

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

Gautam Biswas and Xudong Yu

Dept. of Computer Science

Vanderbilt University

Nashville, TN 37235. U. S. A.

e-mail: biswas, xudongy @vuse.vanderbilt.edu

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 partial 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 system¹[Davis and Hamscher, 1988]. Model-based diagnosis researchers (e.g., [Davis, 1984]) have successfully developed effective and efficient device-independent diagnosis algorithms that consist of two primary subtasks: (i) *candidate generation*, and (ii) *measurement selection*. However, the availability of appropriate system description models that make all the information required for diagnosis explicit is the key to the success of this methodology [Davis and Hamscher, 1988]. Most past efforts (especially *logic-based* approaches to diagnosis) have focused on digital circuits [de Kleer and Williams, 1987; Hamscher, 1988]. Whereas diagnosis of complex circuits provide formidable computational challenges [de Kleer, 1991], the problem becomes even more complicated when one tries to apply these methodologies to dynamic continuous systems. The primary reason for these difficulties can be attributed to the lack of computationally tractable models that are precise enough to be useful for diagnosis [Dague *et al.*, 1987; Davis and Hamscher, 1988].

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

¹An alternate approach is parameter-oriented diagnosis: correct system malfunctions by appropriate control actions (e.g., [Gallanti *et al.*, 1989]).

This paper discusses a formal modeling scheme for continuous physical systems, adapted from the bond graph modeling scheme in systems dynamics [Rosenberg and Karnopp, 1983], that generates appropriate system descriptions for efficient component-oriented, logic-based diagnosis. This is achieved by ensuring that: (i) individual components and component behaviors are explicitly represented, and (ii) relations between measurable parameters and individual components are readily derivable from the models. Our diagnosis algorithm starts off with a set of equations that link output measurements to component parameters, and generates a *partial conflict set* by performing qualitative causal analysis on these output equations. The partial conflict set forms the basis for efficient *candidate generation* and *measurement selection* algorithms.

Adopting this modeling framework provides a unifying framework for studying diagnosis of both continuous and discrete (digital) systems. Our focus in this paper is on modeling for diagnosis. This involves generation of the system description as a set of equations that relate measurements to system components and converting them to partial conflict sets. Candidate generation and measurement selection are discussed in detail elsewhere [Yu and Biswas, 1992]. To illustrate our methodology, we develop diagnosis models of the pneumatic system of DC-10 aircraft. This system regulates air pressure and temperature drawn from a jet engine before it is delivered through the manifold system to different subsystems of the aircraft that constitute loads (e.g., the wing de-icing system). The pneumatic pressure is regulated by a *pressure regulator*, and the temperature is controlled by a *precooler subsystem* (see Fig. 1).

2 From Physical Schematics to Bond Graph Models

Effective problem solving using model-based approaches requires the ability to dynamically construct models of a system that are both parsimonious and adequate for the specific task [Falkenhainer and Forbus, 1992; Nayak *et al.*, 1992]. The overall modeling method is more elaborately described as a four step process:

1. decomposition into *primary mechanisms*,

2. model fragment selection based on mechanisms and *assumption classes*; composition of fragments to generate the overall model,
3. output equation generation, and
4. *partial conflict* generation from measurements using qualitative causal analysis.

Formally, the model construction problem is:

Given: (a) A *schematic description* S that includes components of the system and their interconnections, and a *functional description* F that defines the expected behaviors,

(b) 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 represents a mechanism, and

(c) 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 diagnosis this is the list of possible diagnostic candidates)

Produce: A bond graph model of the overall system for deriving its behavior. Since our focus is on diagnosis, it is also important that given adequate measurements any fault among components (specified by T) should be identifiable.

In our modeling scheme, individual bond graphs usually correspond to different physical domains of system behavior, such as thermodynamics and fluid mechanics. Therefore, the bond graph model of a system may be made up of a number of individual bond graphs.

