FAULT DIAGNOSIS OF PROCESS PLANTS USING CAUSAL MODELS

by

BERNARD L. PALOWITCH JR.

B. S. University of Pittsburgh
1982

B. A. University of Pittsburgh
1982

Submitted in partial fulfillment for the requirements
for the degree of

DOCTOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1987

Signature of Author_____
Department of Chemical Engineering
June 10, 1987

Certified by_____
Dr. Mark A. Kramer
Thesis Supervisor

Accepted by_____
Dr. Robert C. Armstrong
Chairman, Department Graduate Committee

FAULT DIAGNOSIS OF PROCESS PLANTS USING CAUSAL MODELS

by

BERNARD L. PALOWITCH JR.

Submitted to the School of Engineering
on June 10, 1987 in partial fulfillment of the requirements
for the degree of Doctor of Science.

ABSTRACT

The design of a computer-based aid to assist operators in the diagnosis of process failures is investigated. The objectives of this thesis are to develop suitable representations to characterize the physical process and the knowledge necessary for diagnosis, and to develop a general strategy for evaluating the representations. The focus is on qualitative models and model-based diagnostic reasoning.

The process representation used is the causal directed graph (digraph), in which nodes represent process variables and parameters, and arcs represent the causal interactions between them. Although several authors have used causal models for fault diagnosis, none have adequately characterized the causal interactions or described how to develop causal models. In this work, I show how to derive the causal digraph for a set of design equations, present guidelines for developing causal digraphs for fault diagnosis, and develop context-independent causal models for standard system components.

The diagnostic strategy incorporates multiple knowledge representations and problem-solving approaches. Graph search, simulation, qualitative constraints, and heuristics are used within a hypothesis generation and test framework. During candidate generation, nodes causally upstream from abnormal measurements are identified from which fault propagation would cause the observed deviations. Nodes that are locally plausible (i.e., have consistent causal paths to all abnormal measurements) but not consistent with other known information are eliminated during candidate testing. A list of faults is generated from a table that relates the deviations of digraph nodes to specific faults.

Examples from a diagnostic system prototype are presented and implementation issues are discussed.

Thesis Supervisor: Dr. Mark A. Kramer
Assistant Professor of Chemical Engineering

ACKNOWLEDGEMENT

I wish to thank several people who have had a significant influence on me during my years of study at the Massachusetts Institute of Technology:

My advisor, Dr. Mark A. Kramer, was an exceptional mentor. His guidance and direction was sincerely appreciated. Valuable contributions were made by the members of my thesis committee: Dr. Lawrence B. Evans and Dr. George Stephanopoulos of the Department of Chemical Engineering, Dr. Thomas B. Sheridan of the Department of Mechanical Engineering, Dr. Randall Davis of the M.I.T. Artificial Intelligence Laboratory, and Mr. Lowell B. Hawkinson of Gensym Corporation (Cambridge, MA).

I am indebted to Dr. David M. Prett of Shell Development Company (Houston, TX) and Dr. Robert L. Moore of Gensym Corporation for the opportunity to work on artificial intelligence research in industrial settings.

I acknowledge the members of my research group, specifically Jorge Leis, Olayiwola Oyeleye, and F. Eric Finch, as well as the members of the research groups of Dr. Stephanopoulos and Dr. Evans, for the stimulating discussions and helpful suggestions on this work.

Outside of academics, I enjoyed many close friendships. I am grateful to classmates Kevin Joback and Thomas Gandek, and outside of chemical engineering, Stuart Brown, Anne St. Onge, and Frank Fairbairn for many memorable times and their support during the more arduous moments.

My last year and a half at M.I.T. would not have been nearly as happy without the love and support of my fiancee, Karen Cloutier.

Finally, I extend gratitude to my parents, who cultivated in me intellectual curiosity, a desire for academic excellence, and encouraged the development of all my talents (even through swimming lessons).

This research was supported by the National Science Foundation, the Shell Foundation, and the Arthur D. Little Foundation. I received a National Science Foundation Graduate Fellowship (1982-1985), Edwin R. Gilliland Fellowship (1982), and a Lambda Chi Alpha Educational Foundation Scholarship (1982), and N.S.F. Fellowship supplements from the Arthur D. Little Foundation, Atlantic Richfield Corporation, and the E. I. Du Pont de Nemours Corporation. I am grateful to these sponsors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

OVERVIEW

This thesis addresses the design of a computer-based diagnostic aid for real-time process plant fault diagnosis. Specifically, the research investigates model-based diagnostic reasoning. The thesis objectives are (1) to develop suitable representations to characterize the process and the knowledge necessary for diagnosis, and (2) to develop a general strategy for evaluating the representations.

Three themes, which parallel human diagnostic reasoning, underlie these two objectives.

General versus Context-Specific Knowledge

The research focuses on those elements of the diagnostic aid that are general, or independent of the particular context. The context-independent elements, termed the core knowledge, include general strategies for fault diagnosis, models characterizing the behavior of general classes of process equipment, general rules that relate specific failures to process variable deviations, and general knowledge representation formats for the storage of process data and equipment design specifications. Because the behavior of general types of process equipment is identical across plant sites, models of system components are used to form the base of the process representation.

The separation of diagnostic knowledge into core and plant-specific sections adds modularity to the diagnostic aid. Because the core knowledge is transportable between plant sites, implementation involves adding only the context-specific information.

Qualitative Reasoning

The research focuses on developing knowledge representations and a diagnostic procedure that are qualitative. Qualitative reasoning, which involves qualitative component models and qualitative values assigned to the model's parameters and state variables, is sufficient for diagnosing a majority of process failures. Qualitative reasoning overcomes the inherent limitations of quantitative representations.

Multiple Problem-Solving Strategies

The research focuses on incorporating multiple problem-solving strategies into a general solution procedure. Graph search, simulation, heuristics, and qualitative constraints are used within a hypothesis generation and test framework. The use of several strategies improves diagnostic resolution because the additional strategies add knowledge which eliminates infeasible fault candidates.

In summary, the goal of this research is to develop the core knowledge necessary for model-based process plant fault diagnosis. The major constituents of the core knowledge are the plant-independent, qualitative, component-based models for chemical processes and process equipment, and the general solution procedure for evaluating the qualitative models. The research is summarized along these two principal elements.

Knowledge Representation—The Causal Digraph

Description

The causal directed graph (digraph) is a network which represents the cause-and-effect interactions between the parameters and state variables of a physical system. Nodes in the causal digraph represent process variables, parameters, and combinations of these terms. Digraph arcs represent the causal interactions between them. More specifically, a causal arc represents a physical process or mechanism by which a change in one parameter, represented by the node at the arc's tail, is transmitted to, and causes a change in, the parameter represented by the node at the arc's head. The graph is directed because each causal interaction is directed—a cause produces, or is the reason for, the effect. Causal interactions are local interactions, i.e., they exist between parameters and process variables that are in some sense adjacent on a given level of model detail. Reasoning about two nonadjacent nodes is done by constructing a path of digraph arcs between them. The directed path represents how a change in the initial node is propagated to the terminal node. Changes are represented by assigning qualitative values to the digraph nodes.

The design values of process variables, the values of process parameters, and the values of real-time process measurements may be necessary to specify the existence of an arc and the values of its attributes. For a causal arc to accurately characterize a causal influence, the set of conditions associated with the arc must be satisfied.

Summary of Results

Research contributions in developing qualitative representations to characterize chemical processes and process equipment are summarized.

- **Distinction Between the Causal Model and How the Causal Representation is Evaluated**

For a given physical context, the causal interactions are independent of how the qualitative representation is evaluated. This differs from other research in causal modeling because the causal relationships generated by other authors are a function of how a set of equations that model the system is solved. de Kleer and Brown [1984] [1986] solve a set of constraint equations, called confluences. Causal relationships are identified by propagating the effects of a disturbance through the network of constraints. The causal interactions generated depend on the sequence in which the confluences are solved. Similarly, the causal ordering of Iwasaki and Simon [1986a] [1986b] depends on identifying self-contained subsystems in the system of structural equations. Establishing a causal ordering involves finding subsets of variables whose values can be computed independently of the remaining variables. The values of the variables in the subset are used to reduce the system to a smaller set of equations. Specifying different exogenous variables will yield different causal relationships. In both methods, causality becomes identical to the progression of substitutions into the system of equations.

The causal relationships developed here do not depend on the manner in which the causal representation is evaluated. Rather, each physical mechanism generates a specific set of causal arcs. This set is completely independent of any other mechanisms that may exist within the physical

system, and how the values of system parameters are determined. This differs from Iwasaki and Simon, who explicitly state that the knowledge of a single mechanism in isolation does not imply a directed causal interaction between two variables ([1986a], p. 13).

By interpreting the causal digraph as a static representation of the underlying physical mechanisms, there is a clear distinction between the causal digraph as a data structure and the method for assigning values to the nodes to characterize a specific system state. The causal arcs in the digraph are fixed for a given physical context. As long as the context is unchanged, the digraph remains valid. On the other hand, the qualitative values assigned to the nodes, which represent the current state of the system, can vary. Events (deviations in the physical system represented by the qualitative values '+' and '-' assigned to the nodes) are distinguished from the processes by which these events propagate through the system (represented by the digraph arcs).

□ Definition of a Qualitative Value

Reasoning using the causal digraph is accomplished by assigning qualitative values to the digraph nodes. For each continuous variable and parameter, a reference value or range is selected, which represents the expected value or range that characterizes the process variable or parameter under normal operating conditions. The endpoints of the normal range break the value continuum into three regions. In this work, the three qualitative values low '-', normal '0', and high '+' represent the deviation of the numerical value away from the reference. Mathematically, the qualitative value of x, denoted [x], defined as the deviation from a context-specific reference point x_0, is given as $[x] = \mathrm{sgn}(x - x_0)$. When the reference is a range, the qualitative value of x is defined as

$$
\begin{aligned}
[x] &= \text{'-'} \text{ when } x < x_0^- \\
[x] &= \text{'0'} \text{ when } x_0^- \leq x \leq x_0^+ \\
[x] &= \text{'+'} \text{ when } x_0^+ < x,
\end{aligned}
$$

where x_0^- and x_0^+ are the lower and upper range endpoints, respectively. Other authors (e.g., de Kleer [1984], p. 211; Forbus [1984], pp. 95-96;

Williams [1984], pp. 289-290) define [x] = sgn(x). The definition used here is advantageous for two reasons. First, a qualitative value does not represent an absolute quantity. Rather, it depends on the expected operating point. A 'high' temperature in one context may be 'low' in another. The definition as a deviation from a reference captures this context-specific nature. Second, the expected value of a process variable is generally not fixed, but fluctuates within a normal band. The introduction of a normal range, instead of a single reference point, captures this variability.

The reference value or range is stationary or, for a process moving between states, changing with time.

Discrete variables are not modeled explicitly as digraph nodes. A different causal digraph is necessary to model a change in the value of a discrete variable.

▫ Attributes of a Causal Interaction

Causal arcs have a set of attributes that characterize the causal interaction. The attributes sign, magnitude, and time are used in this work. The sign attribute characterizes the direction of deviation of the process variables at the arc's initial and terminal nodes. The qualitative sign '+' indicates that the variables deviate in the same direction; the sign '-' indicates that they vary in opposite directions. The magnitude attribute specifies the strength of the causal interaction. The qualitative value '1' is assigned if any deviation of the causally upstream variable within its entire range of variation can cause the causally downstream variable to deviate outside of its normal range. The magnitude attribute '0' is assigned if all disturbances of the causally upstream variable cannot cause the causally downstream variable to deviate. The time attribute specifies the propagation time for the effect to be transmitted from the initial node to the terminal node along the causal arc. The qualitative value '0' is defined as zero delay (instantaneous transmission); the value '1' indicates positive delay time.

▫ Procedure to Specify Causal Arcs and Arc Attributes

The mathematical relationships and equations used to quantitatively model a physical process do not contain the information needed to specify the relationships between the process variables for causal modeling. Neither do qualitative relationships that do not explicitly represent the directionality of the causal interaction. Mathematical equations specify equality relationships between sets and combinations of parameters, but they contain no information on how changes in an individual process variable or parameter directly affect other system variables. Causal interactions can only be specified from an understanding of the fundamental physical principles and mechanisms that the equations represent.

Causal digraphs were developed from quantitative design equations by identifying and interpreting the underlying physical mechanisms behind each equation. Digraph construction was simplified because, for the example processes investigated, the design equations could be classified into four categories: driving force equations, balance equations, functional relationships, and algebraic equalities. For each category, a standard set of procedures specify the causal arcs and their sign attribute.

▫ Limitations of the Causal Digraph

The causal digraph representation is limited in its ability to fully characterize a system. Among these limitations are:

1. The causal digraph is not unique. Therefore, several different causal digraphs can be developed to represent the same physical system. The differences between them arise from the parameters chosen to be included in the model.

2. The causal digraph does not contain information about discontinuities that may arise in the physical system. Thus, they cannot explicitly handle discrete changes when they occur. Discontinuities require a different set of quantitative equations, and hence, a different digraph, to accurately model the system.

3. Ambiguities may arise when determining the qualitative values of model parameters. On a given level of detail, an arc and its sign attribute serve as a constraint on the behavior between its initial and terminal nodes. At an increased level of detail, these adjacent digraph nodes may become nonadjacent, and multiple paths with opposite net sign may exist between them. When a causal arc at a lower level of detail is a global constraint at a greater level of detail, the constraint is lost on the level of greater detail because the digraph only represents local interactions. Without the constraint, ambiguities arise because the digraph contains no information to specify which path is dominant. When several plausible interpretations for global behavior exist, the causal digraph is not deterministic.

Retaining global knowledge, if it is known, can aid in reducing the number of spurious interpretations. Two methods, a hierarchy of digraph models and qualitative equalities, are suggested.

▫ Guidelines for Developing Digraphs for Fault Diagnosis

My purpose for developing causal digraphs is to construct diagnostic systems. The digraph used, then, should be the one that is most suited for diagnosis. A digraph is suitable for fault diagnosis if it (1) contains a single node representing the primary effect of the fault for every fault desired to be diagnosed, and (2) represents the physical system on the level of detail that minimizes the number of incorrect faults and maximizes the resolution between faults. These objectives suggest the following guidelines for developing and modifying causal digraphs for fault diagnosis.

1. Identify the faults to be diagnosed prior to developing the causal digraph, so that every process variable or parameter that represents the primary effect of a fault is included in the set of quantitative design equations.

2. Greater knowledge about the system improves fault resolution, decreases the number of incorrect fault candidates identified, and retains the fundamental causal relationships. Therefore,

- ¶ Set up quantitative equations for each physical mechanism, unit or piece of equipment, rather than for larger sections of the process.

- ¶ Do not eliminate process variables through the substitution of equations.

3. Modify the causal digraph to handle faults that change the form of the design equations. The digraph must contain the causal arcs that model the behavior of the system when the fault is present.

▫ Context-Independent Component Models

As previously mentioned, the design values of process variables, the values of process parameters, and the values of real-time process measurements may be necessary to specify the existence of an arc and the values of its attributes. For a digraph arc to accurately characterize a causal interaction, the set of conditions associated with the arc must be satisfied. The collection of causal interactions for a system component, together with the rules that specify when the interactions exist, constitute the context-independent component model. The causal digraph is the output from the causal model for a given input set of context-specific parameters.

In a component causal model, the rules for specifying all the causal pathways in the component are grouped into the component rule base. Rule antecedents explicitly state the assumptions and conditions necessary for the existence of the directed arcs. The antecedents reference design values and process measurements related to the specific unit. Rule consequents are the individual causal paths and values of the attributes that are valid for the given context.

Associated with each component rule base is a general component database that stores the design specifications and relevant process

measurements that are required by the rule antecedents. For the digraph to accurately characterize its physical system, the assumptions used to develop the digraph must match the actual physical context. By specifying a set of context-specific numerical and discrete parameters, the rules generate a specific causal digraph for the particular process unit.

Causal models allow the underlying physical behavior of a component to be specified independently from the particular context in which the component will function. A library of context-independent component models facilitates diagnostic system installation.

Strategy for Fault Diagnosis Using the Causal Digraph

Terminology

A fault is any event that causes one or more process variables or parameters to deviate outside the range that represents their normal operation. Therefore, a fault causes the qualitative values of those variables to change from normal '0' to either high '+' or low '-'.

A valid node is a node in the causal digraph that has a measured or assumed nonzero qualitative value. It represents a process variable or parameter that has deviated outside of its range of normal operation. A valid node is a set of two terms: the deviated process variable or parameter and its nonzero qualitative value, e.g., (L, +).

A primary deviation is the deviated process variable or parameter that is the direct result of a fault. Secondary deviations are all other process variable deviations that arise from fault propagation. Both primary and secondary deviations are valid nodes.

A consistent branch is a directed arc between two valid nodes where the product of the values of its initial and terminal nodes equals the sign attribute of the branch. A consistent branch represents a path that may have been involved in the propagation of a failure. A consistent path is a directed path of consistent branches.

A valid tree is a subgraph of the causal digraph that consists of a valid measurement and all the valid nodes causally upstream from the measurement. All the branches in the valid tree are consistent. The valid tree describes the path of fault propagation from any causally upstream, valid node to the particular abnormal measurement.

Description of Diagnostic Strategy

The strategy for diagnosis, based on the use of causal models, is separated into three major steps: candidate generation, candidate testing, and identifying specific faults through a table mapping faults to primary deviations. The candidate generation and testing steps are outlined in detail below. The list of faults is generated from the reduced set of primary deviations through the use of a table mapping faults to primary deviations. The table is created from an expert system using knowledge about the process equipment specifications and the values of design and operating variables.

Candidate Generation

The objective of candidate generation is to rapidly partition the total set of fault origins into a feasible set (i.e., those primary deviations that could cause the observed secondary deviations) and an infeasible set. The criterion for including a node-sign pair in the set of possible primary deviations during candidate generation is that a consistent path must exist from the node to all abnormal measurements. The inputs to the procedure are the causal digraph for the process and the sign attribute of every arc, the controlled and manipulated variables and net sign of the functioning control systems, and the qualitative values of every process measurement.

Faults propagate along causal digraph arcs. Thus, the origin of the failure causing an abnormal process measurement must lie causally upstream from the measurement. Possible primary deviations are identified by searching causally upstream from the abnormal sensor.

The search is accomplished by constructing a valid tree for an individual deviated measurement. The valid tree represents the paths of fault propagation from every possible node in the causal digraph to the abnormal measurement. Beginning with the given valid measurement, qualitative values are assigned to unmeasured, adjacent, causally upstream nodes to make the causal arcs consistent. The assignment of qualitative values is continued until all possible primary deviations that could cause the observed measurement deviation are identified. Thus, the search is exhaustive.

Given the current node in the tree, adjacent nodes causally upstream from the current node are added to the valid tree if

1. The causally upstream node is not already in the path from the current node to the valid measurement, AND

2. If the process variable represented by the causally upstream node is measured, then the measurement must be valid with the qualitative value necessary to make the branch from the measured node to the current node consistent.

Condition 1 eliminates any circuits in the causal digraph which would give rise to cycling during the backward causal search. A node is added to the valid tree only if it does not already appear in the path to the valid measurement. Note that a node can appear more than once in the valid tree; it is only restricted from appearing more than once along any directed path. Condition 2 is used to bound the fault space. One of the assumptions for candidate generation is that if a disturbance is propagating along a causal path and two process variables in the path are measured, the causally upstream measurement will become valid before the downstream measurement. If, during the backward search, a measured node is encountered whose value is normal or opposite of the sign necessary to make the branch consistent, then the fault cannot lie causally above the measurement. The search is discontinued along this arc.

Control systems require a modification to the causal search bounding condition. Because control systems are designed to compensate for

disturbances, the manipulated variable is adjusted to keep the controlled variable at its desired set point. If a disturbance enters a functioning control loop and the magnitude of the disturbance is insufficient to saturate the control system, then the controlled variable remains normal and the disturbance causes a change in the manipulated variable. In the search procedure described above, the search space is bounded by normal measurements. Because a fault can lie causally above a normal measurement if the measurement is used in a control loop, the following modification is necessary: if the manipulated variable is valid (either assumed valid during the search or directly measured) and the controlled variable is normal, then consider the controlled node and nodes causally upstream from the controlled node as possible origins. The value of the manipulated variable and the net sign of the control loop are used to infer the value that the controlled variable would have if no control system were present. The controlled variable is added to the valid tree and the search is continued causally upstream from this node.

Because the probability of multiple, simultaneous, independent events is low, the candidate generation procedure first attempts to explain all the abnormal measurements by a single fault. Under the single fault assumption, a digraph node-sign pair is a primary deviation if a consistent path exists from the primary deviation to **every** abnormal sensor. A primary deviation is the root of a directed tree of consistent branches, which spans the set of valid measurements. Since consistent paths must exist from the primary deviation to every deviated measurement, the converse, that the actual primary deviation must be identified by the backward causal search from every measurement deviation, must also be true. Therefore, given multiple measurement deviations and the single fault assumption, the intersection of the sets of primary deviations generated for each of the valid measurements will identify those primary deviations with consistent paths to every abnormal sensor.

If multiple faults have occurred, set intersection, in most cases, will produce the empty set. A combinatorial intersection procedure is then necessary to determine the minimal set cover, to explain the observed measurement pattern with the fewest number of faults.

Candidate Testing

The purpose of candidate testing is to apply other types of information, beyond the knowledge of causal adjacency used in candidate generation, to eliminate implausible candidates from the set of primary deviations. Nodes that are locally plausible are eliminated if they are not consistent with all other known information. The knowledge considered here is (1) global constraints, (2) fault simulation using time delays, and (3) heuristic rules.

Global Constraints

Because the causal digraph is limited to local interactions, global information may be necessary to constrain spurious interpretations. If global knowledge is known, then it should be retained and incorporated for diagnosis. The global constraints used are that a process variable cannot simultaneously deviate in both directions, given a set of consistent causal paths from a root node to valid measurements, and that global knowledge can be used to specify the dominant causal path when multiple paths of opposite net sign exist in the digraph.

Simulation Using Qualitative Time Delays

A disturbance is instantly propagated from the initial node to the terminal node along an arc with zero time delay. This knowledge can be used to eliminate primary deviations from the set of possible origins. For each root node in the set of primary deviations, fault simulation is performed from the root node along the arcs with zero time delay. Causally downstream nodes are assigned values so that the branches in the simulation tree are consistent. If any node in the tree is measured and the qualitative value of the actual measurement is either normal, if the measured variable is not a controlled variable, or opposite of the sign in the simulation tree, then the root node should be eliminated from the set of primary deviations. If there are no measurement nodes in the tree, or if the actual measurements have the values that match the fault simulation, then the node remains a candidate. A normal measurement causally downstream from the primary deviation is acceptable if it is a controlled

variable, because the control system may compensate for the disturbance and yield a normal value.

Heuristic Rules

Knowledge in the form of rules can be used to reduce the number of primary deviations. Although the term heuristic is used, both experiential and model-based knowledge can be represented in this format. Several examples are presented:

Rule 1: If the controlled variable in a control system is normal, then the control system is working. Therefore, remove any primary deviations associated with the control system (from the controller through the control valve) and the desired set point. [Note: This rule assumes that sufficient time has elapsed for faults within the control system to cause the deviation of the controlled variable.]

Rule 2: If a control system is working, a disturbance propagates into the control system through the control valve, and the control system compensates for the disturbance by closing the valve, then the propagation of the failure is always halted and the controlled variable remains normal. Therefore, any primary deviations causally upstream from the control valve that cause the valve to close can be eliminated.

In addition to referencing qualitative data, rule antecedents can also reference numerical data (e.g., measurement values, reference values, and output from numerical calculations, including rates of change, simulations, statistics, etc.) to reduce the number of primary deviations.

Rule 3: If the measured, numerical value of a concentration is negative, then the sensor is miscalibrated or has failed. Therefore, eliminate all other primary deviations.

Rule 4: If the normal process and measurement noise of a sensor disappears (variance goes to zero), then the sensor has failed. Therefore, eliminate all other primary deviations.

Summary of Results

Research contributions in developing a procedure for evaluating the causal digraph for fault diagnosis are summarized.

- **Diagnostic Procedure Based on a Graphical Search Strategy**

A diagnostic strategy has been presented by Iri and co-workers (Iri et al. [1979] [1980]) that uses a graph representation to model the causal interactions. Their approach uses an iterative procedure to assign '+', '-', and '0' qualitative values to unmeasured and controlled digraph nodes. For each assignment of a qualitative value, the consistent branches of the graph are identified. The subgraph composed of consistent branches is then examined to determine if the subgraph is rooted. Because a single fault is assumed, causal pathways must connect the fault origin to every measurement deviation. If the subgraph becomes disconnected for a particular set of qualitative values, then there cannot be a single fault. The authors used the algorithm on a digraph of 21 nodes and 62 branches, of which six nodes were observed and three nodes were controlled. A purely iterative method requires 3^n assignments, where n is the number of unmeasured and controlled nodes in the causal digraph. Even with a heuristic to reduce the number of subgraphs evaluated, they reported that the algorithm generated about 20,000 subgraphs that were examined for connectivity (Iri et al. [1980]). They note that even with heuristics to reduce the number of graphs examined, the problem grows exponentially with the number of nodes.

Shiozaki et al. [1985] propose modifications to the iterative procedure of Iri et al. to improve computational speed. Improvements to the method are a systematic choice for selecting unmeasured nodes for assigning qualitative values, criteria for terminating the assignment of qualitative values, and the systematic revision of the assigned values on unmeasured nodes.

The premise for this work, identical to the premise of earlier work, is that for a single failure, consistent causal pathways must link the fault origin to all abnormal measurements. But whereas previous authors iteratively assigned and revised the qualitative values of unmeasured

digraph nodes, the diagnostic procedure developed here is based on a graphical search strategy. Because disturbances propagate along causal paths, possible origins of the failure are located by searching causally upstream from abnormal measurements. During candidate generation, a set of fault origins is quickly identified solely on the basis of causal adjacency.

▫ Model-Based Diagnostic Strategy

The diagnostic procedure outlined here is based on models of system components. Model-based approaches rely on an understanding of the system's underlying mechanisms or behavior. In contrast, experience-based strategies relate the pattern of observed abnormal symptoms directly to the fault.

Experience-based diagnostic methods have several limitations. They can only diagnose faults that have been previously observed and coded into the database. Second, the patterns of symptoms are plant-specific. Unless two systems are identical, the knowledge in the database cannot be transferred and used for the diagnosis of the second system. A majority of the diagnostic procedures presented in the literature are experience-based. For example, the rules developed for rule-based expert systems for process plant fault diagnosis are highly dependent on the specific context (Chester et al. [1984], Kumamoto et al. [1984], Andow [1986]). Other experience-based approaches (e.g., alarm trees, fault trees, cause-consequence diagrams, decision tables) are equally plant-specific.

The strengths of model-based diagnostic strategies offset the limitations of experience-based approaches. First, the knowledge of the underlying mechanisms can assist in diagnosing unfamiliar faults. Second, models of system components are plant-independent. Component models can be developed once and used for a variety of processes. Third, the knowledge contained in the models aids fault resolution. Although several database patterns may match the observed symptoms, many patterns may not be consistent with the system's underlying physical behavior.

- **Multiple Problem-Solving Approaches**

The diagnostic strategy incorporates multiple problem-solving approaches into the solution strategy. Graph search, qualitative constraints, simulation, and heuristics are used within a hypothesis generation and test framework. The integration of techniques improves diagnostic resolution because the additional approaches add knowledge which can further eliminate infeasible fault candidates. If the diagnostic strategy relied on a single problem-solving approach, the strategy would be limited to a single knowledge representation and a single evaluation procedure. For example, the causal digraph cannot represent global relationships or heuristics. If the digraph was used alone, many primary deviations that could be eliminated remain as possible fault origins.

- **Context-Independent Rule Bases to Specify the Correct Mapping of Faults to Primary Deviations**

The relationship between faults and primary deviations depends on the context in which the physical system functions. Information about the context is necessary to specify whether a fault should be considered and whether it should be mapped to the + or - deviation of a digraph node. For example, to identify possible faults for low pressure in a vessel, knowledge about the physical characteristics of the unit (e.g., number of flanges, relief valves, rupture disks, drain valves) would be important in identifying possible causes. These "leakage" faults would then be mapped to (P, -) only if the pressure of the vessel was greater than atmospheric pressure. If the vessel pressure was less, then the fault would be associated with the primary deviation (P, +) because the pressure of the vessel would increase as air or other fluid entered the system.

A table mapping faults to primary deviations is used to produce the list of faults from the reduced set of root nodes after candidate testing. Context-independent rules for generating the fault tables can be used to correctly map faults to primary deviations for a given context. Like the context-independent component models for generating the correct causal digraph, these rules can be grouped by process component. Rule antecedents

reference the unit's physical characteristics, design values, and possibly the current process measurements. Rule consequents are the faults for the primary deviations in the given component. Because the values in a specific component's database are used to generate the table mapping faults to primary deviations, the table is only valid for a given set of context-specific conditions and assumptions.

DIEX: A Model-Based Diagnostic System Prototype

DIEX (Diagnostic Expert) is a model-based diagnostic system prototype that builds the causal digraph and executes the diagnostic strategy. DIEX is coded in Franz Lisp, running under UNIX, on a DEC VAX 11-780.

Plant topography and the specific unit design information is entered through an interactive design program. Numerical values of the process parameters are necessary to specify the correct causal interactions from the causal models. Structural information is used to match the ports of interconnected units. The design program creates a data file which is used by a second program to construct the causal digraph. Object-oriented programming is used to specify the digraph. Node and arc flavors are used for instantiating specific process variables and their causal interactions.

General component causal models have been developed for ten types of process equipment and four elementary chemical reactions. The process equipment models include pipe, tee, centrifugal pump, valve (2-port), tank, heat exchanger, vaporizer, continuous stirred tank reactor (CSTR), sensor, and single-input single-output (SISO) control system models.

During candidate generation, DIEX constructs a valid tree for each abnormal measurement from the arcs in the causal digraph. The qualitative values of the measurements can be input into the diagnostic system prototype one at a time or in groups. When more than one measurement is valid, an active set of primary deviations from set intersection is maintained. Thus, a single intersection is performed during each pass. Causal simulation using time delays and heuristic rules were implemented for candidate testing. Rules were incorporated into the prototype as Lisp functions rather than through the use of a general rule interpreter. Global

constraints were not implemented because they are relatively straightforward.

The mapping of faults to primary deviations was done for the particular contexts of the example processes studied. Context-specific rule bases were developed for the process equipment listed above. Plant-independent rule bases were not developed.

Three examples were studied in detail: a tank with a level control system, which contained 21 nodes and 29 arcs; a vaporizer, which contained 45 nodes and 65 arcs; and a CSTR with an exothermic reaction and external cooling, which contained 122 nodes and 173 arcs. For the CSTR example, a dynamic simulator was developed to generate numerical values for the process measurements. The qualitative values entered into DIEX were determined from the simulation output for specific faults.

Plant Implementation

The conclusions generated by the diagnostic system are only valid when the assumptions on which the candidate generation procedure is based, are satisfied. Three of the assumptions, which are listed below, are concerned with implementation issues. Because the second and third assumptions are not always satisfied in practice, when these assumptions are violated, the diagnostic strategy outlined is insufficient to diagnose the failure. Additional knowledge can be incorporated into the diagnostic strategy to handle these cases. Each of the assumptions is reviewed and the research contributions are summarized.

- Assumption: The normal operating ranges for every measurement are selected so that if a fault occurs, one or more measurements will deviate outside of their normal range and become valid.

The limitations of mapping continuous variables into discrete states are presented and the selection of the normal references, used to determine when a node is valid, was investigated.

The mapping of continuous process variables and parameters into discrete qualitative states has two principal drawbacks. First, all

quantitative values that are mapped to a specific qualitative state have the same qualitative value. The relationships $x_1 < x_2$, $x_1 = x_2$, and $x_1 > x_2$ between the parameters in the same qualitative state cannot be determined. Second, the assignment of a qualitative value is sensitive to small changes in a variable's numerical value when the numerical value is near an endpoint of the qualitative range. A differential change in the continuous value can result in a discrete change in the qualitative value. When a measurement crosses an endpoint of the normal range, the diagnosis can change abruptly, due to the discrete logical decision on whether to bound the search space or to continue to search along the causal arc.

Assigning a qualitative value to a process variable depends on the normal range chosen for its reference. If the normal range is too narrow, small disturbances and transients will cause the node to be valid, and activate the diagnostic system. If the range is too wide, a deviated process variable caused by a fault may not be detected; the measurement does not cross the alarm threshold and the node's qualitative value remains '0'. Deviated process measurements that are misclassified as normal because of poorly set normal operating bands adversely affect candidate generation because the search space is bounded by these normal measurements. Therefore, the expected values and bounds must be set correctly to filter out normal process disturbances while being sensitive enough to detect abnormal symptoms.

Alarm thresholds are set from experience and should be statistically determined from historical process data. The range endpoints are chosen so that if the process variable deviates outside the normal range, then a fault has occurred.

- Assumption: Process variables can only deviate in a single direction.

This assumption restricts a process variable to a single direction of deviation, and therefore, a single qualitative value assignment. But variables that return to normal and deviations that change direction (undergo inverse response) do occur.

These cases can be diagnosed correctly if a history of the qualitative values assigned to each digraph node is maintained. Historical values of

variables that return to normal or change qualitative sign allow the causally downstream deviations to be explained by fault propagation through those variables. When historical values are retained, the procedure for constructing the valid tree during candidate generation becomes "a node causally upstream from the current node is added to the valid tree if its qualitative value is or ever was of the the correct sign to make the causal arc consistent." The use of historical values and the digraph modification for inverse response suggested by Oyeleye and Kramer [1987] can eliminate the restriction to single qualitative state changes.

- Assumption: If a disturbance is propagating along a causal path and two process variables in the path are measured, the causally upstream measurement will become valid before the downstream measurement.

This assumption may be violated for faults with small disturbance magnitudes, because the effect of the fault may be insufficient to cause all the measurements along a causal path to be valid. If one or more normal measurements lie in the path, then the causal search from the farthest downstream valid measurement is bounded by a normal measurement and the actual primary deviation is not included in the set of possible fault origins.

Two approaches for diagnosing faults when the disturbance is on the same order of magnitude as the normal process fluctuations are (1) reevaluating the qualitative value assignments of measurements, and (2) bounding the causal search using quantitative values of the maximum arc gains. In the first approach, the qualitative values of the measurements are analyzed by the diagnostic procedure. If the quality of the solution is poor (e.g., several independent failures), normal sensors that are close to the alarm thresholds are reassigned '+' and '-' values. The diagnostic system is rerun with the new valid nodes to see if the solution can be improved. The second approach for addressing faults with with small disturbance magnitudes incorporates the quantitative values of the maximum arc gains. If the quantitative values of the maximum arc gains are known, the gains and the numerical values of the measurements can be used for terminating the causal search during candidate generation. Given the causal arc $X \rightarrow Y$,

where both X and Y are measured, let x and y be the numerical values of the measurement deviation away from the normal references x_0 and y_0, respectively, and G_{XY} be the maximum gain between the two nodes. During the construction of the valid tree, if the current valid node is Y, node X is added to the valid tree only if the deviation at Y can be explained by the observed deviation at node X. Mathematically, node X is added to the valid tree if

$$\text{sgn}(x * G_{XY}) = \text{sgn}(y), \text{ and}$$
$$|x * G_{XY}| \geq |y|.$$

If the deviation at node Y is greater than can be explained by the deviation at node X, the causal search is bounded along this arc. When unmeasured digraph nodes exist between two measurements, the overall path gain is used.

Chapter 1

INTRODUCTION

1.1 Problem Statement

The purpose of process control systems is to maintain process variables within their desired ranges. While control systems compensate for small disturbances, large disturbances and process malfunctions can cause the control system to saturate, and operating conditions will vary outside these design limits. When process variables deviate beyond their desired ranges, not only is product quality in jeopardy, but these variations, if left uncorrected, could result in a catastrophic event such as fire, explosion, or the release of toxic chemicals.

In plants equipped with an information system, measurements of important process variables are collected in a central control room. Abnormal measurements trigger alarms which alert the process operator.

It is the process operator's responsibility to interpret the measurements and take corrective action, either by restoring the plant to normal operation or initiating shutdown procedures. With training and experience, humans can perform the task of troubleshooting quite well. They incorporate multiple problem-solving strategies, can reason with incomplete and inaccurate data, and can view the process from a global perspective.

On the other hand, the operator's ability to accurately diagnose process upsets may be severely limited. The diagnostic process is creative and demands significant cognitive energies. Errors and inaccuracies in reasoning, including poor recall and an inadequate understanding of the plant, will yield incorrect diagnoses. Interferences such as stress, fatigue, and boredom also impair operator performance. Humans have difficulty handling large amounts of data; they minimize or ignore much of the information and focus on a small subset. Although operators are usually well-trained in standard operating procedures, they may have difficulty handling uncommon or unanticipated events. Diagnosis may require an understanding of chemistry and physics, in which they are seldom trained. Experts may not be available for consultation due to work shift, employee

turnover, or vacations. Other factors affecting performance include the operator's training and operating experience. Because time constraints are critical, hesitation as well as inappropriate action could lead to disaster.

The increased complexity of plants in the process industries has made fault diagnosis more difficult. Integrated plant designs with several material recycles, complex utilities distribution, and energy integration increase the number of paths for disturbance propagation. Control and backup systems, designed to compensate for disturbances, tend to obscure the symptoms of faults.

The operator's effectiveness is crucial to the safe and economic operation of the plant. Early failure detection and diagnosis can reduce the number of plant shutdowns. Fewer shutdowns result in greater plant availability and improved operating margins. The losses resulting from a major accident can be staggering: liability for property damage, personal injury and death, the loss of raw materials and equipment, a long period of business interruption, the loss of company goodwill, and reduced employee morale. The early and effective diagnosis of faults can lead to increased safety and profitability.

Today, the process operator requires assistance in diagnosing the cause of upsets. The reasons presented—human factors, plant complexity, economics, and safety—illustrate the urgent need for a diagnostic system to assist the operator in responding to process alarms.

1.2 Research Scope

The design of a computer-based aid to assist operators in the diagnosis of process failures is investigated. The diagnostic aid is intended for the chemical and nuclear industries in which process measurements are collected by fixed instrumentation and displayed in a central control room. The diagnostic methodology described in this thesis is designed to assist the process operator by interpreting the real-time process data and generating a list of possible fault candidates. When a fault occurs, the diagnostic system applies knowledge stored in the system to interpret and diagnose the abnormal measurements. From the list of fault hypotheses

generated, the operator in the control room can then direct field personnel to confirm or reject the individual candidates. This is accomplished by obtaining additional information not available in the control room, e.g., through visual inspection, equipment noise and vibration, etc., and by gathering measurements from equipment-mounted sensors.

The thesis integrates and builds upon research in several disciplines outside of chemical engineering, in particular, the areas of cognitive science, artificial intelligence, and computer science. Research in cognitive science is broadening our understanding of human reasoning and problem solving. Advances in artificial intelligence, specifically in the areas of qualitative modeling of physical systems and knowledge-based expert systems, are fostering the development of computer-based qualitative reasoning. Advances in computer hardware, including list processors and increased data storage, and the continuing improvements in process monitoring and control, will facilitate the development of a commercial quality process diagnostic system.

The goal of this research is to improve operator performance in the diagnosis of faults through the design of a computer-based diagnostic system which will assist the operator during plant upsets. Through the implementation of such a system, it is hoped that the number of faults leading to production down time can be reduced and the number of industrial accidents lessened. The rapid and accurate diagnosis of process failures should lead to greater profitability and increased safe operation in the process industries.

Chapter 2

FAULT DIAGNOSIS

In this chapter, the task of fault diagnosis is investigated from three perspectives. The process environment is first examined. The characteristics of the domain place significant demands on the operator. Second, previous approaches to process fault diagnosis are classified into two general categories and the advantages and limitations of each are summarized. Third, the characteristics of human diagnostic reasoning are studied to identify those that are relevant to the design of a computer-based diagnostic aid. From these viewpoints, a list of several desired attributes of a diagnostic aid are presented and an overall system architecture is proposed. From this investigation, the thesis objectives are formulated.

2.1 Fault Diagnosis in the Process Environment

Diagnosis is the task of identifying the cause or origin of some observed, abnormal behavior in a system. In the process environment, abnormal behavior is identified by the deviations of process measurements away from their normal, expected values. The cause of the abnormal behavior, one of a large number of possible faults, is usually not directly observed. The only information available to the operator in the control room is a pattern of measured process variables that have deviated as a result of fault propagation. Diagnosis, then, is reasoning from these known symptoms to the unknown cause.

Reasoning from a pattern of abnormal measurements in the control room is a difficult task. Diagnosis is demanding because:

- A large portion of the plant is monitored and controlled from a single control room. Hundreds of process measurements are collected.

- Even with the large number of measurements, only a small fraction of the total number of process variables are measured.

- Coupled with the large plant size is complexity. Many paths of interaction exist between process variables through highly integrated process units.

- A large number of faults need to be recognized.

- Fault propagation may be fast and affect several units simultaneously (relative to the sampling rate). Because pressure and flow disturbances propagate at the speed of sound, fault symptoms may appear almost instantaneously far from the fault origin.

- Control and backup systems may be complicated and embody multiple objectives. The intent behind the control scheme design may not be obvious to the operator. Thus, the operator may have difficulty understanding how and why the process responds as it does during a disturbance. Without a sufficient understanding of the control systems, the active alarms may contribute nothing to his resolving the cause of the upset.

- Diagnosis involves synthesizing a large amount of information, including mass and heat transfer, fluid statics and dynamics, thermodynamics, reaction kinetics, physical properties, and process chemistry.

- Measurement uncertainty, sensor degradation, and calibration errors decrease belief in the fault hypotheses generated.

- Severe consequences may result if the failure is not detected, diagnosed and corrected early. The large potential for loss puts pressure on the operator to respond.

Several diagnostic heuristics for locating faults in the process domain are inadequate for a general solution procedure. Among them are

Heuristic: Faults propagate in the direction of bulk fluid flow.

Disturbances that affect pressure and flow rate propagate both with and against the direction of bulk fluid flow. Thus, measurements may deviate physically upstream from the fault origin.

Heuristic: If one or more of the inputs to a process unit are abnormal, then the fault is not located in the unit.

Material recycle loops and feedback control systems can cause process unit inlets to be disturbed as the result of outlet disturbances. Therefore, the fault may lie within the process unit or downstream of the abnormal measurement.

In summary, locating the origin of failures in process plants is quite complex and requires the integration of numerous types of knowledge.

