Diagnosis of Discrete Event Systems Using Ordered Binary Decision Diagrams

Janos Sztipanovits and Amit Misra

Measurement and Computing Systems Laboratory
Vanderbilt University
Nashville, TN 37235
email: [sztipaj | misra]@vuse.vanderbilt.edu
Phone : (615)322-3455
Fax : (615)343-6702

Abstract

Diagnostic methods for engineering systems are typically model-based: functional and/or fault models are used to diagnose the root cause of anomalies. Discrete models are abstractions of systems in a discretized input, output and state space. These models provide an accurate description for a broad category of systems (switching networks, digital systems, etc.) but can also be used for approximating the behavior of continuous or hybrid systems. Diagnosis tasks require the exploration of the discrete state space, which frequently leads to a combinatorial explosion of alternatives. This paper describes an approach, which transforms domain specific discrete failure propagation models or physical models into relational models, and performs diagnosis symbolically. The underlying theory is based on Ordered Binary Decision Diagram (OBDD) representations and related algorithms.

Introduction

Safe and reliable operation is a primary goal in many engineering systems. Equipment has been constructed and operated to satisfy strict safety standards, fail-safe and fool-proof design, ample design margin, inherent safety, automated emergency mechanism, instrumentation with redundancy, etc. During the last two decades, Fault Detection and Diagnosis (FDD) has also been introduced with the aim of minimizing the potential damage by early detection and warning. Although a considerable research activity has focused on the development of FDD techniques in specific system categories, the complex issues of system-wide FDD in large-scale, heterogeneous systems has many unsolved problems.

Our previous experience in the development and testing of real-time diagnostic and monitoring systems for aerospace [MIS92] [MIS94], electric utility [PAD91] and chemical process applications [KAR95] have shown that the selection of a suitable modeling discipline plays critical role in obtaining a practical, usable solution in this system category. Our research focused on problems with the following general characteristics:

- The plants are *complex, heterogeneous* systems. The size of the models is typically very large.
- The fault diagnostic system must handle *component faults* and *input disturbances*.
- The plant is a *dynamic system*. The ultimate goal of the real-time diagnostics is the prevention of the development of critical situations, therefore the diagnosis cannot be based on the observation of a new steady-state behavior after the fault occurred.

In order to satisfy these requirements, we have developed a multiple-aspect modeling approach. The structure of the plants was represented in a functional and a physical hierarchy [MIS94]. The behavior was modeled as a finite state temporal automata using timed failure propagation graphs [SZT93]. The diagnosis and diagnosability analysis algorithms were implemented as efficient graph algorithms leading to acceptable performance even in large-scale applications such as the International Space Station Alpha (ISSA) project [CAR96].

This paper discusses a new approach for diagnostics and diagnosability analysis using finite state automata, or Discrete Event System (DES) models. The essence of the proposed approach is to change system representation into relational models and use Ordered Binary Decision Diagrams (OBDD) based symbolic calculations [REB86] for diagnosis and diagnosability analysis. The result of using symbolic manipulations is a significant improvement in scalability and an opportunity for using discretized models for very large-scale problem domains. In Section 2 we present a brief overview of the relevant approaches. In Section 3 we summarize the OBDD representation. Section 4 discusses system modeling using the OBDD representation and Section 5 discusses the new category of algorithms in the OBDD framework.

Background

Model-based diagnostic systems work with a "model" (a suitable representation) of the system. The level of details in the models is determined by the required diagnostic resolution and mode of operation. Model-based diagnostic systems interpret the observed discrepancies in the context of the system model, and hence, there is no need to generate and store large fault-symptom libraries. There are primarily two approaches that have traditionally been used in diagnosis -- *functional modeling* and *fault modeling*.

Functional models (also called behavioral models) describe the "nominal" behavior of the system, i.e., how the system is supposed to behave when no faults are present. The level of abstraction in the functional models can vary from system to system, depending on the application -- from analytical models using state-space representation to qualitative models. The functionality can be described using just input/output relationships as in [KUI87], using a mathematical description, or using a set of connected components and causal sequences which give a description of how the system behaves [ESC87] [KW87] and [RD84] .

