ISSUES CONCERNING BLACKBOARD-BASED FAULT DIAGNOSIS SYSTEMS

A Masters Bypass Report
Submitted to the Faculty

of

Purdue University

by

Edward Price Gatzke

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

December 10,1996

DISCARD THIS PAGE

TABLE OF CONTENTS

DISCARD THIS PAGE

LIST OF FIGURES

NOMENCLATURE

Blackboard Nomenclature

$AEGIS$	Abnormal Event Guidance and Information System
ASM	Abnormal Situation Management
$DKIT$	Diagnostic Toolkit
KS	Knowledge Source–Any stand–alone diagnosis program
PRM	Plant Reference Model

Horizon–based Estimation Nomenclature

H	Length of estimation horizon
t_c	Current time
$y(i)$	Process measurement at time i
$\hat{y}(i)$	Model estimate at time i

ABSTRACT

Blackboard–based systems are currently provoking interest as a method of performing fault diagnosis in complex systems. The components of the blackboard structure are discussed, including possible knowledge sources and types of process models. A method is proposed for automating process model development by combining process information and low–level unit operation models. Preliminary results are shown, and possible future work is proposed.

1. INTRODUCTION

The chemical process industry loses millions of dollars every year due to loss of production and down time caused by problems in the process. Such problems typically fall into two qualitative categories: abnormal events and catastrophic events. Abnormal events usually result in lost production, wasted resources, defective product, or down-time. Catastrophic events lead to environmental disasters, destruction of expensive facilities, and loss of life. Safeguards and automatic shutoffs typically keep plants from developing catastrophic events, but these safety features are not always sufficient. Abnormal events are typically the gateway to a catastrophic failure. Catastrophic failures rarely happen when everything is running smoothly. Rather, a plant usually moves through a abnormal-type event in the progression towards a catastrophic event. Clearly there is a need for more accurate, well-organized, timely information about a process during the evolution of abnormal events. Early diagnosis of process faults while the plant is still operating in a controllable region in conjunction with better qualitative information about the suspected event can help avoid event progression and reduce the amount of productivity loss during an abnormal event [17].

This problem of fault diagnosis and subsequent control is made much more difficult by the scale and complexity of modern chemical plants. Processes are frequently pushed beyond their designed operating points into more nonlinear operating areas. Some fault events cause a cascading effect, triggering multiple alarms and spawning additional problems. Other events are slow to develop, taking place over the course of many shifts. A slight drift in one of many hundreds of sensors could easily go un-noticed. Operator devised rules for problem solving can break down, especially in the case of never-experienced faults. A method of recognizing problems and organizing information needs to be developed to address these concerns.

One proposed solution to the problem makes use of a blackboard architecture to aid in the diagnosis of plant faults and the management of plant information. Since no one method can accurately work as a diagnostic tool for all types of imaginable faults, a hybrid solution is being developed. The blackboard architecture takes advantage of multiple methods for fault diagnosis, using experts which can work on one specific part of the problem or can work together to solve the problem. The core of such a method is the actual model of the plant. This model can be used by the different diagnostic methods to glean current information about the process or to post current proposed solutions to the problem at hand. To effectively model a chemical process, multiple models should be used. These models include such information as a physical description of the plant (location and size of equipment), a logical hierarchy of the plant structure, a goal-tree representing the objectives of the plant, and a malfunction-tree which represents what can go wrong with the plant. Developing models for a complex system can be extremely time consuming. Qualitative models often are biased by the individual developer of the model. Two different people, working on the same process could develop totally different models. If advanced diagnostic methods are ever to be used in real-world situations, a better method for model development must be developed.

In this report, an overview of work related to blackboard implementation is addressed. This includes a review of types of blackboards and how they are implemented, a review of advanced knowledge sources necessary for diagnosis in blackboard systems, and a review of different methods of model automation. This paper goes on to propose a methodology for the automation of blackboard database models. The proposed method relies upon developing an overall representation from developed models residing in a model library. A case-study is given as an example of this method. Finally, the report suggests directions and related problems for future work.

2. BLACKBOARDS

The blackboard-based approach for collective problem solving developed due to a need to solve complex, computationally intensive problems. This approach is based upon the method often used by human experts to solve a problem. For example, assume that a group of engineers are working on a difficult problem. First, they may write the problem on a blackboard. Each engineer may have an area of expertise applicable to the problem, but each individual engineer lacks enough knowledge to solve the complete problem. Each problem solver would work on parts of the problem, adding proposed solutions or partial solutions to the blackboard. The other problem solvers could look to the blackboard for partial solutions and recent information, eventually adding the product of their own knowledge to the total blackboard information. Typically, there would be a supervisor that regulates who should work on which parts of the problem, controls how the engineers interact while solving the problem, and decides what the final solution should be.

This method can easily be cast as an algorithm for computer-aided problem solving. The engineers correspond to diagnostic applications which have a limited domain of expertise. These knowledge sources can have overlapping areas of knowledge, but they should specialize in certain specific areas. The blackboard corresponds to a computer database representing the problem at hand including all the information currently available. The supervisor would be a scheduler application which allocates computer resources by starting diagnostic applications and ultimately pulling all available blackboard information together to arrive at a conclusion.