2.2 Model-Based and Experience-Based Diagnostic Strategies

At a fundamental level, diagnosis can be considered the task of matching an observed pattern of abnormal symptoms to a reference pattern. Strategies for fault diagnosis can be differentiated on how the reference pattern is obtained. If the reference patterns are obtained from compiled observations of the system over time, such that a specific fault is directly related to an individual pattern in the database, then the method is an experience-based diagnostic strategy. The term "experience-based" or "heuristic-based" is used because the known patterns of observations come directly from experience. During diagnosis, abnormal symptoms from the system are compared to the stored patterns to yield a set of possible faults. The fault considered most probable is the one with the closest match. On the other hand, if a model of the system is used, which represents a deeper understanding of the physical behavior of the system, then the method is classified as a model-based diagnostic strategy. Here, a model is loosely defined to include any representation that involves an understanding of the system's underlying mechanisms or functioning;

model-based strategies incorporate the knowledge of the relationships between the system's parameters and state variables. To illustrate the use of models in diagnosis, consider a model-based hypothesis generation and test strategy. Given a fault hypothesis, the model is used for fault simulation to generate the expected pattern of measurements. The simulation results are compared with the actual known observations to yield possible diagnoses. If the model can be "run in reverse," the model can identify possible fault origins given the abnormal symptoms. Rasmussen and Jensen [1974] and Rasmussen [1978] [1979] [1981] call these strategies symptomatic and topographic, respectively. Note that models can be used to prepare the list of abnormal patterns beforehand, through fault simulation, to match symptoms directly to faults. The distinction between these two basic strategies rests on the use of a system model.

Experience-based and the model-based strategies have advantages and limitations. One drawback of experience-based strategies is that the symptoms of every fault must be known prior to diagnosis. If a particular fault has not been encountered and its pattern of symptoms are not stored in the diagnostic system, then the fault cannot be diagnosed and no information is generated by the system. This limitation is especially dangerous because the faults that the operator will have the most difficulty in identifying, i.e., those failures with a low rate of occurrence, generally are not covered by experience. Second, the patterns stored in the database depend on the particular context. Unless two systems are identical, the knowledge in the database cannot be transferred and used for the diagnosis of the second system. The database of patterns must be created from scratch for every new system. Third, fault resolution is dependent upon the extent of fault propagation. The fault must be significantly developed before the set of fault candidates is reduced to a feasible number.

One advantage of model-based strategies over experience-based methods is that the knowledge of the underlying mechanisms can assist in diagnosing unfamiliar faults. Second, although models of an entire process plant are specific to the plant, models of system components are plant-independent. Models developed for components can be developed once and used for a variety of processes. For example, the functioning of a valve is identical across plant sites. Third, the knowledge contained in the model aids fault

	Experience-Based Strategies	Model-Based Strategies
Advantages	1. A model is not required 2. A solution procedure is not required 3. Computationally faster than model-based strategies	1. Can assist in diagnosing unfamiliar faults 2. Models of system components are plant-independent 3. Knowledge contained in the model aids fault resolution
Limitations	1. Can only diagnose faults that have been previously observed 2. Patterns of symptoms are plant-specific 3. Fault resolution is dependent upon the extent of fault propagation	1. A model is required 2. A solution procedure is necessary to evaluate the model 3. Computationally slower than experience-based strategies

Figure 2-1

Experience-Based and Model-Based Diagnostic Strategies

resolution. Although several database patterns may match the observed symptoms, many patterns may not be consistent with the system's underlying physical behavior.

A major limitation of model-based diagnostic strategies is that a system model is required. Models may be difficult to formulate because it involves gaining an understanding of the underlying physical principles of the device. This is especially true in domains where human understanding of the physical mechanisms is weak or nonexistent (e.g., pathology). The model must also be able to characterize the system over the entire range of operation. Second, solution procedures are necessary to evaluate or solve the model. Third, model-based diagnosis is generally slower than experience-based methods, because solving models entails more effort than comparing observed systems to stored data. This is especially true for common failures which require the repeated use of the model-based procedure to diagnose each occurrence. In experience-based strategies, neither a model nor a solution procedure is required. The strengths and limitations of the two strategies are summarized in Figure 2-1.

Many of the diagnostic systems for the process environment reported in the literature (e.g., fault trees, cause-consequence diagrams, decision tables, rule-based expert systems)[†] use only a single diagnostic strategy. The drawbacks of the individual strategies can be overcome if both experience-based and model-based strategies are incorporated in the diagnostic system. Notice that the strengths of one method offset the limitations of the other.

Quantitative equations have been the dominant method for modeling physical systems. Dynamic models, involving systems of differential and algebraic equations, are well developed and commercial computer programs are available to solve them. Unfortunately, modeling and computational difficulties in the quantitative approach severely restrict this representation for fault diagnosis. The major difficulties include specifying the correct variables and form of the equations, fitting the model parameters, and guaranteeing that the system of equations is valid over the entire

---

[†]Research in fault diagnosis is reviewed in Chapter 7.

range of operation. Also, a fully specified set of inputs is needed to solve the system. The quantitative model, running in parallel with the process, can detect deviations of the process away from the model predictions. But the model cannot be directly used to identify the specific fault after the abnormal measurements are identified. Numerical equations do not explicitly contain causal directionality; therefore, they cannot describe the path of fault propagation through the system. Faults can only be identified indirectly through fault simulation. This requires repeated simulations and that the effects of all possible faults be mathematically characterized. These limitations of quantitative representations provide an impetus for the development of qualitative models for fault diagnosis.

2.3 Characteristics of Human Reasoning During Problem Solving

The ability of man to reason is unparalleled. Even when humans face problems that are not well understood, with data that are inaccurate and incomplete, they still perform exceptionally well. This suggests that the design and development of a diagnostic aid for human problem solving should start with an understanding of the way humans perform diagnostic reasoning. The aim of this section is to study human performance to identify the reasons why man is proficient at problem-solving tasks. This study will serve as a guide for building computer-based reasoning systems. Note that the objective of the diagnostic aid is not to imitate human performance (because human reasoning has its shortcomings). Rather, the goals of the computer-based aid are to capture human knowledge and general reasoning strategies, so that the computer can interact with the human operator and generate understandable explanations, and to allow the computer to enhance human diagnostic ability by compensating for the limitations of human reasoning.

In a review of research in cognitive science, I have identified three characteristics of human problem solving that are relevant to the design of a diagnostic aid: the use of multiple problem-solving strategies, qualitative reasoning, and the application of both general and context-specific knowledge. These characteristics are investigated.

2.3.1 Multiple Problem-Solving Strategies

Human reasoning is characterized by the use of multiple problem solving strategies. These strategies include pattern recognition, the use of simple models (both qualitative and quantitative), hypothesis generation and testing, simulation, cause and effect reasoning, heuristic expertise, procedures and algorithms, trial and error, and reasoning by analogy.

Empirical evidence indicates that people solve problems using both of the general diagnostic strategies discussed in Section 2.2, at different times during problem solving. If given a choice between experience-based and model-based approaches, humans would prefer to act as context-specific pattern recognizers rather than attempting a more analytical approach. If the situation is familiar and experiential knowledge is known, the experience-based strategy is used first. Failure to solve the problem in this mode causes the diagnostician to employ the model-based strategy, which considers the functional topography of the system. Using this strategy, the human must go beyond the surface features of the problem and consider the underlying system structure. Rouse [1983] proposed a model of human problem solving whose architecture is based on the assumption that humans have a clear preference for proceeding on the basis of state-oriented pattern recognition, rather than on the basis of structurally-oriented information. The model first attempts to choose an appropriate action based on the observed symptoms of the malfunction. If the model should fail to recognize a familiar pattern of abnormal state variables, then it selects an action based on the functional structure of the malfunctioning system.

2.3.2 Qualitative Reasoning

Qualitative reasoning is the ability to reason about and solve problems using a qualitative description of the system and qualitative changes in the system's parameters. Humans, when performing diagnostic tasks, do not solve a system of differential and algebraic equations. Rather, they reason by assigning qualitative values to the parameters of a qualitative model. 'Low' and 'high' in "low pressure" and "high temperature" are

examples of qualitative values. Numerical values of system parameters are not required for reasoning.

In many domains, qualitative reasoning is superior to formulating and solving a system of quantitative equations because reasoning based on qualitative models and data is sufficient for performing the desired task and requires less computation. For example, a process operator diagnosing faults in a chemical plant does not need to know the exact, dynamic behavior of a set of process variables; a qualitative description of how the variables change is often satisfactory.

A wide variety of models can be used to represent any given physical system. Model selection depends on the characteristics of the domain, the kind of reasoning to be performed, and the nature of the solution desired. The example below illustrates the wide range of models describing the heating of a plate.

A plate of thickness L, shown in Figure 2-2, is initially at ambient temperature T_∞. The bottom of the plate is then subjected to the uniform heat flux q. Assume that the thickness of the plate is small compared with its other dimensions so that heat loss from the sides may be neglected. The thermal conductivity k is constant.

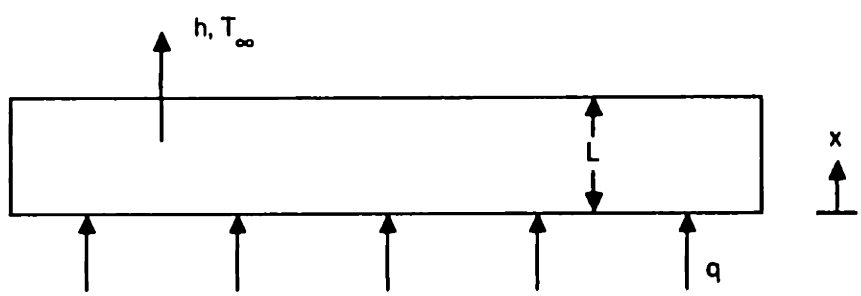


Figure 2-2
Heating a Plate

Model 1: A partial differential equation is required to exactly specify the temperature within the plate over time with respect to the spatial variable x.

$$\rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

The initial and boundary conditions in x measured upward from the bottom of the plate are

$$T(x,0) = T_\infty$$

$$-k \frac{\partial T(0,t)}{\partial x} = q$$

$$-k \frac{\partial T(L,t)}{\partial x} = h[T(L,t) - T_\infty].$$

Model 2: The problem can be simplified if the plate temperature is lumped. An ordinary differential equation results. The initial condition is $T(0) = T_\infty$.

$$\rho c_p L \frac{dT}{dt} = q - h(T - T_\infty)$$

Model 3: The problem can be further simplified with a qualitative model.

$$q \xrightarrow{+} T$$

In this formulation, the arrow denotes 'causes' and the '+' sign indicates that the values of the two terms at the arrow's head and tail change in the same direction. In this representation, the values of q and T are qualitative and can assume the values 'increases' and 'decreases.' The qualitative model is interpreted as "increasing the heat flux increases the plate's temperature." Note that the qualitative model eliminates the rigorous mathematical formulation while retaining the important qualitative features of conductive heating.

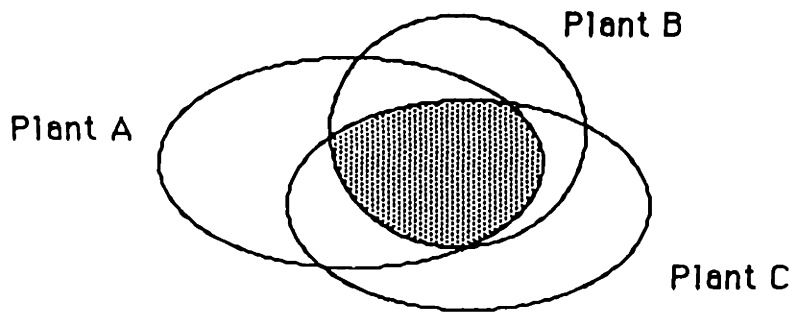
Humans, when performing diagnostic tasks, do not mentally solve a system of differential and algebraic equations. Rather, they reason qualitatively about the deviations in the values of process variables and

parameters. In most cases, the operator diagnosing faults in the process environment does not need to know the exact, dynamic behavior of every changing parameter. Rather, he reasons from a qualitative description of the plant and the qualitative changes in the magnitudes of the process measurements. Model 3 most closely captures this description.

2.3.3 General and Context-Specific Knowledge

Human reasoning is characterized by the use of both general and context-specific knowledge. In the process environment, troubleshooters who lack process-specific knowledge can do a credible job of diagnosis by relying on general diagnostic strategies and process models which capture the underlying physical laws and mechanisms. Local plant experts also do well because they employ specific process knowledge. Thus, the information used in problem solving includes both general strategies for reasoning and general information about the domain, as well as data specific to the current problem.

This dichotomy of diagnostic knowledge is represented in Figure 2-3. The ellipse labeled Plant A represents the total knowledge required to diagnose failures in Plant A. The intersection of this knowledge with the knowledge required to diagnose Plants B and C, represented by the shaded region, is that knowledge which is common among all three plants. This core knowledge characterizes procedural and declarative knowledge that is common among processes. Core knowledge includes general strategies for fault diagnosis, models characterizing the behavior of general classes of process equipment, equipment failure modes, and general knowledge representation formats for the storage of process data. Core knowledge is common among plants, and therefore transportable between processes. This knowledge does not have to be relearned or redeveloped at new plant sites. On the other hand, plant-specific knowledge is valid only within a given context. Plant-specific knowledge includes the plant topography (interconnections between the individual process units), equipment design specifications, context-dependent patterns of abnormal measurements, normal operating ranges for process variables, plant operating procedures, and current



CORE KNOWLEDGE

- General diagnostic strategies
- Models of general categories of process equipment
- Equipment failure modes
- Knowledge representation formats for design data

PLANT-SPECIFIC KNOWLEDGE

- Plant topography
- Equipment design specifications
- Patterns of abnormal measurements (for experience-based diagnosis)
- Normal operating ranges
- Operating procedures
- Current values of process measurements

Figure 2-3

Knowledge Required for Diagnosis:
Core Knowledge and Plant-Specific Knowledge

values of process measurements. Both categories of knowledge are necessary for effective diagnosis.

2.4 Desired Attributes of a Diagnostic System

Several desired characteristics of a diagnostic system are presented below. The list is not exhaustive. Rather, its purpose is to provide an initial direction for system design, as well as serve as criteria for evaluating diagnostic systems.

1. The primary objective of a diagnostic aid is accuracy. The system should produce a list of possible faults that includes the actual fault origin or origins. If the actual fault is not included in the set, the operator will focus his attention on the wrong set of candidates. This may be more harmful than having no diagnostic system at all.

2. The second major objective is to minimize the number of spurious faults that are included in the list. A small set of fault candidates will narrow the operator's focus. Although the number of process measurements will ultimately determine fault resolution, the method should incorporate as much knowledge as possible to eliminate implausible failures.

3. The system should diagnose a wide range of failures and have the maximum resolution between them. It should not be limited to certain classes of faults (for example, only sensor failures).

4. The system should not be brittle at its boundaries. If a specific fault is not identified, the system should at least present the location of the failure in terms of one or more process units, or at a higher level of abstraction, process subsystems, rather than yield no information at all.

5. Because the diagnostic aid must operate on dynamic, real-time processes, the diagnoses must be generated faster than the process so that corrective action can be taken.

6. The system should be easily modified and updated. Changes in the process due to piping and equipment changes should not require extensive reprogramming or data gathering. (Extensive effort includes, for example, constructing a new fault dictionary, building new quantitative models and estimating parameters and coefficients, and rewriting rules in the knowledge base.)

7. The computer code should be flexible and easily portable to a variety of process environments. The diagnostic system should not be designed anew at different plant sites. For example, the diagnosis procedure should be general, and therefore applicable to a wide variety of processes. A modular system design reduces the costs of development and installation.

8. The aid should be able to use the installed instrumentation, without requiring new sensors. This is not to say that additional measurements cannot be added to improve the system's speed and fault resolution, but that it should be able to perform fault diagnosis without specific requirements for the number of sensors and sensor locations.

Implicit in this list is the assumption that multiple types of knowledge and multiple solution strategies will be incorporated if they can improve the system's ability to identify the origin of the failure.

2.5 Overall Diagnostic Architecture

The process of troubleshooting involves three consecutive tasks: detection, diagnosis, and correction. The overall architecture for the diagnostic aid, illustrated in Figure 2-4, follows this sequence. First, measurements of process variables are compared against their normal reference values. Error bounds or thresholds are established for every measured variable so that if any measurement deviates outside its normal range, then

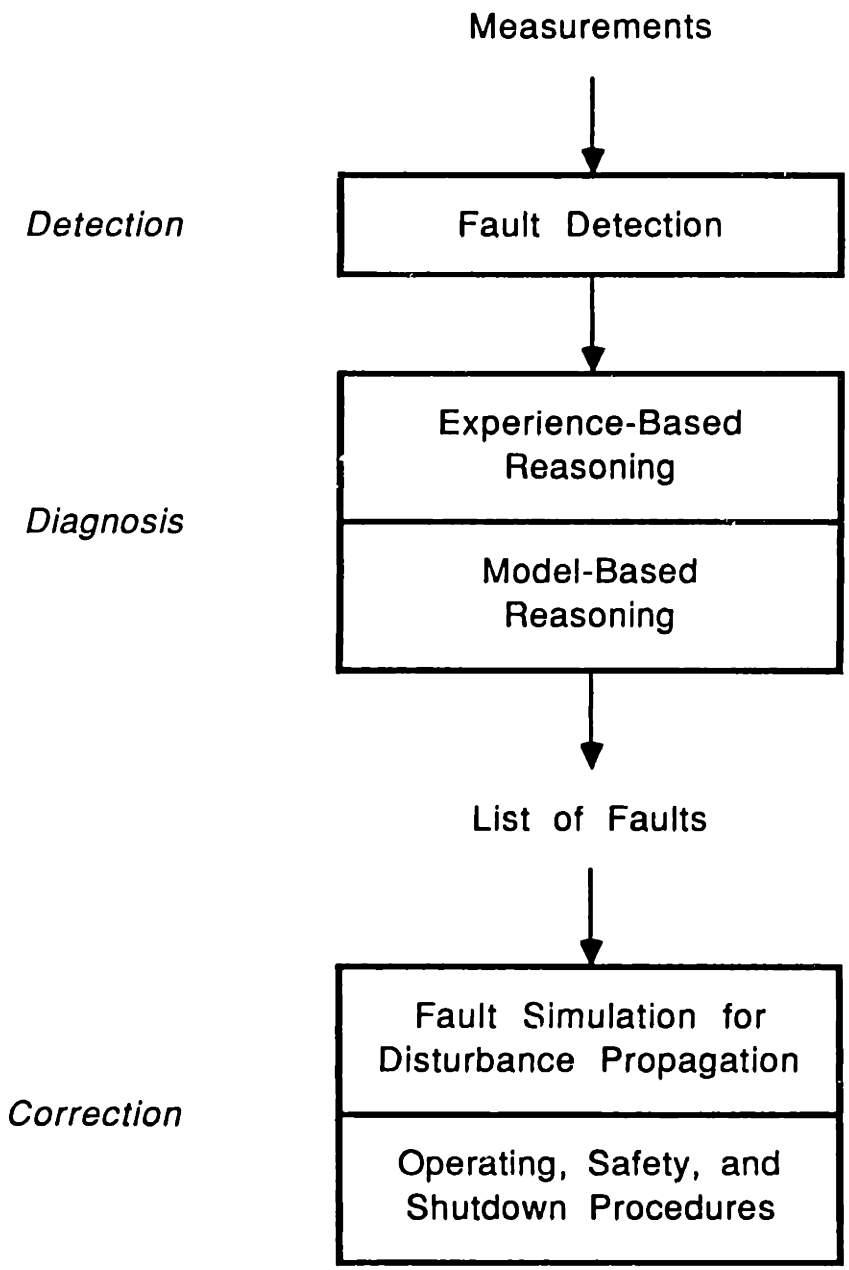


Figure 2-4
Diagnostic System Architecture

a process failure has occurred. Once a deviation is detected, one or more diagnostic strategies can be employed to reason from the abnormal symptoms to specific fault candidates. Following the hierarchy of Rouse, the diagnostic system first attempts to identify the process upset by matching the sensor deviations against the stored measurement patterns in its database. This initial screening of disturbances is used to identify common failures (i.e., those that have been previously diagnosed and stored). If the process deviations do not match the stored patterns, the system then applies a model-based diagnostic strategy. Once the operator determines the actual fault from the list of possible candidates, corrective measures can be taken. This third step involves fault simulation, to analyze the effect of disturbance propagation on the process, and the implementation of remedial action guided by the standard operating, safety, and shutdown procedures found in the plant operating manuals.

2.6 Thesis Objectives

This thesis addresses the design of a computer-based diagnostic aid for real-time process plant fault diagnosis. Specifically, the research investigates model-based diagnostic reasoning. The thesis objectives are (1) to develop suitable representations to characterize the process and the knowledge necessary for diagnosis, and (2) to develop a general strategy for evaluating the representations.

Three themes, which parallel human diagnostic reasoning, underlie these two objectives.

General versus Context-Specific Knowledge

The research focuses on those elements of the diagnostic aid that are general, or independent of the particular context. The context-independent elements, termed the core knowledge, include general strategies for fault diagnosis, models characterizing the behavior of general classes of process equipment, general rules that relate specific failures to process variable deviations, and general knowledge representation formats for the storage of process data and equipment design specifications. Because the behavior of general types of process equipment is identical across plant sites, models

of system components are used to form the base of the process representation.

The separation of diagnostic knowledge into core and plant-specific sections adds modularity to the diagnostic aid. Because the core knowledge is transportable between plant sites, implementation involves adding only the context-specific information.

Qualitative Reasoning

The research focuses on developing knowledge representations and a diagnostic procedure that are qualitative. Qualitative reasoning, which involves qualitative component models and qualitative values assigned to the model's parameters and state variables, is sufficient for diagnosing a majority of process failures. Qualitative reasoning overcomes the inherent limitations of quantitative representations.

Multiple Problem-Solving Strategies

The research focuses on incorporating multiple problem-solving strategies into a general solution procedure. Graph search, simulation, heuristics, and qualitative constraints are used within a hypothesis generation and test framework. The use of several strategies improves diagnostic resolution because the additional strategies add knowledge which eliminates infeasible fault candidates.

In summary, the goal of this research is to develop the core knowledge necessary for model-based process plant fault diagnosis. The two major constituents of the core knowledge are the general, qualitative, component-based models for chemical processes and process equipment, and the general solution procedure for evaluating the qualitative models.

Chapter 3

CAUSAL MODELS FOR FAULT DIAGNOSIS

Causal models are qualitative models based on the cause and effect relationships between a system's parameters and state variables. Although several authors have used causal models for diagnostic reasoning, none have adequately characterized the causal interactions or described how to develop causal models. Because causality is a powerful technique for reasoning about physical systems and the causal digraph is a suitable format for representing causal interactions, a more thorough understanding of causal relationships and a procedure for developing the causal digraph are desired.

In this chapter, I show how to derive the causal digraph for a set of design equations, outline the limitations of qualitative representations, present guidelines for developing causal digraphs for fault diagnosis, and develop general causal models for standard system components.

3.1 Qualitative Models Based on Causality

In this section, a knowledge representation format for specifying the causal relationships between process variables is developed. My focus is on how changes in system parameters, or changes in the system itself, affect other process variables. I use this representation for reasoning about the behavior of a system. Note that the goal is to state or specify the causal interactions, and not to explain or support why the causal interactions exist (e.g., why heat is transferred or why processes approach thermodynamic equilibrium).

Because qualitative reasoning requires qualitative descriptions of system parameters, the values of parameters and attributes in the representation are symbolic, rather than numeric.

3.1.1 Causal Digraph†

Causality is the relationship between a cause and an effect such that the cause produces, or is the reason for, the effect. Within a system, causal relationships exist between the system's parameters and state variables. Causal interactions are local interactions because they exist only between parameters and process variables that are in some sense adjacent on a given level of model detail. This differs from quantitative equations which can express relationships between any set of process variables. Reasoning based on causality attempts to explain the behavior of a system by the cause and effect relationships that exist between the system's parameters and state variables.

The causal relationships are represented in a network structure termed the causal directed graph (digraph). Nodes in the digraph represent process variables (e.g., flow rate, pressure, temperature, species concentration) and process parameters (e.g., resistance to flow, reaction rate constant). Arcs represent the causal interaction between these terms. The graph is directed because each causal influence is unidirectional: the initial node at the arc's tail represents the cause and the terminal node at the arc's head represents the effect.

3.1.1.1 Causal Digraph Arcs

The arcs in the causal digraph represent the causal interactions between the system's variables and parameters. Each arc has a set of attributes that characterize the causal interaction. A set of three attributes is used here: sign, magnitude, and time.

Sign

The sign attribute represents the change in the causally downstream variable due to the deviation in the causally upstream term. The qualitative values '+' and '-' are used to indicate that the process variables

---

†Graph theory terminology is summarized in Appendix A.

represented by the initial and terminal nodes vary in the same direction and in opposite directions, respectively.

Magnitude

This attribute specifies the strength of the causal interaction. The magnitude, or gain, of the causal interaction is the degree of deviation of the causally downstream variable resulting from a change in the upstream variable. The qualitative value '1' indicates that the magnitude of the arc is large enough so that a deviation in the upstream node will cause the downstream node to deviate. The value '0' indicates that the magnitude of the effect is not large enough to cause a deviation at the terminal node.

The magnitude attribute is used to differentiate between the existence of an arc and an arc with a small transmittance. Although a causal arc may exist and be represented in the causal digraph, the effect on the terminal node of the propagation of an influence along the arc may be insignificant. (See Section 3.2.4.2).

Time

The time attribute specifies the delay between a cause and its effect. The delay time is a measure of the propagation time for a deviation in the causally upstream variable to reach the downstream variable along the causal path. The qualitative value '0' for this attribute is defined as zero delay (instantaneous transmission); a value of '1' means that the time delay is positive.

Since both magnitude and time attributes are actually continuous variables, the classification of these values into discrete qualitative states depends on the measurement and time scales chosen by the observer. For example, the ability to detect changes in length will vary depending on whether the measurement scale is in meters or millimeters. The definition of an instantaneous transmission will vary for a system observed once a minute versus once an hour. Thus, the detection of a deviation depends on the granularity of the measurement scale chosen; instantaneous transmission or a positive time delay depends on the granularity of the time scale chosen. The qualitative values of the attributes of a causal arc are summarized in Table 3-1.

Table 3-1

Qualitative Values of the Attributes of a Causal Arc

Attribute	Value	
Sign	+	A positive deviation at the initial node causes a positive deviation at the terminal node.
	-	Deviations at the initial and terminal nodes have opposite sign.
Magnitude	1	Magnitude of the causal influence is large enough to cause the deviation in the causally downstream variable.
	0	Magnitude is not large enough to cause the deviation.
Time	0	Negligible time delay for the transmission of the causal influence.
	1	Positive time delay.

Numerical values of process parameters and design (expected) values of process variables may be required to specify the existence of an arc and the values of its attributes. Therefore, associated with each arc are a set of conditions that must be satisfied before a causal arc can accurately model a causal influence. Conditionals for the existence of causal arcs and arc attribute values are discussed in Section 3.5.2.

3.1.1.2 Causal Digraph Nodes

Nodes in the causal digraph represent process variables, parameters, and combinations of these terms. A node has no attributes that specify causality.

Reasoning using the causal digraph is done by assigning qualitative values to the nodes. Like the magnitude and time delay attributes of

causal arcs, the actual values of state variables and system parameters are continuous. Continuous values are mapped into discrete qualitative states for qualitative reasoning. For each continuous variable and parameter, a range is selected that represents normal operation of the system. The endpoints of this range break the value continuum into three regions, and are chosen so that the intervals represent qualitatively uniform behavior. All quantitative values that lie within a region are assigned the qualitative value of that region. In this work, the three symbolic terms low '-', normal '0', and high '+' are used. Qualitative transitions, for example from normal to high, occur at the boundary endpoints.

Numerical values cannot be mapped directly to qualitative values because qualitative values are meaningless without a specified reference value or normal range. For example, a "high" temperature in one context may be "low" in another. The qualitative value 'high' has meaning only when a reference temperature is specified. Therefore, qualitative values do not represent absolute quantities. Rather, they represent the deviation of a numerical value away from a specific reference. Mathematically, the qualitative value of x, denoted [x], defined as the deviation from a context-specific reference point x_0, is given as $[x] = \mathrm{sgn}(x - x_0)$. When the reference is a range, the qualitative value of x is defined as

$$
\begin{aligned}
[x] &= \text{'-'} \text{ when } x < x_0^- \\
[x] &= \text{'0'} \text{ when } x_0^- \leq x \leq x_0^+ \\
[x] &= \text{'+'} \text{ when } x_0^+ < x,
\end{aligned}
$$

where x_0^- and x_0^+ are the lower and upper range endpoints, respectively.

The reference value is the expected value or range that characterizes the process variable or parameter under normal operating conditions. Within the process environment, the expected value could be stationary (e.g., steady state processes), or be moving with time (e.g., batch or semi-batch processes, changing set points, optimization). Reference values can be obtained from historical data, controller set points, and quantitative process models. For a process moving between states, the reference values should be calculated from dynamic models. Deviations are identified by comparing process data with the predictions of the dynamic models. Because the reference state depends on the given set of process conditions, the reference is context-dependent.

Discrete variables (e.g., a valve is open or closed) change the structure of the causal digraph and are not modeled explicitly as digraph nodes. Discrete variables are used for digraph construction, as will be discussed in Section 3.5.2.

3.1.1.3 Modeling System Behavior

The example presented below shows how system behavior is represented by the causal digraph. The process schematic for a tank with a level control system is presented in Figure 3-1. A causal digraph for this system is illustrated in Figure 3-2. Arcs and nodes related to bulk fluid flow are shown. The + or - sign on each arc represents the sign attribute specifying the direction of influence.

The digraph is a suitable representation format for causality because, in the examples studied by the author, there is usually only a small number of process parameters and variables that directly influence a given process variable. Thus, each factor can be suitably represented and each causal influence can be made explicit.

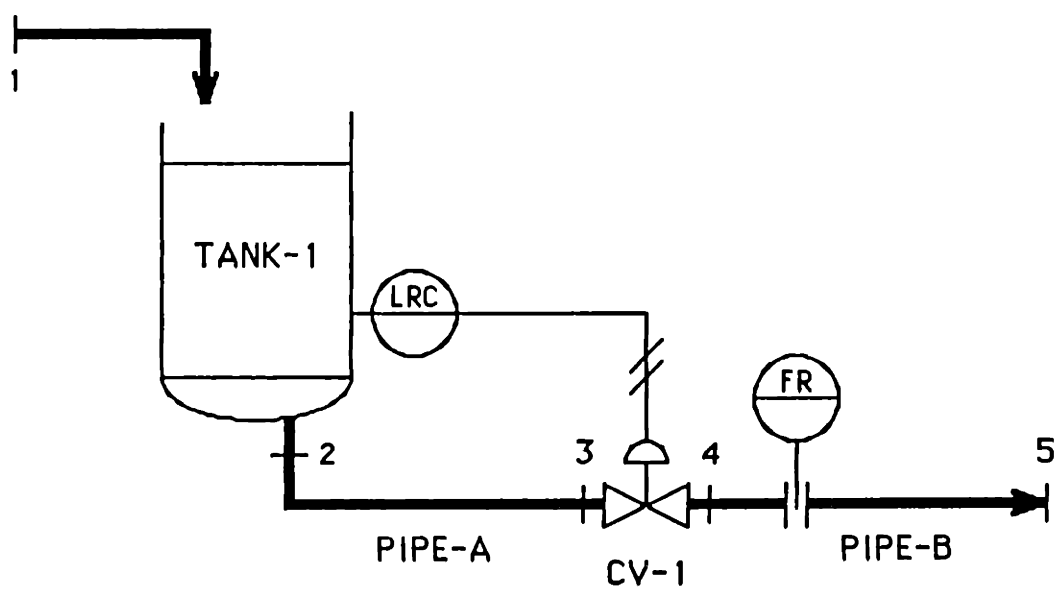


Figure 3-1
Process Schematic of a Tank with a Level Control System

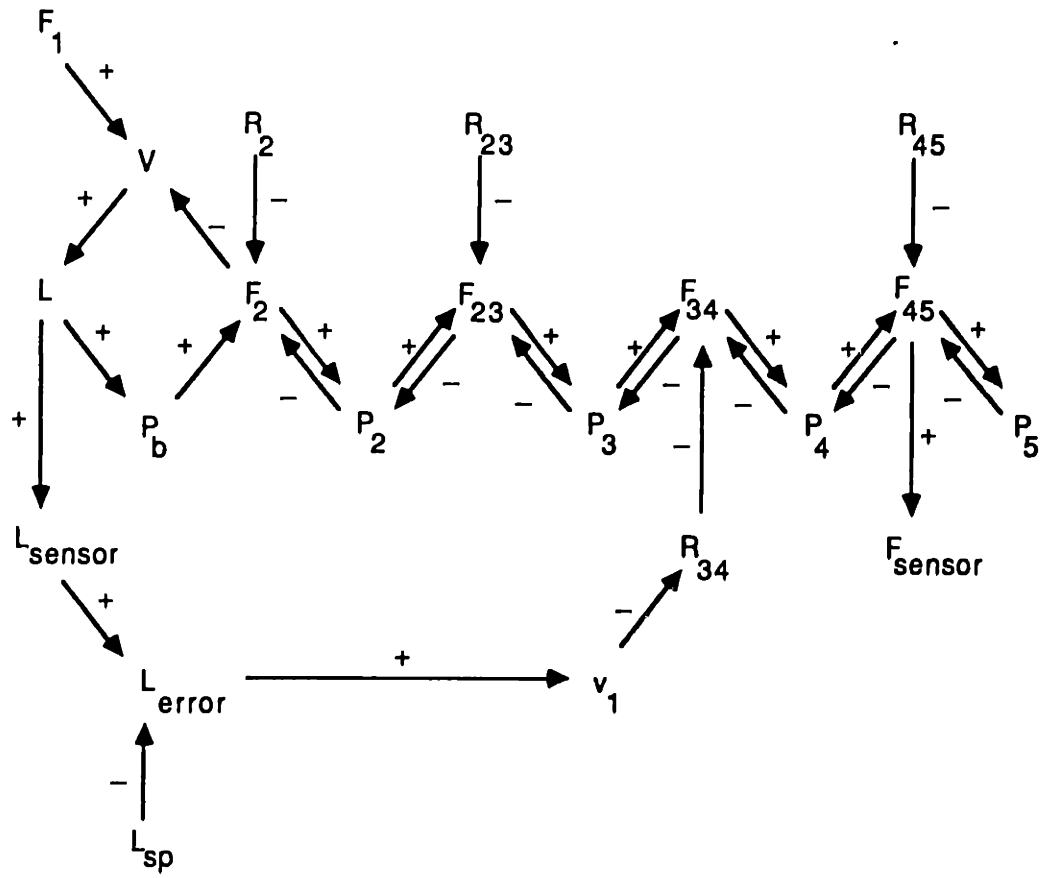


Figure 3-2
Causal Digraph for a Tank with a Level Control System

There is a clear distinction between the causal digraph as a data structure and the values assigned to the nodes to characterize a specific system state. The digraph, which represents the causal interactions between the process variables, is fixed as long as the assumptions and initial conditions used to build the digraph remain valid. On the other hand, the qualitative values assigned to the nodes, which represent the current state of the system, can change. Events (deviations in the physical system represented by the qualitative values '+' and '-' assigned to the nodes) are distinguished from the processes by which these events propagate through the system (represented by the digraph arcs).

3.1.2 Reasoning Using the Causal Digraph

Qualitative reasoning about the behavior of a physical system can be constructed from the causal digraph and the qualitative values assigned to the digraph nodes. A perturbation in one node will affect another node if a directed path from the given initial node to the terminal node exists. The qualitative value of the terminal node can be explained by the deviations that occur causally upstream in the path. Qualitative simulation is accomplished by building a directed tree from a specified root node and assigning consistent qualitative values to the nodes along each path.

To illustrate the explanatory power of the digraph representation, consider the question "Why does the flow rate measurement F_{sensor} increase when F_1 increases?" with the schematic in Fig. 3-1. The answer can be developed directly from the digraph in Fig. 3-2. Two paths are identified from F_1 to F_{sensor}. The process variable and its qualitative value are enclosed in parentheses.

1. An increase in inlet flow rate (F_1, +) increases the fluid volume in the tank (V, +), which increases the liquid level (L, +). Static pressure at the tank bottom increases (P_b, +), which tends to increase the flow rate (F_2, +) and pressure (P_2, +) at the tank's outlet. The pressures and flow rates increase in PIPE-A, CV-1, and PIPE-B. The increased flow rate (F_{45}, +) in PIPE-B causes the flow rate measurement to increase (F_{sensor}, +).

2. An increase in inlet flow rate (F_1, +) increases the fluid volume in the tank (V, +), which increases the liquid level (L, +). The increased fluid level is measured by the level sensor and causes the level measurement to increase (L_{sensor}, +). This causes the controller error to increase (L_{error}, +), which causes the control valve to open (v_1, +), which decreases the flow resistance (R_{34}, -) in CV-1. Flow rate increases in the control valve (F_{34}, +), PIPE-A, and PIPE-B. The increased flow rate (F_{45}, +) in PIPE-B causes the flow rate measurement to increase (F_{sensor}, +).

Although the causal sequence of high and low qualitative values along a path reads as if it is temporally ordered, the values of a set of nodes may change virtually simultaneously. Temporal ordering depends on the delay time attribute of each arc and the time scale chosen to increment the model predictions.

The clarity of the causal explanation is a function of the terms and concepts included in the digraph—the more rigorous the model, the greater its explanatory power. The causal digraph should include all terms (as nodes) that are important for reasoning in a given context. Because causal explanations are constructed from a path of adjacent nodes, if important interjacent nodes are not included, then the causal explanations provided by the digraph are less explicit.

The local nature of the digraph makes causal arguments computationally simple to construct. One limitation of the digraph is that ambiguous qualitative values arise when multiple paths of opposite sign converge at a digraph node. This difficulty will be discussed in Section 3.3.3.

3.2 Developing Causal Models for Engineered Systems

Modeling a physical system by a set of quantitative equations is the traditional approach for describing a system's behavior. Quantitative formulations, though, pose several difficulties. These difficulties include generating the set of equations, identifying the numerical values for system parameters, specifying boundary and initial conditions, and having a solution technique that can solve the quantitative system. One major drawback of quantitative models is that the system cannot be solved (and thus, cannot provide any information) unless it is fully specified.

If an application does not require a rigorous solution, it may be advantageous to develop an alternate representation that does not require the formulation and solution of the system of design equations. Fault diagnosis is one application that falls in this category. A qualitative description of the process and of the changes in the values of the process variables is sufficient for many of the diagnostic subtasks. Causal models overcome the requirement of a fully specified system because the models explicitly show the causal relationships between the process parameters and variables.

My objective is to construct models that will facilitate qualitative reasoning. This section describes how to derive the causal digraph for physical systems that can be characterized by a set of mathematical equations. Knowledge of the underlying physics is necessary to identify the causal relationships because the equations alone do not contain this knowledge.

3.2.1 Engineered Systems

The purpose for developing causal models is to qualitatively describe the behavior of engineered systems. Engineered systems are physical systems in which a thorough understanding of the fundamental principles and mechanisms exist, and which can be fully specified or described by a set of differential and algebraic equations. In the chemical engineering environment, a large fraction of the underlying physical principles are known, and standard quantitative models of heat, mass and momentum transfer, thermodynamics, kinetics, and process chemistry are widely used. Engineered systems differ from other domains like medicine, where a heuristic or rule-based approach is the standard format for knowledge representation. For example, in medical diagnosis, the underlying physical mechanisms are usually unknown, and diagnoses are generated directly from observed patterns of symptoms.

3.2.2 Knowledge of the Underlying Physics

The mathematical relationships and equations used to quantitatively model a physical process do not contain the causal information needed to specify the relationships between the process variables for causal modeling. For example, given the equation that describes the volumetric flow of fluid through a valve, will changing the upstream pressure change the flow rate or increase the valve coefficient? To make this point more clear, consider Equation 1. How do changes in **a** affect the other terms in the equation?

$$a - b = cd \tag{1}$$

The answer cannot be determined because the information necessary to specify how changes in one term affect the other terms is not contained in the equation. Therefore, causality cannot be specified solely from Eq. 1. Mathematical equations specify equality relationships between sets and combinations of parameters, but the equations contain no information on how changes in an individual process variable or parameter directly affect other system variables.

Knowledge of the physical principles and mechanisms behind each equation is necessary to specify causality. To illustrate, let Eq. 1 describe the flow of electricity in a conductor, given by Equation 2.

$$E_1 - E_2 = iR \tag{2}$$

Causal influences can be specified from an understanding of the physical system and the fundamental physical principles that the equation represents. The electromotive force is the driving force for current, i.e., current results from a difference in electric potentials. Resistivity, related to the resistance R, is a physical characteristic of the material and is independent of ΔE or i. From this physical understanding, the causal digraph is developed and presented in Figure 3-3. The causal digraph shows that an increase in emf (an increase in E_1, a decrease in E_2,

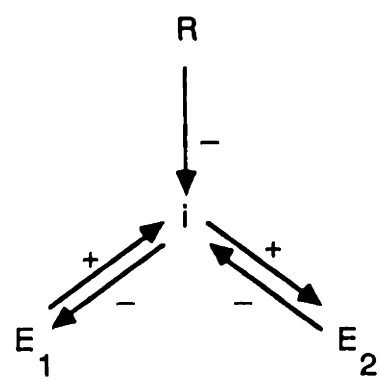


Figure 3-3
Causal Digraph for the Flow of Electricity in a Conductor

or both) increases the current. An increase in resistance decreases current, which has a tendency to increase E_1 and decrease E_2 relative to their nominal values. Because a causal path does not exist from E_1 to R, a change in E_1 cannot affect R. This knowledge (**a** does not affect **d**) is not contained in Eq. 1. The causal relationships are evident only with an understanding of the physical system.

3.2.3 Developing the Causal Digraph

For an engineered system, considerable expertise has usually been invested to develop a set of design equations. The system of design equations is useful for building causal models because they show the parameters and variables that are important (i.e., those that should be included in the causal digraph), but as discussed above, the equations alone provide no information about the causal interactions between the terms.

I have been successful at building causal digraphs by classifying the individual design equations into categories. These categories aid the construction of the digraph because they group together general types of design equations. For each class, a standard set of procedures specify the causal arcs and their sign attribute. The four categories of design equations used to construct the causal digraphs for the examples presented in this thesis are driving force equations, balance equations, functional relationships, and algebraic equalities.

3.2.3.1 Driving Force Equations

Driving force equations describe the steady-state rate of transport of material, momentum, and energy between two locations. These transport processes arise from gradients in pressure, temperature, and concentration, and are governed by the laws of fluid mechanics and by the generalized diffusion equation. Driving force equations include bulk mass flow due to a pressure difference, heat transfer by conduction, and species transfer by diffusion.