Using functional models to diagnose faults has its own problems, the foremost being the accuracy and validity of models, particularly in the presence of faults. This makes them usable primarily for stable systems with a well defined and simple domain theory, which maintains clear relationship between characteristics of physical components and system behavior. Furthermore, while the models are good for identifying the presence of a malfunction (using simulation or analytical methods), they are not necessarily helpful in diagnosing, i.e., locating the faulty component. This is because using functional models can lead to an explosion in the number of possible hypotheses, thereby rendering diagnosis intractable.

Fault models describe the expected system behavior in the presence of faults. This may be done as an extension of functional models to "off-nominal" regions, or by focusing only on the behavior in failure space. Fault models allow expression of *a-priori* knowledge about the typical ways in which components may fail or are designed to fail. Ishida et al. describe a topological approach to fault diagnosis in [IAT85]. Their fault model consists of faults, symptoms and an incidence matrix describing the binary relations between the faults and the symptom. The work done by Kumagai et. al. [KIT86], and Narayanan et al. [NV87] uses directed graphs to represent the relationships between faults and their manifestations.

Modeling methods used in diagnostics can also be categorized according to the selected formalism. Along this dimension, models can be static (steady state) or dynamic, continuous and discrete. *Static or dynamic continuous* models are frequently used in diagnosing incipient faults in systems using parameter identification methods [PFC89]. *Static or dynamic discrete* models describe system behavior in a discretized input/output and state space. Recently, there has been intensive research on diagnosis using discrete event system (DES) formalism [SLS96]. There are several functional modeling approaches that use continuous models to represent nominal behaviors, but the diagnostic reasoning is based on a discretized error space between the model and the observed behavior [BY93].

While discrete models are not suitable for diagnosing incipient failures in continuous systems, they have several advantages that explain their popularity. In diagnosing large-scale, heterogeneous systems, the level of discretization can be conveniently used to adjust modeling accuracy [SZT93] and control the cost of modeling. Discrete models are efficient in diagnosing abrupt component failures if the failure effect is pronounced. Most of the widely used fault analysis techniques, such as fault trees, FMEA analysis techniques are based on static, discrete models. Dynamic discrete models (DES or finite state automaton models) can provide quite an accurate description of system dynamics, therefore they have great significance in diagnosis. The general problem with discrete models in diagnosis and safety analysis is combinatorial explosion. Recently proposed techniques, such as in [SLS96], suffer from serious scalability problems, and cannot be used with complex discrete models, representing realistic systems.

The primary contribution of our research is to provide a formalism and a reasoning technique for static and dynamic discrete models which significantly improve scalability.

Ordered Binary Decision Diagrams (OBDD)

Diagnosis and fault analysis tasks with discrete models are formulated in terms of operations over finite domains. Combinatorial explosion is the result of the exponential increase in the number of discrete elements (states, hypotheses, etc.) during operations, which sooner or later makes the individual access to the elements unfeasible. By introducing a binary encoding , the individual elements, sets of elements, and relations among them can be expressed as Boolean functions. For example, the 2^{100}

states of a finite state automaton can be encoded with binary variables *{s(1),…,s(100)}* forming a binary state vector ***s***. The Boolean functions

$$f_1[s(1),\ldots,s(100)]=s(1)'\wedge s(23)\wedge s(99) \text{ and}$$
$$f_2[s(1),\ldots,s(100)]=s(1)\wedge s(22)\wedge s(89)$$

represent two subsets, S1 and S2, of the 2100 states including 297 elements each. The set $S3=S1\cup S2$ can be derived symbolically as the disjunction of the two Boolean functions:

$$f_3[s(1),\ldots,s(100)]= f_1[s(1),\ldots,s(100)]\vee$$
$$f_2[s(1),\ldots,s(100)]= s(23)\wedge s(99)\vee s(22)\wedge s(89)$$

without the need to enumerate and compare the individual elements - which would be a formidable task otherwise. In general, using Boolean function representations, we can express operations and algorithms in diagnosis and safety analysis in symbolic form, by means of symbolic Boolean function manipulations.

OBDD-s provide a symbolic representation for Boolean functions in the form of directed acyclic graphs [REB86]. They are a restricted, canonical form version of Binary Decision Diagrams (BDD) [LEE59]. Bryant [REB89] described a set of algorithms that implement operations on Boolean functions as graph algorithms on OBDDs. Taking advantage of the efficient symbolic manipulations, researchers have solved a wide range of problems in hardware verification, testing, real-time systems, and mathematical logic using OBDDs that would have been otherwise impossible due to combinatorial explosion. Symbolic model checking is extensively used in hardware design (see, e.g., [BCL93]), and has shown to be efficient in state space sizes 10^{120} and beyond.