One example comes from a simple flow system such as figure (2.1) [14]. In this example, a level controller keeps a tank at a certain level, ensuring that the product

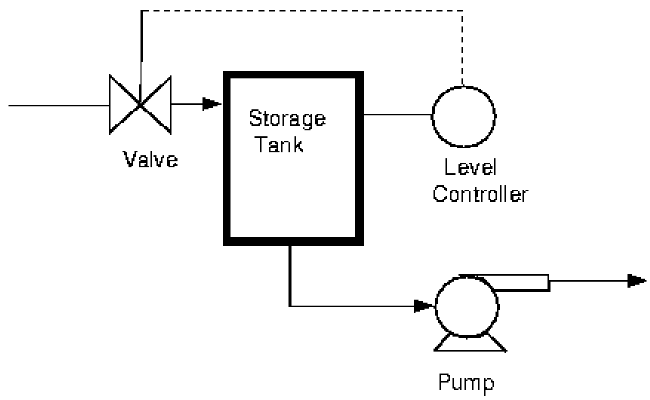

Figure 2.1 Simple flow system

can be supplied by an exit pump. Pump cavitation results if the level drops too low. A goal model of this system can be developed for the blackboard to use. Separate experts can also be developed which diagnose the system (see figure (2.2)). The sensor expert detects sensor bias, the controller expert ascertains the status of the controller, the cavitation expert can find out if the pump is cavitatiing, and the production expert monitors the process output. When a problem occurs messages are sent from the experts to the blackboard. Posting of a message may change the state of the blackboard such that trigger another expert to take action and present a new solution.

There are many advantages to using a blackboard system. A blackboard architecture can be designed such that the system runs in parallel. Since each knowledge source is basically independent of the other knowledge sources, each one may run on separate platforms. This modularization is also good for large-scale development of systems. If the basic structure is specified, the knowledge sources can be developed independently by different groups of people. Poorly developed experts can be replaced and additional experts can be added. Blackboard systems also work well when faced with uncertain problems. When solving the problem, the scheduler takes all the information from the knowledge sources and formulates the best answer. This answer should be the best solution, even though experts may give incomplete

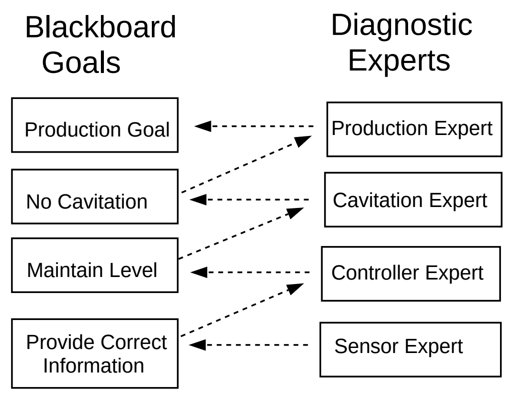

Figure 2.2 Experts and goals

responses, inaccurate responses, or contradicting responses. The scheduler could take into account each expert's domain of knowledge and give it a weight according to how probable the information is correct. The blackboard can also become a good tool for information management. The blackboard may contain models of the actual system or information about the state of the current problem solution. Even if the scheduler breaks down and refuses to find an answer to a problem, a human operator could possibly look at the intermediate blackboard information and draw conclusions from it.

One of the first applications of the blackboard architecture was the Hearsay speech understanding system, developed at Carnegie Mellon University from 1971-1976 [3]. This application was an early attempt at computerized voice recognition. A description and summary of the project is given by Lee Erman [4]. This system had a hierarchy of inter-related problems to solve. At the lowest level, data was formed into segments. Segments form syllables, syllables form words, words form phrases, and phrases form sentences. At each level, expert systems were developed to cope with the problem at hand. Conflicts can arise if a result is erroneous (*i.e.* two proposed words which cannot follow in a sentence). The scheduler works to resolve conflicts and keep the experts working productively. Because of limited computer resources, the focus of attention problem was found. Experts that are computationally intensive should only be called if necessary. The overall results from the Hearsay project were not extremely good, but much was learned from the implementation of this blackboard system.

Another early use for blackboard systems was for use in military target tracking problems. An explanation of the HASP program is given by Penny Nii [16] and a description of the RTBB problem is developed by P. Kersten [9]. The HASP program was a sonar-based problem for sub detection, and the RTBB was a radar tracking system for detecting flying threats. Both problems share similar characteristics. One similarity is that large amounts of data are available for analysis. This includes sensor data and data about known activities in the area of interest. Once a track is formed for a source, the sensor may lose contact. Tracks fade in and out, forcing a overall solution based on partial data. The systems work in systems with large amounts of uncertainty because they are attempting to find a source which may be taking evasive action, obscuring the signal. The blackboard facilitates the interactions of the 40 or more knowledge sources which work together to solve the problem.

Recently, a blackboard systems have been applied to a chemical processes. A. Crespo *et. al.* used a temporal blackboard to reason about control of a cement kiln process [2]. In this application, past data from the process is used to find control actions and the results of the control actions. Numerous predictions about how the system will react are posted to the blackboard. As more information comes in from the process, the probability of each prospective hypothesis can be adjusted up and down. This helps develop an idea of the current sate of the system. One drawback to this approach is that the blackboard typically must wait for enough evidence from the process to evolve in order to weed out false guesses. The only solutions this blackboard system looks at are the few proposed by the prediction experts. The idea of using temporal evidence to develop an idea about the current and future status of the process is obviously similar to fault diagnosis. This type of prediction may be useful to incorporate into fault diagnosis systems.

Mylaraswamy has also done much work related to blackboard systems; in his thesis [14] he describes DKIT, a blackboard-based system for diagnosis of a fluidized catalytic cracking unit (FCCU). This system uses multiple experts to diagnose a variety of faults in a FCCU model. The system contained qualitative models of the FCCU which were accessible by the experts to aid in diagnosis. This type of blackboard was modified and further developed in a project with the Abnormal System Management (ASM) consortium for fault diagnosis involving FCCU units. The new system developed by ASM is called the Abnormal Event Guidance System (AEGIS). This blackboard system uses a more advanced plant model so that the experts can interact more productively with the blackboard. This model is called the Plant Reference Model (PRM). The PRM will be described in detail in the methodology section.