Figure 3-4a: Heat Flow Due To Temperature Gradient ($T_1 > T_2$)

Quantitative Equation

$$Q = UA(T_1 - T_2), \quad R = \frac{1}{UA}$$

Causal Digraph Arcs

$$Q = f(T_1, -T_2, -R)$$

$$T_1 = f(-Q)$$

$$T_2 = f(Q)$$

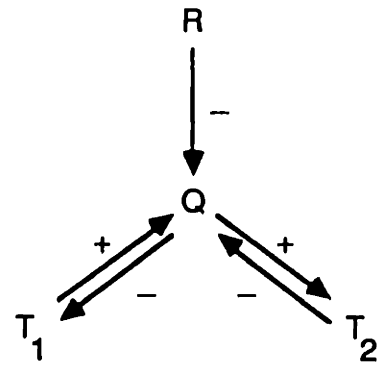


Figure 3-4b: Mass Flow Due To Pressure Gradient ($P_1 > P_2$)

Quantitative Equation

$$F = c_{12} f(P_1 - P_2), \quad R = \frac{1}{c_{12}}$$

Causal Digraph Arcs

$$F = f(P_1, -P_2, -R)$$

$$P_1 = f(-F)$$

$$P_2 = f(F)$$

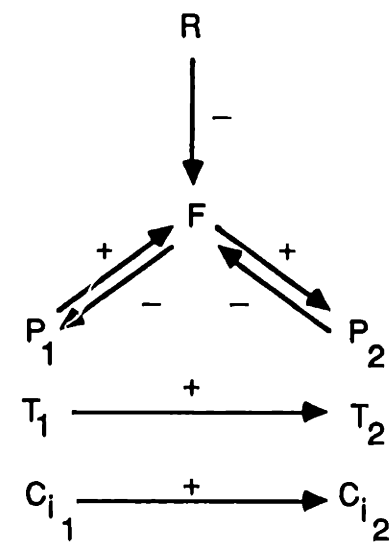


Figure 3-4
Causal Arcs for Driving Force Equations

The general causal model for driving force equations has four terms: upper and lower potentials for the driving force, flow, and resistance. From physics, flow rate increases when the driving force increases, increased resistance decreases the flow rate, and flow tends to reduce the driving force. This knowledge defines the causal arcs for this class of equations. Heat and bulk mass flow examples, presented in Figure 3-4, illustrate the general digraph model. In both examples, the potential at state 1 is greater than the potential at state 2. If state variables and physical properties characterize the flow, then causal arcs are necessary to show their transport. In Fig. 3-4b, causal arcs for temperature and species concentration represent the transport of these quantities in the direction of bulk mass flow. If measurement is on the order of seconds, the time attribute for the arcs characterizing the driving force, flow, and flow resistance is zero. Arcs characterizing the transport of state variables and physical properties usually have positive delay time.

3.2.3.2 Balance Equations

Balance equations describe the conservation of mass, energy, and species within process units. Causal paths are obtained from the unsteady-state form of the balance equation. Arcs exist from the right-hand-side parameters and process variables to the dependent term in the derivative. Expressed mathematically, given $\frac{dx_i}{dt} = f_i(\underline{x})$, a causal arc exists from x_j to x_i if $\frac{\partial}{\partial x_j}\left(\frac{dx_i}{dt}\right) \neq 0, \quad i \neq j$. The sign attribute of the arc is the sign of the partial derivative evaluated at expected (normal) conditions. In Figure 3-5, mass and energy balances around a tank illustrate the derivation of causal arcs for balance equations.

When simplifying and solving balance equations, driving force equations and functional relationships are normally substituted into the balance equation. For example, the substitution of $Q = UA(T - T_r)$ and $r = kC_A^2$ eliminates Q and r, respectively, from the equation. Note that the substitution also eliminates these variables from the digraph. The causal digraph should be developed without these substitutions because, as

Figure 3-5a: Mass Conservation Around Tank With Two Inlets (Subscripts 1 and 2) and One Outlet (Subscript 3)

Assumptions: 1. Constant density of fluid (ρ)
2. $F_1, F_2, F_3 > 0$

Quantitative Equation

$$\frac{dV}{dt} = F_1 + F_2 - F_3$$

Causal Digraph Arcs $V = f(F_1, F_2, -F_3)$

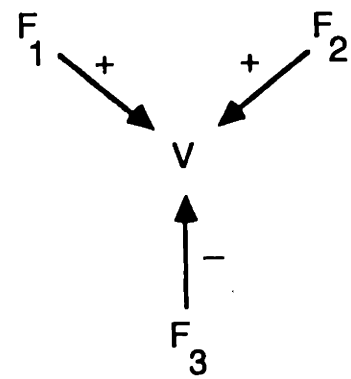


Figure 3-5b: Energy Conservation Around Tank With An Exothermic Chemical Reaction

Assumptions: 3. Well-mixed tank (uniform bulk properties, e.g., $T_3 = T$)
4. Constant heat capacity of fluid (C_p)
5. Constant heat of reaction (ΔH_r)
6. $T > T_1, T > T_2$

Quantitative Equation

$$\frac{dH}{dt} = \rho C_p \frac{d(VT)}{dt} = \rho C_p(F_1T_1 + F_2T_2 - F_3T) + \Delta H_r rV$$

$$\rho C_p \frac{d(VT)}{dt} = \rho C_p\left[V\frac{dT}{dt} + T\frac{dV}{dt}\right] = \rho C_p\left[V\frac{dT}{dt} + F_1T + F_2T - F_3T\right]$$

Figure 3-5

Causal Arcs for Balance Equations

$$\rho C_p \frac{dT}{dt} = \frac{\rho C_p}{V}\left[F_1(T_1 - T) + F_2(T_2 - T)\right] + \Delta H_r r$$

$$\frac{dT}{dt} = -\frac{F_1}{V}(T - T_1) - \frac{F_2}{V}(T - T_2) + \frac{\Delta H_r}{\rho C_p} r$$

Causal Digraph Arcs $\quad T = f(-F_1, T_1, -F_2, T_2, V, r)$

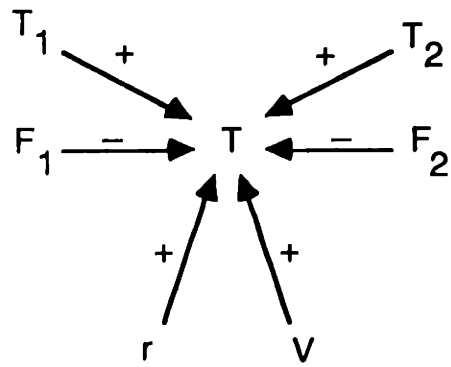


Figure 3-5 (cont.)
Causal Arcs for Balance Equations

will be shown in Section 3.4.2, this elimination of variables may cause spurious interpretations of the digraph.

3.2.3.3 Functional Relationships

Functional relationships explicitly show how one or more independent variables influence a given dependent variable. Examples of functional relationships in physical systems include process measurements, control system outputs, kinetic rate expressions and other empirical relationships. Arcs exist from the independent variables to the dependent variable. Given $x_i = f_i(\underline{x})$, a causal arc exists from x_j to x_i if $\frac{\partial x_i}{\partial x_j} \neq 0$, $i \neq j$. The sign attribute of the arc is the sign of the partial derivative evaluated at expected (normal) conditions. Generally, the time attribute for all arcs derived from functional relationships is '0', except when the function is a time integral.

3.2.3.4 Algebraic Equalities

Because the terms in an equality relationship can be rearranged to solve for any of the parameters or variables in the equation, knowledge of the underlying physics is necessary to specify the correct causal relationships. Causal arcs are identified by evaluating how deviations in each of the terms affect the other terms in the equality. For example, consider density, defined as $\rho = m/V$. Given the values of any two of the terms, the third can be calculated. But to specify the causal interactions, the effects of changes in ρ, m, and V on the other terms in the equation must be understood. Changes in density are caused by changes in mass or volume, yielding an arc with positive sign from m to ρ and an arc with negative sign from V to ρ. Changes in mass cannot be caused by changing ρ or V, and thus, arcs do not exist from these terms to m. A second example is the volume of fluid in a tank, given by $V = AL$. The volume of fluid can be changed only by changing the flow of mass into or out of the tank, and not by changing the cross-sectional area or liquid level. Level can change by changing the apparent volume of fluid (e.g., by adding more mass or dropping a more dense object into the tank), or by changing the cross-sectional area. Therefore, this equation generates two causal arcs, one with positive sign from V to L, and the second with negative sign from A to L.

In the two examples considered, only a single arc connected any two terms. Algebraic equalities can also yield two causal arcs of opposite direction between two variables. Consider Pascal's law: the magnitude of the pressure at any point in a fluid at static equilibrium is equal in all directions. Given two pressures in a static fluid, neither pressure is independent of the other. Because they must change together, two opposite causal arcs, each with a positive arc sign, connect the pressures. A second example is a constant volume tank with both liquid and vapor phases. The tank is described by the equation $V = V_v + V_\ell$. Neither liquid volume nor vapor volume is independent; if one changes, then so must the other. Therefore, both volumes are connected by causal arcs of negative sign.

The rule for specifying the sign attribute of causal arcs derived from algebraic equalities is similar to the rule for functional relationships: the sign attribute is the sign of the partial derivative evaluated at

expected (normal) conditions. The time attribute for algebraic equalities, assuming a time scale of order seconds, is usually '0'.

3.2.4 Removal of Digraph Arcs and Nodes

Modifications of the causal digraph are necessary to accurately model a specific physical system. Context-specific assumptions and values may remove digraph arcs and nodes.

3.2.4.1 Parameters and Process Variables at Fixed Values

If the value of a digraph node is fixed or assumed constant, then the digraph node and the causal arcs terminating on and leaving the node can be removed. Several examples of fixed digraph node values are constant physical properties (e.g., ρ, C_p, ΔH_r, λ), P, V, and T in isobaric, isometric, and isothermal systems, respectively, and physical constants (e.g., g, R). In Fig. 3-5b, density, heat capacity, and heat of reaction were assumed constant. These nodes were removed from the digraph as well as the arcs from these nodes to T.

Faults can change the value of an assumed constant node. Two examples of faults affecting assumed fixed values are catalyst degradation changing the reaction rate constant, and blockage by a foreign object changing the flow resistance in a pipe. Given a node with a fixed value, if it is desired to diagnose faults that can change the node's value, then that node must remain in the digraph. (See Section 3.4.2.1).

3.2.4.2 Arcs With Small Magnitudes

A causal arc is removed from the digraph if the magnitude of the resultant effect is not observable at the terminal node. For example, consider the assumption of an incompressible fluid. If a given liquid is assumed to be incompressible, then changes in pressure have a negligible effect on the volume of the liquid. Although the causal arc from pressure to volume exists, the magnitude of the causal effect is infinitesimal. Note that the volume of the liquid is not fixed—liquid volume can change

by adding or removing fluid. The emphasis is on the magnitude of the causal effect. A second example is atmospheric pressure. Increasing the liquid level in a tank open to the atmosphere will not measurably increase atmospheric pressure.

When the magnitude of transmittance of the causal arc is small, the arc can be directly removed from the causal digraph, or the arc's magnitude attribute can be assigned the value '0'. The assignment of zero is preferred when it is important to show that theoretically, the arc exists, but that the propagation of a disturbance along the arc could not cause a deviation at the terminal node.

3.2.5 Examples of Causal Digraph Construction

Two examples are presented that illustrate the construction of causal digraphs from a set of design equations. In each example, the causal arcs generated for each quantitative equation are presented on the right. Arcs exist from the terms in the functional description to the dependent term. The sign of each functional term is the sign attribute of the causal arc. Note that only arcs with sufficient magnitudes are shown in the figures, and that the removal of digraph nodes and arcs depends on the specific context, determined by the stated assumptions. The rationale for selecting the variables and parameters to include in the digraph, such as reaction rate and space time, will be explained in Section 3.4.2.

Example 1: Isothermal Tank

Construct the causal digraph for an isothermal tank of constant volume, with two inlet ports (subscripts 1 and 2) and one outlet port (subscript 3). Fluid exists in the tank in both liquid and vapor phases. The tank is closed to the atmosphere. Inlet ports are above liquid level. Arcs and nodes related to bulk fluid flow are developed.

Assumptions: 1. Liquid is incompressible

2. Constant physical properties of fluid (ρ)

3. Constant tank cross-sectional area (A) and total volume (V)

4. $F_1, F_2, F_3 > 0$

The schematic is shown in Figure 3-6.

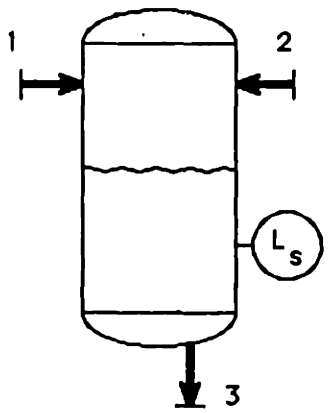


Figure 3-6
Process Schematic of an Isothermal Tank

Quantitative Equations	Causal Digraph Arcs
1. Mass flow rate across each port	
$F = \frac{1}{R} f(\Delta P)$	$F_1 = f(P_1, -P_v, -R_1)$
	$P_1 = f(-F_1)$
	$F_2 = f(P_2, -P_v, -R_2)$
	$P_2 = f(-F_2)$
	$F_3 = f(P_b, -P_3, -R_3)$
	$P_3 = f(-F_3)$
2. Conservation of mass in tank	
$\frac{dV_\ell}{dt} = F_1 + F_2 - F_3$	$V_\ell = f(F_1, F_2, -F_3)$

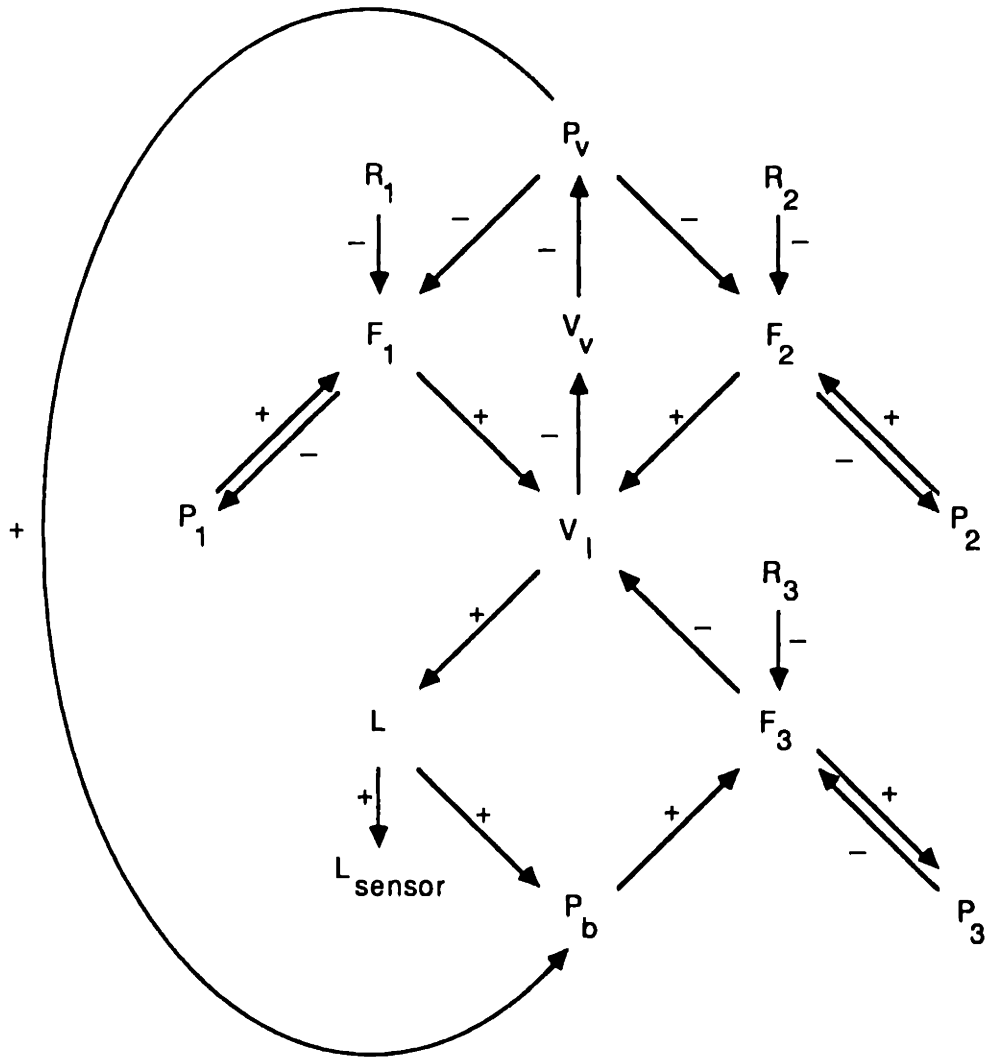


Figure 3-7
Causal Digraph for an Isothermal Tank

3. Static pressure head

$$P_b = \rho g L + P_v \qquad P_b = f(L, P_v)$$

4. Tank level

$$L = \frac{V_\ell}{A} \qquad L = f(V_\ell)$$

5. Level sensor $\qquad L_{sensor} = f(L)$

6. Total tank volume

$$V = V_v + V_\ell \qquad V_\ell = f(-V_v)$$

7. Equation of state for vapor $\qquad P_v = f(-V_v)$

The arcs form the causal digraph in Figure 3-7.

Explanation generated from the causal digraph for a blockage in the tank outlet: "Blockage in the tank outlet (R_3, +) tends to decrease outlet flow rate (F_3, -) and tends to cause the loss of pressure downstream (P_3, -). A decrease in outlet flow rate causes the liquid volume to increase (V_ℓ, +), which increases the level (L, +), the level sensor (L_{sensor}, +), and the pressure at the tank bottom (P_b, +). Increasing the liquid volume decreases the vapor volume (V_v, -) and tends to increase the pressure of the vapor (P_v, +). The inlet flow rates tend to decrease (F_1, -) (F_2, -) and the upstream pressures tend to increase (P_1, +) (P_2, +)."

Example 2: Liquid-Phase Reaction in a CSTR

Construct the causal digraph for the liquid-phase decomposition reaction $A \rightarrow B + C$ in a CSTR. The reaction is exothermic and first order in A. The reactor has one inlet (subscript 1) for reactant entry and one outlet (subscript 2) for product removal. The reaction occurs at atmospheric pressure. Arcs and nodes related to chemical reaction are developed.

Assumptions: 1. Well-mixed CSTR (uniform bulk properties)
2. Constant physical properties of fluid (ρ and C_p)
3. Constant heat of reaction (ΔH_r)
4. F_1, $F_2 > 0$
5. $T_1 < T$

The schematic is shown in Figure 3-8.

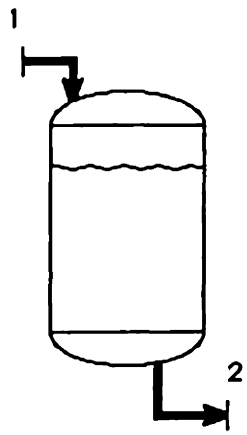


Figure 3-8
Process Schematic of a Liquid-Phase Reaction in a CSTR

Quantitative Equations	Causal Digraph Arcs
1. Conservation of mass	
$\frac{dV}{dt} = F_1 - F_2$	$V = f(F_1, -F_2)$
2. Reactor space time	
$\theta = \frac{V}{F_1}$	$\theta = f(V, -F_1)$
3. Conservation of species	
$\frac{dN_A}{dt} = F_1C_{A_1} - F_2C_A - rV$	

$$\frac{dN_A}{dt} = \frac{d(C_A V)}{dt} = C_A \frac{dV}{dt} + V\frac{dC_A}{dt} = C_A(F_1 - F_2) + V\frac{dC_A}{dt}$$

$$V\frac{dC_A}{dt} = F_1(C_{A_1} - C_A) - rV$$

$$\frac{dC_A}{dt} = \frac{1}{\theta}(C_{A_1} - C_A) - r \qquad C_A = f(-\theta, C_{A_1}, -r)$$

$$\frac{dN_i}{dt} = - F_2 C_i + rV \qquad \text{for } i = B, C$$

$$\frac{dN_i}{dt} = C_i(F_1 - F_2) + V\frac{dC_i}{dt}$$

$$V\frac{dC_i}{dt} = - F_1 C_i + rV$$

$$\frac{dC_i}{dt} = \frac{1}{\theta}C_i + r \qquad C_B = f(\theta, r)$$

$$C_C = f(\theta, r)$$

4. Conservation of energy

$$\frac{dT}{dt} = - \frac{1}{\theta}(T - T_1) + \frac{\Delta H_r}{\rho C_p} r \qquad T = f(\theta, T_1, r)$$

5. Reaction rate

$$r = kC_A \qquad r = f(k, C_A)$$

6. Reaction rate constant

$$k = k_0 e^{- E/RT} \qquad k = f(T)$$

The arcs form the causal digraph in Figure 3-9.

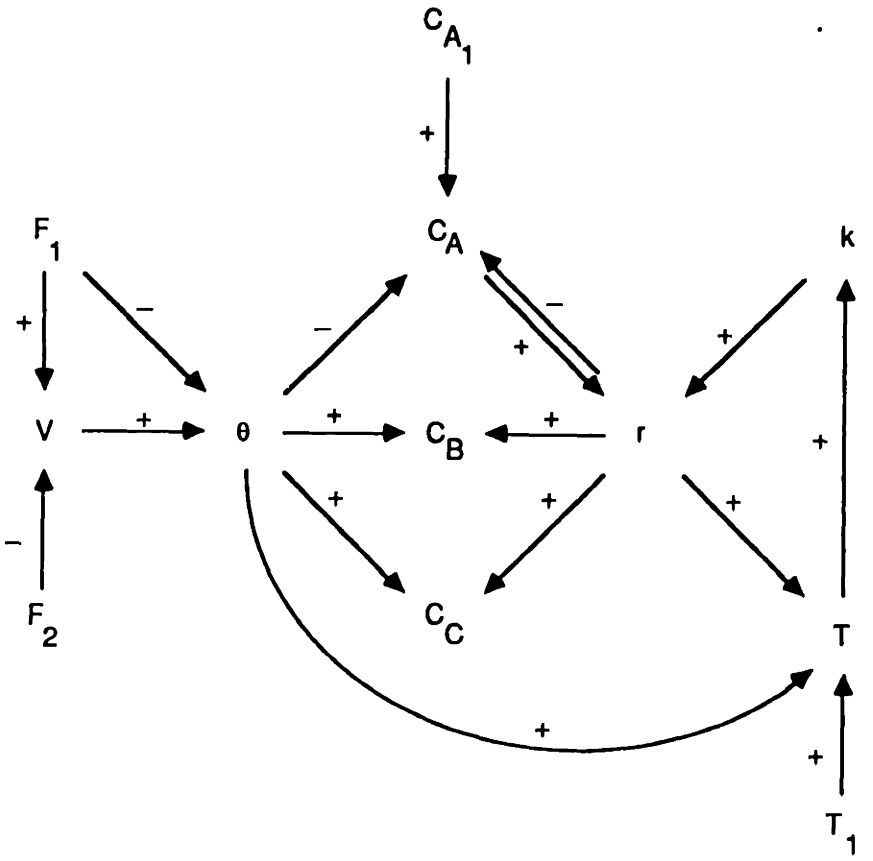


Figure 3-9
Causal Digraph for a Liquid-Phase Reaction in a CSTR

Explanation generated from the causal digraph for catalyst degradation: "Catalyst degradation (k, -) decreases the reaction rate (r, -), and tends to increase the concentration of reactant (C_A, +) and decrease the concentrations of the products (C_B, -) (C_C, -). The reactor temperature decreases (T, -) as reaction rate falls."

Explanation generated from the causal digraph for increased reactor throughput: "An increase in reactor throughput (F_1, +) (F_2, +) decreases the reactor space time (θ, -), causing the reactant concentration to increase (C_A, +) and the product concentrations to decrease (C_B, -) (C_C, -). An increase in reactant concentration has a tendency to increase the reaction rate (r, +), the product concentrations (C_B, +) (C_C, +), the

reactor temperature (T, +), and the rate constant (k, +). A decrease in the space time will tend to decrease the reactor temperature (T, -), causing a decrease in the rate constant (k, -) and the reaction rate (r, -)."

Note that without quantitative information to determine the dominant causal path, the qualitative changes in r, T, k, C_B, and C_C are unknown.

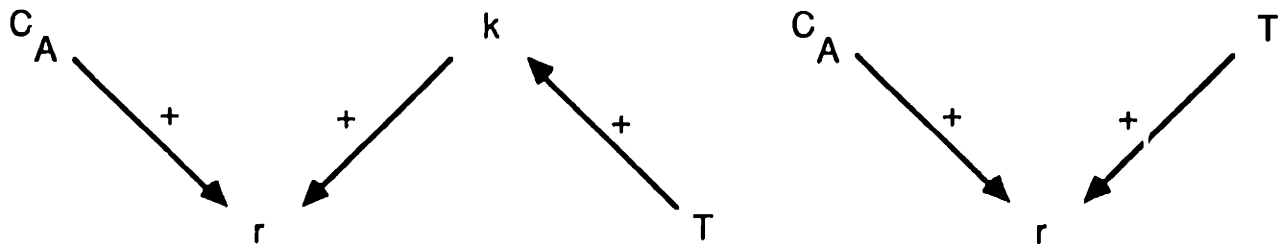
3.3 Limitations of the Causal Digraph

The limitations of the causal digraph are (1) the causal digraph is not unique, (2) the causal digraph cannot explicitly handle discontinuities that occur in the physical system, and (3) the causal digraph introduces ambiguities when determining qualitative parameter states because constraints on system behavior are lost on finer levels of detail. The use of global information is investigated as a means of eliminating ambiguity.

3.3.1 Causal Digraph Uniqueness

The causal digraph is not unique. Several different causal models can be developed to represent the same physical system; the differences between them arise from the parameters chosen to be included in the model. If the quantitative equations selected to model the system are more detailed, then the resulting causal digraph will be more detailed.

Nodes in a given causal model can be removed and causal paths can be combined to obtain less detailed models. This procedure is analogous to combining quantitative equations to eliminate variables. An example illustrating how the causal model is dependent upon the characterization of the physical system by quantitative equations is presented in Figure 3-10. Two models for the chemical reaction rate and reaction rate constant are developed from equations. If the two equations $r = kC_A$ and $k = k_0 e^{-E/RT}$ are used to develop the digraph, then the rate constant k is a digraph node (Fig. 3-10a). If the expression for k is substituted into the rate expression to yield the single equation $r = C_A k_0 e^{-E/RT}$, then k is eliminated from the digraph and a causal arc exists directly from T to r (Fig. 3-10b).



3-10a 3-10b

Figure 3-10
Causal Digraphs for a First-Order Reaction and Reaction Rate Constant

Since the different models characterize the same process, they both must generate qualitative behaviors that match the actual physical system, i.e., for all possible combinations of inputs, they both must produce the outputs actually realized by the physical process. The actual behavior may be one of a set of behaviors predicted by the causal model (see Section 3.3.3).

Both models presented in the example are equally valid representations. Usually, several ways exist to model a system, and, as in quantitative modeling, engineering judgment is required for building the most appropriate model. The parameters chosen to be included in the model depend upon their importance to the objective of the modeling task. The selection of a specific model depends upon the requirements of the individual application.

3.3.2 Discontinuities

The causal digraph cannot handle discontinuities that arise in the physical system. Arcs in the digraph represent the causal influences between the important process variables and parameters, but they do not contain any information about abrupt changes in the physical system. For example, if a liquid is subjected to an influx of heat, the temperature of the liquid will increase. But the causal arc from Q to T contains no information about if or when the liquid will boil.

Discontinuities are discrete changes in the physical system that require a different set of quantitative equations to accurately model the physical system. Thus, when the system changes at a discontinuity, a different digraph is necessary. When vaporization begins, adding heat does not increase the temperature. Rather, when two phases exist, the addition of heat increases the amount of vapor and decreases the amount of liquid. In regions of different qualitative behavior, different causal digraphs are required. If the discrete changes in the physical system are known, then they can serve as preconditions to the existence of causal arcs.

3.3.3 Ambiguous Qualitative Parameter States

Qualitative modeling can be interpreted as the problem of constructing the actual global behavior of a system from the local behavior of its components. A qualitative description contains less information than a quantitative description about the magnitudes of process parameters, and therefore, the causal digraph may not be deterministic. The causal digraph introduces ambiguities when determining qualitative parameter states because constraints on system behavior are lost on finer levels of detail. Thus, there may be several plausible candidates for global behavior.

Ambiguity in Causal Models

Ambiguities arise in causal models when influences of opposite signs act simultaneously on a given process parameter. Because the causal digraph is a qualitative description of behavior, numerical, context-specific information on the magnitudes of causal influences is

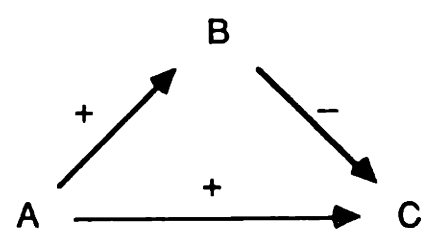


Figure 3-11
Ambiguity in Causal Models

absent. Multiple interpretations for process variable deviations are obtained. Figure 3-11 illustrates this ambiguity. In the figure, there are two causal paths from node A that terminate on node C: one direct path of positive sign, and one indirect path through node B with a resultant negative sign. For a deviation at node A, the resultant deviation at C cannot be determined without the numerical values for the magnitudes of the deviations along each arc. Ambiguity arises from the loss of quantitative information necessary to resolve the summation of multiple pathways of opposite sign.

Level of Detail

An increased level of detail in the causal digraph may eliminate information that constrains qualitative parameter states and adds spurious interpretations to the digraph. As described in Section 3.1.1, causal influences are local interactions, represented by adjacent nodes in the causal digraph. Local interactions are dependent upon the level of detail chosen to model the system. In a coarse level of detail, two nodes may be adjacent, whereas in a model of greater detail, the same two nodes may be separated by intervening nodes. Thus, the level of detail chosen to represent the physical system defines what information is local and what information is global. Local information at one level of detail is global information on a finer level of detail.

Because the digraph represents only local causal interactions, global constraints on the system's behavior, including mass and energy balances over several units, are not included in the digraph. It is this lack of global information that results in ambiguous qualitative parameter values.

An example of how ambiguities arise from finer levels of detail is presented in Figure 3-12. On a coarse level of detail (Fig. 3-12a), a system component is modeled as a black box. Its behavior is represented by a single causal arc $w \rightarrow z$ between its inlet and outlet ports. On a level of greater detail (Fig. 3-12b), the single arc is actually the dominant path $w \rightarrow x \rightarrow z$ of positive sign. On a coarse level of detail, the deviations in w and z are constrained to vary in the same direction. On a greater level of detail, w and z can assume opposite signs. Different qualitative values at node z are consistent with the causal digraph of

Figure 3-12b because the quantitative information necessary to specify a positive resultant magnitude between the inlet and outlet ports is unavailable. Because the predicted model behavior of opposite signs for w and z is not exhibited by the actual physical system, these interpretations of the causal digraph are spurious.

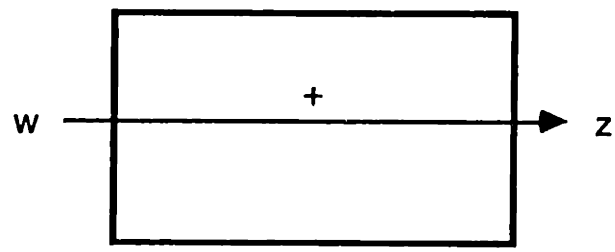


Figure 3-12a: Coarse Detail

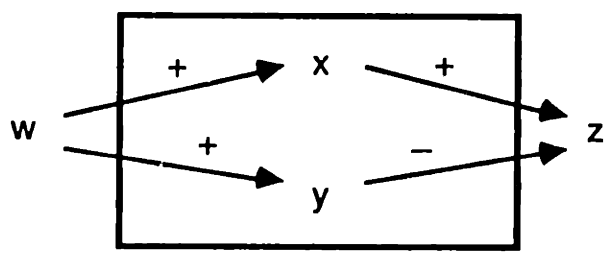


Figure 3-12b: Fine Detail

Figure 3-12
Ambiguities Introduced Through Finer Levels of Detail

In summary, greater detail may increase the number of digraph interpretations because local constraints on behavior become global constraints on finer levels of detail. These global constraints are not represented in the digraph. When trying to construct overall system behavior from individual component models, ambiguities may arise when determining qualitative parameter values because global constraints are unknown. Local models may not contain enough information to specify a unique, dominant causal pathway. Global knowledge, then, is necessary to specify the correct behavior.

3.3.4 Representation of Global Information

Because the causal digraph is limited to local interactions, global information may be necessary to constrain spurious interpretations. If global knowledge is known, then it should be retained and incorporated for qualitative reasoning. Two approaches for incorporating global information are (1) a hierarchy of digraph models, and (2) qualitative equalities.

A hierarchy of digraph models can be used to eliminate spurious interpretations. Given a set of digraphs that characterize the same system, if the predictions of a particular model are not consistent with all the other system models, then the predictions generated from that particular model must be spurious, because all the models characterize the same physical system. Although the two digraphs in Fig. 3-12 characterize the same system, several of the interpretations generated in Fig. 3-12b are not consistent with the digraph in Fig. 3-12a. The digraph of less detail can be used as a filter to eliminate the ambiguities introduced by the digraph of greater detail. Interpretations that are not consistent with all the models are discarded.

A second approach for representing global information is the use of qualitative equalities. A causal arc on a coarse level of detail can be considered a constraint on the behavior of a digraph on a finer level of detail. The arc $w \rightarrow z$ in Figure 3-12a places a restriction on the possible qualitative values of z: the deviation in z must be in the same direction as the deviation in w. This constraint can be expressed by the qualitative equality

$$[w] = [z]. \tag{3}$$

This local equality (local in the context of the digraph in Figure 3-12a) can be used to constrain the behavior in more detailed digraphs (where the equality becomes a global constraint). The equality of Eq. 3 and the digraph of Figure 3-12b yield the same behaviors as the digraph in Figure 3-12a, while providing greater understanding of the causal interactions in the component.

Qualitative equalities are valid only when a deviation causally propagates through the component. If, for example, a fault occurs within the component (the fault affects a node between the parameters of the equality), then the equality is not valid. In Figure 3-12b, if a fault caused y to deviate, and z deviated due to fault propagation along the causal arc from y, then the equality of Eq. 3, which constrains the behavior of w and z, does not apply.

Note that in the qualitative equality, all information about causality (e.g., direction, magnitude, time delay, etc.) is absent. Also, qualitative equalities hold only after the causal effect has reached the terminal node.

3.4 Causal Digraphs for Fault Diagnosis

The procedures for developing causal digraphs, presented in Section 3.2, are general. They can be used to develop digraphs for any system that can be characterized by a set of mathematical equations.

The selection of an appropriate model for a problem depends on the kinds of reasoning to be done and the characteristics of the domain. In Section 3.3.1, it was mentioned that several different digraphs could be constructed to model a given system. Since the purpose for developing causal models is to construct diagnostic systems, the digraph used, then, should be the one that is most suited for diagnosis. In this section, I present guidelines for developing and modifying causal digraphs for fault diagnosis. A digraph is suitable for fault diagnosis if it contains a single node representing the primary effect of the fault for every fault desired to be diagnosed, and it contains the most information about the system to minimize the number of incorrect faults identified and maximize the resolution between faults. These guidelines are explained.

3.4.1 A Digraph Node For Every Fault

The faults desired to be diagnosed must be identified before the digraph is constructed. The faults chosen indicate which process variables

must be included in the quantitative equations so that the nodes corresponding to primary deviations are included in the causal digraph. A primary deviation is defined as the process variable or parameter whose deviation is the primary effect of the fault on the system. Because fault candidates are generated by identifying possible primary deviations (the procedure is described in Chapter 4), if the node associated with the actual fault is not in the digraph, the fault cannot be identified. For example, if faults about species concentration are important, but nodes representing concentration are not included in the digraph, concentration faults cannot be identified.

3.4.2 Greater Knowledge About the Physical System

Greater knowledge about a physical system is represented by a causal digraph with a greater number of digraph nodes. More information about the system can add resolution between individual faults, minimize the number of incorrect fault candidates generated by decreasing the number of spurious digraph interpretations, and retain the concept of causality.

- Add resolution between individual faults

Fault resolution depends on the number of faults mapped to a primary deviation. If ten faults can cause a deviation at a particular node, then the best resolution that can be obtained when that node is identified as a primary deviation is ten faults. If there are ten nodes, each associated with a single fault, then the identification of a single primary deviation identifies one fault. For a given set of measurements, the best possible resolution is obtained if no two faults are associated with a given primary deviation.

- Decrease the number of spurious digraph interpretations

Additional digraph nodes may reduce the number of spurious digraph interpretations, and hence, reduce the number of incorrect fault candidates.

Two examples of reducing spurious interpretations through greater knowledge are presented in Figure 3-13. Assume that the digraphs illustrated in Figs. 3-13a and 3-13c are used to represent two systems whose actual causal interactions can be represented by Figs. 3-13b and 3-13d, respectively. The addition of node E in Fig. 3-13a and node D in Fig. 3-13c to match the real system eliminate spurious interpretations. The qualitative values (C, +) and (D, -) are consistent with two interpretations of the digraph in Fig. 3-13a: (A, +) and (B, +), and (A, -) and (B, -). Neither of these interpretations are consistent with the actual digraph in Fig. 3-13b because node E would have to assume both '+' and '-' qualitative values. Hence, these interpretations are spurious. In Fig. 3-13c, the set (B, +), (A, -), and (C, -) is consistent with the digraph, but these values are inconsistent with the actual physical system represented by Fig. 3-13d. For this set to be valid, node D would have to take both '+' and '-' values. Greater knowledge about the system, represented by more digraph nodes, reduces the number of spurious digraph interpretations.

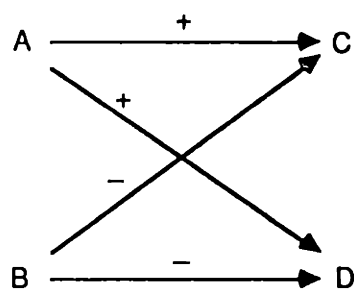


Fig. 3-13a: Digraph Representation

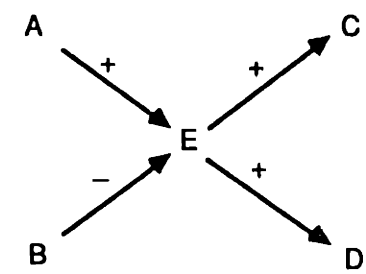


Fig. 3-13b: Actual Causal Interactions

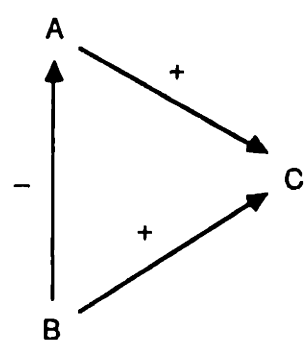


Fig. 3-13c: Digraph Representation

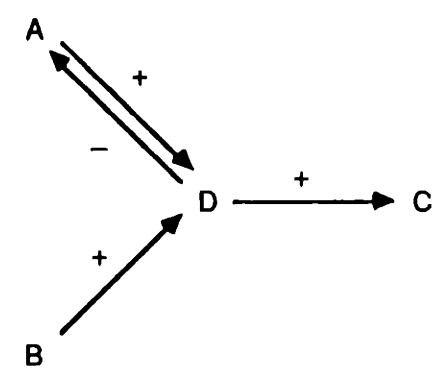


Fig. 3-13d: Actual Causal Interactions

Figure 3-13

Adding Nodes to Decrease the Number of Spurious Interpretations

Increasing model detail (adding digraph nodes) was shown to add spurious interpretations in Section 3.3.3. If the addition of nodes to the digraph results in paths of opposite sign from any one node to another, then spurious interpretations arise. Note though, that global information is generated during the addition which can be used to prune the spurious path. Here, the digraphs in Figs. 3-13a and 3-13c are approximations of the actual physical system. The addition of the digraph node represents greater knowledge about the system because an underlying process variable is made explicit. Spurious interpretations are eliminated because the qualitative value of the node is constrained to a single deviation.

- Retain the concept of causality

The removal of digraph nodes, resulting from the removal of process variables through equation substitution, blurs the concept of causality. For example, in Fig. 3-2, if the nodes along the causal path between F_1 and v_1 were removed through substitution, then these nodes would be adjacent and a causal arc would exist directly from F_1 to v_1. This arc would be interpreted as "increasing inlet flow rate increases the value stem position." The actual, underlying causal relationships are hidden.

In summary, greater knowledge about the physical system, represented by a greater number of digraph nodes, can improve the resolution of individual faults, decrease the number of spurious interpretations, and improve the explanatory power of the causal digraph.

3.4.3 Structural Faults

The causal digraph for a process unit must be able to (1) describe the process variable deviations due to the propagation of a disturbance through the unit, and (2) describe the deviations due to a fault occurring within the unit. Because a disturbance is a change in the magnitude of one or more process variables, the digraph developed from the equations that describe normal operation can always characterize the behavior of fault propagation through the unit. But the digraph developed for normal operation cannot always satisfy the second requirement.

Faults can be grouped into two categories: faults that change the magnitude of a process variable or parameter, and faults that change the form of the system of equations which characterizes normal behavior. Faults in the first category can be diagnosed from the digraph developed for normal operation. The effect of a fault in this class is to change the qualitative value at one of the digraph nodes, while leaving the causal digraph of the unit unchanged. Faults in the second category, called structural faults, change the system of differential and algebraic equations that represent normal operation. Structural faults change the system of equations by changing the form of one or more of the existing equations, or by adding additional equations to the system which are necessary to model the specific fault. If a structural fault occurs within a unit and the fault has not been incorporated into the digraph, then the digraph no longer accurately represents the actual physical behavior, and the digraph predictions of deviations will be incorrect.

Structural faults cannot be diagnosed from the digraph constructed for normal operation because a single primary deviation does not exist. The effect of a structural fault on the digraph characterizing normal operation is to cause two nodes that are not causally connected to simultaneously deviate. Because the fault makes these nodes causally related, the digraph no longer represents the actual system behavior. New nodes and arcs must be added to the normal digraph model to characterize the behavior of the system with the fault.