Discrete Event System and Relational Models for Diagnosis

A broad category of systems, such as digital hardware, switching, distribution, and communication networks, etc., are naturally modeled as DES. Beyond this, the behavior of continuous systems can also be approximated with DES models.

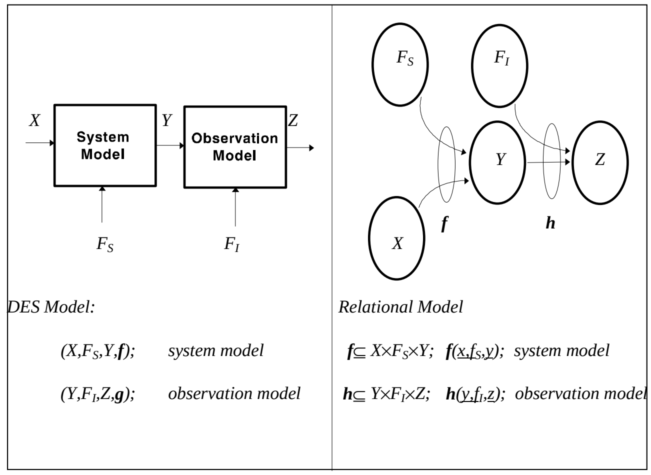
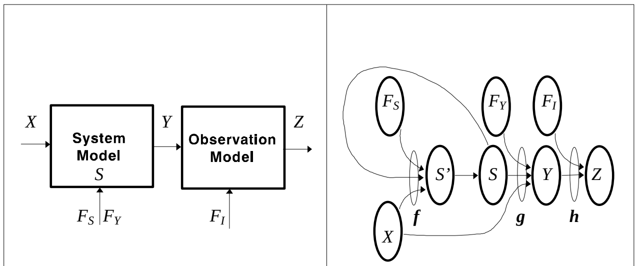


Figure 1: DES and relational models of static systems

The DES model of a static system (system without memory) is shown on the left side of Figure 1. Since our purpose with modeling is sensor-based diagnosis, the model is divided into a *System model* and an *Observation model*. The system model represents a mapping between the elements of the input set X, fault set F_S, and the elements of the output set Y: $\boldsymbol{f}: X\times F_S \rightarrow Y$. In this approach, the component faults are considered as additional inputs to the system. It is also possible to model the abnormal (out of range) inputs as elements of the X input set, creating a ‘normal’ and ‘faulty’ partition in X. The observation model describes a mapping between the Y output set of the system, and the actually observed outputs, Z; $\boldsymbol{h}: Y\times F_I \rightarrow Z$. The set F_I collects the observation faults (or instrumentation faults) that can potentially corrupt the observations. We assume that both ***f*** and ***h*** can represent many-to-many mapping, i.e. they are not necessarily functions. This allows non-deterministic modeling, which is particularly important in large-scale systems. Non-deterministic constructs allow the expression of uncertainties in the outcome of inputs due to noises or non-modeled behaviors.

The right side of Figure 1 shows the equivalent *Relational Model* of a static system. In the relational model, the ***f*** and ***h*** mappings are considered to be the $\boldsymbol{f}\subseteq X\times F_S\times Y$ and $\boldsymbol{h}\subseteq Y\times F_S\times Z$ relations. The significance of the relational representation is that it directly shows that the models can be re-written as Boolean functions by introducing some binary encoding for the sets $X\times F_S\times Y$ and $Y\times F_I\times Z$. The Boolean functions $\boldsymbol{f}(\underline{x},\underline{f_S},\underline{y})$ and $\boldsymbol{h}(\underline{y},\underline{f_I},\underline{z})$ evaluate to *true* for those elements of $X\times F_S\times Y$ and $Y\times F_I\times Z$ (encoded by the Boolean vectors $(\underline{x},\underline{f_S},\underline{y})$ and $(\underline{y},\underline{f_I},\underline{z})$), which are related by the ***f*** and ***h*** relations.