Blackboard systems are emerging as viable options for solving real–time complex problems. For chemical engineering applications involving fault–diagnosis blackboards are proving both useful and necessary. There are many problem areas to be addressed. Better knowledge sources, plant models, and development tools need to evolve before blackboards can be used effectively in many industrial applications.

3. KNOWLEDGE SOURCES

Advanced fault diagnosis systems rely upon procedures for extracting relevant information about a process from actual process data combined with a process model. Development of fault diagnosis procedure requires that the method returns a result quickly, therefore long–running off–line computation is to be avoided because of the quick dynamics found in many chemical processes. The method must be sensitive to small changes in the process, while at the same time false alarms must be avoided. Two different paths are available for development of fault diagnosis system: qualitative approaches and quantitative approaches.

3.1 Qualitative Models

The signed digraph is a simple method used for fault diagnosis. This method relies on the qualitative causal relationship of process variables. If one system variable deviates above normal, other process variables may change accordingly (up or down). Iri gives a basic method for diagnosis using signed digraphs to find the root cause of a disturbance [6]. This method has many limitations. The causality between process variables is assumed fixed. The magnitude of process relationships is ignored. Resolution is typically bad, resulting in many incorrect results. Some causal relationships (such as feedback controllers) can develop looping structures which are difficult to diagnose. Some of these limitations have been addressed [10], but many problems still remain.

Related to signed digraphs are Bayesian belief networks. In these applications, the causal result from one system variable to another is based upon probability.

Developing the probabilistic relationships can be difficult or impossible in systems where humans have little grasp of the relative frequency of events. These probabilities represent the biases of the developer. Cyclic causal relationships cause computational problems.

Traditional rule-based expert systems have also been applied to fault diagnosis. Mandelkern outlines the basic requirements and implementation methods for rule–based expert systems [12]. Expert systems basically use if–then rules to capture process relationships. Rules match their predicates with existing evidence. One matching rule is selected to fire. This may produce new evidence, causing a new rule to fire. Results of this type of system rely heavily upon the influence of the developer. In large complex systems, large numbers of rules may be needed. The many rules can time consuming to develop and difficult to update as the process changes.

3.2 Quantitative Models

Quantitative methods for fault diagnosis involve using process measurements, information about known inputs, and a mathematical model of a process to calculate the extent of a fault acting upon a system. A fault in a process system can be treated as a changing process state, a varying process parameter, or an unknown process input. Once the fault signal is calculated, it is compared against threshold values to determine if a fault actually is present. Frank [5] and Iserman [7] describe two popular methods of fault diagnosis using quantitative models: state estimation and parameter estimation. These methods are typically implemented with linear process models as online filters. Obviously, model mismatch will cause some residual differences to surface, causing false alarms if thresholds are too low or non-diagnosed faults if thresholds are too low.

Most chemical processes have realistic nonlinear models available for use in diagnostic applications. Attempts at developing nonlinear observers and estimators have been limited. For nonlinear systems, multiple linear observers are often used

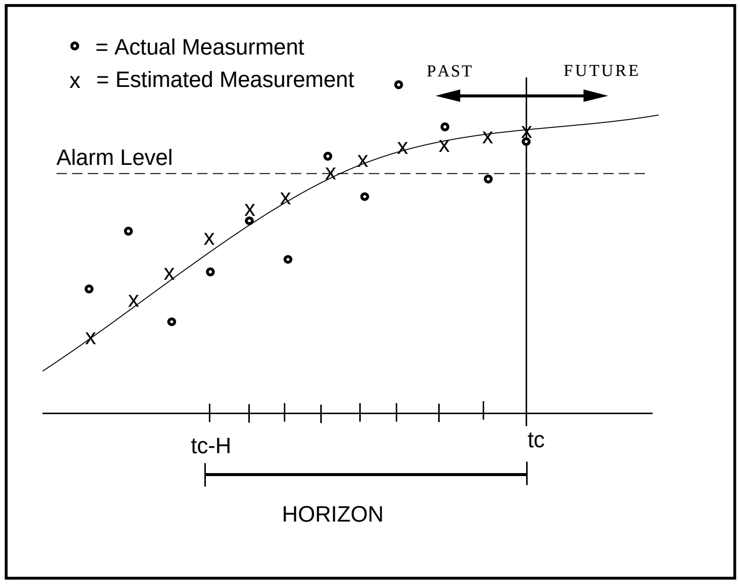

Figure 3.1 Horizon-based optimization to reconcile historical measurements

to capture the nonlinearity of the process [18]. Development of nonlinear observers using actual nonlinear models is achievable, but only for a limited set of problems [8]. As a result, there is much room for development in the realm of advanced observers.

One approach which could be developed is use of horizon-based optimization methods for state and parameter estimation. These methods are much like existing Model Predictive Control (MPC) algorithms in that a optimization problem is solved at each time step. In MPC, the optimal inputs are found to drive a system to a a desired output using process models of the system. Horizon-based optimization for parameter estimation finds the best set of states or parameters which reconcile the past process measurements (see figure (3.1)). A typical problem is posed as a minimization of the 2-norm of the difference between measurements and estimated measurements normalized by V, the variance-covariance.

Objective Function:

$$\min_{\hat{y}(t)} \quad \sum_{i=0=t_c-H}^{t_c} \quad \|y(i) - \hat{y}(i)\|_V^2$$

$$subject \quad to \quad f[\frac{d\hat{y}(t)}{dt} \qquad , \hat{y}(t)] = 0$$

$$h[\hat{y}(t)] = 0, g[\hat{y}(t)] \geq 0$$

Here, H would be the length of the horizon, t_c is the current time, y(i) are the vectors of process measurements, and $\hat{y}(i)$ are the predictions from the model. The minimization is subject to constraints created from the process model, including differential, algebraic, and inequality constraints. A robust linear case has been developed by Tyler and Morari [19]. This implementation uses multiple linear models to represent both process uncertainty and fault response. The use of linear models results in finding a solution to a quadratic programming problem which can be accomplished in real time. Use of nonlinear equations in the optimization problem has been approached by Liebman *et. al.* [11]. Quadratic optimization problems are solved repeatedly to eventually arrive at a solution. Horizon based estimation should become an effective method for extraction of fault information from process measurements and models.