2.1 Bond Graphs as a Modeling Language

The bond graph methodology [Rosenberg and Karnopp, 1983] provides a formal and systematic language for modeling dynamic systems that 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 transformational elements (transformer TF and gyrator GY). These elements take on different physical forms in different domains, but interactions between them are expressed in a domain independent way: as energy transfers represented as *directed bonds*. Each bond has an associated *effort* and a *flow variable*, where $effort \times flow = power$, the rate of energy transfer. Examples of effort and flow variables are temperature and heat flow rate in thermodynamics and force and velocity in mechanics. Connections between elements are established by *common flow* (series) or 1-junctions, and *common effort* (parallel) or 0-junctions.

There exists a systematic procedure for building bond graph models of physical systems [Rosenberg and Karnopp, 1983]. Fig. 2 illustrates the bond graph model of a simple heat exchanger mechanism, which depicts heat flow through a resistive junction R between two materials represented as capacitances C_1 and C_2 . This representation is not unlike the view-process structure, the fundamental basis for modeling in the QPT

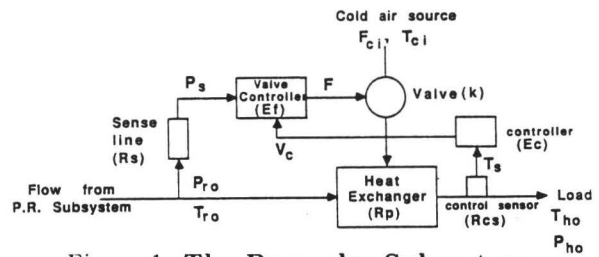


Figure 1: The Precooler Subsystem

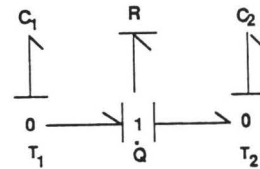


Figure 2: Heat Exchanger Mechanism

framework [Forbus, 1984]. The R element models the relationship between heat flow rate $\dot{Q}$ and temperature difference $T_1 - T_2$. If $T_1 - T_2$ is not large $R \cdot \dot{Q} = T_1 - T_2$. 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., $\Delta Q \cdot C = \Delta T$.

Primitive bond graph elements, such as resistors and capacitors represent mechanisms [Top and Akkermans, 1991] which may or may not be in 1-1 correspondence with individual system components. To extend this framework for *component-oriented diagnosis*, individual components related to measurable parameters are represented explicitly by creating *component* and *co-component parameters*.

Component parameters directly relate to the functionality of components under diagnostic scrutiny, e.g., R (resistance) directly relates to a primary functionality of a heat exchanger component. A component definition may include multiple component parameters that represent different aspects of component functionality. Component parameter definitions also include its Possible Directions of Change (PDC). This is characteristic of specific components, e.g., some resistance values only increase as a system degrades. The PDC of a component parameter takes on one of three values: + (only increases), - (only decreases), and ? (can change in either direction). It determines how that parameter and hence the corresponding component can affect output parameters. This information helps narrow down possible candidates for diagnosis.

Co-component parameters are not directly associated with primitive component functionality. They represent bond graph elements that are introduced to complete system functionality description. The thermal capacitances of the heat exchanger (Fig. 2) in the precooler system (Fig. 1) are not individual components. They model the air masses that exchange heat. The air masses are in turn related to flow rates of the incoming air streams. Thus, co-component parameters are defined in terms of other component parameters.

In addition, the model defines *input* variables associated with source elements of the bond graph. They are exogenous to the system being analyzed. *Output* variables represent values which can be measured as part of the observation set. 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 are considered to be constant. Parameters that express interactions between the system and other subsystems 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 are input variables. 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.

2.2 System Decomposition

In general, system decomposition being task- and viewpoint dependent is hard to automate. For example, if it is sufficient to determine that the cause of a problem is jet engine failure then the engine can be modeled as an effort source. If the diagnosis task requires that the cause of the problem within the engine be determined, then it is important to model explicitly the internal mechanisms and components that govern engine functionality. We make the assumption that the modeler performs the system decomposition task by selecting the set of *primary mechanisms* that define its functionality.

In the bond graph framework, primary mechanisms can be classified as: (i) *energy sources*, (ii) *energy flow and storage mechanisms*, and (iii) *energy transformation mechanisms*. 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 flow sources (e.g., pumps), fluid storage mechanisms (e.g., tanks and pipes), and flow or transport mechanisms (e.g., pipes).