The DES model and relational model of a dynamic system is somewhat more complicated, as shown in Figure 2.



DES Model:

$(X,F_Y,F_S,S,\Gamma,\boldsymbol{f},s_0,Y,\boldsymbol{g})$; *system model*

$(Y,F_I,Z,\boldsymbol{h})$; *observation model*

Relational Model:

$\boldsymbol{f} \subseteq X \times F_S \times S \times S'$; *transition model*
$\boldsymbol{f}(\underline{x},\underline{f_s},\underline{s},\underline{s'})$;
$\boldsymbol{g} \subseteq S \times F_Y \times X \times Y$; *output relation*
$\boldsymbol{g}(\underline{s},\underline{f_y},\underline{x},\underline{y})$
$\boldsymbol{h} \subseteq Y \times F_I \times Z$; *observation relation*
$\boldsymbol{h}(\underline{y},\underline{f_I},\underline{z})$

Figure 2: DES and relation models for dynamic systems

In dynamic systems, the DES model is the $(X,F_Y,F_S,S,\Gamma,\boldsymbol{f},s_0,Y,\boldsymbol{g})$ finite state automata (see e.g. [CC93], where:

X is the input event set,

F_S, F_Y are the sets of transition faults and output faults, both considered to be input events,

S is the state set,

(s) is a set of feasible or enabled events, defined for all $s \in S$ with $\Gamma(s) \in X$,

$\boldsymbol{f}$ is a state transition function, $\boldsymbol{f}: X \times F_S \times S \rightarrow S'$, defined only for $x \in \Gamma(s)$ when the state is s,

s_0 is the initial state,

Y is the output set, and

$\boldsymbol{g}$ is an output function, $\boldsymbol{g}: X \times F_Y \times S \rightarrow Y$, defined only for $x \in \Gamma(s)$ when the state is s.

In order to model partial observations of the state trajectory independently from the outputs of the dynamic system, we use again the $\boldsymbol{h}: Y \times F_I \rightarrow Z$ observation model describing the mapping between the Y outputs, F_I instrumentation faults, and the Z observations. The finite state automaton formalism also allows the representation of non-deterministic state machines, which is an important requirement for modeling large-scale systems.

The right side of Figure 2 shows the equivalent relational models. Similarly to the static system models, the $\boldsymbol{f}(\underline{x},\underline{f_s},\underline{s'},\underline{s})$, $\boldsymbol{g}(\underline{s},\underline{f_y},\underline{x},\underline{y})$ and $\boldsymbol{h}(\underline{y},\underline{f_I},\underline{z})$ functions are the Boolean function representations of the relations over the binary encoded Boolean vectors $\underline{x},\underline{f_s},\underline{s'},\underline{s},\underline{f_y},\underline{y},\underline{f_I}$, and $\underline{z}$.

Although it is not the purpose of this discussion, it is worthwhile to note that DES (or relational) models preserve composability and can be constructed in a modular fashion using either component oriented modeling approach [SLS96] or process-oriented modeling approach [SZT93].

Diagnostic reasoning using OBDD-s

The application of OBDD-s for diagnostic reasoning includes the following steps:

1. Mapping the DES or relational models into OBDD-s: This step can be completed automatically. In the framework of the Multigraph Architecture (MGA), the discrete behavioral models used for diagnosis or safety analysis are usually domain specific [SZT95]. The domain specific models can be translated into an OBDD representation using model interpreters. An example for converting relay logic diagrams into OBDD representations is described in [SZT96].
2. Diagnosability and safety analysis: Diagnosability and safety analysis are accomplished symbolically, using the OBDD representations. Diagnosability and safety criteria are expressed in the form of logic relationships on the discrete state trajectories generated by the models, and these relationships are checked using OBDD algorithms.
3. Diagnosis: In the MGA framework, the diagnostic software is built in two steps. First, a generic run-time support is created. The run-time support includes a diagnostic engine implemented with OBDD algorithms. Second, the software synthesis component of the MGA (one particular form of model interpreters): (a) configures the run-time system using the MGA computational model, and (b) synthesizes the OBDD data structures for the models.

The core components of the diagnostic reasoning are the algorithms that compute the sets using the relational models.