4. MODEL DEVELOPMENT AUTOMATION

Blackboard systems can be used to diagnose system faults by taking advantage of different types of knowledge sources in order to arrive at a collective solution to a problem. Obviously, every element of the blackboard architecture must interact and work properly for accurate answers to the diagnosis problem. The portion of the blackboard which aids in the interaction between knowledge sources, supervisory blackboard applications, and human operators is the process model. The AEGIS system developed by the ASM consortium relies upon the PRM as a backbone of the system. This knowledge representation of a FCCU was developed in an ad-hoc manner. In order to use the blackboard system in further applications, a better method of model development must be found.

Development of digraphs by humans either relies on an expert's opinion of process relationships or a painstaking analysis of model equations. Complex first principles models such as the model IV FCCU model [13] are often available for such analysis. This model contains over one hundred process equations and variables. Automated analysis of such equations should extract the requisite causal relationships needed for digraph construction. Changing operating points requires production of a new digraph. Automatic digraph construction would make such changes less painful and ensure accurate digraph models.

Some work has been done concerning model automation from a signed digraph model. Nam *et. al.* [15] have developed a procedure for extracting symptom-fault associations from a signed digraph. This method basically traces connection back in the digraph until a root node is found, pairing symptoms directly to possible faults. The normal restrictions for digraphs limit the usefulness of this method.

These restrictions include a failure to handle inverse response and problems associated with ignoring the magnitude of process deviation. Paired with automated digraph development, these methods could be helpful for fault diagnosis.

Another approach to model development is the use of a model library to aid in the development of overall process models. Models for all types of process equipment can be joined together to represent an overall process. This type of model development is used in HAZOP expert. One advantage of this type of procedure is that minimal modifications are required of the end-user when implementing a model of a specific process. Some base level models will be modified and new base level models will be created, but the overall amount of work required for developing new process models should be minimized.

The PRM now being used by the ASM consortium exhibits many of the characteristics of the multi-view object database described by Karl–Erik Arzen [1]. Information developed and used by all plant personnel (operators, process engineers, design engineers, maintenance workers, etc.) should be contained and organized in a communal database. This database would contain information on all types of process models, process history, process functions, and general process information. Such a database should automatically be updated as better models are produced or changes in the process are made.

The objective of this work is to develop a procedure that aids the creation of multiple model process representations for use in blackboard applications. The representation will be used by knowledge sources for diagnosis. The model will also offer insight and process information to plant operators. The multiple model knowledge representation will be created from low-level basic models of unit operations.

5. METHODOLOGY

In order to understand the creation of the plant reference model, one must comprehend what the PRM actually is. The PRM is a group of models available on the blackboard for use by the knowledge experts for diagnosis. The PRM is also available to a human user for insight about the plant itself. Four different types of models (or stripes) are currently used in the PRM. These four representations are: the physical stripe, the logical stripe, the goal stripe, and the malfunction stripe.

The physical stripe is a model of the plant containing the physical information about process units. Unit location, size, capacity, and related information are stored here. This is basically a process and instrumentation diagram for the process. The physical stripe is mostly useful for human operators to help understand the process, although experts could be developed to use this representation. For example, a knowledge source cold be developed for steady-state process optimization using the physical stripe. An expert could also be developed to help contain catastrophic disasters, it i.e. if reactor 1 has overflowed, what equipment may directly be affected?

The logical stripe is another type of model used in the PRM. The logical stripe contains a model of the plant hierarchy. This model breaks the process down into systems and sub-systems. This helps show dependencies and relationships in the plant.

The goal stripe is a goal tree representation of the process. This representation is related to the logical stripe. At the top node, the goal would be similar to "Maintain normal operations of overall process". At the next level would be all the goals needed in order to ensure that the higher level goal is met. The goals and subgoals are typically similar to the logical dependencies of the process equipment.

The fourth type of model in the PRM is the malfunction stripe. This representation is also related to the structure of the plant itself. The top level malfunction would be a general plant malfunction. This malfunction could be caused by malfunctions in plant subsystems. A malfunction in a subsystem is caused by malfunctions in further specialized subsystems. Eventually, a single root cause is found in the malfunction tree. Each branching of the malfunction tree is an or connection, representing that one system or another is to blame. Only one system is the root cause.

The automation of the plant reference model is fairly simple. First, the process must be represented on the physical stripe. Using this representation, the logical stripe and hierarchy can be developed with help from the human developer. From this representation, the goal and malfunction stripes can be created.

5.1 Physical Stripe Development

The physical stripe is the basis for model development. This is the actual representation of the physical plant. Typically, detailed CAD drawings of process operations are available. PRM use and development has been done in a Gensym's G2 environment. New software has been developed which can automatically translate CAD renderings into G2 objects. This software would recognize a pump from a P&ID CAD diagram and create a representative G2 pump object. This process would speed the model development for existing plants.

This software was not available for use during creation of the prm–developer. Instead of automatically making the physical stripe from a CAD drawing, the user can create objects from the process equipment library (see figure (5.1)). Pipe and wire connections can easily be added to the process diagram. The process equipment library currently covers the basic types of unit operations (figure (5.2)).

