_{in} is an input parameter (pressure of the flow coming into the pneumatic system), C_3 is constant, and X_h is co-component parameter which is represented by the following equation:

$$X_h = X_{set} - A_2(P_{in} - R_{pt}f_{pt})\frac{E_p}{K_p} \tag{14}$$

where A_2 and f_{pt} are constants, and R_pt, E_p and K_p are component parameters.

After equations for all the output parameters and the co-component parameters are generated, the final form of the output equations may be generated by systematically substituting for co-component parameters till all parameters in the equations are either input parameters or component parameters. The equations would then be in a form where direct relations could be established with component parameters to generate potential conflicts. For example, the output equation for T_{ho} can be generated by combing equations (8), (9), (10), (11), and (12). However, this often produces complicated forms that are difficult to analyze using our automated algorithm. Therefore, we often have multiple equations associated with an output variable rather than one complex equation. For example, our implementation keeps equations (8), (9), (10), (11), and (12) as a set associated with the output variable T_{ho}. These equation sets are then used for diagnostic analysis which is discussed in the next section.

6 Generating Conflicts

Given the current set of observations (values of parameters measured) and the set of output equations that relate these parameters to component parameters, our system identifies the current set of conflicts.

For each variable whose value has been measured, a conflict that depends on the deviation of the measured value is constructed. Our notion of *conflict* extends the existing one, which is defined as a set of assumptions (e.g., a component is normal) that support a symptom, and, therefore, cannot all be true to explain a deviant measurement. For a deviant parameter, the conflict contains the list of component parameters, at least one of which has to be faulty, and whose malfunction could cause the output variable to deviate in the observed direction. For a parameter that is in its normal range, the conflict contains a list of component pairs. Each pair represents component parameters that are both normal, or both faulty so that their combined effect on the output parameter is null. Here, we assume that the normal range of a parameter during steady state operation of the system is given as part of the system description.

The first step in conflict generation analyzes how each candidate component parameter may be linked to the output variable. For example, it may be determined that an increase in the value of a particular component parameter will cause the output to decrease if there are no other changes in the system. To perform this analysis, consider the set of equations associated with each output variable y. For each such equation, $p_k = f_k(w_1, \ldots, w_n, u_1, \ldots, u_m)$, where p_k is either an output parameter or a co-component parameter, u's are input parameters, and w's are component or co-component parameters, we perform the following analysis:

1. Compute the partial derivative $\frac{\partial p_k}{\partial w_i}$.

2. For each term in $\frac{\partial p_k}{\partial w_i}$, which is either a component parameter, co-component parameter, an input variable, or a constant, assign one of $+$, 0, $-$, ? as its qualitative value. The

qualitative value of a variable or parameter is established from knowledge of its numeric value in the operating region of the system. If the value is known to be positive (negative, zero) it is represented as a + (−, 0); if its value is unknown, it is represented as ?. A qualitative algebra similar to that of [6] is applied to determine the qualitative value of $\frac{\partial p_k}{\partial w_i}$[23]. Given the qualitative value of $\frac{\partial p_k}{\partial w_i}$, the relationship between parameters p_k and w_i is determined: a + value indicates a direct proportionality, a − represents an inverse proportionality, and 0 implies that two parameters are not related. A ? implies that the relationship cannot be established due to lack of information.

3. Calculate $PDC(y, z_i)$ using the formula $PDC(y, z_i) = PDC(z_i) * \frac{\partial y}{\partial z_i}$, where * is qualitative multiplication.

For p_k's that are co-component parameters this analysis is repeated recursively, till direct relations are established between the output variable and component parameters. For example, to obtain the relation between R_s and T_{ho}, equation (11) is first evaluated. Step (1) produces: $\frac{\partial X}{\partial R_s} = -\frac{F_s A}{k} E$, where $[A]$, $[k]$, $[E]$, and $[F_s]$, the qualitative values of the respective variables, are known to be +. Therefore, the partial derivative evaluates to −. Since X is not an output parameter, the process repeats through equations (10), (9), and (8), and we get $\frac{\partial T_{ho}}{\partial R_s} = -$, meaning that the relation is an inverse proportionality, i.e., if R_s increases T_{ho} must decrease. Given that $PDC(R_s) = +$, we get $PDC(T_{ho}, R_s) = -$.