Diagnostic reasoning in static systems

Although it seems to be restrictive, static system models are widely used in engineering practice. Fault trees, AND-OR graphs, most of the rule-based models can be considered as some form of the static models. Using the relational model formulation described above, the following calculations can be performed using OBDD algorithms.

a) Observed output calculation

Given the set of input X, and the sets of faults F_S and F_I, the set of outputs Y and the set of observations Z can be calculated by the following formulae:

$$Y=\boldsymbol{f}(X,F_S)=\{y|\exists x, f_S[(x\in X)\wedge(f_S\in F_S)\wedge(x, f_S,y)\in \boldsymbol{f}]\}; \quad (1)$$
$$Z=\boldsymbol{h}(Y,F_I)=\{z|\exists y,f_I[(y\in Y)\wedge(f_I\in F_I)\wedge(y, f_I,z)\in \boldsymbol{h}]\};$$

The required variable quantification and logic operations are executed symbolically. The resulting $\boldsymbol{f}(X,F_S)$ and $\boldsymbol{h}(Y,F_I)$ mappings propagate the elements of the input sets 'forward' in the relational model.

b) Diagnosis

Solution of the diagnosis problem requires the calculation of the hypothesis set D, defined on $X \times F_S \times F_I$ given a set of observations Z:

$$\boldsymbol{d}(Z)=\{x,f_S,,f_I|\exists y,z[(z\in Z)\wedge(y, f_I,z)\in h\wedge(x, f_S,y)\in \boldsymbol{f}]\} \qquad (2)$$

The diagnostic mapping is derived as a combination of the functional composition of the relations $\boldsymbol{f}$ and $\boldsymbol{h}$, and variable quantification. The $\boldsymbol{d}(Z)$ mapping propagates the Z observations 'backwards' to obtain the set of admissible $d\in X\times F_S\times F_I$ elements forming together the diagnosis. The result of the diagnosis, the D hypothesis set, includes all of those $d\in X\times F_S\times F_I$ hypotheses that are consistent with the Z observations. Those elements for which $f_S=\{0\}$ and $f_I=\{0\}$, the corresponding x input values represent inputs for fault free operations. It is interesting to note that the symbolic computation derives in one step the symbolic representation (i.e. the OBDD) of the full $D(Z)$ hypothesis set, including all of the multiple fault combinations.

c) Safety analysis

Safety analysis requires the testing of the models against selected safety criteria. Here we demonstrate the use of symbolic model checking in one particular problem, to test distinguishability of faults. A system and its observation model provide *single-fault distinguishability*, if all possible observations are unique to the single faults. Let $i_{fs}\in X\times F_S\times f_I$ be inputs to the system with a single fault, i.e., each i_{fs} includes exactly one f_S or f_I faults. The condition for single fault distinguishability can be expressed symbolically in the following manner:

$$\boldsymbol{d}^\circ(\boldsymbol{h}^\circ \boldsymbol{f})\,(i_{fs})= i_{fs}\,\boldsymbol{;} \quad \forall\; i_{fs}\in X\times F_S\times F_I \qquad (3)$$

That is, the diagnosis relation $\boldsymbol{d}$ is the inverse of the composition of the $\boldsymbol{h}^\circ \boldsymbol{f}$ relations for all single fault inputs. Similar symbolic expressions can be derived for multiple fault distinguishability, fault masking, fault detectability and other safety characteristics of the models.

Diagnostic reasoning in dynamic systems

The primary difficulty in dynamic systems is that the diagnosis must be performed by using the partial observations of the system trajectory. Particularly in the case of non-deterministic models, the diagnostic reasoning has acute scaling problems. The symbolic form of the algorithms has similar form to that of the static systems.

a) Observed output calculation

Given the set of input X and the sets of faults F_S, F_Y and F_I, the sets of next states S', outputs Y and observations Z can be calculated by the following expression:

$$S=\boldsymbol{f}(X,F_S,S)=\{s'|\exists x,f_S,s[(x\in X)\wedge(f_S\in F_S)\wedge(s\in S)\wedge(x,f_S,s,s')\in \boldsymbol{f}]\};$$
$$Y=\boldsymbol{g}(X,F_Y,S)=\{y|\exists x,f_Y,s[(x\in X)\wedge(f_Y\in F_Y)\wedge(s\in S)\wedge(x,f_Y,s,y)\in \boldsymbol{g}]\};$$
$$Z=\boldsymbol{h}(Y,F_I)=\{z|\exists y,f_I[(y\in Y)\wedge(f_I\in F_I)\wedge(y, f_I,z)\in \boldsymbol{h}]\}; \qquad (4)$$