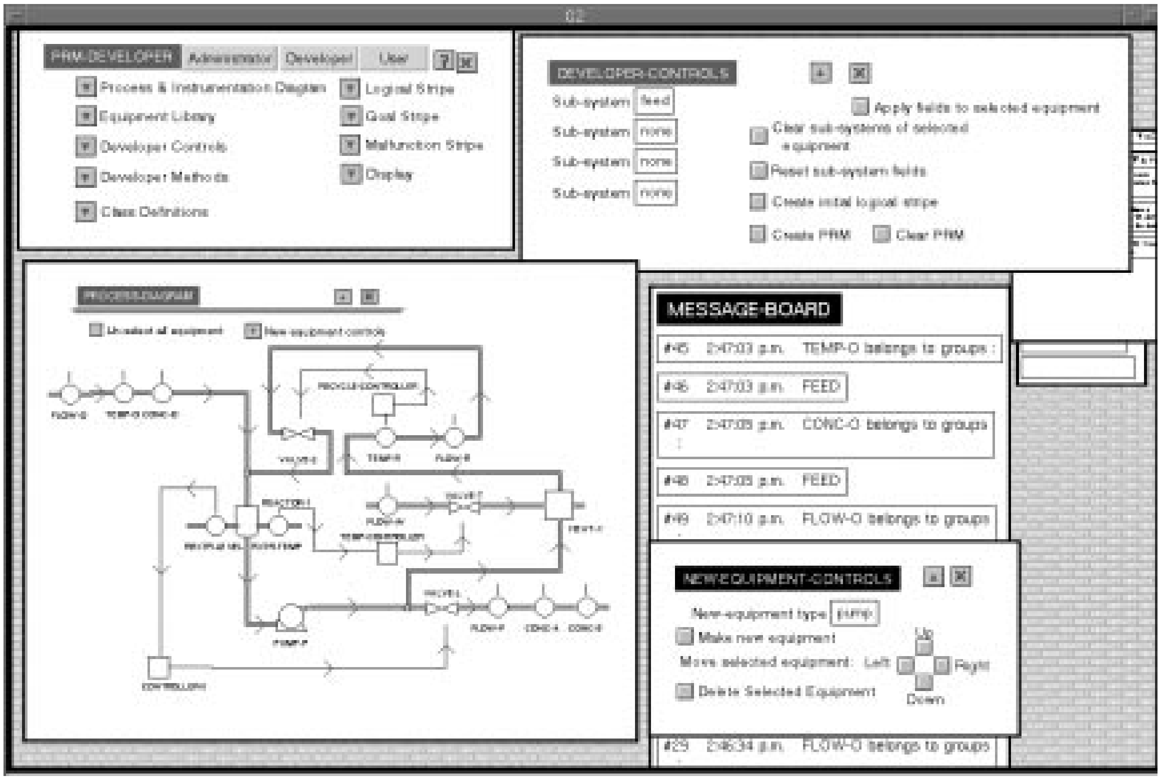

Figure 5.1 Model development interface

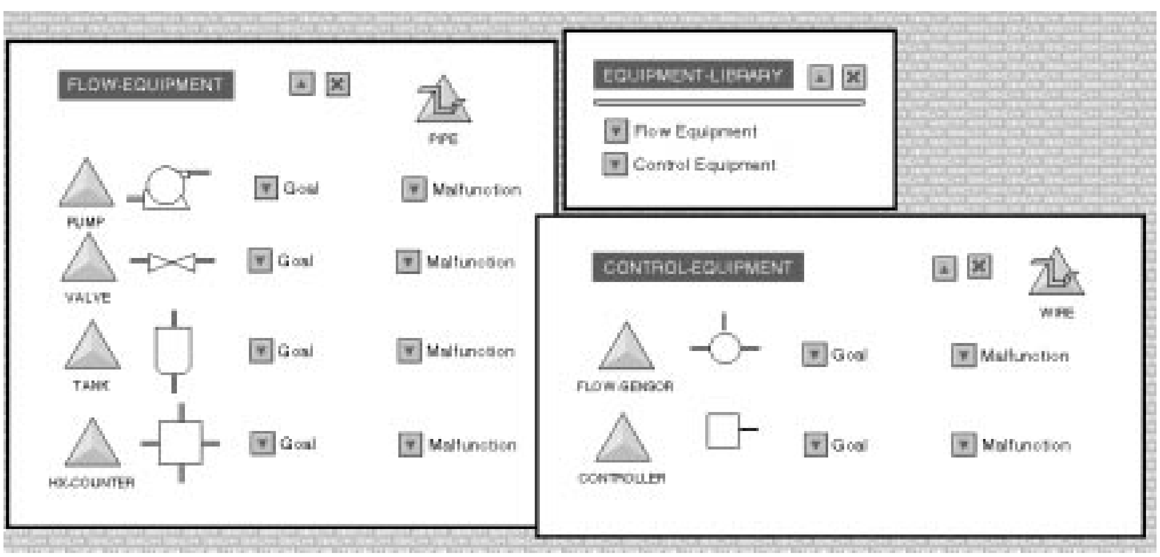

Figure 5.2 Equipment Library

5.2 Logical Stripe Development

The logical stripe is the most important representation for model development. This model must capture the process hierarchy of the entire system. This is the point where human knowledge of the process must be used to aid in the model development. The computer cannot recognize a group of unit operations as a feed system or a reactor as the most important process item. The user is expected to create group representations for the P&ID. The user selects a group of process equipment and assigns the equipment a to a named group. After all groupings are made, a preliminary logical stripe is created.

The algorithm to create the preliminary logical stripe is fairly straightforward.

- Find out the number of basic units in each logical group. Each individual piece of equipment can be considered a group in itself of size 1.
- Make an overall list of groups. These groups should be the minimum number of groups which includes the entire basic equipment list.
- Expand each group i:
  - Form a list of the groups which have all their equipment in group i.
  - From the list, continue to do the following until the list is empty
    - Find the largest group on the list.
    - Make a new logical for the group.
    - Expand the group, if possible.
    - Remove the logical from the list.