The modeler begins the decomposition process by first identifying the different domains that describe system behaviors of interest. 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 to pick 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. The precooler system (Fig. 1) system 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. Note that the system decomposition step produces primary mechanisms and their list of associated components.

2.3 Modeling with the Bond Graph Library

Once primary mechanisms have been identified, model construction takes on a compositional modeling flavor [Falkenhainer and Forbus, 1992; Nayak *et al.*, 1992], which is implemented as a two steps process: (i) selection of bond graph fragments that are derived from primary mechanisms and assumption classes, and (ii) composition of the fragments to build bond graph models of the system under consideration. The modeler's task is facilitated by developing *bond graph libraries* that represent collections of mechanisms in various domains. An example of an element in the library is the heat exchanger mechanism fragment (Fig. 2). 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. 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. For two blocks of metal, the capacitance C of each block is a function of its mass and specific heat. If the heat exchange occurs between two fluids flowing through pipes, the capacitance value computations are more complex.

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? We take the approach proposed by [Falkenhainer and Forbus, 1992] and [Nayak *et al.*, 1992] to describe an indexing mechanism 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. Viscous flow through a pipe with no resistance is inconsistent, and cannot be an assumption class. Assumptions can be further classified into (i) *characteristic assumptions*, and (ii) *component assumptions*.

Characteristic assumptions pertain to behavioral constraints and assumptions that are *global* in nature. For fluid flow the following assumptions are relevant:

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

These assumptions are not all independent, and some combinations may be invalid.

Component assumptions define, for each subsystem, properties of its components that need to be *explicitly represented* in the model. Each property corresponds to a primitive bond graph mechanism. For example, consider a *pipe* as a *fluid flow* mechanism. If its storage capacity is important in the analysis, the pipe is modeled as a combination of R and C elements. If its storage properties are not relevant, the pipe is modeled as a simple resistive element R .

Characteristic and component assumptions are organized into mutually exclusive assumption classes, which

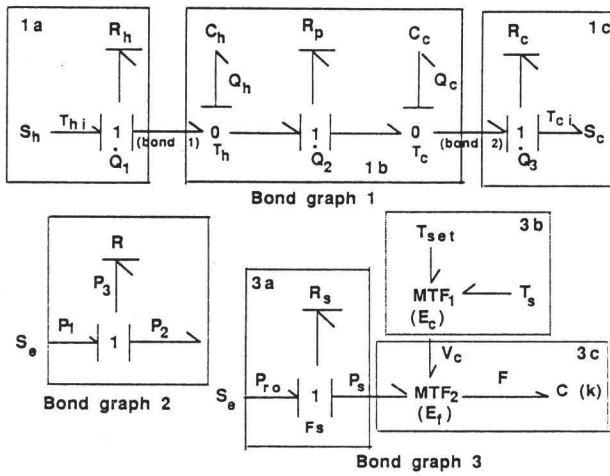


Figure 3: Bond Graph Model: Precooler

form the basis for indexing into the bond graph library and retrieving appropriate model fragments. A model fragment is a bond graph segment and a set of equations, that define effort-flow variable relations for individual bond-graph elements.

A bond-graph segment may have multiple equations to define its behaviors. Each equation set corresponds to a specific *assumption class*. For example, pressure drop in a pipe for incompressible laminar flow is represented by the equation: $\Delta P = \frac{128\mu l}{d^4\pi} Q$, where μ is the viscosity, l is the length, and d is the inside diameter of the pipe. For turbulent flow, the relation becomes: $\Delta P = a_t Q|Q|^{\frac{3}{2}}$, where a_t is a constant that is often determined experimentally.

Selected bond graph segments are then composed to build the bond graph model of the system. This difficult task[Biswas *et al.*, 1992] is much simplified in the bond graph framework, because interactions between components are expressed in one of three domain-independent ways: (i) energy transfers represented as *directed bonds*, (ii) links between segments established by 0- and 1-*junctions*, and (iii) interactions between segments in different domains established by *TF* and *GY* elements.