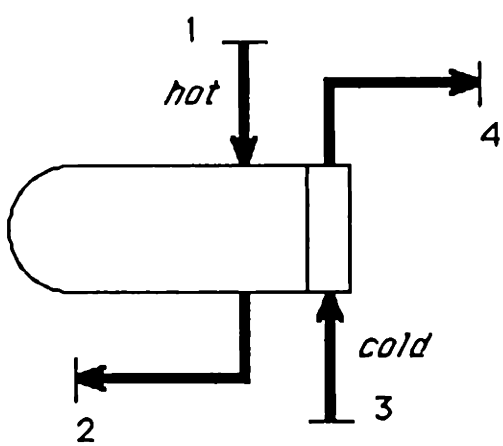
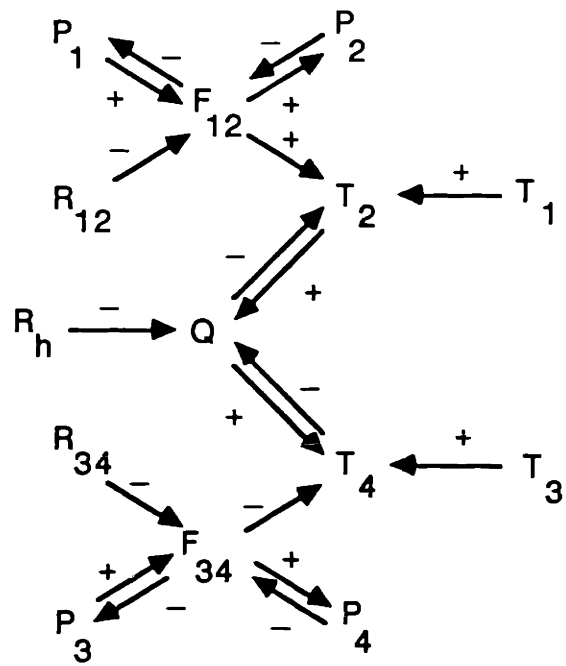
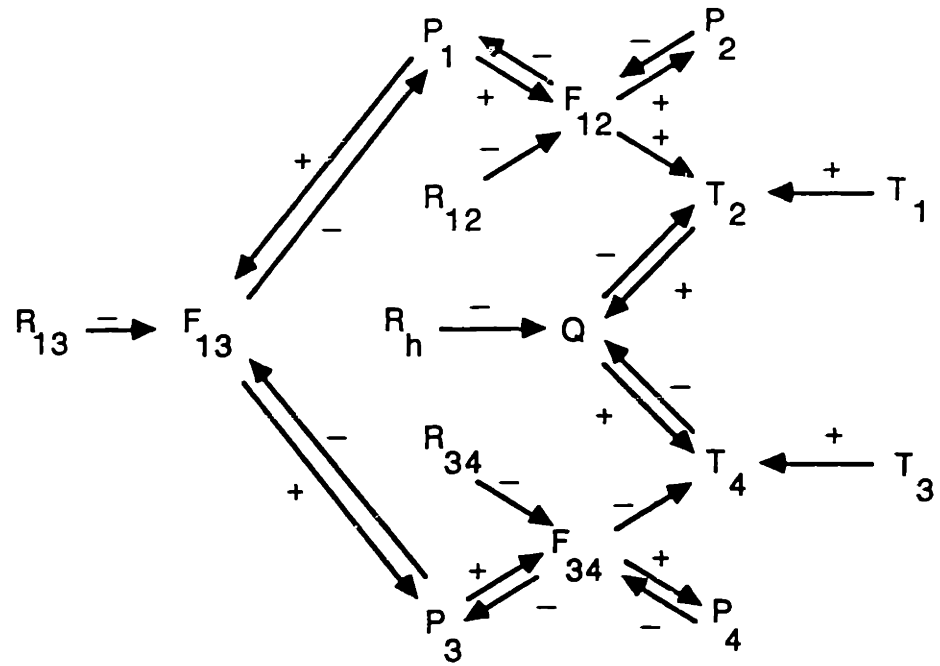


Figure 3-14
Process Schematic of a Heat Exchanger



3-15a



3-15b

Figure 3-15
Causal Digraphs for a Heat Exchanger

An example of a fault that requires modifications to the causal digraph is a leak between the hot and cold streams in a heat exchanger. Consider the heat exchanger shown in the schematic in Figure 3-14. The causal digraph, developed from design equations, is presented in Figure 3-15a. Note that the only expected causal interaction between the two streams is through heat transfer. A leak between the shell and tube sides in the heat exchanger results in bulk fluid flow from higher to lower pressure. If P_1 and P_2 are greater than P_3 and P_4, then a leak tends to decrease P_1 and increase P_3. For this fault to be identified, the causal arcs that represent the leak (bulk flow between the hot and cold streams) must be included in the digraph. These arcs are shown in the digraph in Fig. 3-15b. Under normal operation, the expected value of the resistance R_{13} is infinite, so the value of F_{13} is zero. A leak is a decrease in R_{13}, which causes F_{13} to increase, P_1 to decrease and P_3 to increase.

The guidelines for developing causal digraphs are summarized below.

Guidelines for Developing Causal Digraphs for Fault Diagnosis

1. Identify the faults to be diagnosed prior to developing the causal digraph, so that every process variable or parameter that represents the primary effect of a fault is included in the set of quantitative design equations.

2. Greater knowledge about the system improves fault resolution, decreases the number of incorrect fault candidates identified, and retains the fundamental causal relationships. Therefore,

   - ¶ Set up quantitative equations for each physical mechanism, unit, or piece of equipment, rather than for larger sections of the process.

   - ¶ Do not eliminate process variables through the substitution of equations.

3. Modify the causal digraph to handle structural faults. The digraph must contain the causal arcs necessary to model the behavior of the system when the fault is present.

3.5 Design of Diagnostic Systems

In this section, the focus shifts to the design of diagnostic systems. For any diagnostic framework to be viable for practical use, the issues of portability and the ease of installation and modification need to be addressed. Therefore, I investigate the following design objectives:

- The diagnostic system should be flexible and easily portable to a variety of process environments to reduce the costs of development and installation.

- The process representation should be easily modified and updated. Changes in piping and equipment should not require extensive reprogramming.

- The process representation should easily characterize the plant under changing operating conditions.

Specifically, the following issues are addressed: (1) The equations used for deriving the causal digraph in Section 3.2 could be written for a single process unit or over several units. I propose causal models at the process equipment level. (2) The digraph for a process unit depends on the context in which the unit functions. I introduce conditionals into the qualitative models to separate the underlying physical principles from the supporting context. General context-free component models permit the construction of the causal digraph for a unit in any context. In addition, the conditionals permit real-time changes in the directed arcs and arc attributes due to changes in operating conditions and process equipment. My objective is a set of component-based, context-independent causal models for the purpose of fault diagnosis.

3.5.1 Generic Causal Models

We have chosen to model the behavior of the overall system by modeling the behavior of the individual components that make up the system. General

models for system components are advantageous because the qualitative behavior of an individual process unit is identical across plant sites. Differences in behavior emerge at greater levels of aggregation. For example, the operation of a centrifugal pump is similar in different processing environments, whereas the behavior of a fractionation train, of which the pump is a part, differs between plants (because of different column configurations, different control strategies, etc.). Standard or generic causal models of process equipment add modularity and increase portability of a diagnostic system by reducing the development and installation costs.

Modeling system behavior as the aggregate behavior of its components requires two distinct descriptions. A structural description describes how the process units are physically connected to one another. The behavioral description describes the behaviors of the individual components. Similar divisions of knowledge have been proposed in describing circuits (Davis et al. [1982], Genesereth [1984]) and in medicine (Patil et al. [1981], Kuipers and Kassirer [1984]). Andow and Lees [1975] [1978] present general qualitative models for individual process equipment.

3.5.1.1 Structural Description

The structural description specifies the topology of the process units. Structural knowledge is required (1) to specify the physical interconnections between the individual process units, and (2) to describe the physical orientation of the components to identify spatial adjacency. Process piping and instrumentation diagrams show equipment connections, and the plot plan or diagrams of the plant layout show physical adjacency.

The primitive elements of the structural description are the component models, which represent the physical components that make up the system: tanks, pipes, valves, etc. The internal structure of each process unit (trays, impeller, etc.) is not modeled and the component model is considered a black box at this level of detail. Each physical component has one or more ports through which mass, energy, and information flow.

Process units interact with one another if they are in some sense adjacent. Normal (design) interaction occurs between two pieces of process

equipment if their ports are connected. Components that are not adjacent do not directly interact, although they may interact indirectly through a series of adjacent units. Causal interaction may also occur if the units are not physically connected, but spatially adjacent. For example, a fire or explosion in one unit can affect other units that are not directly connected.

The fluid in the system is not treated as an explicit object. Rather, the properties of the fluid (temperature, pressure, species concentration, etc.) are considered attributes of the piece of equipment in which the fluid exists. The properties of a fluid passing through a port are thought of as attributes of the port.

The vector of attributes that characterizes the fluid at a port represents a set of digraph nodes. Connecting two components by a common port equates the values of the attribute vectors. The individual qualitative values of process variables across the port are identical.

3.5.1.2 Behavioral Description

Causal pathways are the primitive elements of the behavioral description. Each piece of process equipment is represented by a set of causal arcs between its process variables and parameters. The equations that characterize the behavior of the component are used to derive the causal relationships as described in Section 3.2.

Note that causal interactions exist only within a component. Causal paths do not exist across ports.

3.5.1.3 Deriving System Behavior

The causal digraph for the overall system is constructed from the structural description and the general behavioral descriptions of the individual components. The important process components are selected from the process schematic and a block diagram showing the component topology is constructed. For each block, an instance of the component digraph is created. The digraph nodes associated with a port connect the causal arcs between adjacent units. The overall behavior of the process is represented

by the resulting system digraph. As noted in Section 3.3.3, one of the limitations of deriving global behavior from the local interactions of system components is that ambiguities may arise and result in spurious digraph interpretations. In the system digraph, multiple paths of opposite sign are common. If global information is known, it should be included to reduce the number of incorrect interpretations.

The selection of components to include in the structural description is dependent on the purpose of the causal model. For example, in fault diagnosis, conduits (pipes for fluid transport, wires for electrical transport, etc.) should be represented as specific process units if faults associated with the conduit are important in the given context. If conduit faults are insignificant or can be neglected, then the component model for the conduit may be omitted. The process units at the conduit's ports are then considered adjacent.

3.5.2 Context-Independent Causal Models

In the previous sections, causal models have been presented as a set of causal pathways. Actually, the causal digraph is the output from a causal model for a given input set of context-specific parameters.

The causal model is a set of rules that specify the appropriate causal paths and several of the attributes of the paths for a component in a given context. As discussed in Section 3.1.1.1, the assumptions and conditions necessary for the existence of an arc are associated with the arc. For the causal arc to exist in the digraph, these conditions must be satisfied.

In a component causal model, the rules for specifying all the causal pathways in the component are grouped into the component rule base. Rule antecedents contain the conditions necessary for the existence of the directed arc. The antecedents reference design values and process measurements related to the specific unit. Rule consequents are the individual causal paths and values of the attributes that are valid for the given context.

Associated with each component rule base is a general component database that stores the design specifications and relevant process measurements that are required by the rule antecedents. For the digraph to

accurately characterize its physical system, the assumptions used to develop the digraph must match the actual physical context. By specifying a set of context-specific numerical and discrete parameters, the rules generate a specific causal digraph for the particular process unit.

Two examples are presented that show how digraph arcs and arc attributes are dependent on context-specific information. In Example 2, arcs are grouped according to the different operating modes of the unit.

Example 1: Numerical Parameters

This example illustrates how numerical parameters may be necessary for specifying causal arcs and their attributes. Consider a well-mixed tank with a single inlet (subscript 1) and single outlet (subscript 2). Heat can be added or removed from the tank to maintain the relative temperatures $T > T_1$ or $T < T_1$, respectively. The causal arc from space time to bulk temperature is investigated.

From an enthalpy balance around the tank,

$$\frac{dH}{dt} = \rho C_p (F_1 T_1 - F_2 T) + Q$$

$$\frac{dT}{dt} = -\frac{1}{\theta}(T - T_1) + \frac{Q}{\rho C_p} \tag{4}$$

From Eq. 4, three different cases can be identified for the existence of an arc from θ to T and the value of its sign attribute

Case 1: If $T - T_1 > 0$, then $\frac{\partial}{\partial \theta}\left(\frac{dT}{dt}\right) = \frac{1}{\theta^2}(T - T_1) > 0$

An arc from θ to T exists and its sign is +.

Case 2: If $T - T_1 = 0$, then $\frac{\partial}{\partial\theta}\left(\frac{dT}{dt}\right) = 0$

An arc from θ to T does not exist.

Case 3: If $T - T_1 < 0$, then $\frac{\partial}{\partial\theta}\left(\frac{dT}{dt}\right) = \frac{1}{\theta^2}(T - T_1) < 0$

An arc from θ to T exists and its sign is -.

These cases generate two rules that are included in the component rule base.

Rule: If 1. $F_1 > 0$
2. $T_1 > T$

Then $\theta \xrightarrow{+} T$.

Rule: If 1. $F_1 > 0$
2. $T_1 < T$

Then $\theta \xrightarrow{-} T$.

Example 2: Numerical and Discrete Parameters

The causal model for a two-port valve is developed. The subscript 1 denotes the valve's inlet; subscript 2 denotes the valve's outlet. Both numerical, continuous parameters (port pressures) and discrete parameters (valve open or closed) are necessary to specify the digraph. Four cases exist:

Case 1. If the valve is open and $P_1 > P_2$, then bulk fluid flow exists from P_1 to P_2. Valve position changes the flow resistance.

Causal Digraph Arcs: $F = f(P_1, -P_2, -R)$

$P_1 = f(-F)$

$P_2 = f(F)$

$R = f(-v)$

$T_2 = f(T_1)$

$C_{i_2} = f(C_{i_1})$

Case 2: If the valve is open and $P_1 = P_2$, then bulk flow does not exist but Pascal's law is valid.

Causal Digraph Arcs: $P_1 = f(P_2)$

$P_2 = f(P_1)$

Case 3: If the valve is closed and $P_1 > P_2$, then there are no causal interactions between the variables at the valve's inlet and outlet ports.

Case 4: If the valve is closed and $P_1 = P_2$, then there are no causal interactions between the variables at the valve's inlet and outlet ports.

The process for generating a particular causal digraph from a general component model is summarized in Figure 3-16. Given an engineered system that can be modeled by the interactions of its components, the design equations and a knowledge of underlying physics for each component allow the creation of general, context-free component models. These models are a set of rules that explicitly state the assumptions and conditions necessary for each of the causal paths to be valid and specify the values of several of the path's attributes. Associated with each causal model is a database

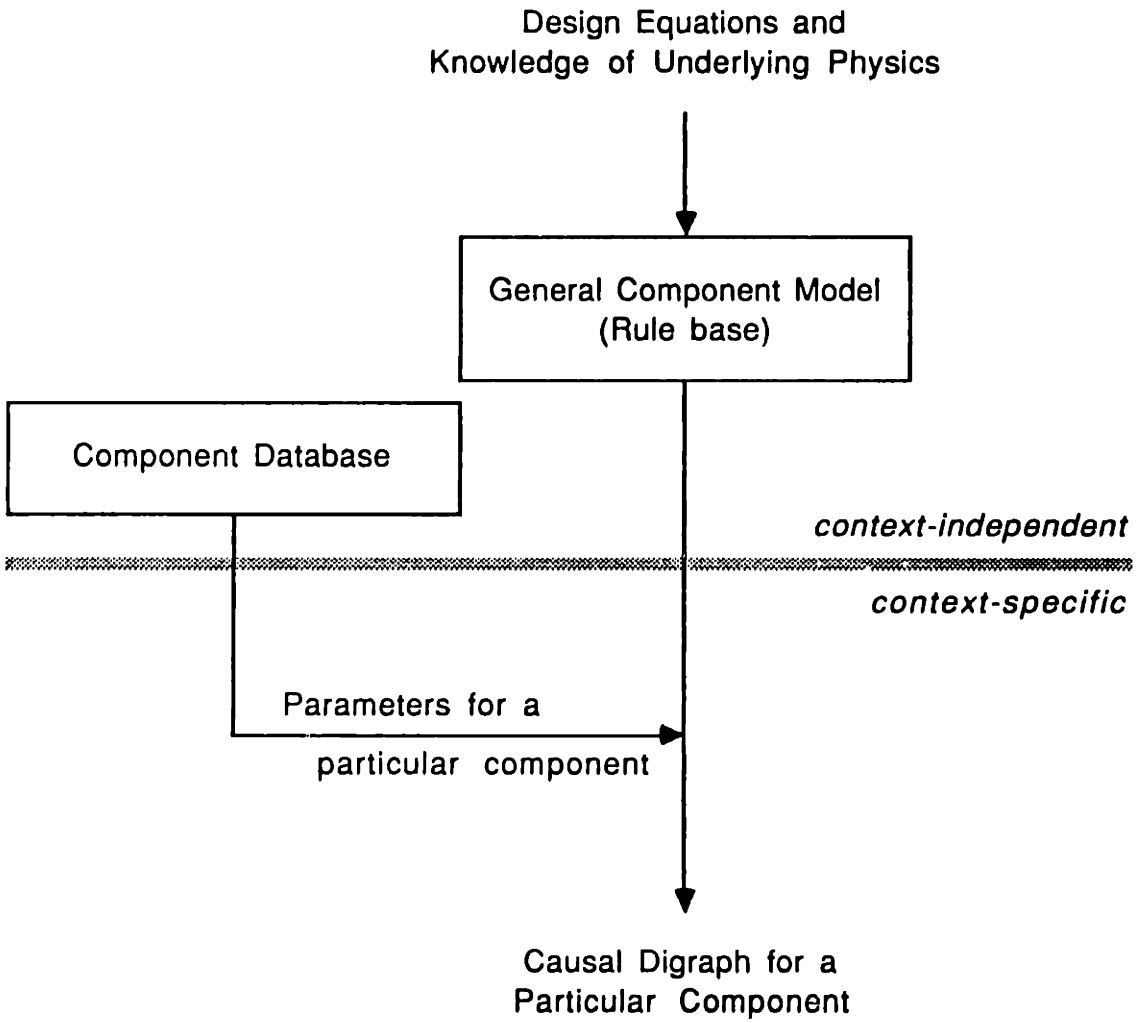


Figure 3-16
Causal Digraph Generation from General Component Models

that holds or references the values needed by the rule antecedents. Both the component model and the component database are context-independent.

When the causal digraph for a particular process unit is desired, an instance of the general component database is created. The database holds design values of parameters and process variables for the particular unit, or for real-time processing, it may reference measurements or equations for calculating the values required by the rules. The rule base is run on these data and a causal digraph is generated for the specific piece of process equipment. The digraph represents the specific behavior of the unit for the given set of conditions. Information from the structural description is used to assign the connections between the ports of different components.

If the system changes (e.g., a flow is rerouted, set points are changed, a tank is emptied, etc.), the values in one or more databases may change. By reevaluating the rule bases associated with each of the units affected by the change, updated component digraphs are produced that accurately model the changed system.

Causal models allow the qualitative behavior of a component to be specified independently from the particular context in which the component will function. A library of general models aids diagnostic system development and installation.

Chapter 4

FAULT DIAGNOSIS BASED ON CAUSAL MODELS

A candidate generation and test strategy for diagnosing faults in process plants is presented. During candidate generation, qualitative values of unmeasured nodes causally upstream from abnormal measurements are assigned so that fault propagation from the upstream nodes would cause the observed deviation. When more than one measurement is abnormal and a single fault is assumed, the intersection of candidate sets generated for each abnormal measurement yields those nodes with consistent causal paths to every deviated measurement. Candidate testing reduces the set of possible fault origins by applying global constraints, knowledge of process dynamics, and heuristic rules. A list of faults is generated from a table that relates the digraph nodes in the reduced candidate set to specific faults.

4.1 Terminology

A fault is any event that causes one or more process variables or parameters to deviate outside the range that represents their normal operation. Therefore, a fault causes the qualitative values of those variables to change from normal '0' to either high '+' or low '-'.

A valid node is a node in the causal digraph that has a nonzero qualitative value. It represents a process variable or parameter that has deviated outside of its range of normal operation. A valid node is a set of two terms: the deviated process variable or parameter and its nonzero qualitative value, e.g., (L, +).

A primary deviation is the deviated process variable or parameter that is the direct result of a fault. Secondary deviations are all other process variable deviations that arise from fault propagation. Both primary and secondary deviations are valid nodes.

A consistent branch is a directed arc between two valid nodes that is a member of the set in Figure 4-1. The arc's sign attribute is listed above the arc, and the qualitative values of the initial and terminal nodes, enclosed in circles, are listed below the nodes. A consistent branch can also be defined as a branch where the product of the signs of its initial and terminal nodes equals the sign of the branch. A consistent branch represents a path that may have been involved in the propagation of a failure. A consistent path is a directed path of consistent branches.

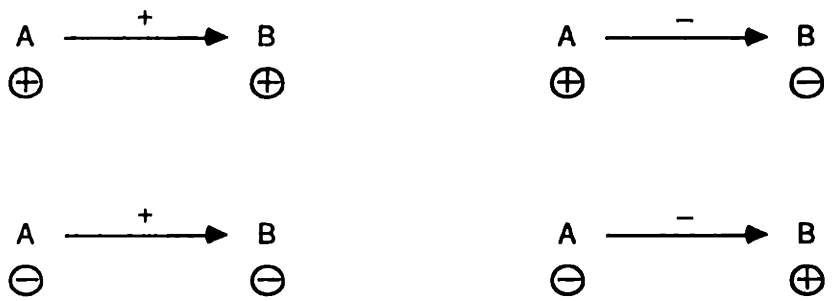


Figure 4-1
Consistent Branches

A valid tree is a subgraph of the causal digraph that consists of a valid measurement and all the valid nodes causally upstream from the measurement. All the branches in the valid tree are consistent. The valid tree describes the path of fault propagation from any causally upstream, valid node to the particular abnormal measurement.

Under the assumption of a single fault, a primary deviation is a root node because a consistent path exists from the root to every valid measurement.

4.2 Overview of Diagnosis Methodology

The strategy for diagnosis, based on the use of causal models, is illustrated in Figure 4-2. The major steps are candidate generation,

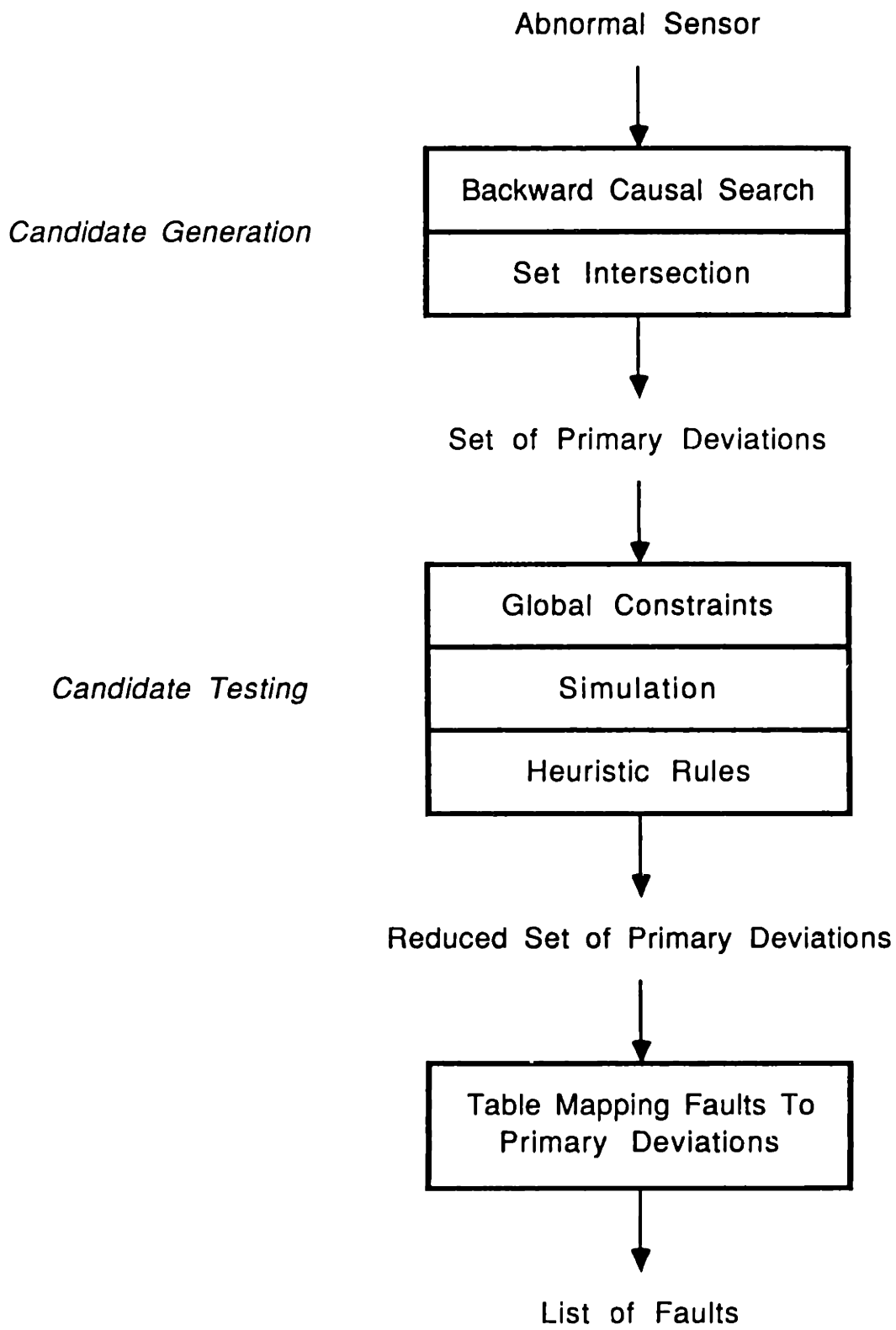


Figure 4-2

Fault Diagnosis Strategy Based on Causal Models

candidate testing, and identifying specific faults through a table mapping faults to primary deviations. When a measurement is deemed abnormal (becomes valid), a failure is assumed to have occurred in the process. The objective of the candidate generation step is to rapidly identify a set of possible primary deviations. A backward search through the causal digraph from the abnormal sensor identifies primary deviations causally upstream from the sensor. The primary deviations are possible fault origins because a consistent path exists from each primary deviation to the deviated measurement. When more than one measurement is valid and a single fault is assumed, set intersection identifies those primary deviations with consistent paths to all abnormal measurements.

In candidate generation, a process variable is a plausible candidate if a path of local interactions exists from the variable to the deviated sensor. Candidate generation is rapid because primary deviations are identified solely on the basis of node adjacency in the digraph and knowledge about control systems. During the candidate testing step, other information about the relationships between digraph nodes is used to reduce the set of candidates. Nodes that are locally plausible are eliminated if they are not consistent with all other known information. Global constraints, simulation using the time delay attribute of the causal arcs, and heuristic rules are used to filter the set of primary deviations.

The list of faults is generated from the reduced set of primary deviations through the use of a table mapping faults to primary deviations. The table is created from an expert system using knowledge about the process equipment specifications and the values of design and operating variables.

Each of these major sections is described in detail.

4.3 Candidate Generation

In this section, the candidate generation procedure is presented. First, several search strategies are described for identifying failure origins in a simple network. Some of the strategies cannot be used for locating faults in the causal digraph because important differences exist between the simple network and the digraph. The search strategy chosen identifies failure origins by tracing backward through the digraph along

the causal arcs from abnormal measurements. When a single fault is assumed, set intersection of the primary deviations generated from the causal search from each deviated measurement yields those primary deviations with consistent paths to all abnormal measurements. The candidate generation procedure is illustrated with an example.

4.3.1 Rouse Network

Rouse [1978] [1981] used a network to investigate human problem solving performance in fault diagnosis tasks. The focus of his research

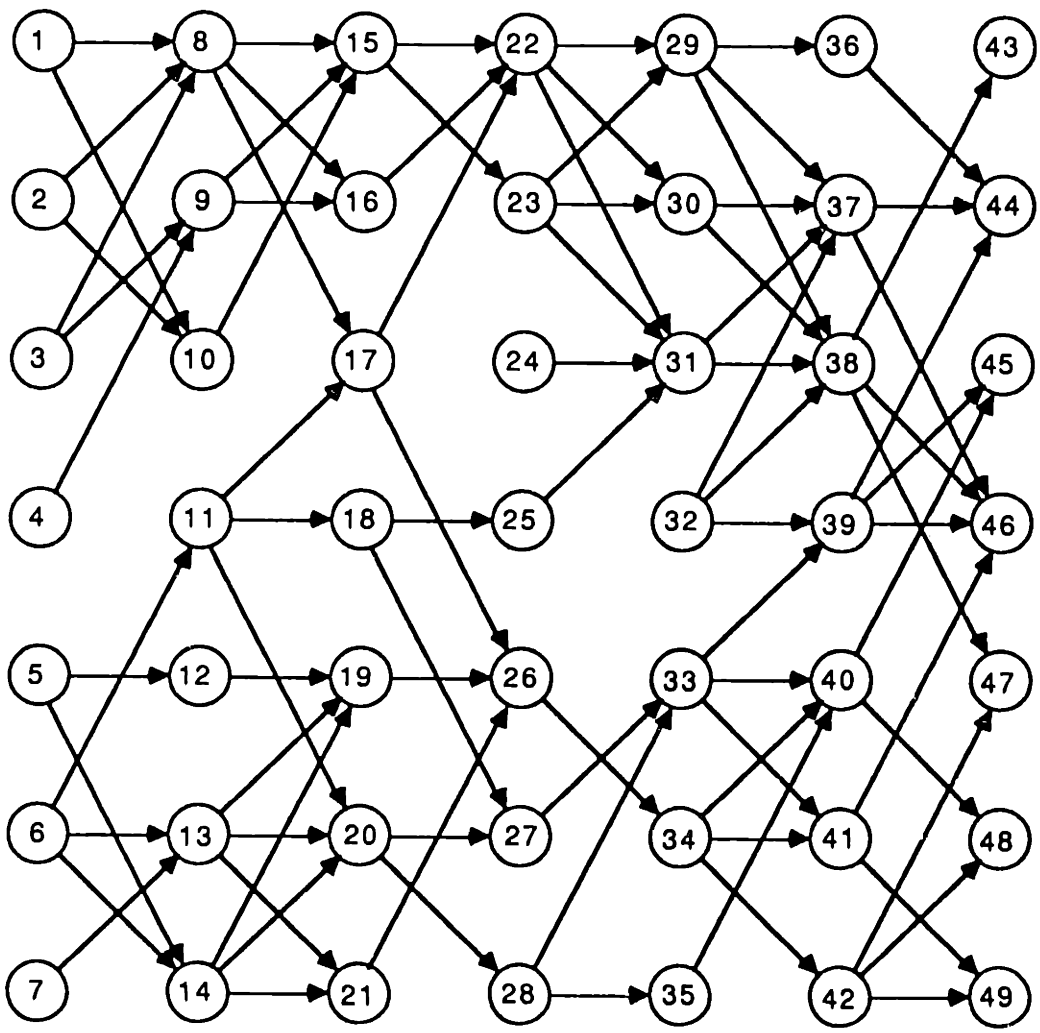


Figure 4-3
Rouse Network

was to use the network to study human performance as a basis for developing training methods. One of the networks he used is presented in Figure 4-3. The problem presented to the test subjects was to locate the fault in the network. A fault is defined as a node whose inputs are all normal (have the value '1') and whose outputs are all abnormal (have the value '0'). Each node behaves like an AND gate, and thus, can produce the outputs zero and one. A node will have the value '1' if (1) all inputs to the node are '1', and (2) the fault is not located at the node; otherwise the node will produce a '0'. All outputs emanating from a node carry the value of the node. Because a node with a zero input transmits a zero to all its outputs, the effect of the fault is propagated throughout the network. The human test subjects were told that a single fault had been introduced into the network and were given the values of the nodes in the far right-hand-side column. The subject's task was to identify the location of the fault by obtaining from the proctor the values of any desired connections between the nodes. A smaller number of queries for information implied that the test subject used a better reasoning strategy and/or more fully utilized the information in the network.

This network problem is examined here to investigate a variety of search strategies. From these strategies and the differences between the simple network and the causal digraph, a procedure for fault diagnosis using the digraph is developed. My objective differs from Rouse because in his study, additional information is obtained from the proctor to locate the single faulty node. Here, I narrow the search space from the given information only. In a control room setting, obtaining additional information from the plant, beyond that gathered through an automatic data acquisition system, is both time-intensive and requires the attention of several plant personnel. As a first step in diagnosis, the objective is to reduce the set of fault candidates as much as possible without operator intervention or additional process measurements.

One possible solution strategy for the Rouse network is presented below. Given the right-hand-side column of outputs,

Step 1: **Eliminate infeasible candidates causally upstream from the known, normal nodes.** For each node with the value '1' in the

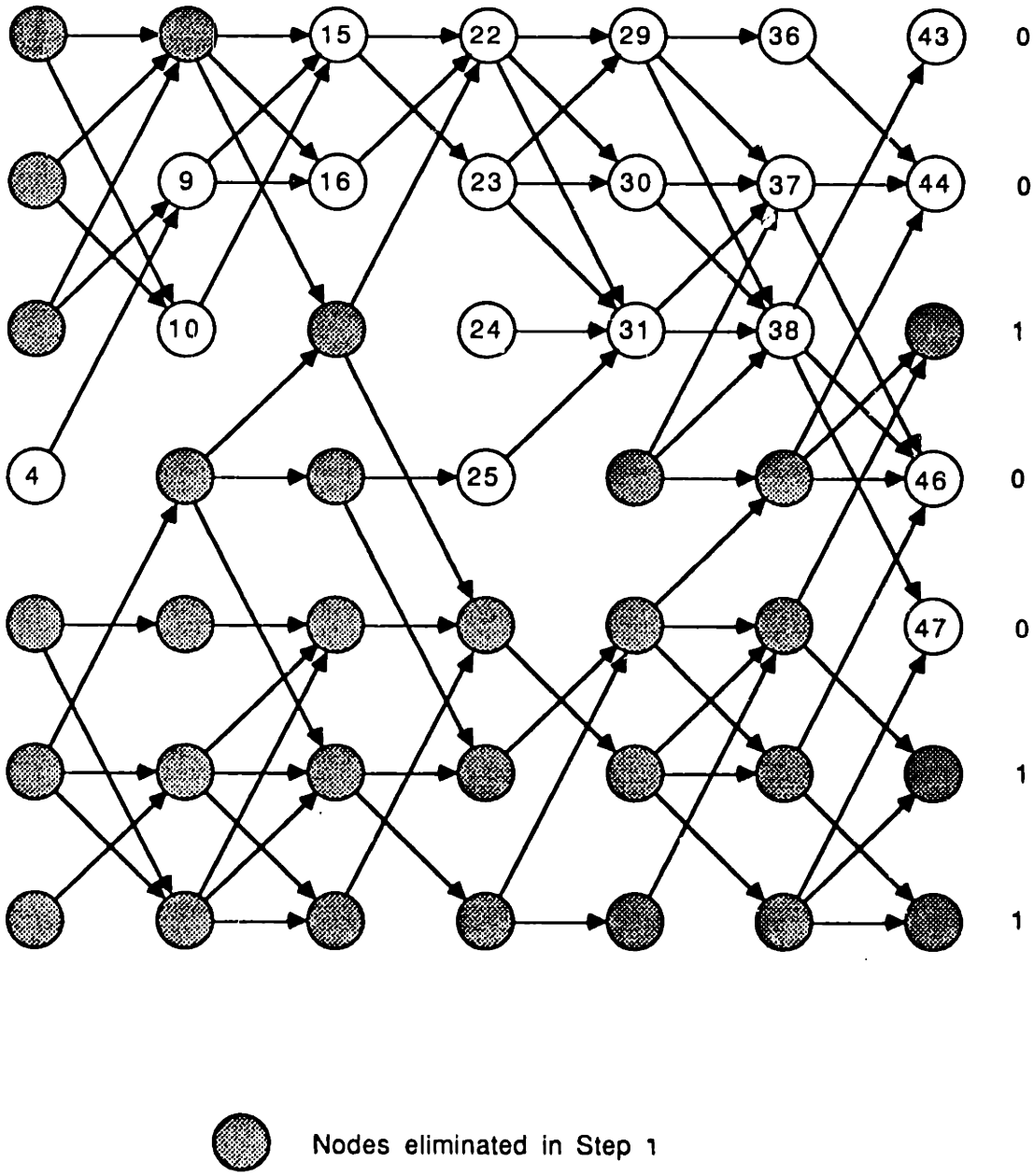


Figure 4-4

Set of Fault Candidates After Step 1

right-hand-side column, work causally upstream (from right to left) along the arcs to eliminate nodes. These nodes can be discarded because if any of these nodes were the location of the fault, the propagation of the fault would cause the observed node to have the value '0'. Since the known node has the value '1', those nodes causally upstream cannot be faulty.

Step 2: **Eliminate infeasible candidates through disturbance simulation.** For each remaining node, assume that the fault has occurred at the node and propagate the effect of the fault in the direction of the arcs. If fault propagation from the node cannot explain all the observed deviations, it cannot be a fault candidate because for a single fault, directed paths must connect the fault to all observed deviations.

If we start at the right-hand side by performing the simulation at the known observations, we can minimize the number of simulation tests by realizing that once a node is found such that the propagation of zeros from the node can explain all the known deviations, then all nodes causally upstream from that node can also cause all the known deviations because paths from the causally upstream nodes pass through the node.

This strategy is illustrated with an example. Given the set of values for the nodes in the right-hand-side column in Figure 4-4, Step 1 would eliminate the nodes shaded in grey. In Step 2, disturbance simulation begins with node 43. If node 43 is assumed to be faulty, disturbance propagation from the node cannot explain the deviations observed at nodes 44, 46, and 47; thus, it is eliminated. Once nodes 29, 30, and 31 are tested, no further simulation is necessary because (1) deviations at each of these nodes can explain all the observed deviations, and (2) paths from all other nodes to the observed deviations pass through these nodes. The final set of fault candidates is presented in Figure 4-5.

Note that disturbance simulation involves more effort to eliminate a node from the set of possible candidates than does the causal upstream

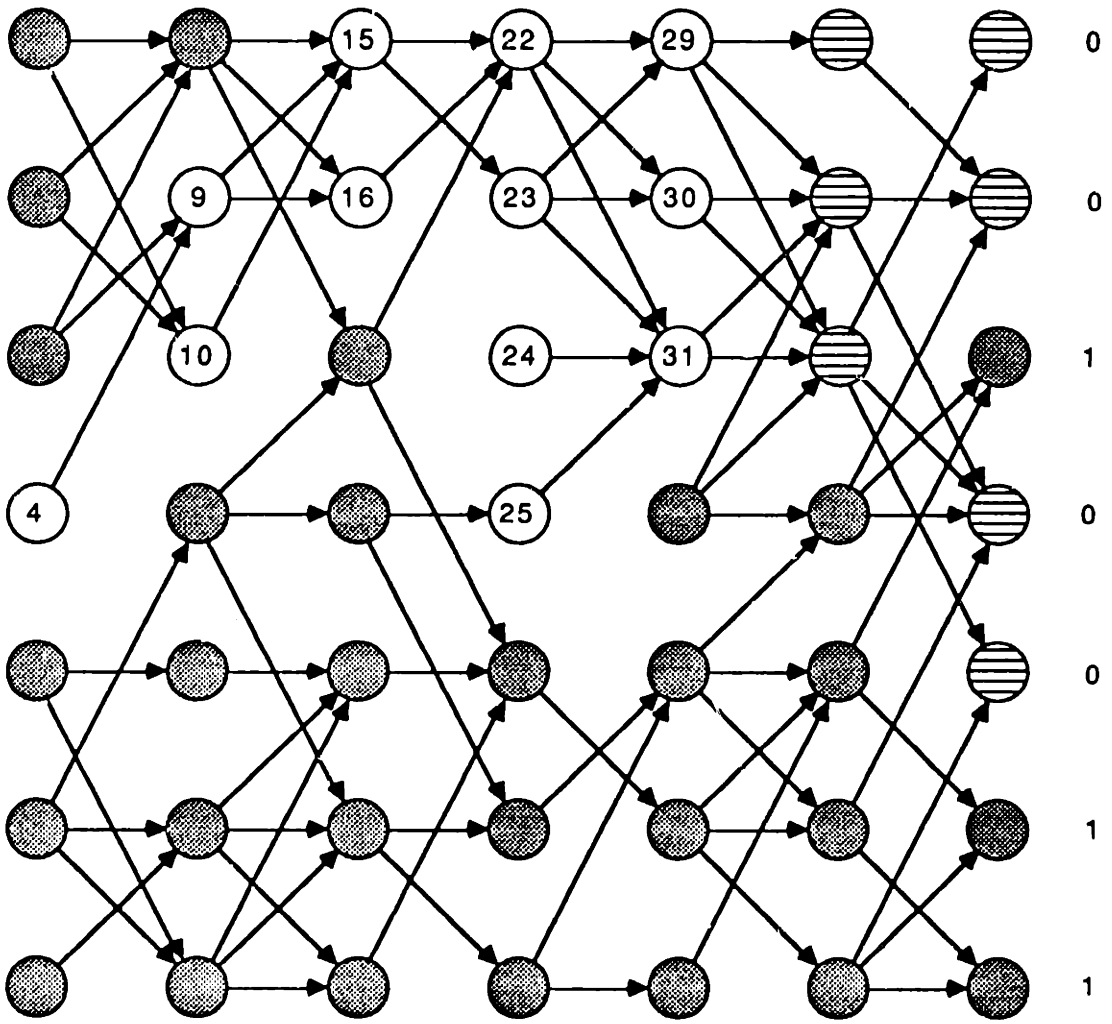


Figure 4-5
Set of Fault Candidates After Step 2

search from the normal nodes. Only the knowledge of adjacency is required in Step 1, whereas both adjacency and the simulation of the disturbance through the network is required in Step 2.

A less computationally-intensive procedure for disturbance simulation in Step 2 is based on the knowledge that, for a single fault, causal paths must exist from the faulty node to **all** the observed deviations. For each right-hand-side node with the value zero, a causally upstream search identifies all possible nodes whose deviation could cause the deviation at that node. The set of fault candidates that can explain all the observed deviations is the intersection of the sets generated for each observed deviation. For the same example, the set of fault origins generated from a backward causal search from node 44 includes nodes 36 and 44 as possible candidates. The set for node 46 includes nodes 37 and 46. The intersection of these two sets eliminates all four of these nodes because they are not contained in the other set.

The search strategy presented may not have the fewest number of processing operations for a graph with a large number of nodes and a small number of observed deviations. Instead of eliminating candidates causally upstream from normal nodes, a procedure that relies on a causal upstream search from any abnormal node and simulation may be more efficient. For example, consider the network in Figure 4-6 in which only two deviations are observed. Starting with node 45, fault simulations from node 45, and then from causally upstream node 39, eliminate these nodes as possible candidates because the deviated node 48 is not reached. Simulation from node 40 provides a candidate. Simulations from nodes 33 and 34, causally upstream from node 40, terminate on normal nodes. Node 35 becomes a second candidate. Simulation from node 28 fails. Two candidates are obtained. This procedure, based on a combination of causally searching upstream from deviated nodes and simulation, is more efficient when the number of observations is small because it does not search the entire graph to eliminate normal nodes.