After relationship between component parameters and output parameters are established, the current set of conflicts are generated as discussed below:

- For each deviant parameter y, select all z_i's such that $PDC(y, z_i)$ is consistent with the current observed deviation of y. A PDC is defined to be consistent with a deviation if (i) they have the same value (both + or −), or (ii) the PDC has value ? (e.g., it can either be + or −).

- For each normal parameter y, form a propositional formula: $(\neg z_i \wedge \ldots \wedge \neg z_n) \vee (z_1 \wedge z_2) \vee \ldots \vee (z_{n-1} \wedge z_n)$, where for each pair (z_i, z_j), $PDC(y, z_i)$ and $PDC(y, z_j)$ are complementary and $z_i \neq z_j$. A pair of PDCs are defined to be complementary if one is consistent with + and the other is consistent with −. This formula suggests that the z_i's are either normal or at least two of them are deviant, so that their combined effect is null.

Suppose the observed deviation for T_{ho} is + (above-normal), a conflict for T_{ho} is generated by analyzing $PDC(T_{ho}, z)$ for each component parameter z. For example, given $PDC(T_{ho}, R_s) = -$ we know that the possible change in the resistance of the sense line (e.g., a blockage) is not consistent with the observation, and, therefore, R_s is excluded from the conflict for T_{ho}.

The resulting conflict for T_{ho} is:
$F(T_{ho}) = R_p + \vee E_f + \vee K - \vee E_c - \vee R_{cs} -$. Note that for parameters that can change in both directions, the *specific* direction of change that explains the particular deviation is explicitly recorded in the list. This information can be used to prune candidate sets in the diagnosis algorithm. As we mentioned earlier, when the entire pneumatic system is considered, P_{ro} becomes co-component and the effects of components in the pressure regulator subsystem on T_{ho} also need to be considered. As a result, the conflict for T_{ho} now becomes:
$F(T_{ho}) = R_p + \vee E_f + \vee k - \vee E_c - \vee R_{cs} - \vee K_p + \vee E_p - \vee R_p t +$. Using the same method, the conflict for a normal parameter P_{ho} (the pressure at the output of the pneumatic system) is

generated as:
$F(P_{ho}) = (\neg K_p \wedge \neg R_p \wedge \neg E_p \wedge \neg R_{pt}) \vee (K_p + \wedge E_p +) \ldots$
Note that for each z_i, $\neg z_i$ implies $(\neg z_i + \wedge \neg z_i -)$. For example, $\neg k_p$ represents $(\neg k_p + \wedge \neg k_p -)$.

7 Summary

In this paper, we presented a bond-graph based modeling scheme that focuses on the diagnosis task by converting an analytic equation-based model of the system into conflict sets that are generated from observations and measurements made on the system. Our overall modeling philosophy mirrors the compositional modeling approach presented by Falkenhainer and Forbus[9] and Nayak et al.[17]. The primary difference is that our modeling framework is based on the more formal bond graph language, and, therefore, we are better able to characterize and formalize the system decomposition and model composition tasks. Besides, the advantage of starting with an analytic equation-oriented model, provides the opportunity to introduce successively more precise information (such as orders of magnitude information, and quantitative values for parameters) if available, and derive more accurate diagnostic results without altering our framework or modeling methodology. This, as well as our focus on diagnosis, possibly differentiates our work from other bond graph applications in model based reasoning (e.g., Top and Ackermans, Linkens, etc.).

To demonstrate the general capabilities of our modeling and diagnosis methodology in handling complex continuous-valued systems, we are currently expanding and refining the bond graph libraries to accommodate the space station thermal bus system. We have reused a number of models created for the pneumatic system, such as heat exchange mechanisms, and fluid flow through pipes. A number of new mechanisms have also been created, e.g., a fluid flow source to model a cavitating venturi, and heat exchangers that involve materials in two phases (liquid and vapor). We are also working on extending our diagnosis schemes to make it more efficient. Our goal is to use results from system-level diagnosis[18] so that given a partial set of measurements, one can select the minimum set of additional measurements that will guarantee a complete diagnosis in polynomial time.

Acknowledgements: We acknowledge the work of Stefanos Manganaris who contributed to the development of the bond graph modeling methodology and the development of the pneumatic system model.

References

[1] G. Biswas, S. Manganaris and X. Yu, "Extending Component Connection Modeling for Analyzing Complex Physical Systems", *to appear, IEEE Expert*, 1992.