The bond graph modeler has been implemented in X window and C with a simple graphics- and menu-based interface[Yu, 1992]. As part of the modeling system, we have developed adequate bond graph libraries in the thermodynamic and fluid domains.

2.4 Bond Graph Model – Precooler Subsystem

As discussed earlier, the temperature regulating part of the pneumatic system (Fig. 1) is composed of the heat exchanger, the pneumatic flow valve, the feedback control subsystems. Three primary mechanisms are identified for the heat exchanger subsystem: the heat exchanger mechanism, and two source mechanisms. The bond graph fragment for the heat exchange mechanism (fragment 1b, Fig. 3) models uniform heat exchange be-

tween the two air masses through a resistive junction R_p . Each mass is represented as a thermal capacitor. Bond graph fragments for the hot and cold sources (fragments 1a and 1c) provide 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, Fig. 3).

Bond graph 2 for the valve subsystem connects the bond graph fragment for resistive flow (model of the valve) with the fragment for an ideal pneumatic source. The amount of pneumatic flow is determined by the resistance in the system, a function of the *opening* of the valve, which is controlled by the feedback system.

The feedback subsystem has four primary mechanisms: (i) the pneumatic source S_e , (ii) resistive flow in the *sense line* (fragment 3a, Fig. 3), (iii) energy transformation from temperature difference $T_s - T_{set}$ (T_s : sensed temperature, T_{set} : desired temperature) to a voltage signal V_c (fragment 3b, Fig. 3), and (iv) energy transfer (fragment 3c, Fig. 3). 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* (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).

3 Equation Generation

Output equations that relate output measurements to component parameters are generate in three steps: (i) assign causal strokes to bonds in the bond graph, (ii) generate state equations, and (iii) generate initial output equations and manipulate them algebraically to convert them to the desired form. The first step is discussed elsewhere [Top and Akkermans, 1991] and not repeated here. The algorithms 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.

3.1 State Equation Generation

In the bond graph framework, an n th order system is modeled as a set of n first order differential equations in the following form:

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

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 standardized procedure for generating state equations[Rosenberg and Karnopp, 1983] involves formulation of initial equations associated with the I, R, and C elements. For example, the hot air mass is represented as a capacitor element satisfies the equation² $T_h = T_{hi} + \frac{Q_h}{C_h}$. First order differential equations are then formulated for each state variable. The $\dot{Q}_h$ variable in

²The two air masses are modeled as lumped systems. More detailed piecewise models can also be created.

bond graph 1 can be expressed as: $\dot{Q}_h = \dot{Q}_1 - \dot{Q}_2$. Making appropriate substitutions using equations for the I, R, and C elements we get:

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

3.2 Output Equation Generation

The general form of an output equation is:

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

where y_k is an output parameter. Equation generation is based on the following assumptions: (i) the physical systems we deal with are linear or are modeled by linear approximations, 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).

For generating output equations in the proper form, assumption (ii) is exploited to set $\dot{x}_i$ to 0 (no change in x_i) or a constant (x_i changes at a constant rate). The state variables are then solved for and substituted into the output equations to generate: $y_i = f'_i(z_1, \dots, z_i, u_1, \dots, u_r, c_1, \dots, c_m)$. To relate co-component parameters to component parameters output equations are generated from bond graphs where the co-component parameters are treated as output parameters. This is repeated till all equations for co-component parameters are of the form: $c_i = f''_i(z_1, \dots, z_i, u_1, \dots, u_r)$. The algebraic manipulations and solution of linear equations are executed using the Mathematica package.