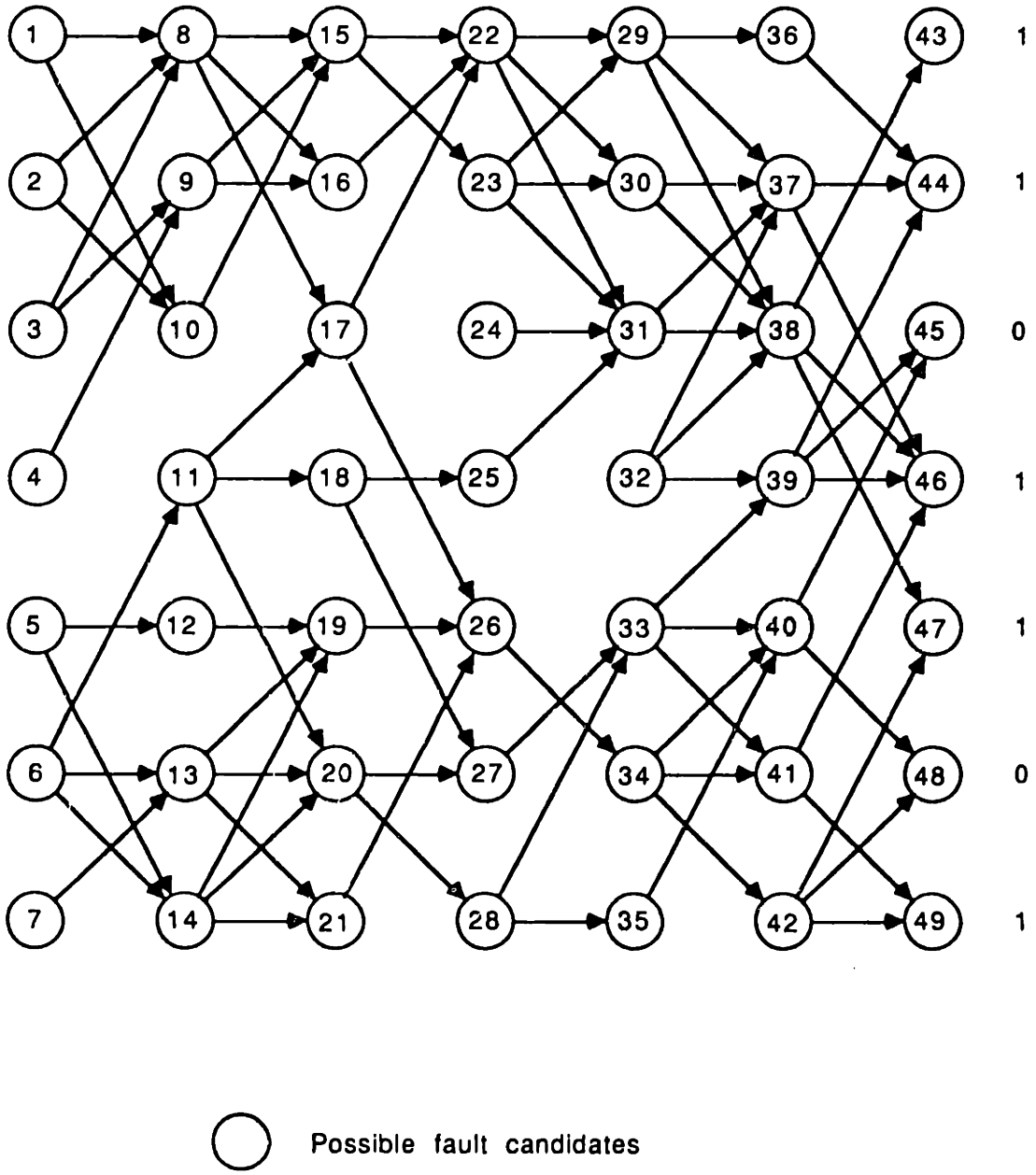


Figure 4-6

Set of Fault Candidates Generated From a Backward Causal Search From Abnormal Nodes and Fault Simulation

4.3.2 Differences Between the Rouse Network and the Causal Digraph

The differences between the Rouse network and the causal digraph are summarized below.

Rouse Network

1. Two discrete values ('0' and '1') representing faulty and normal states, respectively
2. Arcs have no attributes
3. No circuits
4. Fully propagated fault
5. No attenuation of fault disturbance

Causal Digraph

1. Three discrete values ('-', '0', '+') representing negative, zero, and positive deviations, respectively, from a specified reference state
2. Arcs have attributes, including sign, magnitude, and time
3. Circuits (both process circuits and feedback loops)
4. Dynamic fault propagation
5. Possible attenuation and compensation of fault disturbance

The characteristics of the causal digraph eliminate some of the search strategies that were used with the Rouse network. Specifically, because fault diagnosis is usually performed before the disturbance is fully propagated, measurements causally downstream from the failure may be normal. Downstream measurements may also be normal because the magnitude of the disturbance is attenuated or control systems have halted the disturbance propagation. Thus, fault candidates causally upstream from normal measurements cannot be eliminated. Similarly, without knowledge about dynamics and attenuation, qualitative disturbance simulation, accomplished by propagating deviations causally forward through the digraph, cannot be

performed because the disturbance may not have reached the downstream sensors or be too weak to cause measurement deviations.

Therefore, candidate generation must be accomplished by searching causally upstream from abnormal measurements. For a single fault, set intersection can also be used because consistent causal paths must exist from the faulty node to all the observed deviations. The other differences between the simple network and the causal digraph will necessitate modifications to the search strategy (e.g., the use of consistent branches and testing for circuits). These will be examined.

4.3.3 Assumptions for Candidate Generation

The following assumptions are necessary to guarantee that the actual fault origin is included in the set of fault candidates:

1. The digraph represents the actual causal interactions between the process variables.

   If a given pathway is omitted from the causal digraph, the actual fault origin may not be included in the set of primary deviations. If extraneous paths are included, then the causal search generates spurious candidates. The operator may lose confidence in the diagnostic system if it displays many inconsistent fault hypotheses or if logical fault candidates are omitted.

2. The normal operating ranges for every measurement are selected so that if any of the faults desired to be diagnosed occur, one or more measurements will cross their normal threshold and become valid.

   If the range representing normal operation is too narrow, non-failure disturbances may initiate the diagnostic system. If the range is too wide, failures that cause disturbances with small magnitudes may not be detected because the measured process variable deviation may not cross the alarm threshold. The ranges are selected so that if any digraph node is valid, then the system is in failure.

3. If a disturbance is propagating along a causal path and two process variables in the path are measured, the causally upstream measurement will alarm before the downstream measurement.

   This result is used to bound the fault space during candidate generation. When a measured process variable deviates, a causal search upstream from the deviated measurement attempts to identify possible fault candidates. If the value of a measured process variable causally upstream is normal, the fault cannot lie causally above this normal measurement. If the fault was located above this normal measurement and the actual path for fault propagation included this measured variable, then this measurement would have deviated before the downstream measurement. Therefore, the location of the failure lies within the digraph bounded by the normal measurements.

4. Process variables can only deviate in a single direction. Controlled variables can also return to normal.

   Inverse response occurs when a variable's initial deviation is opposite from its long-term deviation. When the direction of deviation of a variable changes, its qualitative value also changes, and the graph of valid nodes and consistent branches can become disconnected. If the graph becomes disconnected, a consistent causal path does not exist from the primary deviation to deviated measurements causally downstream from the process variable exhibiting inverse response. When a process variable lies within a circuit with a net negative sign, the restriction of a single change in sig. can be stated in terms of a negative feedback heuristic: a feedback effect along a causal circuit can never dominate its cause.

   Controlled variables can deviate outside their normal range and return to normal.

5. Each failure is represented by the deviation of a single node in the causal digraph.

   The digraph must be constructed so that every failure causes the deviation of a single digraph node. Since the candidate generation procedure involves a search for primary deviations, i.e., single digraph nodes that can explain all other secondary deviations by fault propagation, if a single origin for the failure is not in the digraph, the fault cannot be identified.

Under these assumptions, the method guarantees that the actual fault origin is included in the candidate set.

4.3.4 Candidate Generation Procedure

The objective of candidate generation is to rapidly partition the total set of fault origins into a feasible set (i.e., those primary deviations that could cause the observed secondary deviations) and an infeasible set. The criterion for including a node-sign pair in the set of possible primary deviations is that a consistent path must exist from the node to all abnormal measurements. The inputs to the procedure are the causal digraph for the process and the sign attribute of every arc, the controlled and manipulated variables and net sign of the functioning control systems, and the qualitative values of every process measurement. Quantitative plant data are compared with the expected, normal references so that every measurement has a '+', '0', or '-' qualitative value.

Faults propagate along causal digraph arcs. Thus, the origin of the failure causing an abnormal process measurement must lie causally upstream from the measurement. Possible primary deviations are identified by searching causally upstream from the abnormal sensor.

The search is accomplished by constructing a valid tree for an individual deviated measurement. The valid tree represents the paths of fault propagation from every possible node in the causal digraph to the abnormal measurement. Beginning with the given valid measurement, qualitative values are assigned to unmeasured, causally upstream nodes to make the

causal arcs consistent. The assignment of qualitative values to adjacent, causally upstream nodes is continued until all possible primary deviations that could cause the observed measurement deviation are identified. Thus, the search is exhaustive.

Given the current node in the tree, adjacent nodes causally upstream from the current node are added to the valid tree if

1. The causally upstream node is not already in the path from the current node to the valid measurement, AND

2. If the process variable represented by the causally upstream node is measured, then the measurement must be valid with the qualitative value necessary to make the branch from the measured node to the current node consistent.

Condition 1 eliminates any circuits in the causal digraph which would give rise to cycling during the backward causal search. A node is added to the valid tree only if it does not already appear in the path to the valid measurement. Note that a node can appear more than once in the valid tree; it is only restricted from appearing more than once along any directed path. Condition 2 is used to bound the fault space. If, during the backward search, a measured node is encountered whose value is normal or opposite of the sign necessary to make the branch consistent, the search is discontinued along this arc.

4.3.4.1 Control Systems

Control systems are different from other process equipment because they are designed to compensate for disturbances: the manipulated variable is adjusted to keep the controlled variable at its desired value. If a disturbance enters a functioning control loop and the magnitude of the disturbance is insufficient to saturate the control system, then the controlled variable remains at its set point and the disturbance causes a change in the manipulated variable. In terms of the digraph, the failure propagates through a normal, measured node, and a consistent path does not

exist between the fault origin and the abnormal measurements causally downstream of the controlled variable. In the search procedure described above, the search space is bounded by normal measurements. Because a fault can lie causally above a normal measurement if the measurement is used in a control loop, the following modifications to the causal search strategy are necessary. (The controlled variable is assumed to be directly measured.)

Manipulated Variables

If the origin of the failure is causally upstream from a control system and the controlled variable was normal, a search from causally below the control system will encounter the normal measurement within the loop and bound the fault space. Thus, the fault origin would not be included in the set of possible primary deviations. Therefore, during the backward causal search, if the manipulated variable is valid (either assumed valid during the search or directly measured) and the controlled variable is normal, then consider the controlled node and nodes causally upstream from the controlled node as possible origins. The value of the manipulated variable and the net sign of the control loop are used to infer the value that the controlled variable would have if no control system were present. The search is continued causally upstream from the controlled variable.

Measurements

A measurement can only be encountered in the backward search if it is within a control system. When a measurement is encountered, the measurement node is added to the valid tree only if the actual measurement is abnormal and of the correct qualitative sign to make the branch between the measurement and the control system error consistent.

Controlled Variables

If a controlled variable is encountered during the backward search and the measurement of the controlled variable is valid, then the controlled variable is added to the valid tree regardless of whether the assumed sign necessary to make the causal arc valid matches the measurement. When the assumed direction of deviation of the controlled variable is not the same as the actual measurement, including the node for the controlled variable

is necessary to identify sensor failures within the control loop. For example, in a negative feedback loop, if the sensor fails high and the control loop is operational, then the actual value of the controlled variable would be low.

4.3.4.2 Eliminating Arcs With Small Magnitudes

When the causal digraph is used for fault diagnosis, the digraph arc is interpreted as a possible path of fault propagation. A consistent path from a primary deviation to an abnormal measurement represents the actual path of fault propagation. Implicit in the causal search is the assumption that the magnitude of the causal interaction along each arc is large enough so that the terminal node becomes valid when the initial node is valid.

Primary deviations may not be capable of causing the observed measurement pattern if the magnitude of one or more arcs along any path to a valid measurement is insufficient to transmit the disturbance. Therefore, information about an arc's magnitude attribute can be used to bound the fault space during candidate generation and reduce the number of possible fault origins. If the magnitude attribute of an arc is small (has a qualitative value of '0'), the arc is not included during the construction of the valid tree.

4.3.5 Intersection of Root Node Sets

The construction of the valid tree identifies the possible primary deviations for each individual abnormal measurement. No assumptions were made during the causal search procedure about the actual number of faults that have occurred in the process.

Because the probability of multiple, simultaneous, independent events is low, the candidate generation procedure first attempts to explain all the abnormal measurements by a single fault. Under the single fault assumption, a digraph node-sign pair is a primary deviation if a consistent path exists from the primary deviation to **every** abnormal sensor. A primary deviation is the root of a directed tree of consistent branches, which spans the set of valid measurements.

The intersection of the sets of primary deviations generated for each valid measurement yields those primary deviations from which consistent paths exist to all the valid measurements. Since consistent paths must exist from the primary deviation to every deviated measurement, the converse, that the actual primary deviation must be identified by the backward causal search from every measurement deviation, must also be true. Therefore, given multiple measurement deviations and the single fault assumption, the intersection of the sets of primary deviations generated for each of the valid measurements will identify those primary deviations with consistent paths to every abnormal sensor.

Under the assumptions presented in Section 4.3.3, the root node set will contain the actual fault origin. In practice, set intersection will usually reduce the number of primary deviations generated for a single valid measurement. If there is no reduction in the size of the root node set, then an existing valid measured variable is a node on the fault propagation path from the root to the new valid measurement.

When new measurements become valid, set intersection should be performed with the sets of primary deviations generated for each abnormal measurement rather than the reduced root node set from candidate testing. The primary deviations generated for each abnormal measurement are used because possible fault origins that are eliminated during candidate testing may later need to be considered. For example, the validity of the rule antecedents in the heuristic rules, evaluated during candidate testing, may change when new measurements become valid. If the rules eliminate primary deviations and this reduced set is used during set intersection when a new sensor becomes valid, the intersection may not contain all the possible origins.

If multiple faults have occurred, set intersection, in most cases, will produce the empty set. A combinatorial intersection procedure is then necessary to explain the observed measurement pattern with the fewest number of faults. For example, consider the three sets of primary deviations A, B, and C, produced from the causal search for three valid measurements. If the intersection of A and C yields common origins, while the intersections of sets A and B and sets B and C are null, then the measurement pattern can be explained by two independent faults. If multiple

faults have occurred and the faults lie along a consistent path, then the faults can be explained by a single primary deviation.

4.3.6 Example

The candidate generation procedure is illustrated using the causal digraph for a tank with a level control system shown in Fig. 3-2. Given the valid measurement (F_{sensor}, +), the valid tree in Figure 4-7 is constructed from the arcs in the causal digraph. The signs of unmeasured nodes causally upstream from the flow sensor are assigned so that consistent branches exist from the nodes to the valid measurement. The dotted arrow represents the causal path through the normal controlled variable L to its manipulated variable v_1. The valid tree ends at nodes R_2, R_{23}, R_{45}, and L_{sp} because there are no causal arcs that terminate at these nodes, at nodes F_1 and P_5 because they are at the boundary of the process, at P_b because L, its causally upstream node, is measured and normal, and at P_3 in the right-half branch because F_{34}, its causally upstream node, is already in the path from (P_3, +) to (F_{sensor}, +).

Notice that several process variables appear in the tree more than once, with both the same and opposite signs, but no node is found more than once along any path to the abnormal measurement. Nodes with opposite signs are found in the valid tree when the information contained in the causal digraph and the qualitative values of the measurements is not adequate to constrain the node to a single value. For example, consider the paths from from (F_2, +) and (F_2, -) to (F_{sensor}, +). The paths are interpreted as follows:

Path from (F_2, +) to (F_{sensor}, +)

Increasing the flow rate from the tank outlet increases the flow rates F_{23}, F_{34}, and F_{45} in PIPE-A, control valve CV-1, and PIPE-B, respectively, causing the flow measurement F_{sensor} to increase.

Path from (F_2, -) to (F_{sensor}, +)

Decreasing the flow rate from the tank outlet increases the fluid volume and level in the tank. The level control system opens the

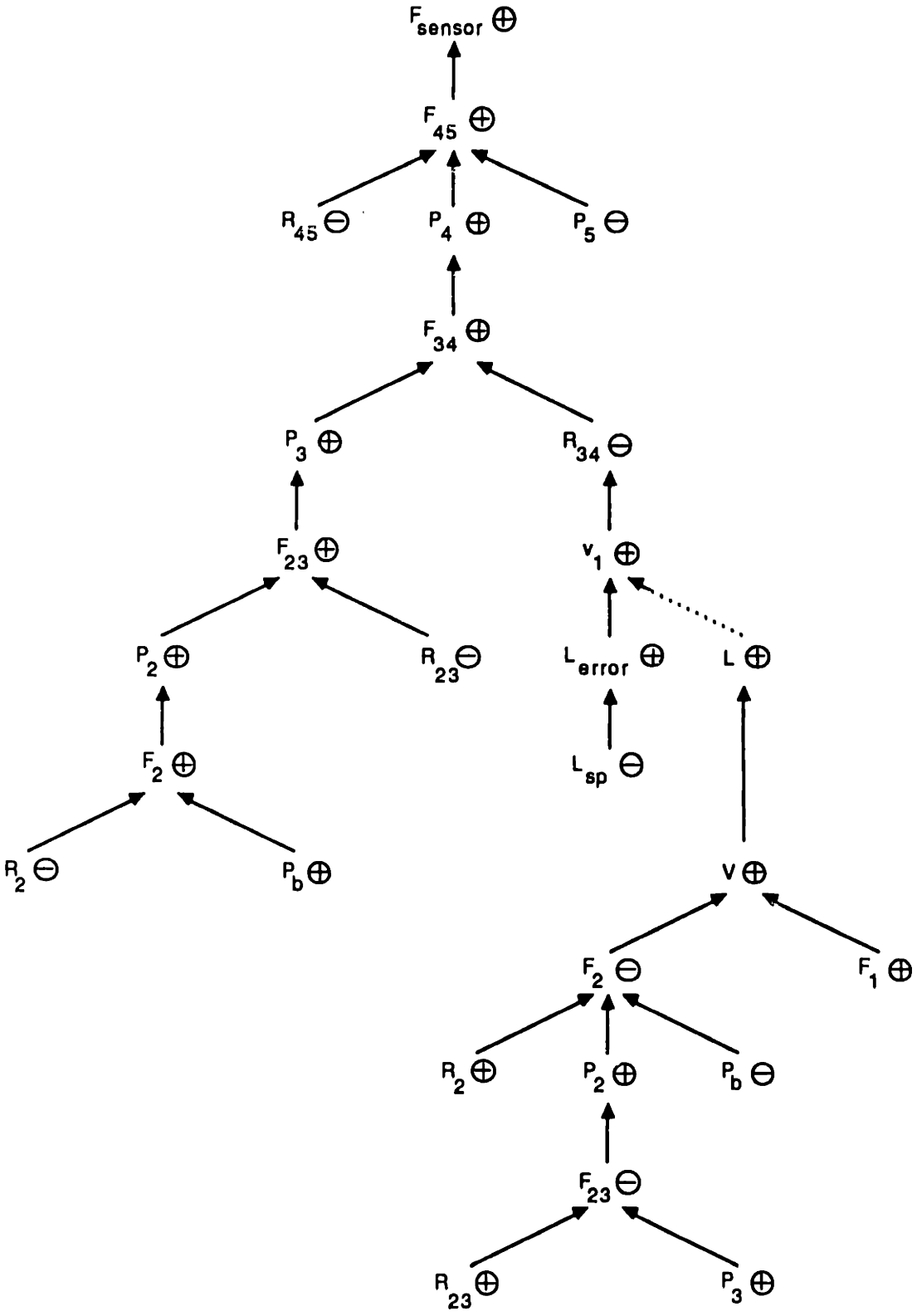


Figure 4-7

Valid Tree for (F_{sensor}, +)

control valve CV-1, increasing the flow rates F_{34} and F_{45}, causing the flow measurement F_{sensor} to increase.

Although the second argument is not globally consistent because the conservation of mass between the tank outlet and PIPE-B is not satisfied, both arguments are consistent with local, causal interactions. In Section 4.4, candidate testing will be used to eliminate primary deviations that are inconsistent with global knowledge.

Twenty five primary deviations are identified in the valid tree. A table relating faults to primary deviations, which will be discussed in Section 4.5, is used to generate the list of faults. Nine faults, presented in Table 4-1, are identified.

Table 4-1

List of Primary Deviations and Possible Faults for F-SENSOR High after Candidate Generation

Measurements:
F-SENSOR high

List of Primary Deviations: ((L-SP -) (L-ERROR +) (F23 -) (R23 +) (PB -) (R2 +) (F1 +) (F2 -) (V +) (L +) (V1 +) (PB +) (R2 -) (F2 +) (P2 +) (R23 -) (F23 +) (P3 +) (R34 -) (F34 +) (P4 +) (P5 -) (R45 -) (F45 +) (F-SENSOR +))

Possible Faults:
1> The set point of LEVEL_CONTROL_SYSTEM set low.
2> Control system LEVEL_CONTROL_SYSTEM failed high.
3> Blockage in pipe PIPE-A.
4> Outlet blockage in tank TANK-1.
5> Control valve CV-1 failed open.
6> Leak in pipe PIPE-B.
7> Sensor F-SENSOR failed high.
8> High flow rate F1 entering tank TANK-1.
9> Low pressure downstream of pipe PIPE-B.

When a second measurement becomes abnormal, a valid tree is constructed for the new measurement. The valid trees for (L_{sensor}, +) and (L_{sensor}, -) are shown in Figures 4-8 and 4-9, respectively. The search

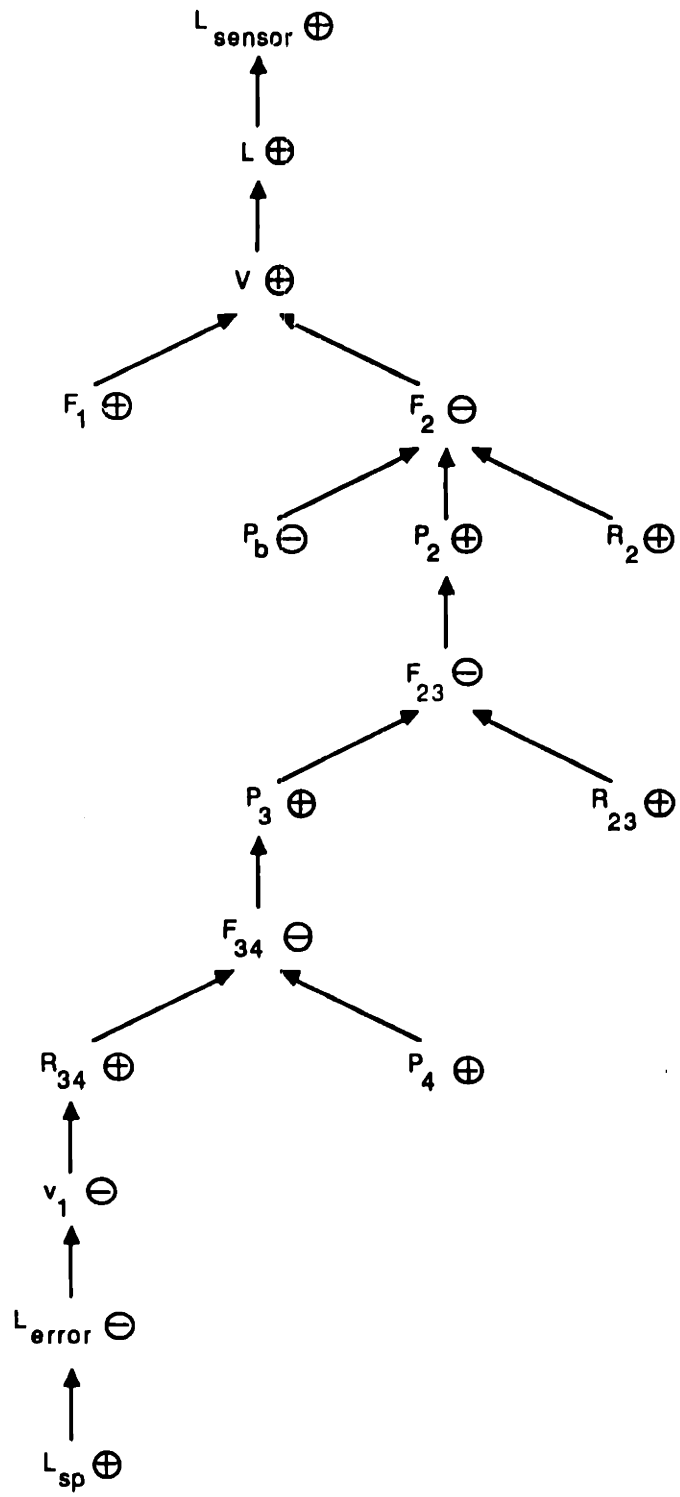


Figure 4-8

Valid Tree for (L_{sensor}, +)

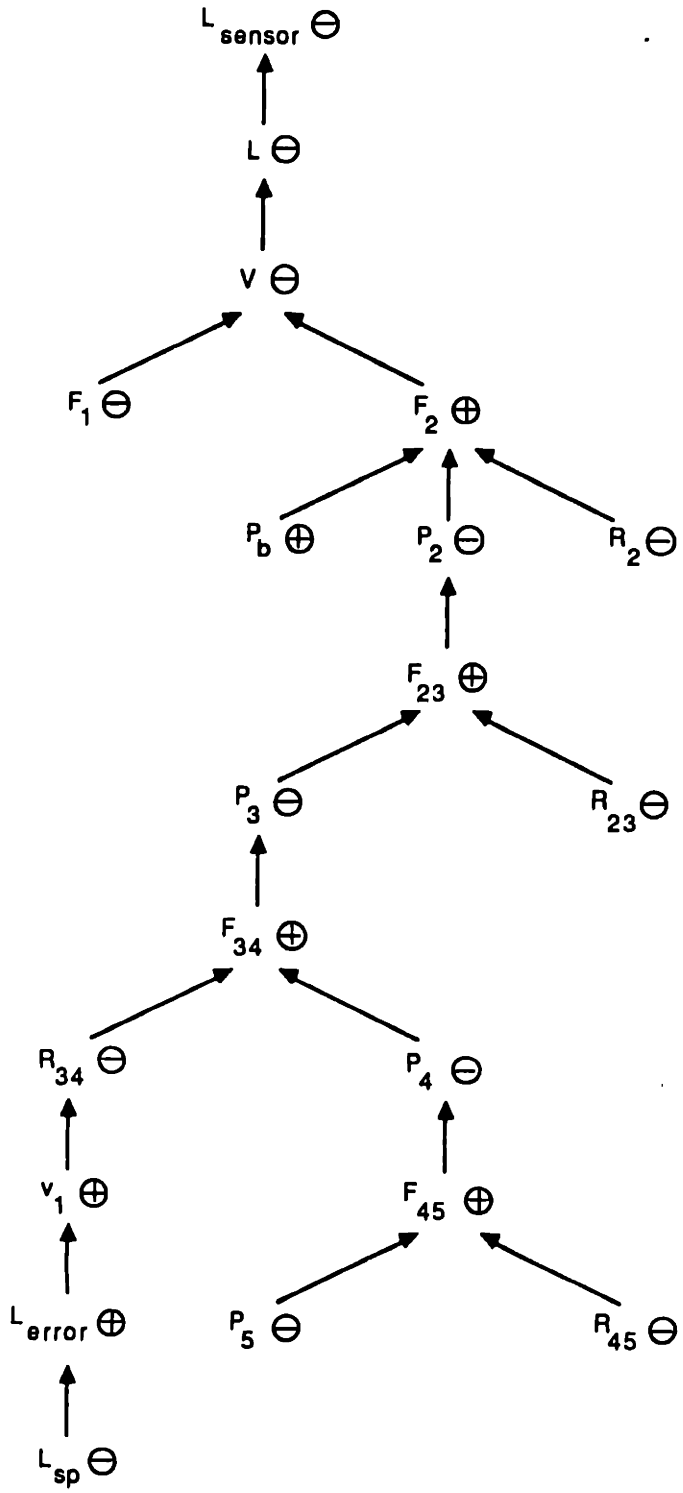


Figure 4-9

Valid Tree for (L_{sensor}, -)

ends at P_4 in Fig. 4-8 because F_{45}, causally upstream from P_4, requires the assignment of the qualitative value '-' to make the branch consistent. The negative value is inconsistent with the actual positive deviation at F_{sensor}.

If a single failure is assumed, the intersection of the sets of primary deviations identified for each measurement yields those primary deviations from which consistent paths exist to all the valid measurements. The intersection of the set of primary deviations identified for the abnormal measurement (F_{sensor}, +) with the sets for (L_{sensor}, +) and (L_{sensor}, -) are shown in Tables 4-2 and 4-3, respectively. The number of primary deviations and the number of associated faults are reduced as the pattern of symptoms develops.

Table 4-2

List of Primary Deviations and Possible Faults for
F-SENSOR High and L-SENSOR High after Candidate Generation

Measurements:
  F-SENSOR high
  L-SENSOR high

List of primary deviations: ((F23 -) (R23 +) (PB -) (R2 +) (F1 +) (F2 -)
 (V +) (L +) (P2 +) (P3 +) (P4 +))

Possible Faults:
  1> Blockage in pipe PIPE-A.
  2> Outlet blockage in tank TANK-1.
  3> High flow rate F1 entering tank TANK-1.

Table 4-3

List of Primary Deviations and Possible Faults for
F-SENSOR High and L-SENSOR Low after Candidate Generation

Measurements:
F-SENSOR high
L-SENSOR low

List of primary deviations: ((L-SP -) (L-ERROR +) (V1 +) (PB +) (R2 -) (F2 +) (R23 -) (F23 +) (R34 -) (F34 +) (P5 -) (R45 -) (F45 +))

Possible Faults:
1> The set point of LEVEL_CONTROL_SYSTEM set low.
2> Control system LEVEL_CONTROL_SYSTEM failed high.
3> Control valve CV-1 failed open.
4> Leak in pipe PIPE-B.
5> Low pressure downstream of pipe PIPE-B.

4.3.7 Sequence of Alarms

A smaller set of primary deviations is identified when the causal search is performed on each valid measurement in the order in which it occurs, rather than when the fault is more fully developed, because building a valid tree with fewer abnormal measurements tends to reduce the size of the search space. Consider the valid tree constructed for the first abnormal measurement. The tree is bounded causally upstream by normal measurements. If the valid tree is constructed for the same abnormal measurement when several sensors are abnormal, the causal search may encounter valid measurements causally upstream that are consistent with the current search path. The causal search along these paths is not bounded and the construction of the valid tree is continued. The sections of the valid tree above consistent, measured nodes add additional fault origins to the set of primary deviations. The following example illustrates how evaluating the alarms in sequence reduces the number of primary deviations. Consider the causal digraph in Figure 4-10, in which nodes C and F are measured and valid. If the deviation (C, +) is known to have occurred first, candidate generation from node C would identify the four

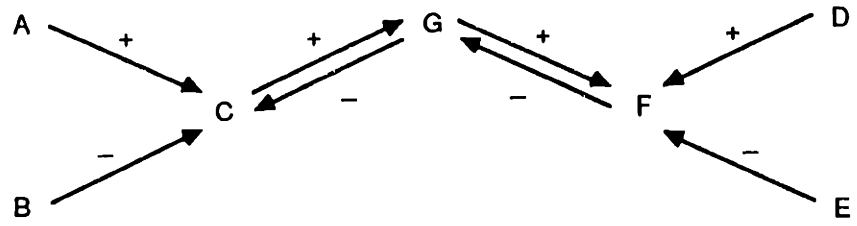
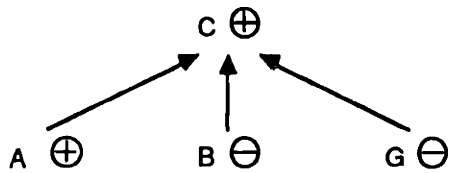
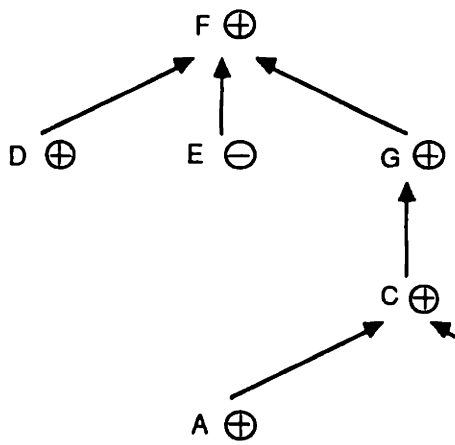


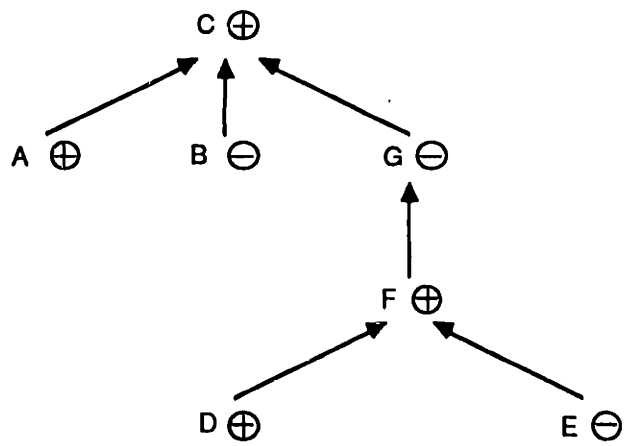
Figure 4-10
Causal Digraph for Illustrating Sequence of Alarms



4-11a



4-11b



4-11c

Figure 4-11
Valid Trees for (C, +) and (F, +)

primary deviations shown in Figure 4-11a, because the valid tree is bounded at (G, -) when node F is normal. When the deviation (F, +) occurs, candidate generation would produce the valid tree in Fig. 4-11b. Set intersection of the nodes in Figs. 4-11a and 4-11b identifies three primary deviations: ((A, +) (B, -) (C, +)). If no information about the order of the alarms was available and both nodes C and F were valid, then the search from (C, +) would not be bounded at (G, -). The valid tree in Fig. 4-11c would be generated for (C, +), rather than the tree in Fig. 4-11a. Intersection of the nodes in Figs. 4-11b and 4-11c yield six primary deviations: ((A, +) (B, -) (C, +) (D, +) (E, -) (F, +)). By evaluating each measurement in the order in which it becomes valid, the set of primary deviations identified for each measurement tends to be smaller because the search space is bounded by normal measurements.

4.4 Candidate Testing

The purpose of candidate testing is to apply other types of information, beyond the knowledge of causal adjacency used in candidate generation, to eliminate implausible candidates from the set of primary deviations. Because this information is more complex, candidate generation is first performed to identify a small set of possible origins. Computation time is minimized, and hence, the speed of the diagnosis is increased, because candidate testing is performed on this smaller set.

Because we have been investigating qualitative reasoning, the knowledge employed during the candidate testing step is qualitative. The knowledge considered here is (1) global constraints, (2) fault simulation using time delays, and (3) heuristic rules.

It is important to note that the primary deviations eliminated during candidate testing are not permanently removed from further consideration; they are only removed for the current pattern of abnormal sensors. After a new measurement deviation, the set of primary deviations from candidate generation is reexamined because the new valid measurement may change the simulation results and/or the validity of the antecedents in the heuristic rules. Primary deviations removed during one pass of candidate testing may not be eliminated during another.

4.4.1 Global Constraints

Because the causal digraph is limited to local interactions, global information may be necessary to constrain spurious interpretations. If global knowledge is known, then it should be retained and incorporated for diagnosis. The global constraints considered here are that a process variable cannot simultaneously deviate in both directions, given a set of consistent causal paths from a root node to valid measurements, and that global knowledge can be used to specify the dominant causal path when multiple paths of opposite net sign exist in the digraph.

4.4.1.1 Eliminating Roots That Yield Nodes With Multiple Values

During candidate generation, the intersection of the sets of primary deviations generated for each valid measurement was done to explain the observed measurement pattern with the fewest number of faults. If two or more faults have occurred, it may be possible to identify single fault origins that can explain the observed measurement pattern. When ambiguity exists in the digraph[†], set intersection alone is not sufficient to guarantee that the primary deviations obtained are plausible fault origins.

It is possible to eliminate some of the spurious candidates by analyzing the paths from each root to every deviated sensor. During the causal search, the process variables along a path were examined. After set intersection, the group of consistent paths from each root to every deviated sensor should also be examined. A root is eliminated from the set of primary deviations if any digraph node is assigned both '+' and '-' qualitative values along different paths from the root to abnormal measurements. The assignment of multiple values corresponds to a process variable simultaneously having both positive and negative deviations.

[†]Multiple paths of opposite net sign can be identified from the set of primary deviations identified from the valid tree. If a node appears twice in the set of primary deviations with both '+' and '-' qualitative values, then ambiguity in the digraph exists because both directions of deviation can explain the same valid measurement.

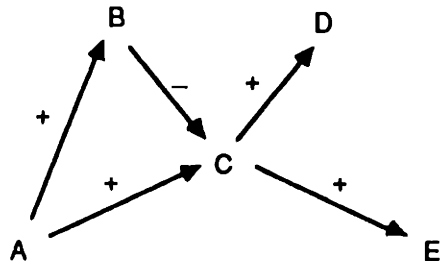


Figure 4-12
Causal Digraph With Ambiguity

The removal of primary deviations because nodes in the consistent paths were assigned multiple values is demonstrated with the causal digraph in Figure 4-12. Nodes D and E are measured and valid. The valid trees for the measurement deviations (D, -) and (E, +) are presented in Figures 4-13a and 4-13b, respectively. A single fault is assumed because set intersection is not empty: the primary deviations (A, +) and (A, -) are identified. An analysis of the paths from each root to the abnormal measurements shows that neither of these node-sign pairs can be the actual location of the fault because both fault candidates require that node C simultaneously hold '+' and '-' values. When these candidates are eliminated, the set of primary deviations is empty and the conclusion

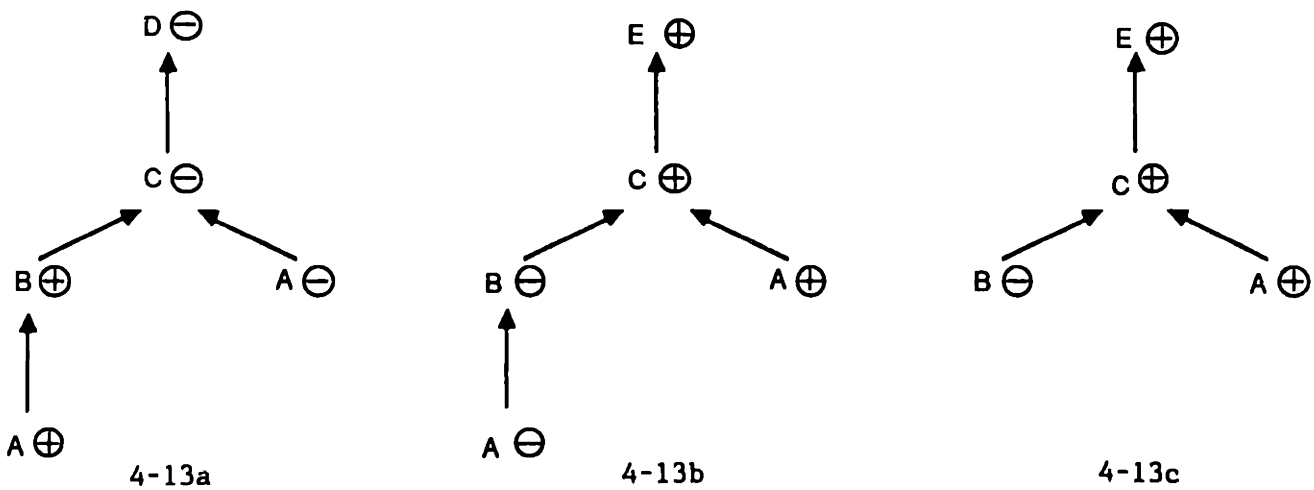


Figure 4-13
Valid Trees for (D, -) and (E, +)

reached is that multiple faults have occurred. In summary, the intersection of candidate sets identifies those root nodes that have a consistent path to every deviated measurement. Analysis of the causal paths is required to show that each process variable has a single value.

4.4.1.2 Dominant Causal Paths

If the behavior of nonadjacent process variables is known, it can be used to eliminate spurious primary deviations. The use of qualitative equalities is one approach for representing these constraints on system behavior. A qualitative equality represents the dominant causal pathway when paths of both positive and negative net sign exist between two digraph nodes. For example, in Fig. 4-12, multiple paths of opposite sign exist between nodes A and C. If the qualitative equality [A] = [C] is known and the fault propagates from A to C, the deviation at C is known to be in the same direction as the deviation at A. The positive path AC is dominant over the negative path ABC.

Qualitative equalities can be incorporated while checking the consistent paths for digraph nodes with multiple value assignments, as discussed in Section 4.4.1.1. The equality must hold when the consistent path between the origin and the deviated sensor contains both the nodes in the equality. Qualitative equalities can also be incorporated into the candidate generation procedure. The equalities are used to bound the valid tree during the causal search. The valid tree constructed for (E, +), when the constraint is known, is shown in Fig. 4-13c. The valid tree terminates at (B, -) because the qualitative equality between nodes A and C would be violated if the consistent node (A, -) was added. Therefore, with the global constraint, (A, -) is not identified as a primary deviation. Note that the equality bounds the search space only when the fault propagates through both nodes of the equality.

4.4.2 Simulation Using Qualitative Time Delays

Qualitative information characterizing process dynamics can be used to reduce the set of primary deviations.

4.4.2.1 Qualitative Modeling of Dynamics

Information about a system's dynamic behavior is implicit in the equations chosen to model the system. If a differential equation is chosen to model the relationship between a group of process variables, then the dynamics between the terms of the equation are important, i.e., they are on the same order of magnitude as the dynamics of the overall system and/or the scanning frequency of the measurements. If an algebraic equation is selected, the time lag between the change in one variable and the change in another is negligible; the process variables in the model are assumed to change together. To illustrate this idea further, consider a differential mass balance equation around a process unit containing an incompressible fluid. If the mass of the fluid remains constant over time, as in a pipe, then the system model becomes $F_{in} = F_{out}$. Changes in inlet flow rate are instantly propagated to the outlet because the time constant for this equality relationship is assumed to be zero. If the volume of the fluid in the unit can change with time, as in a tank, and F_{out} is a function of the fluid volume, then changes in F_{out} due to changes in F_{in} are delayed by a positive time constant.

The qualitative approximations of numerical time constants used in this paper are the two qualitative values '0' and '1', representing zero delay between a cause and its effect, and positive delay, respectively. As a first approximation in assigning qualitative values, causal arcs developed from balance equations in which the derivative is not constant, functional relationships with integral action, and arcs characterizing transportation lag have positive time delay. Driving force equations, non-integral functional relationships, and equalities yield arcs which have zero time delay.