Obviously, depending upon how the user decides on groupings, vastly different logical representations can come about. After the initial logical stripe is produced, the user can modify the organization by copying, cloning, and reconnecting logical group representations.

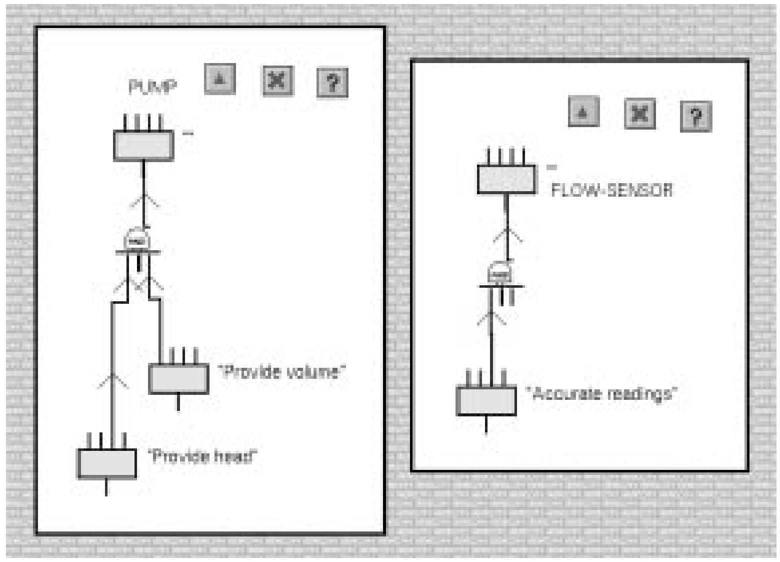

Figure 5.3 Basic equipment goals

5.3 Goal and Malfunction Stripe Development

The goal and malfunction stripes can be created directly from a combination of the logical stripe and the basic models in the equipment library. Some examples of base level goals are given in figure (5.3). Examples of base level malfunction models are shown in figure (5.4). New models for equipment not in the library can easily be developed by cloning existing base level models.

The combination of logical stripe and base level malfunction models to form the malfunction stripe is straightforward. Each logical representing a group of unit operations should have a corresponding malfunction. The malfunction would be summarized as deviation of the system from the normal operating conditions. Each logical group representing on a single unit operation would have corresponding malfunctions based on the low level malfunction model found in the equipment library. The hierarchy of the malfunction tree is the same as the logical stripe. The group malfunctions must be correctly connected with the low–level malfunction models from the equipment library.

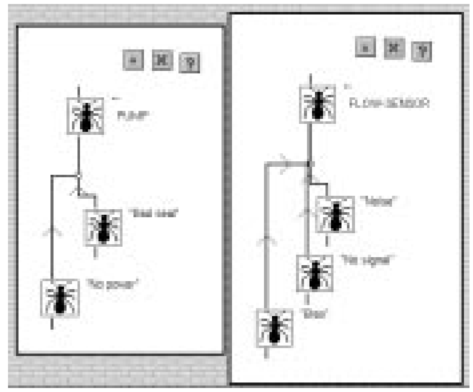

Figure 5.4 Basic equipment malfunctions

The goal tree development is similar to the malfunction tree development. The only real difference is found in the connectivity of the tree levels. In a goal tree, goals are dependent upon all subgoals being met. Therefore, and connections are used to represent the connectivity of goals from system to subsystem. Each goal must know what goals are direct superior goals and direct subordinate goals.

To facilitate use of the various types of models, each goal has a pointer to corresponding logical and malfunction objects. This model to model connectivity helps understand the results of faults. Once a malfunction in a system is diagnosed or proposed, the affected goals and systems can be found easily. Also, if a goal is not being met, the corresponding malfunction would be a likely first place to look during diagnosis.

5.4 Kramer Case Study

The first case study is an exothermic stirred–tank reactor system with a product recycle (figure (5.5)). This example is taken from a paper on signed digraph models by Kramer and Palowitch [10]. Fresh feed and cooled product are combined in a stirred tank reactor. The hot product is taken from the reactor and split into a product stream and a recycle stream. The recycle is cooled in a counter-current heat exchanger and fed back to the reactor.