The symbolic expression above calculates a one-step propagation forward in the state automata. The result is a new set of possible states S', and the related Y outputs and Z observations. The set of reachable states can be found by computing the transitive closure of $\boldsymbol{f}$ using fixed-point calculation, i.e. to find an S for which $\boldsymbol{f}(X,F_S,S)=S$. This is particularly important in safety analysis, where safety requirements frequently impose constraints on the reachability set of the state automata.

b) Diagnosis

There are several ways to perform diagnosis in dynamic systems. In off-line diagnosis, observations are collected and analyzed independently from the operation of the system. In on-line diagnosis, the diagnostic system runs parallel with the system, and refines the hypothesis set as new observations are collected. As an example, we describe an on-line diagnosis method using symbolic expressions. The on-line diagnostic system re-calculates the hypothesis set $D\subset X\times F_S\times F_Y\times F_I\times S$ whenever a new observation event(s) arrives:

$$D_{j+1}=\boldsymbol{d}(D_j,Z_{j+1})=$$
$$\{x,f_S,,f_Y,f_I,s'|\exists s,y,z[(z\in Z_{j+1})\wedge(y,f_I,z)\in \boldsymbol{h}\wedge(f_I\in F_{I,j})\wedge$$
$$\wedge(\underline{s},\underline{f_y},\underline{x},\underline{y})\in \boldsymbol{g}\wedge(x\in X_j)\wedge(f_Y\in F_{Y,j})\wedge(x, f_S,,s,s')\in \boldsymbol{f}\wedge(f_S\in F_{S,j})]\} \quad (5)$$

The $(x\in X_j)$, $(f_S\in F_{S,j})$, $(f_Y\in F_{Y,j})$, and $(f_I\in F_{I,})$ conditions assume that during the observation the faults (including possibly faulty x inputs) are persistent. If this assumption cannot be made, the conditions must be eliminated from the reasoning. Receiving newer and newer observations, the diagnostic algorithm will converge to a hypothesis set which includes all possible explanations for the observed trajectory. These explanations extend to the multiple fault hypotheses as well.

Criteria for fault distinguishability can be derived the same way as it was described for static systems.

It is important to mention that the diagnostic algorithms of *(2)* and *(5)* (and all of the other expressions above) are computed symbolically. Symbolic computation here means that all of the sets and relations are represented as OBDDs and the logic and quantification operators are executed by manipulating the OBDDs by means of a small set of efficient algorithms [REB86].

Summary

Diagnosis and safety analysis using discrete event system models is a difficult problem due to the combinatorial explosion of the state and event sets derived during the analysis. The problem can be significantly reduced by using relational models and OBDD representations. Although symbolic manipulations offer tremendous advantage in diagnostic reasoning, scalability remains an important issue in analyzing large-scale systems. The size of OBDD data structures is sensitive to the ordering of the Boolean variables, which indicates the need for the development of good heuristics while mapping the models into OBDD representations. Our experiences with diagnosing a discrete controller implemented with a relay logic network [BSS95] has shown the feasibility of the approach.

References

[BCL93] Burch, J. R., Clarke, E.M., Long, D. E., "Symbolic Model Checking for Sequential Circuit Verification," Technical Report, CMU-CS-93-211, Carnegie Mellon University, July 15, 1993.

[BSS95] Bapty, T., Sztipanovits, J., Scott, J., Ball, B., "Plant-Wide Preventative Maintenance and Monitoring," Final Report for AFOSR, Measurement and Computing Systems Laboratory, Vanderbilt University, December, 1995.

[BY93] Biswas, G., Yu, X. "A Formal Modeling Scheme for Continuous Systems: Focus on Diagnosis," *Proc. of the IJCAI-93*, pp. 1474-1479, France, 1993.

[CAR96] Carnes, R., Misra, A., Sztipanovits, J., "Model-Integrated Toolset for Fault Detection, Isolation and Recovery (FDIR)" *Proc. of the 1996 IEEE Symposium and Workshop on Engineering of Computer-Based Systems*, pp. 356-363, Friedrichshafen Germany, March, 1996

[CC93] Cassandras, C. G., *Discrete Event Systems*, IRWIN Inc. 1993.