4.4.2.2 Using Qualitative Time Delays to Eliminate Primary Deviations

If there were multiple paths of opposite sign between a primary deviation and a sensor, the initial deviation of the sensor could be predicted solely on the basis of the delay time: the deviation would be in the direction that would make the path with the shortest delay time

consistent. Without the numerical values for positive time delays, the time when a measurement will alarm cannot be specified. But for paths with zero time delay, the disturbance is instantly propagated from the initial node to the terminal node in the path. This knowledge can be used to eliminate primary deviations from the set of possible origins.

For each root node in the set of primary deviations, fault simulation is performed from the root node along the arcs with zero time delay. Causally downstream nodes are assigned values so that the branches in the simulation tree are consistent. If any node in the tree is measured and the qualitative value of the actual measurement is either normal, if the measured variable is not a controlled variable, or opposite of the sign in the simulation tree, then the root node should be eliminated from the set of primary deviations. If there are no measurement nodes in the tree, or if the actual measurements have the values that match the fault simulation, then the node remains a candidate. A normal measurement causally downstream from the primary deviation is acceptable if it is a controlled variable, because the control system may compensate for the disturbance and yield a normal value.

Returning to the tank example, three arcs in the causal digraph illustrated in Fig. 3-2 have positive time delay: $F_1 \rightarrow V$, $F_2 \rightarrow V$, and $L_{error} \rightarrow v_1$, assuming an integral control mode. All other arcs have the value '0'. Two faults, "Blockage in pipe PIPE-A" and "Outlet blockage in tank TANK-1," identified during candidate generation and presented in Table 4-1, are eliminated when delay times are considered. Simulation trees from the primary deviations (R_{23}, +) and (R_2, +) reach the flow sensor. The paths from these nodes to the sensor represent the instantaneous pressure/flow rate propagation along the piping length. The two faults are eliminated because the qualitative value '-' is required at the flow sensor to make the paths from each root consistent. The actual value of the measurement is '+'.

4.4.3 Heuristic Rules

Knowledge in the form of rules can be used to reduce the number of primary deviations. Although the term heuristic is used, both experiential

and model-based knowledge can be represented in this format. Several illustrative examples are presented. These rules were derived from analyzing case studies.

Rule 1: If the controlled variable in a control system is normal, then the control system is working. Therefore, remove any primary deviations associated with the control system (from the controller through the control valve) and the desired set point. [Note: This rule assumes that sufficient time has elapsed for faults within the control system to cause the deviation of the controlled variable.]

Rule 2: If a control system is working, a disturbance propagates into the control system through the control valve, and the control system compensates for the disturbance by closing the valve, then the propagation of the failure is always halted and the controlled variable remains normal. Therefore, any primary deviations causally upstream from the control valve that cause the valve to close can be eliminated.

In addition to referencing qualitative data, rule antecedents can also reference numerical data (e.g., measurement values, reference values, and output from numerical calculations, including rates of change, simulations, statistics, etc.) to reduce the number of primary deviations.

Rule 3: If the measured, numerical value of a pressure sensor is negative, then the sensor has failed. Therefore, eliminate all other primary deviations.

Rule 4: If the normal process and measurement noise of a sensor disappears (variance goes to zero), then the sensor has failed. Therefore, eliminate all other primary deviations.

4.4.4 Tank Example Revisited

Candidate testing is demonstrated on the tank example investigated in Section 4.3.6. Several of the faults presented in Tables 4-1, 4-2, and 4-3

are not consistent with higher-level knowledge, although they are consistent with the local knowledge used during candidate generation.

Table 4-4 lists the primary deviations and possible faults for "F-SENSOR High" after candidate generation and testing. Global constraints were not specified for the problem, and hence, no primary deviations were eliminated due to constraints. Fault simulation using time delays removes five primary deviations: ((F23, -) (R23, +) (PB, -) (R2, +) (F2, -)). Heuristic Rule 1 eliminates three primary deviations: ((L-SP, -) (L-ERROR, +) (V1, +)), and heuristic Rule 2 removes one primary deviation: ((P5, -)). The nine faults in Table 4-1 are reduced to two.

Table 4-4

List of Primary Deviations and Possible Faults for
F-SENSOR High after Candidate Generation and Testing

Measurements:
F-SENSOR high

List of Primary Deviations: ((F1 +) (V +) (L +) (PB +) (R2 -) (F2 +) (P2 +) (R23 -) (F23 +) (P3 +) (R34 -) (F34 +) (P4 +) (R45 -) (F45 +) (F-SENSOR +))

Possible Faults:
1> Sensor F-SENSOR failed high.
2> High flow rate F1 entering tank TANK-1.

Table 4-5 lists the primary deviations and possible faults for F-SENSOR high and L-SENSOR high after candidate generation and testing. Fault simulation using time delays removes five primary deviations: ((F23, -) (R23, +) (PB, -) (R2, +) (F2, -)). The three faults in Table 4-2 are reduced to a single fault.

Table 4-5

List of Primary Deviations and Possible Faults for
F-SENSOR High and L-SENSOR High after Candidate Generation and Testing

Measurements:
  F-SENSOR high
  L-SENSOR high

List of primary deviations: ((F1 +) (V +) (L +) (P2 +) (P3 +) (P4 +))

Possible Faults:
  1> High flow rate F1 entering tank TANK-1.

Table 4-6 lists the primary deviations and possible faults for F-SENSOR high and L-SENSOR low after candidate generation and testing. Heuristic Rule 2 removes one primary deviation: ((P5, -)). The five faults in Table 4-3 are reduced to three.

Table 4-6

List of Primary Deviations and Possible Faults for
F-SENSOR High and L-SENSOR Low after Candidate Generation and Testing

Measurements:
  F-SENSOR high
  L-SENSOR low

List of primary deviations: ((L-SP -) (L-ERROR +) (V1 +) (PB +) (R2 -)
 (F2 +) (R23 -) (F23 +) (R34 -) (F34 +) (R45 -) (F45 +))

Possible Faults:
  1> The set point of LEVEL_CONTROL_SYSTEM set low.
  2> Control system LEVEL_CONTROL_SYSTEM failed high.
  3> Control valve CV-1 failed open.

4.5 Mapping Faults to Primary Deviations

A table mapping faults to primary deviations is used to produce the list of faults from the reduced set of root nodes after candidate testing. To keep the diagnostic system modular, tables are constructed for system components. Therefore, given a primary deviation and the class of process equipment, the fault is identified. An example of the mapping of faults to primary deviations for a centrifugal pump is presented in Table 4-7. Given the primary deviation (R, +) and the component class "centrifugal pump", the faults "impeller suction or discharge opening partially plugged" and "suction strainers clogged" are identified. Note that not all primary deviations have faults associated to them; for example, there are no faults whose primary deviations are (F, +), (R, -), and (C_A, +), even though these nodes could be identified as possible origins.

The relationship between faults and primary deviations depends on the context in which the physical system functions. Information about the context is necessary to specify whether a fault should be considered and whether it should be mapped to the + or - deviation of a digraph node. For example, to identify possible faults for low pressure in a vessel, knowledge about the physical characteristics of the unit (e.g., number of flanges, relief valves, rupture disks, drain valves) would be important in identifying possible causes. These "leakage" faults would then be mapped to (P, -) only if the pressure of the vessel was greater than atmospheric pressure. If the vessel pressure was less, then the fault would be associated with the primary deviation (P, +) because the pressure of the vessel would increase as air entered the system.

Context-independent rules for generating the fault tables can be used to correctly map faults to primary deviations for a given context. Like the context-independent component models for generating the correct causal digraph, these rules can be grouped by process component. Rule antecedents reference the unit's physical characteristics, design values, and possibly the current process measurements. Rule consequents are the faults for the primary deviations in the given component. Plant-specific faults can be added to the table.

Table 4-7

Mapping of Faults to Primary Deviations For a Centrifugal Pump[†]

(Subscript 1 denotes inlet, subscript 2 denotes outlet)

Assumptions:
1. Newtonian liquid
2. Strainers
3. P_1, $P_2 > P_{atm}$
4. Electric motor
5. Insulation
6. T_1, $T_2 > T_{atm}$

Primary Deviation	Possible Faults
(P_2, -)	Leakage of fluid to external environment
(F, -)	Entrained vapor in fluid a. air leak in suction line b. stuffing box packing worn or liquid seal plugged, allowing leakage of air into pump casing
	Change in liquid physical properties (increased density, increased viscosity, decreased vapor pressure)
	Temperature increase causing cavitation of hot or volatile liquid
(R, +)	Impeller suction or discharge opening partially plugged
	Suction strainers clogged
(T, +)	External fire
(T, -)	Insulation removed
(ω, -)	Broken impeller shaft; broken coupling
	Impeller damaged
	Impeller key missing
(i, -)	Loss of power to electric motor
(i, +)	Power too high to electric motor

[†]Faults obtained from Centrifugal Pumps (Newtonian Liquids), AIChE Equipment Testing Procedure, 1984.

When the causal digraph for a particular process unit is desired, an instance of the general component database is created. The database holds design values of parameters and process variables for the particular unit, or for real-time processing, it may reference measurements or equations for calculating the values required by the rules. The values in this component database are used to generate the table mapping faults to primary deviations. The table relates faults to their primary deviations for a given set of conditions and assumptions about the operating state and equipment design specifications.

If the system changes (e.g., a flow is rerouted, set points are changed, a tank is emptied, etc.), the values in one or more databases may change. By reevaluating the rule bases associated with each of the units affected by the change, the appropriate faults and the correct direction of deviation are maintained in the table.

The actual mapping of faults to primary deviations comes from experience and expertise with a particular component's operation. For example, only someone intimately familiar with the mechanical design and operation of a centrifugal pump knows that air can enter through worn packing, causing a loss in flow rate. The assignment of a fault to a primary deviation can be checked by fault simulation. The causal digraph predictions, generated by fault propagation from the primary deviation, must match the actual physical system when the fault is present. All faults desired to be diagnosed by the diagnostic system must be included in the fault table.

If the number of fault candidates is large, the faults can be ranked based on failure rate data. Then, only the most probable faults would be initially presented to the operator. This reduction allows the operator to focus his attention on the subset of fault candidates with the greatest likelihood. If these faults are investigated and none are found to be the actual cause, the operator can then review candidates with lower probabilities. For example, the median failure rate for a pump is 3×10^{-5} faults/hr (with upper and lower bounds 3×10^{-6} to 3×10^{-4}), whereas the median failure rate for a rupture in a pipe with an inner diameter less than three inches is 1×10^{-9} faults/hr (with upper and lower bounds 3×10^{-11} to 3×10^{-8}) (AEC [1974]). If both of these faults appeared in

the list of possible candidates, the operator, without any additional knowledge about the failure, should investigate the pump failure first, because its failure rate is four orders of magnitude greater than that of a pipe rupture.

If the standard operating, safety, and shutdown procedures are stored as data in the computer, then the operator can input the true cause of the upset into the system and generate the appropriate corrective action.

Chapter 5

DIEX: A MODEL-BASED DIAGNOSTIC SYSTEM PROTOTYPE

The causal digraph and the diagnostic strategy, developed in Chapters 3 and 4, were implemented in a computer program. In this chapter, DIEX (Diagnostic Expert), a model-based diagnostic system prototype, is described and tested on three example processes of increasing complexity. For each example, the system digraph is developed from general component models for a specific set of context-specific assumptions. The performance of the prototype is then demonstrated on each of the processes. For each example, several faults are selected and the qualitative measurement deviations for each fault are entered into DIEX. The list of faults identified by the diagnostic system is checked so that (1) the actual fault is contained in the list, and (2) all faults that are inconsistent with other available information are eliminated.

5.1 Description

DIEX (Diagnostic Expert) is a model-based diagnostic system prototype. DIEX has been implemented in Franz Lisp running under UNIX on a DEC VAX 11-780. Major portions of the computer code are listed in Appendix B. The description of the prototype is separated into two sections: knowledge representation and diagnostic strategy.

Although the implementation effort focused on flexibility rather than on efficient code, the diagnostic system performed quickly. For the largest example studied (a continuous stirred tank reactor with an exothermic reaction and external cooling, which contained 122 nodes and 173 arcs), the maximum system response time for candidate generation and compiling the list of faults from the set of primary deviations was under 5 seconds per valid measurement.

5.1.1 Knowledge Representation

Plant topography and the specific unit design information is entered through an interactive design program. In many cases, numerical values of the process parameters are necessary to specify the correct causal interactions from the causal models. Structural information is used to match the ports of interconnected units. The design program creates a data file which is used by a second program to construct the causal digraph. Object-oriented programming is used to specify the digraph. Node and arc flavors are used for instantiating specific process variables and their causal interactions. The flavor definitions are presented in Appendix B.

General component causal models have been developed for ten types of process equipment and four elementary chemical reactions. The process equipment models include pipe, tee, centrifugal pump, valve (2-port), tank, heat exchanger, vaporizer, continuous stirred tank reactor (CSTR), sensor, and single-input single-output (SISO) control system models. Rules in the models are similar to those presented in Example 2 in Section 3.5.2.

Data structures for storing global information (Section 3.3.4) and for representing the physical adjacency of process units (Section 3.5.1.1) were not implemented.

5.1.2 Diagnostic Strategy

During candidate generation, the system constructs a valid tree for each abnormal measurement from the arcs in the causal digraph, as discussed in Section 4.3.4. The qualitative values of the measurements can be input into the diagnostic system prototype one at a time or in groups. When more than one measurement is valid, an active set of primary deviations from set intersection is maintained. Thus, a single intersection is performed during each pass. Causal simulation using time delays and heuristic rules were implemented for candidate testing. Rules are incorporated into the prototype as Lisp functions rather than through the use of a general rule interpreter. Global constraints were not implemented because they are relatively straightforward. Computer code for the major sections of the prototype is presented in Appendix B.

The mapping of faults to primary deviations was done for the particular contexts of the example processes studied. Context-specific rule bases were developed for the following process equipment: pipe, tee, centrifugal pump, valve (2-port), tank, heat exchanger, vaporizer, CSTR, sensor, and SISO control system. These are also included in Appendix B. Plant-independent rule bases were not developed.

5.2 Examples

Three examples are presented to illustrate the construction of the causal digraph from the component models and the performance of the DIEX system prototype. The digraphs are based on the methodology and guidelines presented in Chapter 3. In each example, context-specific design data and the interconnections of system components were necessary to obtain the overall process digraph. The system digraphs developed for each example are presented in Appendix C. In the schematics, the perpendicular lines across major flow streams denote equipment ports. The number at each port denotes the subscript used for the process variables associated with the port.

5.2.1 Tank With a Level Control System

The schematic for a tank with a level control system, illustrated in Fig. 3-1, is repeated in Figure 5-1. A liquid feed enters the tank, which is at atmospheric pressure. A SISO control system maintains the tank level at the desired set point. The outlet flow rate is also measured. Arcs related to bulk fluid flow are considered.

The schematic was decomposed into general process components for digraph construction. The schematic contains the following components: one tank with one inlet port and one outlet port, one control valve, two sensors, one SISO control system, and two pipes.

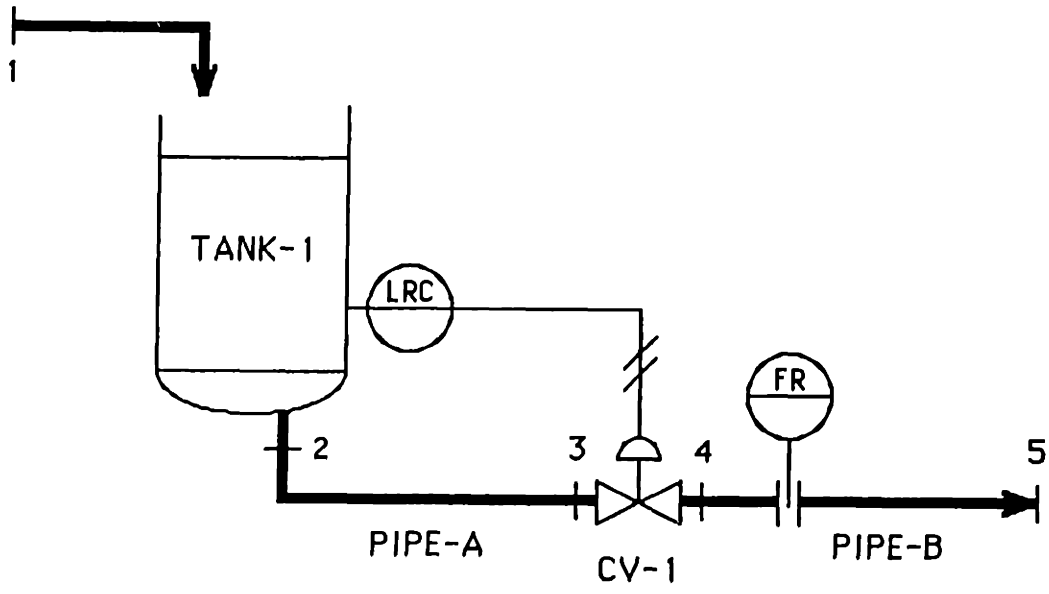


Figure 5-1
Process Schematic of a Tank with a Level Control System

The system digraph was constructed from the general component causal models and process design data. The context-specific knowledge used to build the digraph is

1. The valve is open
2. All flow rates are positive and in the directions assumed
3. Liquid inlet in the tank is above liquid level
4. Tank is at atmospheric pressure.

The system digraph, presented in Appendix C-1, contains 21 nodes and 29 arcs.

Actual system output after candidate generation and testing was presented in Section 4.4.4. Table 4-4 presents the possible fault candidates for (F_{sensor}, +), Table 4-5 lists possible origins for (F_{sensor}, +) and (L_{sensor}, +), and Table 4-6 lists fault candidates for (F_{sensor}, +) and (L_{sensor}, -).

5.2.2 Vaporizer

The schematic of a vaporizer is illustrated in Figure 5-2. The pure-component, liquid feed is vaporized by heat transferred from the heating medium through the coiled tubes in the vaporizer. Two SISO control systems maintain the vaporizer liquid level and pressure at their desired set points. Inlet flow rate is also measured.

The schematic was decomposed into general process components for digraph construction. The schematic contains the following components: one vaporizer, two control valves, three sensors, two SISO control systems, and two pipes.

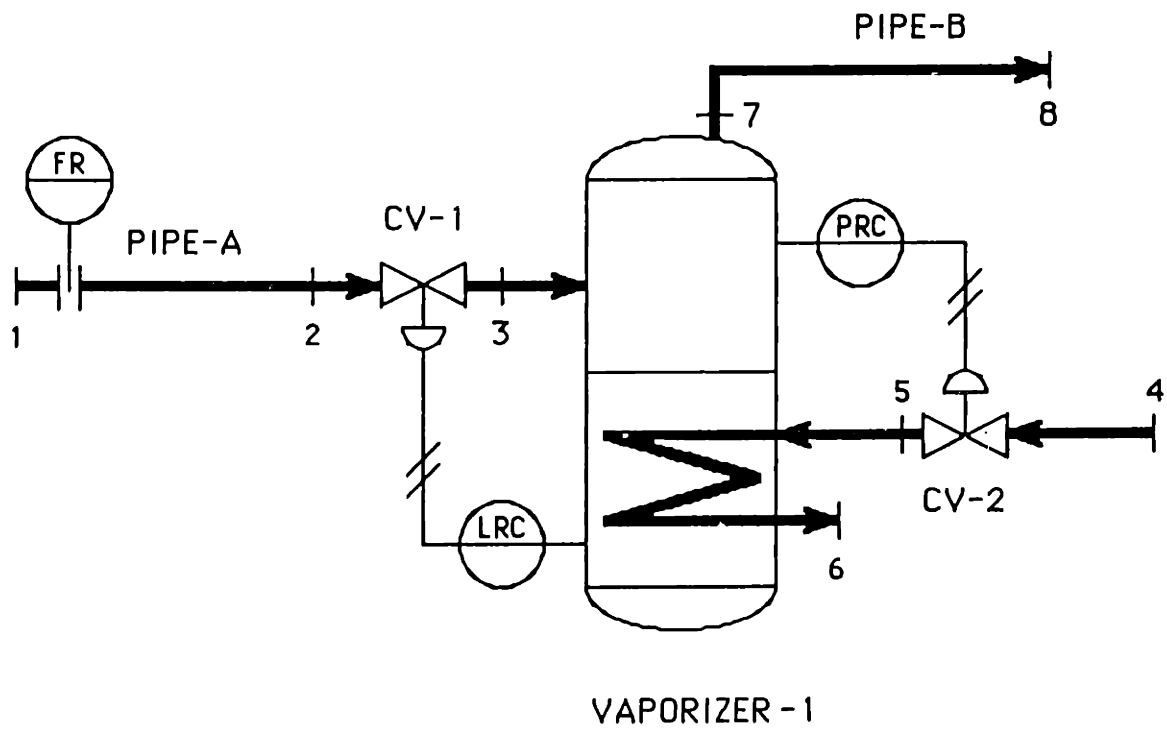


Figure 5-2
Process Schematic of a Vaporizer

The system digraph was constructed from the general component causal models and process design data. The context-specific knowledge used to build the digraph is

1. All valves are open
2. All flow rates are positive and in the directions assumed
3. $T_3 < T < T_5$
4. Constant fluid properties (ρ, C_p, λ)
5. Liquid inlet in the vaporizer is above liquid level
6. No structural faults in the vaporizer.

The system digraph, presented in Appendix C-2, contains 45 nodes and 65 arcs. The assumptions for constructing the table mapping faults to primary deviations are $P > P_{atm}$, $T > T_{atm}$, and the system components that handle fluid are insulated.

Four failures (level control system failed low, low temperature heating fluid, vapor leak from the vaporizer, and fire around the vaporizer) are examined. For each pattern of abnormal measurements, the reduced set of primary deviations and the list of possible fault origins generated by DIEX are presented.

5.2.2.1 Level Control System Failed Low

Candidate generation for the valid measurement (F_{sensor}, +) yielded 25 faults (45 primary deviations). Candidate generation for (L_{sensor}, +) and set intersection reduced the number of faults to four (ten primary deviations). Heuristic Rule 2 eliminated the primary deviation (P_1, +) during candidate testing. The third valid sensor (P_{sensor}, -) added no additional information. The final set of faults is presented in Table 5-1.

Table 5-1

Faults Identified for Level Control System Failed Low

Measurements:
  F-SENSOR high
  L-SENSOR high
  P-SENSOR low

List of primary deviations: ((L-SP +) (L-ERROR -) (V1 +) (R3T -) (F3T +) (R23 -) (F23 +) (R12 -) (F12 +))

Possible Faults:
  1> The set point of LEVEL_CONTROL_SYSTEM set high.
  2> Control system LEVEL_CONTROL_SYSTEM failed low.
  3> Control valve CV-1 failed open.

5.2.2.2 Low Temperature Heating Fluid

Candidate generation for the valid measurement (P_{sensor}, -) identified 24 faults (38 primary deviations). Candidate generation for (F_{sensor}, -) and set intersection reduced the number of possible fault candidates to 20 (32 primary deviations). Fault simulation during candidate testing eliminated eight primary deviations ((P_8, -) (R_{78}, -) (F_{78}, +) (P_7, -) (R_{T7}, -) (F_{T7}, +) (VAPOR_RATE, -) (P_T, -)) and four faults. The remaining sixteen faults are presented in Table 5-2.

Table 5-2

Faults Identified for Low Temperature Heating Fluid

Measurements:
  P-SENSOR low
  F-SENSOR low

List of primary deviations: ((P2 +) (P3 +) (R3T -) (P-SP -) (P-ERROR +) (V2 -) (P4 -) (R45 +) (F45 -) (P5 -) (P6 +) (R56 +) (T4 -) (T5 -) (F56 -) (T6 -) (RH +) (T1 -) (T2 -) (T3 -) (Q -) (F3T +) (T -) (P-SENSOR -))

Possible Faults:
  1> The set point of PRESSURE_CONTROL_SYSTEM set low.
  2> Control system PRESSURE_CONTROL_SYSTEM failed high.
  3> Control valve CV-2 failed closed.
  4> Blockage in control valve CV-2.
  5> Leak in control valve CV-2.
  6> Blockage in heating coils in vaporizer VAPORIZER-1.
  7> Insulation removed on control valve CV-2.
  8> Severe fouling in heating coils in vaporizer VAPORIZER-1.
  9> Insulation removed on pipe PIPE-A.
  10> Insulation removed on control valve CV-1.
  11> Insulation removed on vaporizer VAPORIZER-1.
  12> Sensor P-SENSOR failed low.
  13> Low pressure upstream of control valve CV-2.
  14> High pressure downstream of P6 in vaporizer VAPORIZER-1.
  15> Low temperature fluid entering control valve CV-2.
  16> Low temperature fluid entering pipe PIPE-A.

5.2.2.3 Vapor Leak From Vaporizer

As in the previous example, candidate generation for the valid measurement (P_{sensor}, -) generated 38 primary deviations and 24 faults. The valid measurement (F_{sensor}, +) eliminated two primary deviations and no faults. Twenty four faults (36 primary deviations) remain after candidate generation and set intersection for the two valid measurements. Heuristic Rule 1 eliminated three primary deviations ((L_{sp}, +) (L_{error}, -) (V_1, +)) during candidate testing. The final set of fault candidates is presented in Table 5-3.

Table 5-3

Faults Identified for Vapor Leak From Vaporizer

```
Measurements:
  P-SENSOR low
  F-SENSOR high

List of primary deviations: ((L -) (R23 -) (F23 +) (R3T -) (P-SP -)
 (P-ERROR +) (V2 -) (P4 -) (R45 +) (F45 -) (P5 -) (P6 +) (R56 +) (T4 -)
 (T5 -) (F56 -) (T6 -) (RH +) (T1 -) (T2 -) (T3 -) (Q -) (F3T +) (T -)
 (P8 -) (R78 -) (F78 +) (P7 -) (RT7 -) (FT7 +) (VAPOR_RATE -) (PT -)
 (P-SENSOR -))

Possible Faults:
  1> Liquid leak from vaporizer VAPORIZER-1.
  2> The set point of PRESSURE_CONTROL_SYSTEM set low.
  3> Control system PRESSURE_CONTROL_SYSTEM failed high.
  4> Control valve CV-2 failed closed.
  5> Blockage in control valve CV-2.
  6> Leak in control valve CV-2.
  7> Blockage in heating coils in vaporizer VAPORIZER-1.
  8> Insulation removed on control valve CV-2.
  9> Severe fouling in heating coils in vaporizer VAPORIZER-1.
  10> Insulation removed on pipe PIPE-A.
  11> Insulation removed on control valve CV-1.
  12> Insulation removed on vaporizer VAPORIZER-1.
  13> Leak in pipe PIPE-B.
  14> Leak at outlet [P7] in vaporizer VAPORIZER-1.
  15> Vapor leak from vaporizer VAPORIZER-1.
  16> Sensor P-SENSOR failed low.
  17> Low pressure upstream of control valve CV-2.
  18> High pressure downstream of P6 in vaporizer VAPORIZER-1.
  19> Low temperature fluid entering control valve CV-2.
  20> Low temperature fluid entering pipe PIPE-A.
  21> Low pressure downstream of pipe PIPE-B.
```

5.2.2.4 Fire at Vaporizer

The valid measurements (P_{sensor}, +) and (F_{sensor}, +) were entered concurrently. Candidate generation identified 19 faults (32 primary deviations). The valid level sensor eliminated the single deviation (P_{sensor}, +) and the fault "Sensor P-SENSOR failed high." Heuristic Rule 2 eliminated five faults (11 primary deviations) and simulation removed an additional six faults (13 primary deviations) through candidate testing. Seven faults and seven primary deviations, presented in Table 5-4, remain.

Table 5-4

Faults Identified for Fire at Vaporizer

```
Measurements:
  P-SENSOR high
  F-SENSOR high
  L-SENSOR low

List of primary deviations: ((P-SP +) (P-ERROR -) (V2 +) (T1 +) (T2 +)
 (T3 +) (T +))

Possible Faults:
  1> The set point of PRESSURE_CONTROL_SYSTEM set high.
  2> Control system PRESSURE_CONTROL_SYSTEM failed low.
  3> Control valve CV-2 failed open.
  4> Fire at pipe PIPE-A.
  5> Fire at control valve CV-1.
  6> Fire at vaporizer VAPORIZER-1.
  7> High temperature fluid entering pipe PIPE-A.
```

5.2.3 Continuous Stirred Tank Reactor

The schematic for a continuous flow stirred tank reactor (CSTR) with an external heat exchanger is illustrated in Figure 5-3. Reactant A is fed into the CSTR and reacts to form product B in the first-order, elementary reaction $A \rightarrow B$. Both product and reactant are in the liquid phase. The reaction is exothermic, and the inlet temperature of the reactant is less than the bulk temperature of the reactor. The reactor is well-mixed so that the state variables and physical properties of the fluid in the reactor are assumed to be uniform. The reactant mixture is recycled through a water-cooled heat exchanger to remove heat. Three SISO control systems maintain the reactor liquid level, the reactor temperature, and the recycle flow rate at their respective set points. Additional measurements include the reactor pressure, recycle temperature, cooling water flow rate, product flow rate, and product concentration.

The schematic was decomposed into general process components for digraph construction. The schematic contains the following components: one CSTR with two inlet ports and one outlet port, one heat exchanger, one centrifugal pump, three control valves, nine sensors (the concentration sensor measures both C_A and C_B), three SISO control systems, five pipes, and one tee.

The system digraph was constructed from the general component causal models and process design data. The context-specific knowledge used to build the digraph is

1. All valves are open
2. All flow rates are positive and in the directions assumed
3. T_2, $T_{12} < T_R$
4. $T_6 > T_{14}$
5. $P_6 > P_{14}$ (for heat exchanger structural fault)
6. Constant fluid properties (ρ, C_p)
7. Exothermic reaction
8. Constant heat of reaction (ΔH_r)
9. Recycle concentrations C_A and C_B are identical to the reactor concentrations.

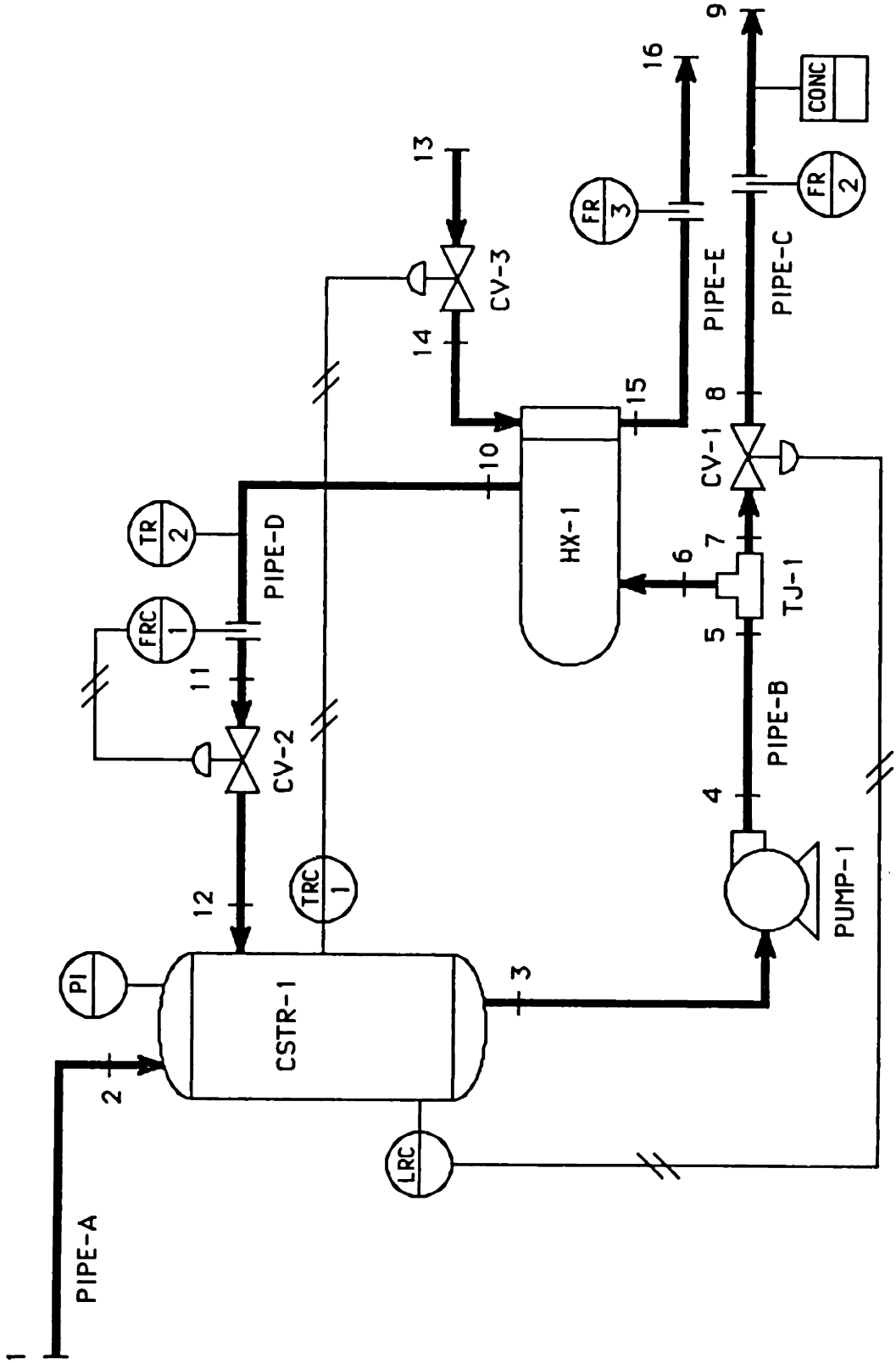


Figure 5-3

Process Schematic of a Continuous Stirred Tank Reactor

The system digraph, presented in Appendix C-3, contains 122 nodes and 173 arcs. The assumptions for constructing the table mapping faults to primary deviations are $P > P_{atm}$, $T > T_{atm}$, a centrifugal pump with an electric motor drive, and the system components that handle fluid are insulated.

Five failures (control valve failed closed, blockage in pump, temperature sensor fails high, low inlet concentration of reactant, and catalyst fouling) are examined. For each pattern of abnormal measurements, the reduced set of primary deviations and the list of possible fault origins generated by DIEX are presented.

For this example, a quantitative, dynamic model was developed to generate the deviations away from steady state. Faults were introduced into the model and the system's dynamic response was simulated. Qualitative values were assigned to the observed measurement deviations. These values were entered into the diagnostic system prototype.

5.2.3.1 Control Valve CV-2 Failed Closed

Candidate generation for the valid measurements ($F1_{sensor}$, -) and ($F2_{sensor}$, +) identified 27 faults (64 primary deviations). Candidate generation for the three additional valid measurements ($F3_{sensor}$, +), (P_{sensor}, +), and ($T1_{sensor}$, +), entered together, and set intersection reduced the number of possible fault candidates to 14 (30 primary deviations). Fault simulation during candidate testing eliminated the following 19 primary deviations: ((CURRENT +) (P4 -) (R34 -) (OMEGA +) (F34 +) (PB +) (P3 -) (RR3 -) (FR3 +) (VL -) (THETA-1 -) (P4 +) (P1 -) (R12 +) (F12 -) (P2 -) (R2R +) (F2R -) (THETA-1 +)). Seven faults (11 primary deviations), listed in Table 5-5, are identified by DIEX.

Table 5-5

Faults Identified for Control Valve CV-2 Failed Closed

Measurements:
  F1-SENSOR low
  F2-SENSOR high
  F3-SENSOR high
  P-SENSOR high
  T1-SENSOR high

List of primary deviations: ((F-SP -) (F-ERROR +) (V2 -) (R12R +) (F12R -) (R1112 +) (F1112 -) (R1011 +) (F1011 -) (R610 +) (F610 -))

Possible Faults:
  1> The set point of FLOW_CONTROL_SYSTEM set low.
  2> Control system FLOW_CONTROL_SYSTEM failed high.
  3> Control valve CV-2 failed closed.
  4> Inlet blockage [R12R] in reactor CSTR-1.
  5> Blockage in control valve CV-2.
  6> Blockage in pipe PIPE-D.
  7> Blockage in hot stream in heat exchanger HX-1.

5.2.3.2 Blockage in Pump

Candidate generation for the valid measurements ($F1_{sensor}$, -) and ($F2_{sensor}$, -) identified 41 faults (75 primary deviations). Candidate generation for (L_{sensor}, +) and set intersection reduced the number of possible fault candidates to 38 (70 primary deviations). The two valid measurements ($T1_{sensor}$, +) and (P_{sensor}, +) eliminated seven additional faults and 10 primary deviations. Thus, after candidate generation and set intersection of the five measurements, 31 faults (60 primary deviations) remained. Fault simulation during candidate testing eliminated 24 faults (51 primary deviations). The remaining seven faults are presented in Table 5-6.

Table 5-6

Faults Identified for Blockage in Pump

Measurements:
  F1-SENSOR low
  F2-SENSOR low
  L-SENSOR high
  T1-SENSOR high
  P-SENSOR high

List of primary deviations: ((R45 +) (CURRENT -) (R34 +) (OMEGA -) (F34 -) (RR3 +) (FR3 -) (F1-SENSOR -) (F45 -))

Possible Faults:
  1> Blockage in pipe PIPE-B.
  2> Loss of power to electric motor on pump PUMP-1.
  3> Blockage of section or discharge in pump PUMP-1.
  4> Broken shaft or coupling in pump PUMP-1.
  5> Entrained vapor or change of physical properties in pump PUMP-1.
  6> Outlet blockage [RR3] in reactor CSTR-1.
  7> Sensor F1-SENSOR failed low.

5.2.3.3 Temperature Sensor T_1 Failed High

Candidate generation for the valid measurement ($T1_{sensor}$, +) identified 59 faults (129 primary deviations). Fault simulation during candidate testing eliminated 128 primary deviations, because a causal path of zero time delay exists from the measured variable T_R to the pressure sensor. Therefore, any fault whose causal path contains T_R must also cause the pressure measurement to be valid.

All subsequent measurement deviations, caused by the temperature control system, can be traced back to the temperature sensor failure. Only the single fault, presented in Table 5-7, is identified.

Table 5-7

Faults Identified for Temperature Sensor T_1 Failed High

```
Measurements:
  T1-SENSOR high
  F3-SENSOR high
  T2-SENSOR high
  P-SENSOR low
  CA-SENSOR high
  CB-SENSOR low

List of primary deviations: ((T1-SENSOR +))

Possible Faults:
  1> Sensor T1-SENSOR failed high.
```

5.2.3.4 Low Inlet Concentration C_A

Candidate generation for the valid measurement ($F3_{sensor}$, -) and ($T2_{sensor}$, +) identified 51 faults (102 primary deviations). Candidate generation for (CA_{sensor}, -) and (CB_{sensor}, -), and set intersection reduced the number of possible fault candidates to 40 (71 primary deviations). Fault simulation during candidate testing eliminated all but two faults (three primary deviations). They are presented in Table 5-8.

Table 5-8

Faults Identified for Low Inlet Concentration C_A

Measurements:
  F3-SENSOR low
  T2-SENSOR high
  CA-SENSOR low
  CB-SENSOR low

List of primary deviations: ((CA1 -) (CA2 -) (CAR -))

Possible Faults:
  1> Side reaction occurring in reactor CSTR-1, depleting reactant.
  2> Low concentration of species A entering pipe PIPE-A.

5.2.3.5 CSTR Catalyst Fouling

Candidate generation for the valid measurements (P_{sensor}, -) and ($T1_{sensor}$, -) identified 62 faults (131 primary deviations). Candidate generation for the two additional valid measurements (CA_{sensor}, +) and (CB_{sensor}, -), and set intersection reduced the number of possible fault candidates to 60 (127 primary deviations). Fault simulation during candidate testing eliminated all but a single fault (two primary deviations). The fault and primary deviations are presented in Table 5-9.

Table 5-9

Faults Identified for Catalyst Fouling

Measurements:
  T1-SENSOR low
  P-SENSOR low
  CA-SENSOR high
  CB-SENSOR low

List of primary deviations: ((K -) (REACTION_RATE -))

Possible Faults:
  1> Catalyst fouling in reactor CSTR-1.

5.2.4 Discussion of Examples

Two desired attributes of a diagnostic system, presented in Section 2.4, are (1) the actual fault origin is included in the list of fault candidates, and (2) that the number of spurious fault candidates included in the list is minimized. For the examples investigated, DIEX exhibited both of these attributes: for every fault considered, the actual origin was included in the set of fault candidates, and the maximum resolution between the candidates was achieved for the given number and position of the measurements. No further resolution between the fault candidates can be obtained without additional information.

The largest number of fault candidates is found in Tables 5-2 and 5-3. The relatively large number of candidates arises from ambiguity in the causal digraph and a small number of measurements. Two paths of opposite net sign exist between the vaporizer temperature T and F_{sensor}. The paths are interpreted for (T, -):

Path from (T, -) to (F_{12}, +)

Decreasing the vaporizer temperature (T, -) causes a decrease in the rate of vaporization (VAPOR_RATE, -), which decreases the vaporizer pressure (P_T, -). Decreasing the vaporizer pressure increases the inlet flow rate (F_{3T}, +), the flow rate through the control valve CV-1 (F_{23}, +), and the flow rate in pipe PIPE-A (F_{12}, +).

Path from (T, -) to (F_{12}, -)

Decreasing the vaporizer temperature (T, -) causes a decrease in the rate of vaporization (VAPOR_RATE, -), which increases the liquid level in the vaporizer (L, -). The level control system closes the control valve (V_1, -), which decreases the flow rate through the valve (F_{23}, -) and the flow rate in pipe PIPE-A (F_{12}, -).

Because two paths of opposite net sign exist, the measurement of the inlet flow rate F_{12} provides no information for discriminating between possible fault candidates that lie causally above the vaporizer temperature. The 16 faults in Table 5-2 are associated with primary deviations

at, and causally above, the digraph node representing the vaporizer temperature. The ambiguity in the digraph is the reason why all 16 faults in Table 5-2 are also found in Table 5-3, even though the deviation at F_{sensor} is in the opposite direction.

If the knowledge that the dominant causal path was the net positive path, from T to F_{sensor} through the control system, then for a high flow rate measurement, those faults that have the valid node (T, -) on the causal path to the measurement are eliminated. For a low inlet flow rate, faults with (T, +) on the causal path are removed. If this additional knowledge was applied as a global constraint, one fault candidate (Pressure sensor failed low) would be eliminated from Table 5-2, and fifteen fault candidates would be eliminated from Table 5-3. The 21 faults in Table 5-3 are reduced to six:

1> Liquid leak from vaporizer VAPORIZER-1.
2> Leak in pipe PIPE-B.
3> Leak at outlet [P7] in vaporizer VAPORIZER-1.
4> Vapor leak from vaporizer VAPORIZER-1.
5> Sensor P-SENSOR failed low.
6> Low pressure downstream of pipe PIPE-B.