This method applied to the output temperature T_{ho} produces the initial output equation:

$$T_{ho} = T_{hi} - C_h(T_{hi} - T_{ci})\left(\frac{v}{l} + \frac{1}{C_c R_c D}\right) \quad (1)$$

Co-component parameters are solved for recursively until they are represented as a function of component and input parameters only. For the pneumatic system, the following set of equations are generated. (Details of their derivation appear in [Yu, 1992].)

$$C_c = Vc\rho = \Delta t F_c c\rho = \frac{l}{v} F_c c\rho \quad (2)$$

$$F_c = \sqrt{\frac{P_3}{C_3}}(X_{max} - X) \quad (3)$$

$$X = \frac{A}{k}(P_{ro} - R_s F_s)E \quad (4)$$

$$E = E_f + E_c(T_{set} - R_{cs} q_r) \quad (5)$$

$$P_{ro} = P_{in}\left(1 - \frac{R_p}{R_p + \frac{C_3}{X_h}}\right) \quad (6)$$

$$X_h = X_{set} - A_2(P_{in} - R_{pt} f_{pt})\frac{E_p}{K_p}. \quad (7)$$

A number of these parameters are constants; some are measurable parameters (e.g., P_{ro}) and others are component parameters (e.g., R_s). 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 are then in a form where direct relations could be established with component parameters to generate potential conflicts.

4 Generating Partial Conflicts

For each output variable whose value is known, a *partial conflict* which depends on the deviation of the measured value is constructed. Partial conflict extends the notion of *conflicts*[de Kleer and Williams, 1987], which are defined in terms of a set of assumptions (e.g., a component is normal) related to a symptom that produces an inconsistency. In their ATMS-based methodology, conflicts were generated *incrementally* as new measurements are made. Earlier measurement results were combined with a new one to generate an additional conflict. We take a slightly different but equivalent approach and introduce the concept of a *partial conflict*. A partial conflict, associated with a measurement, takes one of two forms:

1. if the measurement is deviant, the partial conflict is defined as a disjunction of PDC's for component parameters that are consistent with the direction of change of the measured value, indicating that at least one of them must be deviant in that indicated direction, and
2. if the measurement is within its normal range, the partial conflict is basically a disjunction of 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. (We assume that the normal range of a parameter during steady state operation is given as part of the system description.)

The first step in partial 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 equations each output variable y_k . For each such equation, generate the partial derivative $\frac{\partial y_k}{\partial z_i}$, where z_i is a component parameter, and compute its qualitative value (+, -, 0, or ?). Then compute $PDC(y, z_i +)$ and $PDC(y, z_i -)$ using: $PDC(y, z_i) = PDC(z_i) * \frac{\partial y}{\partial z_i}$, where * is qualitative multiplication.

For 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 (2) is first evaluated to produce: $\frac{\partial X}{\partial R_s} = -\frac{F_s A}{k} E = -$, because the qualitative values $[A]$, $[k]$, $[E]$, and $[F_s]$ are known to be +. Since X is not an output parameter, the process repeats through equations 4-8, and we get $\frac{\partial T_{ho}}{\partial R_s} = -$, i.e., if R_s increases T_{ho} must decrease.

After relationship between component parameters and output parameters are established, the current set of conflicts are generated: (a) for each deviant parameter y , select all z_i 's such that $PDC(y, z_i)$ is consistent with the current observed deviation in y ; (b) for each normal parameter y , generate: $(\neg z_i \wedge \dots \wedge \neg z_n) \vee (z_1 \wedge z_2) \vee \dots \vee (z_{n-1} \wedge z_n)$, where for each pair (z_i, z_j) , $PDC(y, z_i)$ and $PDC(y, z_j)$ are complementary and $z_i \neq z_j$ is generated. (PDC's are complementary if one is consistent

with + and the other is consistent with -).

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, if $PDC(T_{ho}, R_s) = -$ the conflict for T_{ho} is: $F(T_{ho}) = R_p + \vee E_f + \vee K - \vee R_s - \vee E_c - \vee R_{cs}$. Similarly, the conflict for a normal parameter P_{ho} (the pressure at the output of the pneumatic system) is generated as: $F(P_{ho}) = (\neg K_p \wedge \neg R_p \wedge \neg E_p \wedge \neg R_{pt}) \vee (K_p \wedge E_p) \dots$. Each $z_i, \neg z_i$ implies $(\neg z_i + \wedge \neg z_i -)$. For example, $\neg K_p$ represents $(\neg K_p + \wedge \neg K_p -)$. Combining the two partial conflicts $F(T_{ho}) \wedge F(P_{ho})$ produces the current set of possible candidates. Details of the rest of the diagnosis algorithm appear in [Yu and Biswas, 1992].