[ESC87] Scarl, E., et al., "Diagnosis and Sensor Validation through Knowledge of Structure and Function," *IEEE Trans. Syst., Man and Cybernetics*, vol. SMC-17, no. 3, May./June 1987, pp. 360-368.

[IAT85] Ishida, Y., Adachi, N. and Tokumaru, H., "A Topological Approach to Failure Diagnosis of Large-Scale Systems," *IEEE Trans. Syst., Man and Cybernetics*, vol. SMC-15, no. 3, May./June 1985, pp. 327-333.

[KAR95] Karsai, G., Padalkar, S., Franke, H., Sztipanovits, J., "A Practical Method for Creating Plant Diagnostic Applications," *Integrated Computer-Aided Engineering* to be published in 1996.

[KIT86] Kumagai, N., Ishida, Y. and Tokumaru, H., "A Knowledge Representation for Diagnosis of Dynamical Systems," {\em *IFAC Real-Time Programming*}, Lake Balaton, Hungary, 1986, pp. 27-37.

[KUI87] Kuipers, B., "Qualitative Simulation as Causal Explanation," *IEEE Trans. Syst., Man and Cybernetics*, vol. SMC-17, no. 3, May./June 1987, pp. 432-444.

[KW87] de Kleer, J. and Williams, B., "Diagnosing Multiple Failures," *Artificial Intelligence*, vol. 32, 1987.

[LEE59] Lee, C. Y., "Representation of Switching Circuits by Binary Decision Programs," *Bell System technical Journal* pp. 985-999, 1959.

[MIS92] Misra, A., Sztipanovits, J., Underbrink, A., Carnes, R., Purves, B., "Diagnosability of Dynamical Systems," *Proc. of the Third International Workshop on Principles of Diagnosis*, pp. 239-244, Rosario, WA 1992.

[MIS94] Misra, A., Sztipanovits, J., Carnes, R., "Robust Diagnostic System: Structural Redundancy Approach," *Proc. Of the SPIE International Symposium on Knowledge-Based Artificiaal Intelligent Systems in Aerospace Systems*, Orlando Florida, April 5-6, 1994.

[NV87] Narayanan, N. H. and Vishwanadham, N., "A Methodology for Knowledge Acquisition and Reasoning in Failure Analysis of Systems," *IEEE Trans. Syst., Man and Cybernetics*, vol. SMC-17, no. 2, Mar./Apr. 1987, pp. 274-288.

[PAD91] Padalkar, S., Karsai, G., Biegl, C., Sztipanovits, J., Okuda, K., Miyasaka, N., "Real-Time Fault Diagnostics," *IEEE Expert*, pp. 75-85, June, 1991.

[PFC89] Patton, R., Frank, P., Clark, R., *Fault Diagnosis in Dynamic Systems: Theory and Applications* Prentice Hall International, 1989

[RD84] Davis, R., "Diagnostic Reasoning Based on Structure and Behavior," *Artificial Intelligence*, vol. 24, 1984.

[REB86] Bryant, R. E., "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, C-35(8), 1986.

[SLS96] Sampath, M. et al., "Failure Diagnosis Using Discrete-Event Models," *IEEE Transactions on Control Systems Technology*. Vol. 4, No. 2, pp. 105-124, March, 1996.

[SZT93] Sztipanovits, J., Carnes, R., Misra, A., "Finite-State Temporal Automata Modeling for Fault Diagnosis," *Proc. Of the 9th AIAA Conference on Computing in Aerospace*, San Diego, CA, October, 1993.

[SZT95] Sztipanovits, et al., "MULTIGRAPH: An Architecture for Model-Integrated Computing" *Proc. of the IEEE ICECCS'95* Ft. Lauderdale, Florida, Nov. 6-10. 1995.

[SZT96] Sztipanovits, J., "Safety Analysis and Diagnosis in a Model-Integrated Computing Framework," *Proc. of the Workshop on Model-Integrated Computing*, DuPont, Old Hickory, TN June 3-4, 1996. URL: http://mcsl.vuse.vanderbilt.edu/pub/mgadocs/micws.ps.Z