Additional measurements can be added to discriminate between the possible origins. For example, consider the addition of a temperature sensor on the inlet of the heating fluid to the vaporizer after the control valve CV-2. The valid temperature measurement (T, -) would reduce the 16 faults in Table 5-2 to only two faults: insulation removed on control valve CV-2 and low temperature fluid entering control valve CV-2. The additional measurements reduce the number of fault candidates because they add knowledge about the current state of the physical system. Additional measurements tend to bound the valid tree sooner during candidate generation and eliminate more candidates through simulation and heuristic rules during candidate testing.

Quantitative information about the process can also be coded in a plant-specific rule base and used to eliminate primary deviations. For example, in Table 5-2, if it was known that the loss of insulation could not cause a sizable heat loss, then the four faults (nos. 7, 9, 10, and 11) concerning loss of insulation could be eliminated from the list.

Excellent resolution was obtained in Tables 5-8 and 5-9 because the majority of primary deviations were eliminated during candidate testing. Because F_1 and F_2 sensors were normal, all faults that caused pressure and flow disturbances were eliminated.

5.2.5 Comparisons Between the Example Processes

Five factors affect the number of fault candidates identified: (1) the number of possible faults (size), (2) the number of digraph arcs (degree of causal interaction), (3) digraph ambiguity, (4) the number of measurements, and (5) the position of measurements within the digraph.

Number of Possible Failures

The number of possible failures was 32 for the tank with the level control system (28 process failures and four disturbances through the system boundary), 56 for the vaporizer (44 process failures and 12 disturbances through the system boundary), and 110 for the CSTR (96 process failures and 14 through the system boundary). Given two primary deviations per digraph node, the ratio of failures to primary deviations is 0.76, 0.62, and 0.45, respectively, for the three examples.

Digraph Size and Degree of Interconnection

The sizes of the causal digraphs examined were 21 nodes and 29 arcs, 45 nodes and 65 arcs, and 122 nodes and 173 arcs, respectively, for the three examples. The ratio of arcs to nodes was 1.38, 1.44, and 1.42, respectively. The degree of interconnection is similar because the overall digraphs were constructed from many of the same general component models.

Number of Measurements

The number of measurements used was two for the tank example, three for the vaporizer example, and nine for the CSTR example. The ratio of the number of digraph nodes to the number of measurements was 10.5, 15, and 13.6, respectively, for the three examples.

Diagnostic resolution improves with a decreased number of possible faults candidates, a decreased number of causal arcs, decreased digraph ambiguity, an increased number of measurements, and improved strategic positioning of sensors. Because the physical process is given (e.g., the physical mechanisms represented by the digraph are fixed), the first three factors usually cannot be varied. Additional information about the system, such as qualitative constraints and additional measurements, can improve resolution. Optimal sensor placement is investigated in Chapter 6.

Chapter 6

PLANT IMPLEMENTATION

The focus of this chapter is on putting the causal models and diagnostic strategy into practice, i.e., connecting the diagnostic system to the physical process. The most important implementation issues are interpreting the process measurements and satisfying the assumptions on which candidate generation is based. Five assumptions, which were presented in Section 4.3.3, are necessary for the causal digraph search. Several examples presented in this chapter show how some of these assumptions may be violated. Modifications to the diagnostic search strategy are suggested that employ additional knowledge to handle these cases. The added information includes the history of each node's qualitative value assignments, and the quantitative values of process measurements and causal arc gains. The placement of sensors for maximum fault resolution is also discussed and the knowledge required for diagnosis is summarized.

6.1 Setting the Normal Operating Range

A valid node represents a process variable or parameter that has deviated outside its range of normal operation. If any measured variable crosses its normal range endpoints, then a fault has occurred.

Assigning a qualitative value to a process variable depends on the normal range chosen for its reference. If the normal range is too narrow, small disturbances and transients will cause the node to be valid, and activate the diagnostic system. If the range is too wide, then a deviated process variable caused by a fault may not be detected. Because the measured value of the variable may not cross the alarm threshold, the node's qualitative value remains '0'. Deviated process measurements that are misclassified as normal because of poorly set normal operating bands also adversely affect candidate generation because the search space is bounded by these normal measurements. Therefore, the expected values and

bounds must be set correctly to filter out normal process disturbances while being sensitive enough to detect abnormal symptoms.

Alarm thresholds are set from experience and should be statistically determined from historical process data. The range endpoints are chosen so that if the process variable deviates outside the normal range, then a fault has occurred. The endpoints are determined by analyzing prior measurement deviations when faults were known to have been present.

Note that for a measurement to yield information, its normal operating range must be specified. Without a reference, the measurement is not useful because its qualitative value cannot be determined. As mentioned in Section 3.1.1.2, the range can be stationary or be moving with time.

6.2 Problems With Discrete States

The mapping of continuous process variables and parameters into a discrete set of qualitative states is an important issue. As Long [1983] points out, the qualitative classification of states simplifies the reasoning problem, but places more burden on the interpretation of measurements.

The use of qualitative states has two principal drawbacks. First, all quantitative values that are mapped to a specific qualitative state have the same qualitative value. If two temperatures, $T_1 = 200°C$ and $T_2 = 350°C$, lie above a high temperature reference T_0^+, then they both are assigned the qualitative value '+'. This is a limitation because, given only that T_1 = '+' and T_2 = '+', the relationship between the two temperatures (i.e., $T_1 < T_2$, $T_1 = T_2$, or $T_1 > T_2$) and their relationships with other temperatures in the same qualitative state cannot be determined. Qualitative values only specify that both temperatures are greater than the reference T_0^+. Assigning qualitative values loses information because both temperatures are mapped to the same qualitative state.

The second drawback of qualitative states is that the assignment of a qualitative value is sensitive to small changes in a variable's numerical value when the numerical value is near an endpoint of the qualitative range. A differential change in the continuous value can result in a discrete change in the qualitative value. For example, if the high temperature reference was $T_0^+ = 80°C$, then the value 79.999°C is considered

normal, and 80.001°C is considered high. When a measurement crosses an endpoint of the normal range, the diagnosis can change abruptly, due to the discrete logical decision on whether to bound the search space or to continue to search along the causal arc. Andow [1981] concludes that this problem does not become more tractable if additional qualitative states are added.

6.3 Review of the Assumptions for Candidate Generation

The conclusions generated by the diagnostic system are valid only when the assumptions outlined in Section 4.3.3 are satisfied. These assumptions guarantee that, for a single failure, the causal search from each valid measurement will include the actual fault origin in its set of primary deviations. But the assumptions are not always satisfied in practice.

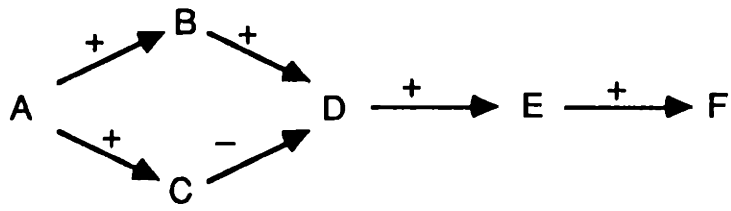
In this section, the assumptions concerning a single change of state and failure propagation along a causal path are investigated. Several examples show that these assumptions can be violated. When the assumptions are not satisfied, the diagnostic strategy presented in Chapter 4 is insufficient to diagnose the failure. Additional knowledge must be incorporated into the diagnostic strategy.

6.3.1 Changing Qualitative Values

In Assumption 4, process variables were restricted to a single direction of deviation, and therefore, a single qualitative value assignment. But variables that return to normal and deviations that change direction (undergo inverse response) do occur. Two examples illustrate how this assumption is violated.

- Parallel paths of opposite net sign between digraph nodes when the dominant path has a larger time delay.

In Figure 6-1, parallel paths of opposite net sign exist between nodes A and D. The path through node C is dominant and has a larger time delay. The qualitative dynamic response for a positive deviation at node A is



A	B	C	D	E	F
[+]	0	0	0	0	0
+	[+]	0	0	0	0
+	+	0	[+]	0	0
+	+	0	+	[+]	0
+	+	[+]	+	+	[+]
+	+	+	0	+	+
+	+	+	[-]	0	+

time ↓

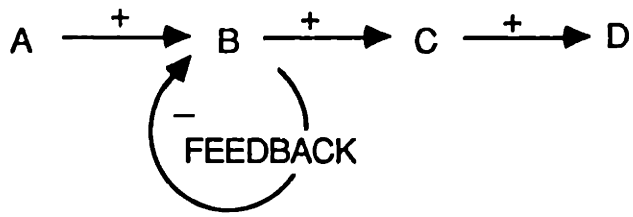
Figure 6-1
Inverse Response From Parallel Paths of Opposite Net Sign

shown in the table. The valid trees constructed from node F for the last two patterns in the table are bounded at nodes D and E, respectively, because these nodes have normal qualitative values. Because the valid trees are bounded at these nodes, node A, the failure origin, is not identified as a primary deviation.

- A control system with a slow compensating response.

Figure 6-2 illustrates a negative feedback control loop. The compensating response of the control system is slower than the disturbance propagation,

so that deviations causally downstream from the loop occur before the controlled variable (node B) is returned to normal. A valid tree constructed from node D for the last pattern in the table is bounded by the normal measurement at node C. Again, because the valid tree is bounded, the actual origin at node A is not identified.



	A	B	C	D
	[+]	0	0	0
	+	[+]	0	0
time	+	+	[+]	0
	+	0	+	[+]
	+	0	0	+

Figure 6-2
Inverse Response From Slow Control Systems

In each example, the intersection of the sets of primary deviations is empty for the abnormal measurement patterns analyzed. The diagnosis "multiple faults" is generated.

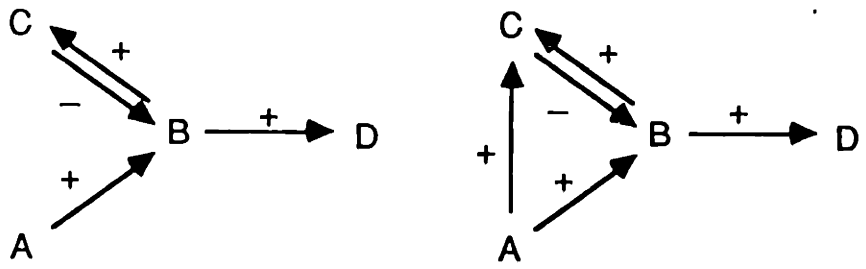
6.3.2. History of a Node's Qualitative Value Assignments

The two examples can be diagnosed correctly if a history of the qualitative values assigned to each digraph node is maintained. The boxed values in Figs. 6-1 and 6-2 represent the historical values saved and used during fault diagnosis. The procedure for constructing the valid tree during candidate generation is modified to include historical values: a node causally upstream from the current node is added to the valid tree if its qualitative value is or ever was of the the correct sign to make the causal arc consistent. For example, consider building a valid tree from node F for the last pattern in the table in Fig. 6-1. Without historical values, the valid tree from F is bounded at node E. When historical information is used, node D maintains both the values '+' and '-', and node E maintains the value '+', even though the current values of the nodes are '-' and '0', respectively. The valid tree from node F now includes the valid nodes (E, +), (D, +), (B, +), and (A, +).

When inverse response occurs in a digraph without separate parallel paths, the causal digraph can only predict the initial direction of deviation. The digraph cannot explain a measurement pattern when the long-term qualitative state of a node is opposite from its initial state. Consider the digraph in Figure 6-3a, in which node B exhibits inverse response. For a positive deviation at node A, the final steady-state set of valid nodes is (A, +), (B, -), (C, +), and (D, -). This set cannot be explained by the digraph. Oyeleye and Kramer [1987] suggest a modification to the digraph to handle this type of inverse response. The modification, shown in Fig. 6-3b, is the addition of a causal arc from A to C. This arc, which represents the positive path ABC, makes explicit two parallel paths of opposite net sign between nodes A and B. With this modification and the use of historical values, all the measurement patterns in the table can be explained.

Note that the qualitative values in the historical file need to be discarded after fault correction returns the plant to normal operation.

In summary, the use of historical values and the modification for inverse response can eliminate the restriction of multiple changes of qualitative values imposed by Assumption 4 (Section 4.3.3). Historical



6-3a 6-3b

	A	B	C	D
	[+]	0	0	0
	+	[+]	0	0
time	+	+	[+]	[+]
	+	0	+	+
	+	[-]	+	0
	+	-	+	[-]

Figure 6-3
Digraph Correction To Handle Inverse Response

values of variables that return to normal or change qualitative sign allow the causally downstream deviations to be explained by fault propagation through those variables.

6.3.3 Fault Propagation Along a Causal Path

Assumption 3 stated that if a disturbance is propagating along a causal path and two process variables in the path are measured, the causally upstream measurement will alarm before the downstream measurement. The following example illustrates that this assumption may be violated for faults with small disturbance magnitudes.

- Faults with small disturbance magnitudes.

In Figure 6-4, the measurement deviations for the process variables A, B, and C are presented. The points represent the deviation of the sensor

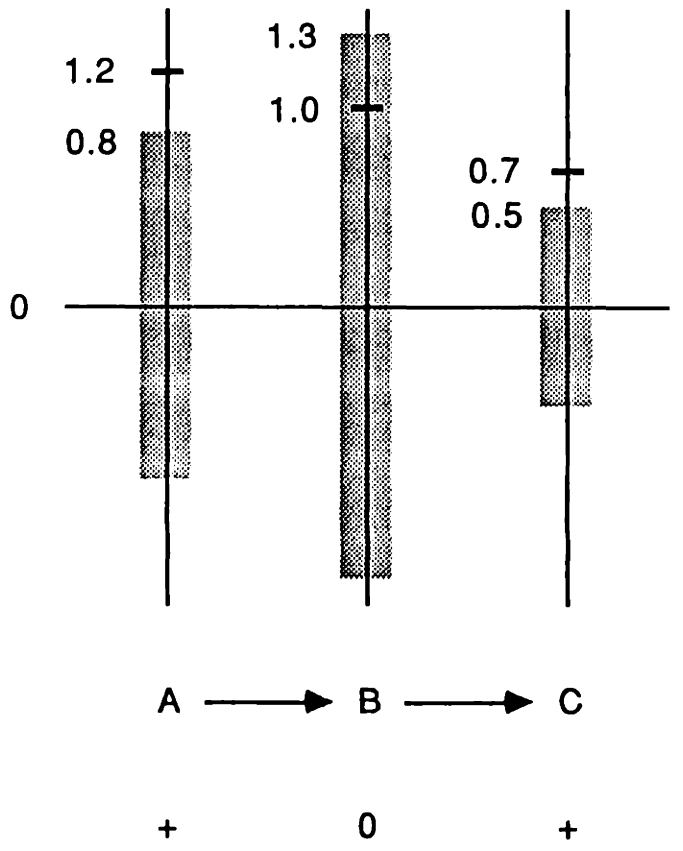


Figure 6-4
Fault Propagation for Small Disturbances

values away from their expected reference, and the bands around zero for each measurement represent the ranges of normal operation. A fault has occurred at node A and the disturbance has propagated to nodes B and C. Since the deviations of A and C fall above their normal ranges, they are assigned the qualitative value '+'. Node B is assigned the value '0' because its measurement deviation (1.0) is below the upper normal range threshold (1.3). The magnitude of the disturbance from node A is insufficient to cause the value of B to deviate outside its normal range, although the fault has propagated through B to cause the observed deviation at node C.

During candidate generation, the valid tree constructed from node C is bounded at B because the value of B is normal. Because node A is also valid, an empty set is obtained during set intersection. The criterion for bounding the search space during the construction of the valid tree is based on the assumption that the causally upstream alarm will be valid if the fault lies causally above the alarm. Misdiagnosis may occur when the effect of the fault is small, so that the measurement deviation due to the fault is on the same order of magnitude as the normal process fluctuations. When the effect is small, measured nodes along the propagation path may not be valid.

6.3.4 Incorporating the Quantitative Values of Measurements and Arc Gains

Two approaches for addressing the problem of diagnosing faults with small disturbance magnitudes are investigated. The methods presented are (1) reevaluating the qualitative value assignments of measurements, and (2) bounding the causal search using quantitative values of the maximum arc gains.

6.3.4.1 Reevaluating the Qualitative Value Assignments

When a fault's disturbance is on the same order of magnitude as the normal process fluctuations, process variables that should be valid may lie

within the normal range. When one or more of these measured variables are normal, no single fault can explain the abnormal measurement pattern.

Reevaluating the qualitative value assignments of process measurements is one approach for diagnosing faults with small disturbance magnitudes. The qualitative values of the measurements are first analyzed, as usual, by the diagnostic procedure. Then, the quality of the solution is evaluated to determine if the solution is poor (e.g., several independent failures). One possible rationale for the poor solution is that a fault with a small disturbance magnitude has occurred. Normal sensors are reassigned '+' and '-' values and the diagnostic system is rerun with the new valid nodes to see if the solution can be improved.

One method for determining which sensor values to change is to calculate the deviation index for normal measurements. The deviation index (DI) is a normalized measure of a process variable's deviation away from its steady-state reference x_0. For a measured variable x above its reference, the deviation index is given by

$$DI = \frac{x - x_0}{x_0^+ - x_0},$$

where x_0^+ is the upper range endpoint. For deviations below x_0, the upper range endpoint is replaced by the lower range endpoint x_0^-. For normal operation, the values of the DI range from -1.0 to 1.0. For the example in Fig. 6-4, the DI for node B is 1.0/1.3 = 0.77. If this value is judged sufficiently close to 1.0, the qualitative value is changed to '+' and the diagnosis is repeated. In the example, when B is '+', a consistent causal path exists from node A to node C; a single failure at node A can explain the observed measurement deviations.

6.3.4.2 Bounding the Causal Search Using Arc Gains

A second approach for addressing faults with small disturbance magnitudes incorporates the quantitative values of the maximum arc gains. In the procedure for candidate generation developed in Chapter 4, the valid tree is bounded at measured nodes with normal qualitative values. If the

quantitative values of the maximum arc gains are known, the gains and the numerical values of the measurements can be used for terminating the causal search.

To illustrate this approach, consider the causal arc $X \rightarrow Y$, where both X and Y are measured. Let x and y be the numerical values of the measurement deviation away from the normal references x_0 and y_0, respectively, and G_{XY} be the maximum gain between the two nodes. G_{XY} is defined as the maximum value of the process transfer function between the parameters X and Y. For a step change at X and overdamped response, G_{XY} is the steady-state gain. For underdamped response, G_{XY} is evaluated at the maximum overshoot.

The procedure for candidate generation is modified to incorporate the numerical values of process gains. During the construction of the valid tree, if the current valid node is Y, node X is added to the valid tree only if the deviation at Y can be explained by the observed deviation at node X. Expressed mathematically, the initial node of a causal arc is added to the valid tree if

$$\text{sgn}(x * G_{XY}) = \text{sgn}(y), \text{ and} \tag{5a}$$

$$|x * G_{XY}| \geq |y|. \tag{5b}$$

If the deviation at node Y is greater than can be explained by the deviation at node X, the causal search is bounded along this arc and node X is not added to the valid tree. Returning to Fig. 6-4, let $G_{AB} = 1.0$ and $G_{BC} = 0.8$. Node B is added to the valid tree from node C because

$$\text{sgn}(1.0 * 0.8) = \text{sgn}(0.7)$$

$$|1.0 * 0.8| \geq |0.7|.$$

Similarly, node A is added because

$$\text{sgn}(1.2 * 1.0) = \text{sgn}(1.0)$$

$$|1.2 * 1.0| \geq |1.0|.$$

Therefore, a consistent causal path exists from node A to node C. When unmeasured digraph nodes exist between two measurements, the maximum gain used is the overall maximum path gain.

The device topography may be required for determining the quantitative values of the maximum gains. For example, given two parallel paths, the disturbance propagation along each path may be insufficient to explain the observation. But the overall gain, representing the sum of the effects from the node where the individual paths branched, may be able to explain the observed deviation. A second example is a control system, where frequency response is necessary to determine the overall maximum gain for the loop.

Assumption 3 (Section 4.3.3) required that for the propagation of a fault along a path in which two process variables are measured, the causally upstream measurement will become valid before the downstream measurement. If all the arc gains are known, this assumption is not necessary, because the bounding rule during candidate generation can be based solely on satisfying Eqs. 5a and 5b. Note that normal ranges are still necessary for initiating the diagnostic procedure (i.e., for detecting that a failure has occurred).

6.4 Sensor Placement for Optimal Resolution

Process control objectives dictate a set of process variables to be measured during the design of a process plant. In addition to the variables chosen as inputs to control systems, other measurements are selected to provide further information to the process operator. The causal digraph can be used to guide the placement of additional sensors. Thus, the digraph, in addition to real-time fault diagnosis, can also serve as a tool to improve plant diagnosability.

The following analysis assumes that all faults are equally important and that every node is a primary deviation for at least one fault.

The objectives for sensor placement for fault diagnosis are first, to detect all possible faults, and second, to discriminate between a set of fault candidates. For the first objective, the sensor should be positioned at the downstream ends of causal paths. The disturbance from any fault in

the path will propagate along the causal path and be detected at the causally downstream sensor. After the minimum number of sensors are positioned for fault detection, the second objective is to discriminate between fault candidates. A divide-and-conquer strategy is employed to position each additional sensor, to minimize the maximum number of indistinguishable faults.

Two examples illustrate the placement of sensors. In Figure 6-5, the first sensor is positioned at node F. This one sensor is able to detect all possible faults. Note that if the sensor was placed at any other node,

Figure 6-5
Sensor Placement: Example 1

faults occurring at any node causally downstream from the sensor are not detectable. The position of the second sensor is determined by minimizing the maximum number of indistinguishable faults. The three tables below show the primary deviations identified for individual valid sensors, when the placement of the second sensor is at nodes B, C, and D, respectively.

Sensors F	B	Primary Deviations
+	0	C D E F
0	+	A B

Sensors F	C	Primary Deviations
+	0	D E F
0	+	A B C

Sensors F	D	Primary Deviations
+	0	E F
0	+	A B C D

Node C is chosen as the location of the second sensor because the maximum number of indistinguishable faults is minimized at three. Sensor placement at nodes B or D would result in four indistinguishable faults.

In the example above, a fault was assumed to be associated with every node. Sensors should be positioned with respect to faults, rather than

digraph nodes. For example, if there were no faults associated with nodes D and F, the first sensor would be placed at node E, to eliminate the time delay of fault propagation along the arc from E to F. The second sensor would be placed at node B to divide the remaining faults into two sets of two faults: nodes A and B, and nodes C and E.

In Figure 6-6, again assuming faults at every node and faults of equal importance, the first and second sensors are assigned to nodes C and H.

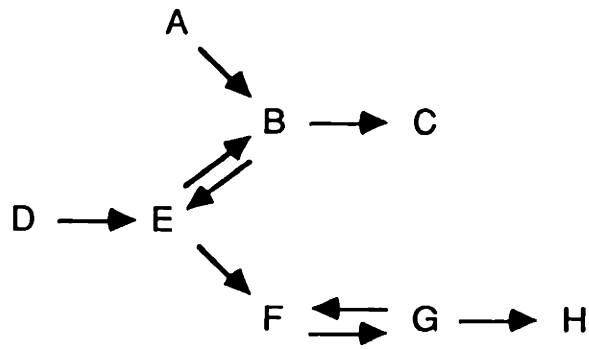


Figure 6-6
Sensor Placement: Example 2

Seven primary deviations are identified for an abnormal measurement at H.

Sensors C H	Primary Deviations
+ 0	A B C D E
0 +	A B D E F G H

The next sensor should add resolution between the seven primary deviations. Sensor placement at nodes B, E, and F is presented.

Sensors C H B	Primary Deviations
+ 0 0	C
0 + 0	D E F G H
0 0 +	A B D E

Sensors C H E	Primary Deviations
+ 0 0	A B C
0 + 0	F G H
0 0 +	A B D E

Sensors C H F	Primary Deviations
+ 0 0	A B C D E
0 + 0	G H
0 0 +	A B D E F

The next measurement should be located at node E.

When the consequences of one fault are more severe, more measurements should be positioned at or around the primary deviation to detect the presence of the fault.

6.5 Summary of Knowledge Requirements for Fault Diagnosis

The knowledge requirements for model-based fault diagnosis are summarized.

Constructing the Overall Process Causal Digraph

1. General component causal models
2. Context-specific design information, for specifying causal arcs and arc attributes
3. Plant topography, for specifying the interconnections between process units

Specifying Qualitative Values

1. Values of process measurements
2. Context-specific normal references (expected value and normal range of operation) for every measurement

Candidate Generation

1. Causal digraph
2. Qualitative values of process measurements
3. Information about control systems (measurement, manipulated variable, and net sign of control loop)
4. Previous values of process measurements, for handling inverse response and variables that return to normal
5. Numeric arc gains and the numeric values of measurement deviations, for bounding the fault space for faults with small disturbances

Candidate Testing

1. Causal digraph
2. Qualitative values of process measurements
3. Global constraints (dominant causal paths)
4. Heuristic rules for eliminating primary deviations
5. Context-specific design information for specifying which global constraints and heuristics to apply

Fault Generation

1. General rules relating equipment failure modes to primary deviations
2. Context-specific design information, for building the table mapping faults to primary deviations and for specifying specific faults

Chapter 7

RESEARCH IN QUALITATIVE MODELING AND DIAGNOSIS

Key research in the following three areas is summarized: (1) causal reasoning about physical systems, (2) diagnosis using the directed graph process representation, and (3) a rule-based approach to fault diagnosis. The reader interested in a broader review of the fault detection and diagnosis literature should consult Lees [1983], O'Shima [1983], Isermann [1984], Pau [1981] and Himmelblau [1978]. Rouse [1983] reviews models of human problem solving.

7.1 Causal Reasoning About Physical Systems

de Kleer and Brown [1984] [1986] base their notion of causality on the way a disturbance is propagated through a network of constraint equations. Constraint equations, called confluences, are based on the definition of a component, and are written in terms of the qualitative derivatives of variables. Because the causal relationships are identified by propagating the effects of a disturbance through the network of constraints, the causal interactions generated depend on the sequence in which the confluences are solved.

Iwasaki and Simon [1986a] [1986b] establish a causal ordering by analyzing the structure of the system of equations that models the physical system. Causal ordering is determined by finding subsets of variables whose values can be computed independently of the remaining variables. The values of the variables of the subset are used to reduce the structure to a smaller set of equations containing only the remaining variables. Causal paths are dependent upon which variables are chosen as exogenous.

The result of both de Kleer & Brown and Iwasaki & Simon is that the causal relationships generated are a function of how a set of equations that model the system is solved. Causality becomes identical to the progression of substitutions into the system of equations.

Forbus [1984] attempts to reason about physical changes by characterizing the process through which change occurs. In qualitative process theory, processes are characterized by (1) the individuals to which the process applies, (2) a set of preconditions about the individuals and their relationships, (3) a set of quantity conditions (e.g., the relative values of two temperatures to determine if heat transfer occurs), (4) a set of relations (functional relationships), and (5) a set of influences, which specify what can cause a quantity to change. Causal changes are direct changes caused by processes, or the propagation of the effects of direct changes through functional dependencies.

Bobrow [1985] compiled eight papers on qualitative reasoning that appeared in the Artificial Intelligence Journal.

7.2 Diagnosis Using the Directed Graph Process Representation

A model-based diagnostic strategy has been presented by Iri and co-workers (Iri et al. [1979] [1980]) that uses a graph representation to model the causal interactions. Their approach uses an iterative procedure to assign '+', '-', and '0' qualitative values to unmeasured and controlled digraph nodes. For each assignment of a qualitative value, the consistent branches of the graph are identified. The subgraph composed of consistent branches is then examined to determine if the subgraph is rooted. Because a single fault is assumed, causal pathways must connect the fault origin to every measurement deviation. If the subgraph becomes disconnected for a particular set of qualitative values, then there cannot be a single fault. The authors used the algorithm on a digraph of 21 nodes and 62 branches, of which six nodes were observed and three nodes were controlled. A purely iterative method requires 3^n assignments, where n is the number of unmeasured and controlled nodes in the causal digraph. Even with a heuristic to reduce the number of subgraphs evaluated, they reported that the algorithm generated about 20,000 subgraphs that were examined for connectivity (Iri et al. [1980]). They note that even with heuristics to reduce the number of graphs examined, the problem grows exponentially with the number of nodes.

Shiozaki et al. [1985] proposed modifications to the iterative procedure of Iri et al. to improve computational speed. Improvements to the method are a systematic choice for selecting unmeasured nodes for assigning qualitative values, criteria for terminating the assignment of qualitative values, and the systematic revision of the assigned values on unmeasured nodes.

Instead of changing the signs on nodes with time, Umeda et al. [1980] introduced the staged causal digraph. The continuous system of equations is discretized and repeated at a frequency of the smallest process lag. The iterative method of Iri et al. is used to identify consistent branches. At each stage, a single failure source is not identified. Rather, all the consistent branches from the previous stage to the current stage are determined.

The advantage of this method is that it can handle multiple state changes (e.g., inverse response and measurements returning to zero). The major drawback is that if the cycle time for updating process measurements is not short enough, the sequence of state changes cannot be identified. Implementation difficulties include a large amount of data and a heavy computation load.

Tsuge et al. [1985] introduced process delay information to obtain a multi-staged signed directed graph with delay. These authors argue that information on delay time is able to further reduce the set of candidates generated from the digraph.

Kokawa (Kokawa and Shingai [1982], Kokawa et al. [1983]) proposed a fault location method in which digraph nodes represent process units rather than system variables. The purpose of their method is to aid operators in blocking off failure propagation paths and for preventative maintenance design. They assume that failure propagation occurs solely in the direction of bulk fluid flow. Thus, the plant topology represents the failure propagation network. Under this assumption, the method cannot accommodate flow and pressure disturbance propagation against bulk flow, recycle flows and control systems.

Andow and Lees [1975] [1978] used a network of process variables to construct a process alarm network. The purpose of the alarm network is to

analyze the sequence of alarm firings and explain downstream alarms in terms of fault propagation.

Finch and Kramer [1986] use a directed graph to represent the interactions between process subsystems. They decompose the physical system by function, defined as groups of systems components that perform a single system objective, and assign the functions to the digraph nodes.

Kramer and Palowitch [1986] present a method to translate the paths of fault propagation in the causal digraph into a set of logic statements, which can be used within an expert system environment.

7.3 Rule-Based Approach to Process Diagnosis

Several researchers are developing rule-based expert systems for real-time fault diagnosis of process plants. The FALCON (Fault Analysis Consultant) expert system is a joint research project between du Pont de Nemours Co., Foxboro Co., and the University of Delaware for exploring the application of expert systems technology for diagnosing chemical plant malfunctions (Chester, et al. [1984], Lamb, et al. [1985], Shirley [1985], Rowan [1986]). Both qualitative and quantitative information are incorporated in the system in a plant-specific production rule format.

Shum et al. [1986] present an expert system architecture based on a hierarchy of malfunction hypotheses. The hierarchical structure allows the fault classification task to proceed using a top-down strategy, from generality to detail. Kumamoto et al. [1984] use the classification methodology to diagnose an engine cooling system.

A prototype expert system presented by Andow [1985] is based on the search of fault trees and cause-consequence diagrams. In [1986], Andow presents context-dependent production rules for the diagnosis of a utility cooling water system. The rules relate patterns of observed symptoms to specific faults.

Rule-based systems are receiving significant attention in the nuclear power industry. Cain et al. [1985] reviews artificial intelligence research at the Electric Power Research Institute (EPRI) for nuclear power applications. Nelson [1984] describes six diagnostic expert systems for

nuclear reactor operations. Knowledge representation in the systems described include networks, logic trees, and rules.

LISP Machines Inc. has introduced PICON, an expert system development environment intended for real-time process monitoring and control (Moore et al. [1985]).

Chapter 8

CONCLUSIONS

This thesis demonstrates the feasibility of using causal models for fault diagnosis. In the first part of the thesis, the causal digraph was developed to characterize the cause-and-effect interactions between process variables and parameters. Causal digraphs were derived for sets of design equations, guidelines were presented for creating causal digraphs for fault diagnosis, and context-independent causal models for standard system components were developed. In the second part of the thesis, the diagnostic strategy presented incorporates the causal models with other knowledge representations and multiple problem-solving approaches. Graph search, simulation, qualitative constraints, and heuristics were used within a hypothesis generation and test framework. The strategy, which was implemented in a computer program, achieved excellent performance on the example processes examined. The prototype demonstrates that a computer can interpret the sensor values and generate a list of fault candidates to assist the process operator during plant upsets.

The diagnostic strategy developed in this thesis satisfies the criteria for evaluating diagnostic systems presented in Section 2.4.

- The system produces a list of possible faults that includes the actual fault origin or origins. The causal search during candidate generation is exhaustive, so all possible primary deviations are identified.

- The system minimizes the number of spurious faults that are included in the set of possible candidates by incorporating multiple types of knowledge.

- The system is able to diagnose a wide range of failures, as demonstrated by the examples in Chapter 5.

- The system generates diagnoses faster than the dynamics of real-time chemical and nuclear processes.

- The diagnostic system is portable to a variety of process environments. The knowledge representation formats and diagnostic strategy are general, so that system implementation is not done from scratch at each new plant site. The modular component architecture adds flexibility and reduces the costs of development and installation.

- The system is easily modified and updated. Changes in the process due to piping and equipment changes does not require extensive reprogramming or data collection. An updated causal digraph is generated by changing the context-specific inputs into general component models.

- The system is able to use the installed instrumentation, without specific requirements for the number of sensors and sensor locations.

This research provides a foundation for qualitative, model-based fault diagnosis. It forms one part of a comprehensive diagnostic system for process operations. The other major components are fault detection, experienced-based fault diagnosis, failure correction, and the design of an operator interface.

NOTATION

A = cross-sectional area
C = concentration
c = flow coefficient
C_p = specific heat at constant pressure
E = electromotive force
E_a = activation energy
F = volumetric flow rate
g = acceleration due to gravity
H = enthalpy
ΔH_r = heat of reaction
i = current
k = first order reaction rate constant
L = fluid level
m = mass
N = moles
P = pressure
Q = heat flow rate
R = resistance to volumetric flow
resistance to heat flow
resistance to current flow
$\mathbf{R}$ = Universal gas constant
r = reaction rate
T = temperature
t = time
U = overall heat transfer coefficient
V = volume
v = valve stem position

λ = heat of vaporization
ρ = density
θ = reactor space time
ω = angular velocity
Δ = finite difference in quantity
f = function

Subscripts

A, B, C	species
b	bottom
error	controller error
ℓ	liquid
r	reference
R	reactor
sensor	sensor
sp	set point
v	vapor

LITERATURE CITED

Andow, P. K.
[1981] Fault Trees and Failure Analyses: Discrete State Representation Problems, Transactions of the Institute of Chemical Engineering 59, 125-128.
[1985] Fault Diagnosis Using Intelligent Knowledge Based Systems, PSE '85: The Use of Computers in Chemical Engineering (IChE Symposium Series No. 92), Cambridge, England, 145-156.
[1986] Improvement of Operator Reliability Using Expert Systems, Reliability Engineering 14, 309-319.

Andow, P. K. and Lees, F. P.
[1975] Process Computer Alarm Analysis: Outline of a Method Based on List Processing, Transactions of the Institute of Chemical Engineering 53, 195-208.
[1978] Real Time Analysis of Process Plant Alarms (paper presented at NATO Advanced Study Institute on "Synthesis and Analysis Methods for Safety and Reliability Studies," Urbino, Italy).

Bobrow, D. G. (ed.)
[1985] Qualitative Reasoning About Physical Systems, The MIT Press, Cambridge, Massachusetts.

Cain, D. G., Sun, W., and Faught, W. S.
[1985] Artificial Intelligence Research at the Electric Power Research Institute for Nuclear Power Application, Proceedings of the 9th Int. Conf. on Modern Power Stations (Liege, Belgium) 79.1-79.7.

Centrifugal Pumps (Newtonian Liquids), AIChE Equipment Testing Procedure, 1984.

Chester, D. L., Lamb, D. E., and Dhurjati, P.
[1984] Rule-Based Computer Alarm Analysis in Chemical Process Plants, Proceedings of the 7th Annual Micro-Delcon, 22-29.

Davis, R., Shrobe, H., Hamscher, W., Wieckert, K., Shirley, M., and Polit, S.
[1982] Diagnosis Based on Description of Structure and Function, Proceedings of AAAI-82 (Pittsburgh, PA), 137-142.

de Kleer, J. and Brown, J. S.
[1984] A Qualitative Physics Based On Confluences, Artificial Intelligence 24, 7-83.
[1986] Theories of Causal Ordering, Artificial Intelligence 29, 33-61.

Finch, F. E. and Kramer, M. A.
[1986] Narrowing Diagnostic Focus Using Functional Decomposition, AIChE Journal, submitted.

Forbus, K. D.
[1984] Qualitative Process Theory, Artificial Intelligence 24, 85-168.

Genesereth, M. R.
[1984] The Use of Design Descriptions in Automated Diagnosis, Artificial Intelligence 24, 411-436.

Himmelblau, D. M.
[1978] Fault Detection and Diagnosis in Chemical and Petrochemical Processes, Chemical Engineering Monograph 8, Elsevier, Amsterdam.

Iri, M., Aoki, K., O'Shima, E., and Matsuyama, H.
[1979] An Algorithm for Diagnosis of System Failures in the Chemical Process, Computers & Chemical Engineering 3, 489-493.
[1980] A Graphical Approach to the Problem of Locating the Origin of the System Failure, Journal of the Operations Research Society of Japan 23, 295-311.

Isermann, R.
[1984] Process Fault Detection Based on Modeling and Estimation—A Survey, Automatica 20, 387-404.

Iwasaki, Y. and Simon, H. A.
[1986a] Causality in Device Behavior, Artificial Intelligence 29, 3-32.
[1986b] Theories of Causal Ordering: Reply to de Kleer and Brown, Artificial Intelligence 29, 63-72.

Kokawa, M. and Shingai, S.
[1982] Failure Propagating Simulation and Nonfailure Paths Search in Network Systems, Automatica 18, 335-341.

Kokawa, M., Miyazaki, S., and Shingai, S.
[1983] Fault Location Using Digraph and Inverse Direction Search with Application, Automatica 19, 729-735.

Kramer, M. A. and Palowitch, B. L.
[1987] A Rule-Based Approach to Fault Diagnosis Using the Signed Directed Graph, AIChE Journal, in press.

Kuipers, B. J. and Kassirer, J. P.
[1984] Causal Reasoning in Medicine: Analysis of a Protocol, Cognitive Science 8, 363-385.

Kumamoto, H., Ikenchi, K., Inoue, K., and Henley, E.
[1984] Application of Expert System Techniques to Fault Diagnosis, Chemical Engineering Journal 29, 1-9.

Lamb, D. E., Dhurjati, P., Chester, D. L., and Hale, J. C.
[1985] An Academic/Industry Project to Develop and Expert System for Chemical Process Fault Detection, Proceedings of AIChE (Chicago, IL).

Lees, F. P.
[1983] Process Computer Alarm and Disturbance Analysis: Review of the State of the Art, Computers & Chemical Engineering 7, 669-694.

Long, W. J.
[1983] Reasoning About State From Causation and Time in a Medical Domain, Proceedings of AAAI-83 (Austin, TX), 251-254.

Moore, R. L., Hawkinson, L. B., Levin, M. E., and Knickerbocker C. G.
[1985] Expert Control, Proceedings of ACC-85 (Boston, MA), 885-888.

Nelson, W. R.
[1984] Response Trees and Expert Systems for Nuclear Reactor Operators, Report NUREG/CR-3631, Washington, DC.

Oyeleye, O. O. and Kramer, M. A.
[1987] Correction to the Single-staged Signed Directed Graph Representation of System Dynamics, Chemical Engineering Science, submitted.

O'Shima, E.
[1983] Computer Aided Plant Operation, Computers & Chemical Engineering 7, 311-329.

Patil, R. S., Szolovits, P., and Schwartz, W. B.
[1981] Causal Understanding of Patient Illness in Medical Diagnosis, Proceedings of IJCAI-7 (Vancouver, British Columbia), 893-899.

Pau, L. F.
[1981] Failure Diagnosis and Performance Monitoring, Marcel Dekker, New York.

Rasmussen, J.
[1978] Notes on Diagnostic Strategies in Process Plant Environment, Riso National Laboratory, Roskilde, Denmark, Report RISO-M-1983.
[1979] On the Structure of Knowledge—A Morphology of Mental Modes in a Man-machine Context, Riso National Laboratory, Roskilde, Denmark, Report RISO-M-2192.
[1981] Models of Mental Strategies in Process Plant Diagnosis, in J. Rasmussen and W. B. Rouse (eds.), Human Detection and Diagnosis of System Failures, Plenum Press, New York, 241-258.

Rasmussen, J. and Jensen, A.
[1974] Mental Procedures in Real Life Tasks: A Case Study of Electronic Troubleshooting, Ergonomics 17, 293.

Rouse, W. B.
[1978] A Model of Human Decision Making in a Fault Diagnosis Task, IEEE Transactions on Systems, Man, and Cybernetics SMC-8, 357-361.
[1981] Experimental Studies and Mathematical Models of Human Problem Solving Performance in Fault Diagnosis Tasks, in Rasmussen, J. and Rouse, W. B., eds., Human Detection and Diagnosis of System Failures, Proceedings of a NATO Symposium in Roskilde, Denmark on August 4-8, 1980, Plenum Press, New York, 199-216.
[1983] Models of Human Problem Solving: Detection, Diagnosis, and Compensation for System Failures, Automatica 19, 613-625.

Rowan, D. A.
[1986] Chemical Plant Fault Diagnosis Using Expert Systems Technology: A Case Study, IFAC Kyoto Workshop on Fault Detection and Safety in Chemical Plants (Kyoto, Japan), 81-87.

Shiozaki, J., Matsuyama, H., O'Shima, E., and Iri, M.
[1985] An Improved Algorithm for Diagnosis of System Failures in the Chemical Process, Computers & Chemical Engineering 9, 285-293.

Shirley, R. S.
[1985] Status Report: An Expert System to Aid Process Control, Proceedings of ISA 85 (Philadelphia, PA), 1463-1470.

Shum, S. K., Davis, J. F., Punch III, W. F., and Chandrasekaran, B.
[1986] An Expert System for Diagnosing Process Plant Malfunctions, IFAC Kyoto Workshop on Fault Detection and Safety in Chemical Plants (Kyoto, Japan), 116-120.

Tsuge, Y., Shiozaki, J., Matsuyama, H., and O'Shima, E.
[1985] Fault Diagnosis Algorithms Based on the Signed Directed Graph and its Modifications, PSE '85: The Use of Computers in Chemical Engineering (IChE Symposium Series No. 92), Cambridge, England, 133-144.

Umeda, T., Kuriyama, T., O'Shima, E., and Matsuyama, H.
[1980] A Graphical Approach to Cause and Effect Analysis of Chemical Processing Systems, Chemical Engineering Science 35, 2379-2388.

U.S. Atomic Energy Commission
[1974] Reactor Safety Study, An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, Appendix III, Failure Data, WASH-1400, Washington, D.C.

Williams, B.
[1984] Qualitative Analysis of MOS Circuits, Artificial Intelligence 24, 281-346.