[2] T. Crews, S. Manganaris, and G. Biswas. "A Graphics Interface for Bond Graph Modeling" *Tech. Report*, Dept. of Computer Science, Vanderbilt University, Nashville, TN, 1991.

[3] P. Dague, O. Raiman, and P. Deves, "Troubleshooting: when modeling is the difficulty", *Proceedings of 6th National Conference on Artificial Intelligence*, pp. 600-605, Seattle, WA, August, 1987.

[4] R. Davis. "Diagnostic reasoning based on structure and behavior", *Artificial Intelligence*, pp. 347-410, 1984.

[5] R. Davis and W.C. Hamscher, "Model-Based Reasoning: Troubleshooting", in *Exploring Artificial Intelligence: Survey Talks from the National Conferences on AI*, H.E. Shrobe, ed., pp. 297-346, Morgan Kaufmann, San Mateo, CA, 1988.

[6] J. de Kleer and J. S. Brown, "A qualitative physics based on confluences", *Artificial Intelligence*, vol. 24, pp. 7-83, 1984.

[7] J. de Kleer and B.C. Williams. "Diagnosing multiple faults", *Artificial Intelligence*, pp. 97-130, 1987.

[8] J. de Kleer. "Focusing on Probable Diagnoses", *Proceedings of the 9th AAAI*, pp. 842-848, 1991.

[9] B. Falkenhainer and K. Forbus. "Compositional Modeling: Finding the Right Model for the Job", *Artificial Intelligence (Fall 1991), special issue on Qualitative Physics*, pp. 95-143, 1991.

[10] K.D. Forbus. "Qualitative Process Theory", *Artificial Intelligence*, vol. 24, pp. 85-168, 1984.

[11] M. Gallanti, M. Roncato, R. Stefanini, and G. Tornielli. "A diagnostic algorithm based on models at different levels of abstraction", *Proceedings of 11th IJCAI*, pp. 1350-1355, 1989.

[12] M.R. Genesereth. "The use of design descriptions in automated diagnosis", *Artificial Intelligence*, pp. 411-436, 1984.

[13] W.C. Hamscher. "Model-based Troubleshooting of Digital Systems", *Tech. Report*, AI Lab, MIT, 1988.

[14] W.C. Hamscher. "Principles of Diagnosis: Current Trends and a Report on the First International Workshop", *AI Magazine*, pp. 15-22, 1991.

[15] S. Manganaris and G. Biswas. "Modeling and Qualitative Reasoning about Perturbations in Mechanics Problem Solving", *Tech. Report*, CS-90-16, Vanderbilt University, Nashville, TN, 1990.

[16] W. Morris, R.L. Boyer, and T. Hill. "Thermal Control System Automation Project (TCSAP)", *Interim Report CTSD-SS-527*, McDonnell Douglas Space Systems Company, Houston, TX, Dec. 1991.

[17] P. P. Nayak, L. Joskowicz, and S. Addanki. "Aumomated Model Selection using Context-Dependent Behaviors". *Proceedings of the Tenth National Conference on Artificial Intelligence*, July, 1992.

[18] V. Raghavan and A.R. Tripathi, "Improved Diagnosability Algorithms", *IEEE Trans. on Computers*, vol. 40, pp. 143-153, 1991.

[19] R. Reiter. "A Theory of Diagnosis from First Principles", *Artificial Intelligence*, pp. 57-95, 1987.

[20] R.C. Rosenberg, and D.C. Karnopp. *Introduction to Physical System Dynamics*, McGraw-Hill, 1983.

[21] J. Top and H. Akkermans. "Computational and Physical Causality", *Proceedings 12th IJCAI*, pp. 1171-1176, 1991.

[22] X. Yu, S. Manganaris, and G. Biswas. "A Component Connection (CC) Modeling Case Study: Results and Future Directions", *Working Paper of the Qualitative Reasoning Workshop*, Austin, TX, May 1991. pp. 326-338.

[23] X. Yu and G. Biswas, "A Candidate Generation Method for Model-Based Diagnosis of Continuous-valued System", *Working Papers: AAAI Model-Based Reasoning Workshop*, Anaheim, CA, July, 1991.

[24] X. Yu and G. Biswas. "A method for Diagnosis of Continuous-valued Systems", *submitted to International Workshop on Principles of Diagnosis*, 1992.