5 Summary

This paper discusses a compositional modeling approach that exploits formal properties of the bond graph modeling language to build models of complex continuous-valued physical systems. Our overall modeling philosophy mirrors the compositional modeling approach presented by [Falkenhainer and Forbus, 1992] and [Nayak et al., 1992]. The primary difference is that the more formal properties of the bond graph language allow us to better characterize and formalize the system decomposition and model composition tasks. Besides, bond graph components are grounded in physical reality, therefore, models formed by composing bond graph models are guaranteed to be meaningful and physically realizable. The accuracy of the model can be linked to the explicit assumptions that have to be made in linking system mechanisms to bond graph components.

To focus on diagnosis, we demonstrate how partial conflicts are generated from analytic bond graph models. This leads to efficient candidate generation and measurement selection techniques [Yu and Biswas, 1992]. We have also shown that starting with an analytic equation-oriented model successively more precise information (such as orders of magnitude information, and quantitative values for parameters) can be introduced, if available, to derive more accurate diagnostic results. 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 [Top and Akkermans, 1991]).

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, and a number of new mechanisms have also been created, e.g., heat exchangers that involve materials in two phases (liquid and vapor). We are also developing control structures for automated model switching to accommodate situations where changes in operating conditions of the physical system are drastic enough to require dynamic changes of bond graph models of the system.

References

- [Biswas et al., 1992] G. Biswas, S. Manganaris and X. Yu. Extending Component Connection Modeling for Analyzing Complex Physical Systems. *IEEE Expert*, pp. 48-57, Feb. 1993.
- [Dague et al., 1987] P. Dague, O. Raiman, and P. Deves, Troubleshooting: when modeling is the difficulty, *Proc. 6th AAAI*, pp. 600-605, Seattle, WA, August, 1987.
- [Davis, 1984] R. Davis. Diagnostic reasoning based on structure and behavior, *Artificial Intelligence*, pp. 347-410, 1984.
- [Davis and Hamscher, 1988] 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.
- [de Kleer and Williams, 1987] J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, vol. 32, pp. 97-130, 1987.
- [de Kleer, 1991] J. de Kleer. Focusing on Probable Diagnoses. *Proc. 9th AAAI*, Anaheim, CA, pp. 842-848, 1991.
- [Falkenhainer and Forbus, 1992] B. Falkenhainer and K. Forbus. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence*, vol. 51, pp. 95-143, 1991.
- [Forbus, 1984] K.D. Forbus. Qualitative Process Theory. *Artificial Intelligence*, vol. 24, pp. 85-168, 1984.
- [Gallanti et al., 1989] M. Gallanti, M. Roncato, R. Stefanini, and G. Tornielli. A diagnostic algorithm based on models at different levels of abstraction. *Proc. 11th IJCAI*, Detroit, MI, pp. 1350-1355, 1989.
- [Hamscher, 1988] W.C. Hamscher. Modeling Digital Circuits for Troubleshooting. *Artificial Intelligence*, vol. 51, pp. 223-271, 1991.
- [Nayak et al., 1992] P. P. Nayak, L. Joskowicz, and S. Addanki. Automated Model Selection using Context-Dependent Behaviors. *Proc. 10th AAAI*, San Jose, CA, pp. 710-716, July 1992.
- [Rosenberg and Karnopp, 1983] R.C. Rosenberg, and D.C. Karnopp. *Introduction to Physical System Dynamics*, McGraw-Hill, 1983.
- [Top and Akkermans, 1991] J. Top and H. Akkermans. Computational and Physical Causality. *Proc. 12th IJCAI*, Sydney, Australia, pp. 1171-1176, 1991.
- [Yu and Biswas, 1992] X. Yu and G. Biswas. A method for Diagnosis of Continuous-valued Systems. *Working Papers - Third Intl. Workshop on Principles of Diagnosis*, Rosario, WA, pp. 57-66, October 1992.
- [Yu, 1992] X. Yu. Multi-level Reasoning and Diagnosis for Complex Continuous-valued Systems. *Ph.D. Thesis*, Dept. of Computer Science, Vanderbilt University, Nashville, TN, 1992.