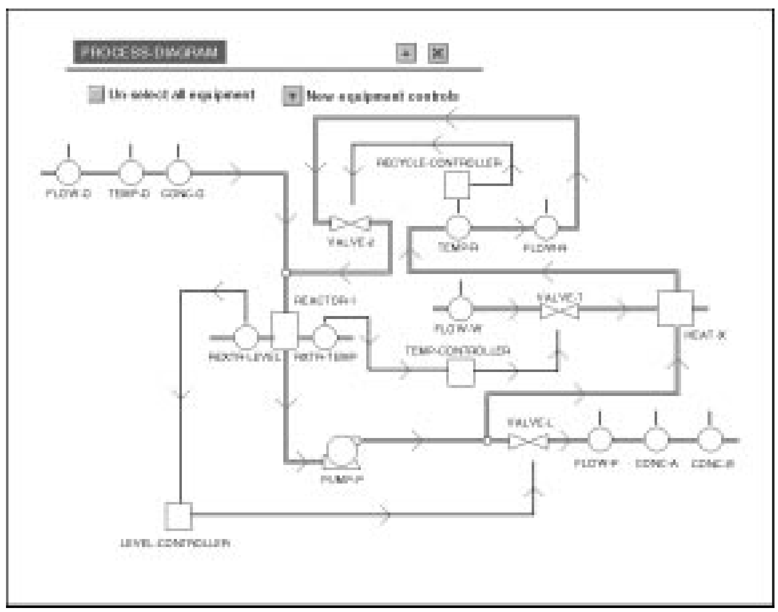

Figure 5.5 Kramer Case Study

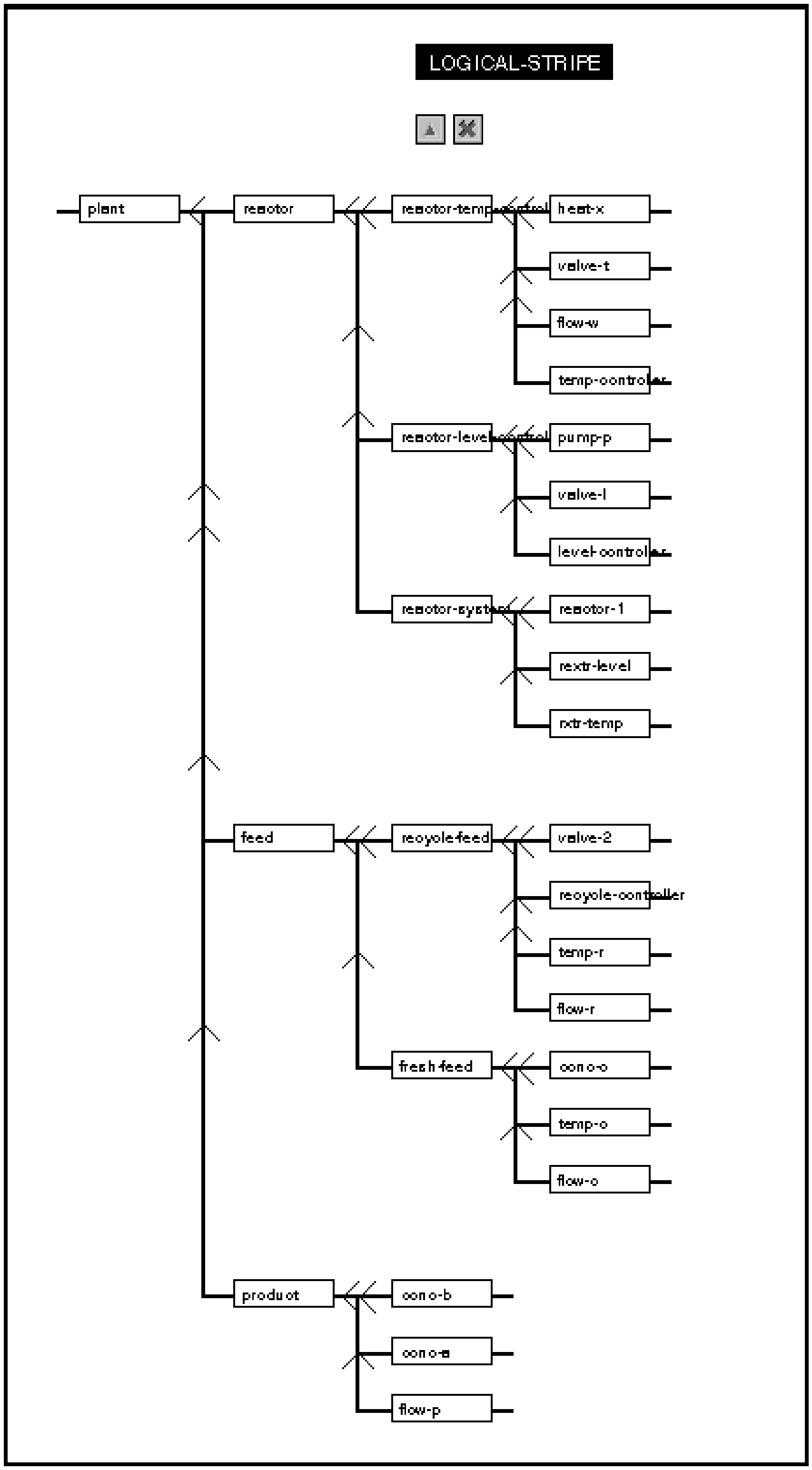

Figure 5.6 Kramer Case Study-logical groupings

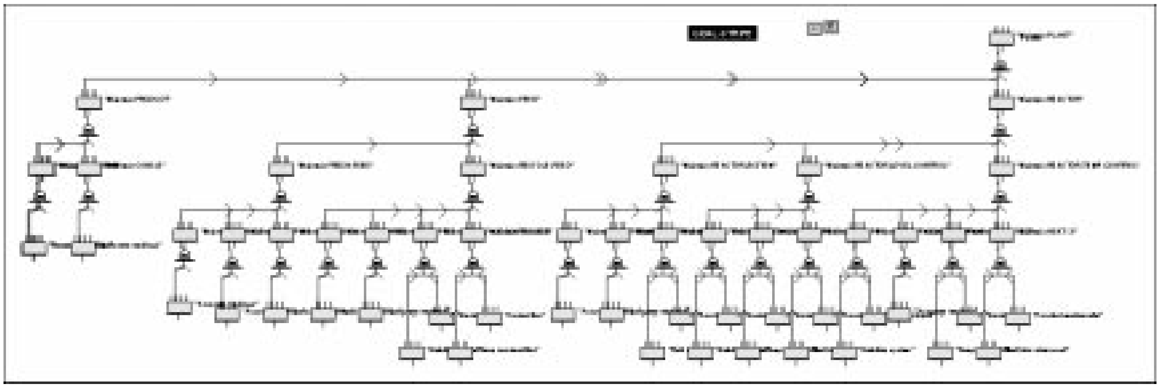

Figure 5.7 Kramer Case Study-goal stripe

The resulting logical groupings are seen in figure (5.6). One should notice that the reactor feed system includes both the fresh feed and the recycle feed. The reactor system includes the reactor itself, the level control system, and the temperature control system. The heat-exchanger can be considered part of the temperature control system. In this representation, the feedback loop is cut after the heat-exchanger and before the product is fed back into the reactor.