Appendix A

Graph Theory Terminology

Graphs

A graph G is a set (V(G), E(G)), where V(G) is a non-empty set of elements called vertices (nodes, points) and E(G) is a set of unordered pairs of distinct elements of V(G) called edges (lines, curves). V(G) is called the vertex set and E(G) is called the edge set of G. If both V and E are finite sets, then G is called a finite graph.

Two vertices of G are said to be adjacent if there is an edge that joins them. The two adjacent vertices are said to be incident to the edge. The vertices incident with an edge are called its end points, and are said to be joined by the edge.

Two distinct edges of G are adjacent if they have at least one common vertex. The degree of a vertex is the number of edges incident to it (i.e. the number of edges which have that vertex as an end point).

A loop is an edge that joins a vertex to itself. A simple graph is a graph that contains no loops or multiple edges (edges with the same two end points).

A subgraph of a graph G is a graph whose vertices belong to V(G) and whose edges belong to E(G).

An edge sequence is a finite sequence of adjacent edges. An edge sequence in which all the edges are distinct is called a trail (chain). If the vertices are also distinct (except, possibly the initial and final vertices of the chain), then the trail is called a path. A path is closed if the initial vertex and final vertex of the path are identical; open otherwise. A closed path containing at least one edge is called a circuit.

A graph is connected if every pair of vertices is joined by at least one chain. A connected graph cannot be expressed as the union of two graphs.

A tree is a connected graph which has no circuits. Thus, a graph is a tree if and only if every pair of distinct vertices are joined by exactly one path. A tree with n vertices is a simple graph with precisely n-1 edges. A spanning tree is a subgraph of a connected graph G which is a tree and which includes all vertices of G.

Directed Graphs

A directed graph D, also called a digraph, is a graph where every edge is oriented (given a direction). An arc is directed edge.

The digraph D is a simple digraph if the arcs of D are all distinct and D contains no loops. The arcs of a simple digraph can be represented without ambiguity by ordered pairs of vertices, since at most one arc joins a given pair of vertices in a specified direction.

The outdegree od(v) of a vertex v is the number of arcs in D whose initial vertex is v; the indegree id(v) is the number of arcs whose terminal vertex is v.

An arc sequence is a finite sequence of adjacent, similarly-oriented arcs. An arc sequence in which all the arcs are distinct is called a ditrail (directed trail). If the vertices are also distinct (except, possibly the initial and final vertices of the ditrail), then the ditrail is called a dipath (directed path). A dipath is closed if the initial vertex and final vertex of the path are identical. A closed directed path is also called a cycle. A directed graph D is said to be cyclic if it contains at least one cycle; acyclic otherwise.

A digraph is strongly connected if, for every pair of vertices v and w, there exists a dipath from v to w as well as one from w to v (i.e. every two points are mutually reachable). A digraph is weakly connected if it cannot be expressed as the union of two disjoint digraphs.

A rooted directed tree is a directed graph which (1) is a tree in the undirected sense, and (2) has a vertex v such that there exists a directed path from v to every other vertex. The tree is said to be rooted at the vertex v.

Appendix B

DIEX Computer Code

The computer code for major sections of the DIEX system prototype is presented. The files, along with a short description of the functions contained in each file, are listed below.

defns.l
: Flavor definitions for arc, node, valid_node, and control_system; simple functions.

diex.l
: Top level function for entering DIEX. Prompts for inputs; calls lower-level functions.

generate_candidate_set.l
: Identifies primary deviations by constructing a valid tree for a given deviated measurement.

node_test.l
: Checks that the current node is not already in the path to the valid measurement, before it is added to the set of primary deviations during candidate generation.

simulate.l
: Simulation using qualitative time delays.

identify_faults.l
: Generates a list of faults for a set of primary deviations; formats and prints output.

generic_rulebase.l
: Rule bases relating faults to primary deviations.

vapor_boundary.l
: Defines the function roots_at_boundaries.

Before DIEX can be run,

1. the causal digraph must be created;

2. the function roots_at_boundaries must be defined;
   (This function, specific for a given digraph, defines the possible disturbances that could affect the process through the system boundaries.)

3. a control system object must be instantiated for every control system;

4. the variables controller_list, measurement_list, and boundary_list must be set.

For the tank example, the last three items are accomplished by the following Lisp code.

```
(defun roots_at_boundaries (node-sign)
  (setq fault_description nil)
  (setq boundary_node (member (car node-sign) boundary_list))
  (cond (boundary_node
         (setq node (car node-sign))
         (setq sign (cadr node-sign))
         (caseq node
           (F1 (cond ((eq sign '+) (setq fault_description
                 '|High flow rate F1 entering tank TANK-1.  |))
                     ((eq sign '-) (setq fault_description
                 '|Low flow rate F1 entering tank TANK-1.  |))))
           (P5 (cond ((eq sign '+) (setq fault_description
                 '|High pressure downstream of pipe PIPE-B.  |))
                     ((eq sign '-) (setq fault_description
                 '|Low pressure downstream of pipe PIPE-B.  |)))))
         )))
;
;
(setq level_control (make-instance 'control_system
  :measurement 'L-SENSOR
  :manip 'V1
  :setpoint 'L-SP
  :description '+))
;
;
(setq controller_list '(level_control))
(setq measurement_list '(L-SENSOR F-SENSOR))
(setq boundary_list '(F1 P5))
```

```
;
;
; file defns.l
;
; Flavor definitions for arc, node, control_system, and valid_node; simple
; functions.
;
;
(declare (lambda validate forward_nodes backward_nodes forward_valid_nodes
                 backward_valid_nodes intersection)
         (special current_element))
;
;
; Description of flavor: arc
;   initial_node     initial node of the arc
;   terminal_node    terminal node of the arc
;   sign             +/-
;   magnitude        magnitude of the interaction
;   time             fault propagation time
;
(defflavor arc (initial_node
                terminal_node
                sign
                magnitude
                time)
               ()
               :gettable-instance-variables
               :settable-instance-variables
               :inittable-instance-variables)
;
;
; Description of flavor: node
;   variable_type    generic process variable class
;   equip_name       name of equipment instance
;   equip_type       generic equipment class
;   arcs_to          a list of arcs terminating at the node
;   arcs_from        a list of arcs leaving the node
;   controlled       yes/nil
;   manip            yes/nil
;   measured         name of measurement/nil
;   setpoint         yes/nil
;   control_system   name of the control system/nil
;
(defflavor node (variable_type
                 equip_name
                 equip_type
                 arcs_to
                 arcs_from
                 controlled
                 manip
                 measured
                 setpoint
                 control_system)
                ()
                :gettable-instance-variables
                :settable-instance-variables
                :inittable-instance-variables)
```

```
;
;
; Description of flavor: control_system
;   measurement      name of measurement
;   manip            name of the manipulated variable
;   setpoint         name of the setpoint
;   description      +/-  (net direction of the manipulated variable, given
;                           a positive deviation in the measurement)
;
(defflavor control_system (measurement
                           manip
                           setpoint
                           description)
                          ()
                          :gettable-instance-variables
                          :settable-instance-variables
                          :inittable-instance-variables)
;
;
; Description of flavor: valid_node
;   node_name      print name of the valid node
;   sign           valid node sign
;   forward_nodes  list of adjacent nodes causally downstream from the valid
;                    node
;   path           the consistent path between the node and the valid
;                    measurement
;
(defflavor valid_node (node_name
                       sign
                       forward_nodes
                       path)
                      ()
                      :gettable-instance-variables
                      :settable-instance-variables
                      :inittable-instance-variables)
;
;
(defun validate (sign1 sign2)
  (cond ((and (eq sign1 '+) (eq sign2 '+)) '+)
        ((and (eq sign1 '+) (eq sign2 '-)) '-)
        ((and (eq sign1 '-) (eq sign2 '+)) '-)
        ((and (eq sign1 '-) (eq sign2 '-)) '+)))
;
;
(defun forward_nodes (current_node)
  (mapcar #'(lambda (arc) (symeval-in-instance (symeval arc) 'terminal_node))
    (symeval-in-instance (symeval current_node) 'arcs_from)))
;
;
(defun backward_nodes (current_node)
  (mapcar #'(lambda (arc) (symeval-in-instance (symeval arc) 'initial_node))
    (symeval-in-instance (symeval current_node) 'arcs_to)))
```

```
;
;
(defun forward_valid_nodes (node-sign)
  (mapcar #'(lambda (arc)
              (list (symeval-in-instance (symeval arc) 'terminal_node)
                    (validate (cadr node-sign)
                              (symeval-in-instance (symeval arc) 'sign))))
          (symeval-in-instance (symeval (car node-sign)) 'arcs_from)))
;
;
(defun backward_valid_nodes (node-sign)
  (mapcar #'(lambda (arc)
              (list (symeval-in-instance (symeval arc) 'initial_node)
                    (validate (cadr node-sign)
                              (symeval-in-instance (symeval arc) 'sign))))
          (symeval-in-instance (symeval (car node-sign)) 'arcs_to)))
;
;
; Function intersection finds the intersection of two lists.  If an element
; appears in either input list several times, it only appears in the output
; list once.
;
(defun intersection (list1 list2 temp)
  (cond ((null list1) (reverse temp))
        (t (setq current_element (car list1))
           (cond ((and (member current_element list2)
                       (not (member current_element temp)))
                  (setq temp (cons current_element temp))))
           (intersection (cdr list1) list2 temp))))
;
;
```

```
;
;
; file diex.l
;
; Top level routine for DIEX (Diagnostic Expert), a model-based diagnostic
; system based on causal models of chemical processes and process equipment.
;
; To run the system, the directed graph of causal interactions must exist.
; Flavor definitions can be found in the file defns.l.
;
;
; rns           root node set for the current measurement.
; current_rns   root node set generated by the intersection of the rns for
;                 every valid measurement.
; reduced_rns   root node set after rules and simulation.  This set is used
;                 to identify the fault candidates.
;
;
(setq valid_hash (make-hash-table :size 250 :test #'equal))
;
(include generate_candidate_set.o)
(include node_test.o)
(include identify_faults.o)
(include generic_rulebase.o)
(include simulate.l)
;
;
(defun diex ()
  (setq current_rns nil)
  (setq valid_meas_list nil)
  (exec clear)
  (patom '|                    DIEX -- Fault Diagnosis Expert System|) (terpri)
  (patom '|Fault diagnosis based on causal models of chemical process |)
  (patom '|equipment.|) (terpri) (terpri) (terpri)
  (do ((num nil))
      ((equal num 'stop) (patom '|List of primary deviations: |) current_rns)
    (patom
      '|Enter the number of new valid measurements, or 'stop' to quit. => |)
    (setq num (read)) (terpri)
    (cond ((and (fixp num) (plusp num))
           (get_new_measurements num)
           (do ((i 0 (+ i 1)))
               ((equal i num))
               (setq rns (generate_candidate_set (nth i valid_meas_list)))
               (cond ((null current_rns) (setq current_rns rns))
                     (t (setq current_rns
                              (intersection current_rns rns ()))))))
           (setq reduced_rns current_rns)
           (setq reduced_rns (heuristic_rules reduced_rns))
           (setq reduced_rns (simulate reduced_rns))
           (cond ((null reduced_rns) (patom
             '|Measurement pattern cannot be explained by a single origin.|)
             (terpri) (return nil))
                 (t (identify_faults valid_meas_list reduced_rns)
                    (terpri)))))))
```

```
;
;
(defun get_new_measurements (num)
  (do ((i 0 (+ i 1)))
      ((eq i num))
      (do ((flag t))
          ((null flag))
          (patom '|Enter valid measurement name => |)
          (setq mname (read))
          (cond ((member mname measurement_list) (setq flag nil))
                (t (patom mname)
                   (patom '| is NOT a measurement!|) (terpri) (terpri))))
      (do ((flag t))
          ((null flag))
          (patom '|Enter measurement sign (+ or -) => |)
          (setq msign (read))
          (cond ((member msign '(+ -)) (setq flag nil))
                (t (patom msign)
                   (patom '| is NOT a '+' or '-'!|) (terpri) (terpri))))
      (setq valid_meas_list (cons (list mname msign) valid_meas_list))
      (terpri)))
;
;
```

```
;
;
; file generate_candidate_set.l
;
; Function generate_candidate_set identifies all nodes causally upstream from
; a given deviated sensor that could be the origin of the fault.  The search
; space is bounded by normal measurements causally upstream from the abnormal
; measurement.
;
; When the node being tested is a manipulated variable in a control system,
; and the controlled variable is normal, then the disturbance may be passing
; through the control system.  The controlled node is added to the valid
; tree and the search continues causally upstream from the controlled variable.
;
; This function returns a list of candidate root nodes for the given valid
; measurement.
;
(declare (lambda generate_candidate_set backward_valid_nodes node_test)
         (localf next_node make_valid_node_instance
                 node_is_measurement node_is_measured node_is_manipulated)
         (special valid_hash meas stack rns current_term nodes_to_test
                 already_instance init_node_measurement init_node_measured
                 measurement_list init_node_manipulated fault_path cs_name
                 measurement_node controlled_node controlled_sign
                 valid_meas_list init_node-sign))
;
;
(defun generate_candidate_set (meas_node-sign)
  (clrhash valid_hash)
  (setq meas (car meas_node-sign))
  (setq stack (list meas_node-sign))
  (setq rns (list meas_node-sign))
  (addhash meas_node-sign valid_hash (gensym 'N))
  (set (gethash meas_node-sign valid_hash) (make-instance 'valid_node
    :node_name meas
    :sign (cadr meas_node-sign)))
  (do ()
      ((null stack) rns)
    (setq current_term (pop stack))
    (setq nodes_to_test (backward_valid_nodes current_term))
    (cond (nodes_to_test (mapcar 'next_node nodes_to_test)))))
```

```
;
;
(defun next_node (init_node-sign)
  (setq already_instance (gethash init_node-sign valid_hash))
  (setq init_node_measurement (member (car init_node-sign) measurement_list))
  (setq init_node_measured
    (symeval-in-instance (symeval (car init_node-sign)) 'measured))
  (setq init_node_manipulated
    (symeval-in-instance (symeval (car init_node-sign)) 'manip))
  (cond ((eq (car init_node-sign) meas))
        (already_instance
           (cond ((node_test init_node-sign current_term)
                    (set-in-instance (symeval already_instance) 'forward_nodes
                    (cons current_term (symeval-in-instance
                    (symeval already_instance) 'forward_nodes))))))
        (t (cond (init_node_measurement (node_is_measurement))
                 (init_node_measured (node_is_measured init_node_measured))
                 (t (make_valid_node_instance init_node-sign current_term)))
           (cond (init_node_manipulated (node_is_manipulated))))))
;
;
; Function make_valid_node_instance makes a new flavor instance for a valid
; node.
;
(defun make_valid_node_instance (init_node-sign term_node-sign)
  (setq fault_path (node_test init_node-sign term_node-sign))
  (cond (fault_path
           (push init_node-sign stack)
           (setq rns (cons init_node-sign rns))
           (addhash init_node-sign valid_hash (gensym 'N))
           (set (gethash init_node-sign valid_hash) (make-instance 'valid_node
             :node_name (car init_node-sign)
             :sign (cadr init_node-sign)
             :forward_nodes (list term_node-sign)
             :path fault_path)))))
;
;
(defun node_is_measurement ()
  (cond ((member init_node-sign valid_meas_list)
           (make_valid_node_instance init_node-sign current_term))))
;
;
(defun node_is_measured (measurement_node)
  (cond ((and (not (member (list measurement_node '+) valid_meas_list))
              (not (member (list measurement_node '-) valid_meas_list))))
        ((symeval-in-instance (symeval measurement_node) 'control_system)
           (make_valid_node_instance init_node-sign current_term))
        ((member (list measurement_node (cadr init_node-sign)) valid_meas_list)
           (make_valid_node_instance init_node-sign current_term))))
```

```
;
;
; Function node_is_manipulated is used to validate nodes causally upstream
; from a controlled variable.  This function is called when the manipulated
; variable is valid and the controlled variable is normal.  The control
; system is assumed working and not saturated.
;
(defun node_is_manipulated ()
  (setq cs_name (symeval-in-instance (symeval (car init_node-sign))
    'control_system))
  (setq measurement_node (symeval-in-instance (symeval cs_name)
    'measurement))
  (cond ((and (not (member (list measurement_node '+) valid_meas_list))
              (not (member (list measurement_node '-) valid_meas_list)))
         (setq controlled_node (car (backward_nodes measurement_node)))
         (setq controlled_sign (validate (symeval-in-instance
           (symeval cs_name) 'description) (cadr init_node-sign)))
         (make_valid_node_instance (list controlled_node controlled_sign)
           init_node-sign))))
;
;
```

```
;
;
; file node_test.l
;
; Function node_test checks the consistent path from the given node to the
; valid measurement to guarantee that the given node does not already lie on
; the path.
;
; "meas" is a global variable and must be set before this function is called.
; Its value is the print name of the valid measurement.
;
;
; Function popit returns
;   nil if the stack is empty;
;   the print name of the variable on the top of the stack.
;
(declare (lambda node_test)
         (localf popit node_not_in_path)
         (special new_name new_nodes meas path stack7 last_node valid_hash))
;
;
(defun popit ()
  (setq last_node (pop stack7))
  (setq path (cdr path))
  (cond ((null stack7) nil)
        (t (setq new_nodes (cdr (member last_node (symeval-in-instance
             (symeval (gethash (car stack7) valid_hash)) 'forward_nodes))))
           (cond ((null new_nodes) (popit))
                 (t (setq new_name (car new_nodes)))))))
;
;
; Function node_not_in_path returns
;   t if the current node is not already in the path;
;   nil if the the current node is in the path, or if the stack is empty
;
(defun node_not_in_path ()
  (cond ((member (car new_name) path)
           (setq new_nodes (cdr new_nodes))
           (cond ((null new_nodes)
                    (cond ((popit) (node_not_in_path))))
                 (t (setq new_name (car new_nodes))
                    (node_not_in_path))))
        (t)))
```

```
;
;
; Function node_test returns
;   nil if a consistent path is not found;
;   the list of nodes in the path if a consistent path exists.
;
(defun node_test (init_node-sign term_node-sign)
  (setq path (list (car term_node-sign) (car init_node-sign)))
  (setq stack7 (list term_node-sign))
  (do ()
      ((null stack7) nil)
      (cond ((member meas path)
               (return (cons init_node-sign (reverse stack7)))))
      (setq new_nodes (symeval-in-instance (symeval
        (gethash (car stack7) valid_hash)) 'forward_nodes))
      (setq new_name (car new_nodes))
      (cond ((node_not_in_path)
               (push new_name stack7)
               (setq path (cons (car new_name) path))))))
;
;
```

```
;
;
; file simulate.l
;
; Function simulate removes root nodes from reduced_rns.  Fault simulation
; from each root node is done on arcs with zero time delay.
;
; A node is removed if
;  1. A path exists with zero delay lag to normal measurements,
;  2. A non-consistent path exists to a valid sensor with zero delay lag.
;
;
(declare (lambda simulate)
         (localf tsimul)
         (special node_list nodestack arcstack terminal_node node-sign
                  measurement valid_meas_list reduced_rns current_node-sign
                  current_arc terminal_sign))
;
;
(defun simulate (reduced_rns)
  (mapcar 'tsimul reduced_rns)
  reduced_rns)
;
;
(defun tsimul (node-sign)
  (setq node_list (list (car node-sign)))
  (setq nodestack (list node-sign))
  (do () ((null nodestack) t)
    (setq current_node-sign (pop nodestack))
    (setq arcstack (symeval-in-instance (symeval (car current_node-sign))
      'arcs_from))
    (do () ((null arcstack) t)
      (cond ((equal (symeval-in-instance (symeval (car arcstack)) 'time) '0.0)
                (setq current_arc (pop arcstack))
                (setq terminal_node (symeval-in-instance
                  (symeval current_arc) 'terminal_node))
                (setq terminal_sign (validate (cadr current_node-sign)
                  (symeval-in-instance (symeval current_arc) 'sign)))
                (setq measurement (symeval-in-instance (symeval terminal_node)
                  'measured))
                (cond ((member terminal_node node_list))
                      (measurement
                         (cond ((member (list measurement terminal_sign)
                                   valid_meas_list) (push
                                   (list terminal_node terminal_sign) nodestack)
                                (push terminal_node node_list))
                               (t (setq reduced_rns
                                    (delete node-sign reduced_rns))
                                  (setq nodestack nil)
                                  (return))))
                      (t (push (list terminal_node terminal_sign) nodestack)
                         (push terminal_node node_list))))
            (t (pop arcstack))))))
;
;
```

```
;
;
; file identify_faults.l
;
; This set of functions accepts a list of node-sign pairs and returns, for
; each node, those faults that would cause the node to be a primary deviation.
; This function must be called after the node attributes have been set in the
; node instances.
;
;
(declare (lambda identify_faults roots_at_boundaries
           control_system_rulebase
           control-valve_rulebase
           cstr_rulebase
           heat-exchanger_rulebase
           pipe_rulebase
           pump_rulebase
           sensor_rulebase
           t-junction_rulebase
           tank_rulebase
           vaporizer_rulebase)
         (localf print_output_header print_the_fault get_fault_description)
         (special valid_meas_list deviation counter fault_description
           node sign name variable_type equip_name equip_type))
;
;
(defun print_output_header ()
  (terpri)
  (patom "Measurements:") (terpri)
  (mapcar #'(lambda (node-sign)
              (cond ((eq (cadr node-sign) '+) (setq deviation 'high))
                    (t (setq deviation 'low)))
              (patom "  ") (patom (car node-sign))
              (patom " ") (patom deviation)
              (terpri))
            (reverse valid_meas_list))
  (terpri)
  (patom "Possible Faults:")
  (terpri))
;
;
(defun identify_faults (valid_meas_list reduced_rns)
  (print_output_header)
  (setq counter 0)
  (mapcar #'(lambda (node-sign)
              (setq fault_description nil)
              (get_fault_description node-sign)
              (cond (fault_description (print_the_fault)))
              t)
            reduced_rns)
  (mapcar #'(lambda (node-sign)
              (setq fault_description nil)
              (roots_at_boundaries node-sign)
              (cond (fault_description (print_the_fault)))
              t)
            reduced_rns)
  (terpri) (terpri))
```

```
;
;
(defun get_fault_description (node-sign)
  (setq node (car node-sign))
  (setq sign (cadr node-sign))
  (setq name (symeval node))
  (setq variable_type (symeval-in-instance name 'variable_type))
  (setq equip_name (symeval-in-instance name 'equip_name))
  (setq equip_type (symeval-in-instance name 'equip_type))
  (caseq equip_type
    (CONTROL_SYSTEM  (control_system_rulebase))
    (CONTROL-VALVE   (control-valve_rulebase))
    (CSTR            (cstr_rulebase))
    (HEAT-EXCHANGER  (heat-exchanger_rulebase))
    (PIPE            (pipe_rulebase))
    (PUMP            (pump_rulebase))
    (SENSOR          (sensor_rulebase))
    (T-JUNCTION      (t-junction_rulebase))
    (TANK            (tank_rulebase))
    (VAPORIZER       (vaporizer_rulebase))
  ))
;
;
(defun print_the_fault ()
  (setq counter (add1 counter))
  (patom "  ") (print counter) (patom "> ") (patom fault_description)
  (terpri))
;
;
```

```
;
;
; file generic_rulebase.l
;
; This file contains component rule bases that relate faults to primary
; deviations for the stated assumptions.  The variables variable_type and
; equip_name must be set before a component rule base is called.
;
;
(declare (lambda control_syst m_rulebase
                 control-valve_rulebase
                 cstr_rulebase
                 heat-exchanger_rulebase
                 pipe_rulebase
                 pump_rulebase
                 sensor_rulebase
                 t-junction_rulebase
                 tank_rulebase
                 vaporizer_rulebase)
         (special node sign name variable_type equip_name equip_type
                 fault_description))
;
;
(defun pipe_rulebase ()
;
; Assumptions:  P > P atm; T > T atm
;
  (caseq variable_type
    (PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak in pipe | equip_name '|.  |)))))
    (FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage in pipe | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at pipe | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on pipe | equip_name '|.  |)))))
  ))
;
;
(defun t-junction_rulebase ()
;
; Assumptions:  P > P atm; T > T atm
;
  (caseq variable_type
    (PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak in t-junction | equip_name '|.  |)))))
    (FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage in t-junction | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at t-junction | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on t-junction | equip_name '|.  |)))))
  ))
```

```
;
;
(defun control-valve_rulebase ()
;
; Assumptions:  P > P atm; T > T atm
;
  (caseq variable_type
    (PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak in control-valve | equip_name '|.  |)))))
    (FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage in control-valve | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at control-valve | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on control-valve | equip_name '|.  |)))))
    (VALVE-STEM
      (cond ((eq sign '+) (setq fault_description (concat
        '|Control-valve | equip_name '| failed open.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Control-valve | equip_name '| failed closed.  |)))))
  ))
;
;
(defun pump_rulebase ()
;
; Assumptions: Centrifugal pump; electric motor; Newtonian liquid;
;  P inlet, P outlet > P atm; T > T atm
;
  (caseq variable_type
    (PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak in pump | equip_name '|.  |)))))
    (FLOWRATE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Entrained vapor or change of physical properties in pump |
       equip_name '|.  |)))))
    (FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage of section or discharge in pump | equip_name '|.  |)))))
    (RPM
      (cond ((eq sign '-) (setq fault_description (concat
        '|Broken shaft or coupling in pump | equip_name '|.  |)))))
    (CURRENT
      (cond ((eq sign '+) (setq fault_description (concat
        '|Power too high to electric motor on pump | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Loss of power to electric motor on pump | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at pump | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on pump | equip_name '|.  |)))))
  ))
```

```
;
;
(defun sensor_rulebase ()
  (cond ((eq sign '-) (setq fault_description (concat
    '|Sensor | node '| failed low.  |)))
       ((eq sign '+) (setq fault_description (concat
    '|Sensor | node '| failed high.  |)))))

 ))

;
;
(defun control_system_rulebase ()
  (caseq variable_type
    (ERROR
      (cond ((eq sign '+) (setq fault_description (concat
        '|Control system | equip_name '| failed high.  |)))
           ((eq sign '-) (setq fault_description (concat
        '|Control system | equip_name '| failed low.  |)))))
    (SETPOINT
      (cond ((eq sign '+) (setq fault_description (concat
        '|The setpoint of | equip_name '| set high.  |)))
           ((eq sign '-) (setq fault_description (concat
        '|The setpoint of | equip_name '| set low.  |)))))
 ))
;
;
(defun tank_rulebase ()
;
; Assumptions: Liquid inlet ports above liquid level; tank at atmospheric
;  pressure (P = P atm); T > T atm
;
  (caseq variable_type
    (VOLUME
      (cond ((eq sign '-) (setq fault_description (concat
        '|Liquid leak from tank | equip_name '|.  |)))))
    (TANK-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Vapor leak from tank | equip_name '|.  |)))))
    (INLET-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Inlet blockage in tank | equip_name '|.  |)))))
    (OUTLET-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak at outlet in tank | equip_name '|.  |)))))
    (OUTLET-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Outlet blockage in tank | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at tank | equip_name '|.  |)))
           ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on tank | equip_name '|.  |)))))
 ))
```

```
;
;
(defun cstr_rulebase ()
;
; Assumptions:  P vapor > P atm; T reactor > T atm
;
  (caseq variable_type
    (VOLUME
      (cond ((eq sign '-) (setq fault_description (concat
        '|Liquid leak from reactor | equip_name '|.  |)))))
    (REACTOR-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Vapor leak from reactor | equip_name '|.  |)))))
    (INLET-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Inlet blockage [| node '|] in reactor | equip_name '|.  |)))))
    (OUTLET-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak at outlet [| node '|] in reactor | equip_name '|.  |)))))
    (OUTLET-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Outlet blockage [| node '|] in reactor | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at reactor | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on tank | equip_name '|.  |)))))
    (CONC-A
      (cond ((eq sign '-) (setq fault_description (concat
        '|Side reaction occuring in reactor | equip_name
        '|, depleting reactant.  |)))))
    (RX_RATE_CONSTANT
      (cond ((eq sign '-) (setq fault_description (concat
        '|Catalyst fouling in reactor | equip_name '|.  |)))))
))
```

```
;
;
(defun heat-exchanger_rulebase ()
;
; Assumptions:  P hot > P cold > P atm; T hot > T cold       .
;
  (caseq variable_type
    (HOT-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak in hot stream in heat exchanger | equip_name '|.  |)))))
    (HOT-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage in hot stream in heat exchanger | equip_name '|.  |)))))
    (HOT-TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at heat exchanger | equip_name '|.  |)))))
    (COLD-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak in cold stream in heat exchanger | equip_name '|.  |)))))
    (COLD-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage in cold stream in heat exchanger | equip_name '|.  |)))))
    (HX_RATE_CONSTANT
      (cond ((eq sign '-) (setq fault_description (concat
        '|Severe fouling in heat exchanger | equip_name '|.  |)))))
    (SHELL-TUBE-FLOW-RESIST
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak between shell and tube sides in heat exchanger |
        equip_name '|.  |)))))
  ))
```

```
;
;
(defun vaporizer_rulebase ()
;
; Assumptions:  P > P atm; T > T atm; leak of hot fluid from heating coils
;  into vaporizer not included in the list of faults.
;
  (caseq variable_type
    (LEVEL
      (cond ((eq sign '-) (setq fault_description (concat
        '|Liquid leak from vaporizer | equip_name '|.  |)))))
    (VAPORIZER-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Vapor leak from vaporizer | equip_name '|.  |)))))
    (INLET-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Inlet blockage [| node '|] in vaporizer | equip_name '|.  |)))))
    (OUTLET-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Leak at outlet [| node '|] in vaporizer | equip_name '|.  |)))))
    (OUTLET-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Outlet blockage [| node '|] in vaporizer | equip_name '|.  |)))))
    (TEMPERATURE
      (cond ((eq sign '+) (setq fault_description (concat
        '|Fire at vaporizer | equip_name '|.  |)))
            ((eq sign '-) (setq fault_description (concat
        '|Insulation removed on vaporizer | equip_name '|.  |)))))
    (HOT-FLOW_RESIST
      (cond ((eq sign '+) (setq fault_description (concat
        '|Blockage in heating coils in vaporizer | equip_name '|.  |)))))
    (HOT-PRESSURE
      (cond ((eq sign '-) (setq fault_description (concat
        '|Hot stream leak at exit of vaporizer | equip_name '|.  |)))))
    (HX_RATE_CONSTANT
      (cond ((eq sign '+) (setq fault_description (concat
        '|Severe fouling in heating coils in vaporizer | equip_name '|.  |)))))
  ))
;
;
```

```
;
;
; file vapor_boundary.l
;
; This function, specific to the vaporizer example, contains the possible
; faults for primary deviations at the process boundaries.  The function
; takes a root node and checks to see if it is on the boundary.  If it is,
; it assigns fault_description.
;
;
(declare (lambda roots_at_boundaries)
         (special fault_description boundary_list))
;
;
(defun roots_at_boundaries (node-sign)
  (setq fault_description nil)
  (setq boundary_node (member (car node-sign) boundary_list))
  (cond (boundary_node
          (setq node (car node-sign))
          (setq sign (cadr node-sign))
          (caseq node
            (P1 (cond ((eq sign '+) (setq fault_description
                  '|High pressure upstream of pipe PIPE-A.  |))
                      ((eq sign '-) (setq fault_description
                  '|Low pressure upstream of pipe PIPE-A.  |))))
            (T1 (cond ((eq sign '+) (setq fault_description
                  '|High temperature fluid entering pipe PIPE-A.  |))
                      ((eq sign '-) (setq fault_description
                  '|Low temperature fluid entering pipe PIPE-A.  |))))
            (P4 (cond ((eq sign '+) (setq fault_description
                  '|High pressure upstream of control-valve CV-2.  |))
                      ((eq sign '-) (setq fault_description
                  '|Low pressure upstream of control-valve CV-2.  |))))
            (T4 (cond ((eq sign '+) (setq fault_description
                  '|High temperature fluid entering control-valve CV-2.  |))
                      ((eq sign '-) (setq fault_description
                  '|Low temperature fluid entering control-valve CV-2.  |))))
            (P6 (cond ((eq sign '+) (setq fault_description
              '|High pressure downstream of P6 in vaporizer VAPORIZER-1.  |))
                      ((eq sign '-) (setq fault_description
              '|Low pressure downstream of P6 in vaporizer VAPORIZER-1.  |))))
            (P8 (cond ((eq sign '+) (setq fault_description
                  '|High pressure downstream of pipe PIPE-B.  |))
                      ((eq sign '-) (setq fault_description
                  '|Low pressure downstream of pipe PIPE-B.  |)))))
          )))
;
;
```

Appendix C-1

Component Digraphs for the Process Schematic in Figure 5-1: Tank With a Level Control System

Arcs exist from the terms in the functional description to the dependent term. The sign attribute of the arc is the sign of the independent term.

Context-Specific Assumptions:

1. The valve is open
2. All flow rates are positive and in the directions assumed
3. Liquid inlet in the tank is above liquid level
4. Tank is at atmospheric pressure

TANK-1

$V = f(F_1, -F_2)$

$L = f(V)$

$P_b = f(L)$

$F_2 = f(P_b, -P_2, -R_2)$

$P_2 = f(F_2)$

CV-1

$F_{34} = f(P_3, -P_4, -R_{34})$

$P_3 = f(-F_{34})$

$P_4 = f(F_{34})$

$R_{34} = f(-v_1)$

PIPE-A

$F_{23} = f(P_2, -P_3, -R_{23})$

$P_2 = f(-F_{23})$

$P_3 = f(F_{23})$

PIPE-B

$F_{45} = f(P_4, -P_5, -R_{45})$

$P_4 = f(-F_{45})$

$P_5 = f(F_{45})$

Measurement

$$L_{sensor} = f(L)$$

$$F_{sensor} = f(F_{45})$$

Level Control System

$$L_{error} = f(L_{sensor}, -L_{sp})$$

$$v_1 = f(L_{error})$$

Appendix C-2

Component Digraphs for the Process Schematic in Figure 5-2: Vaporizer

Arcs exist from the terms in the functional description to the dependent term. The sign attribute of the arc is the sign of the independent term.

Context-Specific Assumptions:

1. All valves are open
2. All flow rates are positive and in the directions assumed
3. $T_3 < T < T_5$
4. Constant fluid properties (ρ, C_p, λ)
5. Liquid inlet in the vaporizer is above liquid level
6. No structural faults in the vaporizer

VAPORIZER $F_{3T} = f(P_3, -P_T, -R_{3T})$

$P_3 = f(-F_{3T})$

$L = f(F_{3T}, -VAPOR\_RATE)$

$F_{T7} = f(P_T, -P_7, -R_{T7})$

$P_T = f(-F_{T7}, VAPOR\_RATE)$

$P_7 = f(F_{T7})$

$VAPOR\_RATE = f(-P_T, T)$

$F_{56} = f(P_5, -P_6, -R_{56})$

$P_5 = f(-F_{56})$

$P_6 = f(F_{56})$

$T_6 = f(T_5, F_{56}, -Q)$

$T = f(T_3, -F_{3T}, Q, -VAPOR\_RATE)$

$Q = f(T_6, -T, -R_h)$

$T_7 = f(T)$

CV-1 $F_{23} = f(P_2, -P_3, -R_{23})$

$P_2 = f(-F_{23})$

$P_3 = f(F_{23})$

$R_{23} = f(-v_1)$

$T_3 = f(T_2)$

CV-2 $F_{45} = f(P_4, -P_5, -R_{45})$

$P_4 = f(-F_{45})$

$P_5 = f(F_{45})$

$R_{45} = f(-v_2)$

$T_5 = f(T_4)$

PIPE-A $F_{12} = f(P_1, -P_2, -R_{12})$

$P_1 = f(-F_{12})$

$P_2 = f(F_{12})$

$T_2 = f(T_1)$

PIPE-B $F_{78} = f(P_7, -P_8, -R_{78})$

$P_7 = f(-F_{78})$

$P_8 = f(F_{78})$

$T_8 = f(T_7)$

Measurement

$F_{sensor} = f(F_{12})$

$L_{sensor} = f(L)$

$P_{sensor} = f(P_T)$

Level Control System

$L_{error} = f(L_{sensor}, -L_{sp})$

$v_1 = f(L_{error})$

$P_{error} = f(P_{sensor}, -P_{sp})$

$v_2 = f(P_{error})$

Appendix C-3

Component Digraphs for the Process Schematic in Figure 5-3: Continuous Stirred Tank Reactor

Arcs exist from the terms in the functional description to the dependent term. The sign attribute of the arc is the sign of the independent term.

Context-Specific Assumptions:

1. All valves are open
2. All flow rates are positive and in the directions assumed
3. T_2, $T_{12} < T_R$
4. $T_6 > T_{14}$
5. $P_6 > P_{14}$ (for heat exchanger structural fault)
6. Constant fluid properties (ρ, C_p)
7. Exothermic reaction
8. Constant heat of reaction (ΔH_r)
9. Recycle concentrations C_A and C_B are identical to the reactor concentrations.

CSTR-1

$F_{2R} = f(P_2, -P_v, -R_{2R})$

$P_2 = f(-F_{2R})$

$F_{12R} = f(P_{12}, -P_v, -R_{12R})$

$P_{12} = f(-F_{12R})$

$F_{R3} = f(P_b, -P_3, -R_{R3})$

$P_3 = f(F_{R3})$

$V_\ell = f(F_{2R}, F_{12R}, -F_{R3})$

$V_v = f(-V_\ell)$

$P_v = f(-V_v, T_R)$

$L = f(V_\ell)$

$P_b = f(L, P_v)$

$\theta_1 = f(V_\ell, -F_{2R})$

$\theta_2 = f(V_\ell, -F_{12R})$

$C_{A_R} = f(C_{A_2}, -\theta_1, -r)$

$C_{B_R} = f(\theta_1, r)$

$T_R = f(T_2, T_{12}, \theta_1, \theta_2, r)$

$r = f(k, C_{A_R})$

$k = f(T_R)$

$C_{A_3} = f(C_{A_R})$

$C_{B_3} = f(C_{B_R})$

$T_3 = f(T_R)$

HX-1

$F_{610} = f(P_6, -P_{10}, -R_{610})$

$P_6 = f(-F_{610})$

$P_{10} = f(F_{610})$

$T_{10} = f(T_6, F_{610}, -Q)$

$C_{A_{10}} = f(C_{A_6})$

$C_{B_{10}} = f(C_{B_6})$

$F_{1415} = f(P_{14}, -P_{15}, -R_{1415})$

$P_{14} = f(-F_{1415})$

$P_{15} = f(F_{1415})$

$T_{15} = f(T_{14}, -F_{1415}, Q)$

$Q = f(T_{10}, -T_{15}, -R_h)$

$F_{614} = f(P_6, -P_{14}, -R_{614})$

$P_6 = f(-F_{614})$

$P_{14} = f(F_{614})$

PUMP-1

$F_{34} = f(P_3, -P_4, -R_{34}, \omega)$

$P_3 = f(-F_{34})$

$P_4 = f(F_{34})$

$T_4 = f(T_3)$

$C_{A_4} = f(C_{A_3})$

$C_{B_4} = f(C_{B_3})$

$\omega = f(i)$

TJ-1

$F_{56} = f(P_5, -P_6, -R_{56})$

$P_5 = f(-F_{56}, -F_{57})$

$P_6 = f(F_{56})$

$T_6 = f(T_5)$

$C_{A_6} = f(C_{A_5})$

$C_{B_6} = f(C_{B_5})$

$F_{57} = f(P_5, -P_7, -R_{57})$

$P_7 = f(F_{57})$

$T_7 = f(T_5)$

$C_{A_7} = f(C_{A_5})$

$C_{B_7} = f(C_{B_5})$

PIPE-A

$F_{12} = f(P_1, -P_2, -R_{12})$

$P_1 = f(-F_{12})$

$P_2 = f(F_{12})$

$T_2 = f(T_1)$

$C_{A_2} = f(C_{A_1})$

PIPE-B

$F_{45} = f(P_4, -P_5, -R_{45})$

$P_4 = f(-F_{45})$

$P_5 = f(F_{45})$

$T_5 = f(T_4)$

$C_{A_5} = f(C_{A_4})$

$C_{B_5} = f(C_{B_4})$

PIPE-C $F_{89} = f(P_8, -P_9, -R_{89})$

$P_8 = f(-F_{89})$

$P_9 = f(F_{89})$

$T_9 = f(T_8)$

$C_{A_9} = f(C_{A_8})$

$C_{B_9} = f(C_{B_8})$

PIPE-D $F_{1011} = f(P_{10}, -P_{11}, -R_{1011})$

$P_{10} = f(-F_{1011})$

$P_{11} = f(F_{1011})$

$T_{11} = f(T_{10})$

$C_{A_{11}} = f(C_{A_{10}})$

$C_{B_{11}} = f(C_{B_{10}})$

PIPE-E $F_{1516} = f(P_{15}, -P_{16}, -R_{1516})$

$P_{15} = f(-F_{1516})$

$P_{16} = f(F_{1516})$

$T_{16} = f(T_{15})$

CV-1 $F_{78} = f(P_7, -P_8, -R_{78})$

$P_7 = f(-F_{78})$

$P_8 = f(F_{78})$

$R_{78} = f(-v_1)$

$T_8 = f(T_7)$

$C_{A_8} = f(C_{A_7})$

$$C_{B_8} = f(C_{B_7})$$

CV-2

$$F_{1112} = f(P_{11}, -P_{12}, -R_{1112})$$

$$P_{11} = f(-F_{1112})$$

$$P_{12} = f(F_{1112})$$

$$R_{1112} = f(-v_2)$$

$$T_{12} = f(T_{11})$$

$$C_{A_{12}} = f(C_{A_{11}})$$

$$C_{B_{12}} = f(C_{B_{11}})$$

CV-3

$$F_{1314} = f(P_{13}, -P_{14}, -R_{1314})$$

$$P_{13} = f(-F_{1314})$$

$$P_{14} = f(F_{1314})$$

$$R_{1314} = f(-v_3)$$

$$T_{14} = f(T_{13})$$

Measurements

$$P_{sensor} = f(P_v)$$

$$L_{sensor} = f(L)$$

$$C_{A_{sensor}} = f(C_{A_9})$$

$$C_{B_{sensor}} = f(C_{B_9})$$

$$T_{1_{sensor}} = f(T_R)$$

$$T_{2_{sensor}} = f(T_{11})$$

$$F_{1_{sensor}} = f(F_{1011})$$

$$F_{2_{sensor}} = f(F_{89})$$

$$F_{3_{sensor}} = f(F_{1516})$$

Level Control System

$$L_{error} = f(L_{sensor}, -L_{sp})$$

$$v_1 = f(L_{error})$$

Recycle Control System

$$F_{error} = f(F_{1_{sensor}}, -F_{sp})$$

$$v_2 = f(-F_{error})$$

Temperature Control System

$$T_{error} = f(T_{1_{sensor}}, -T_{sp})$$

$$v_3 = f(T_{error})$$