Goal development relies upon the logical groupings. Figure (5.7) shows the overall goal–tree for the Kramer case study. Obviously, this is a fairly complex structure, even for such a relatively simple process. A complete process plant would have many more systems, resulting in a much larger goal-tree representation. Figure (5.8) shows the feed system of the goal–tree. The feed system depends on the fresh feed and the recycle feed. Each subsystem is dependent upon the individual unit operation goals, which are taken from the equipment library. The resulting malfunction–tree is very similar and just as complex as the goal–tree (figure (5.9)). Here, a malfunction in the plant can be caused by a feed system fault, a reactor system fault, or a product system fault. Figure (5.10) shows the reactor system malfunction tree. A malfunction in the reactor system can be attributed to a malfunction in the reactor level control system, the reactor temperature control system, or the reactor itself. At a lower level, a problem with the reactor temperature control system can be caused by a

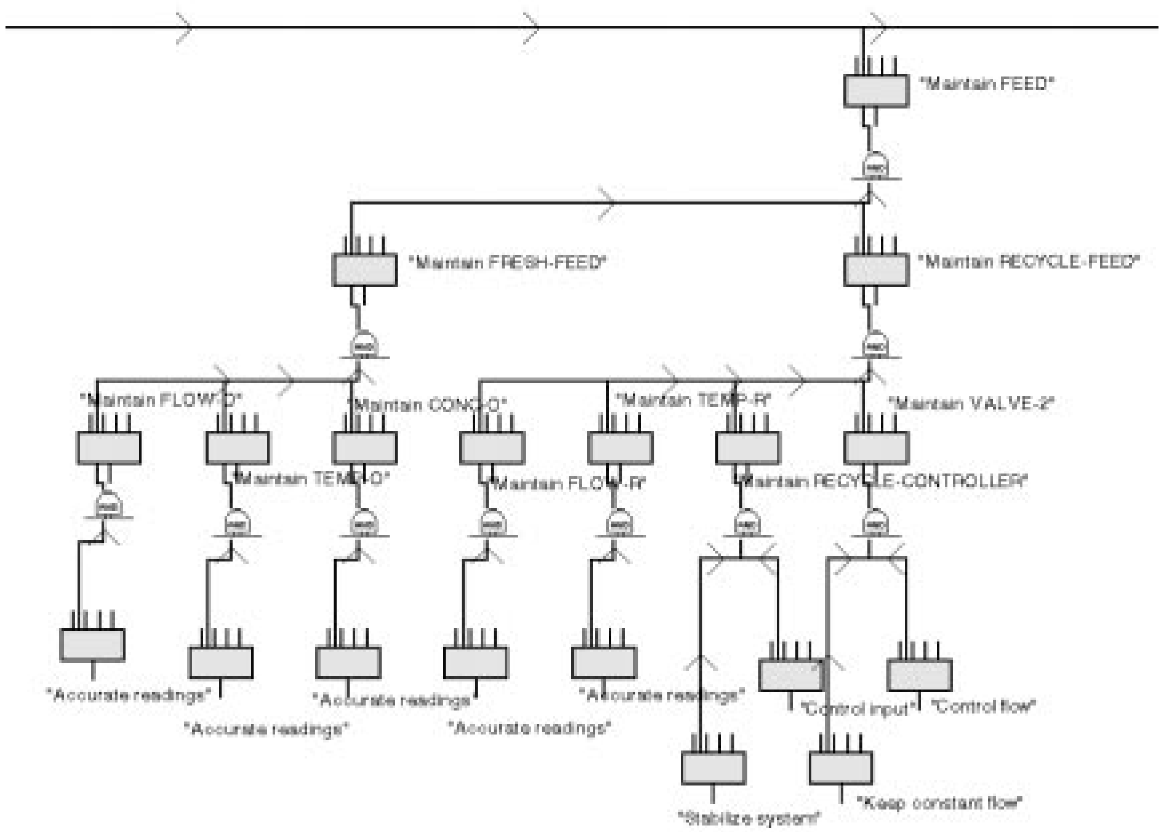

Figure 5.8 Kramer Case Study-feed system goals

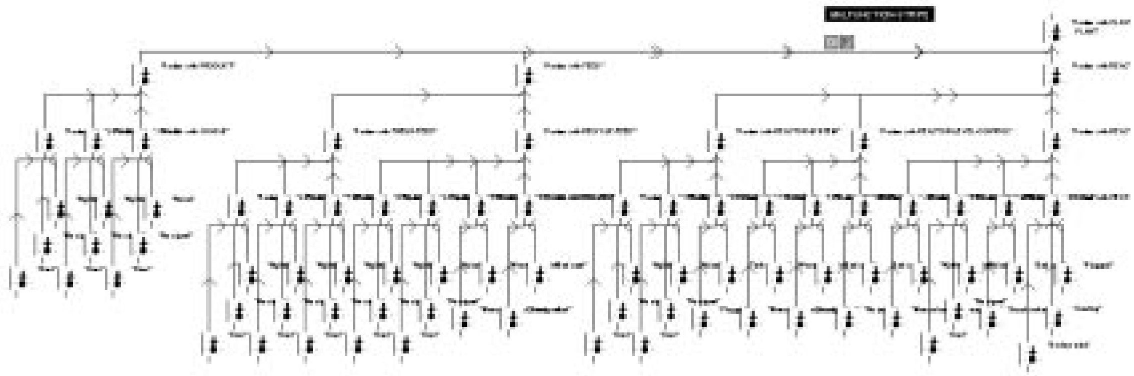

Figure 5.9 Kramer Case Study-malfunction stripe

malfunction of the temperature controller, a problem with the coolant feed flow, a malfunction of the coolant control valve, or a fault involving the heat exchanger. The base-level malfunctions of the heat exchanger include such malfunctions as scaling, plugging, and broken tubes.

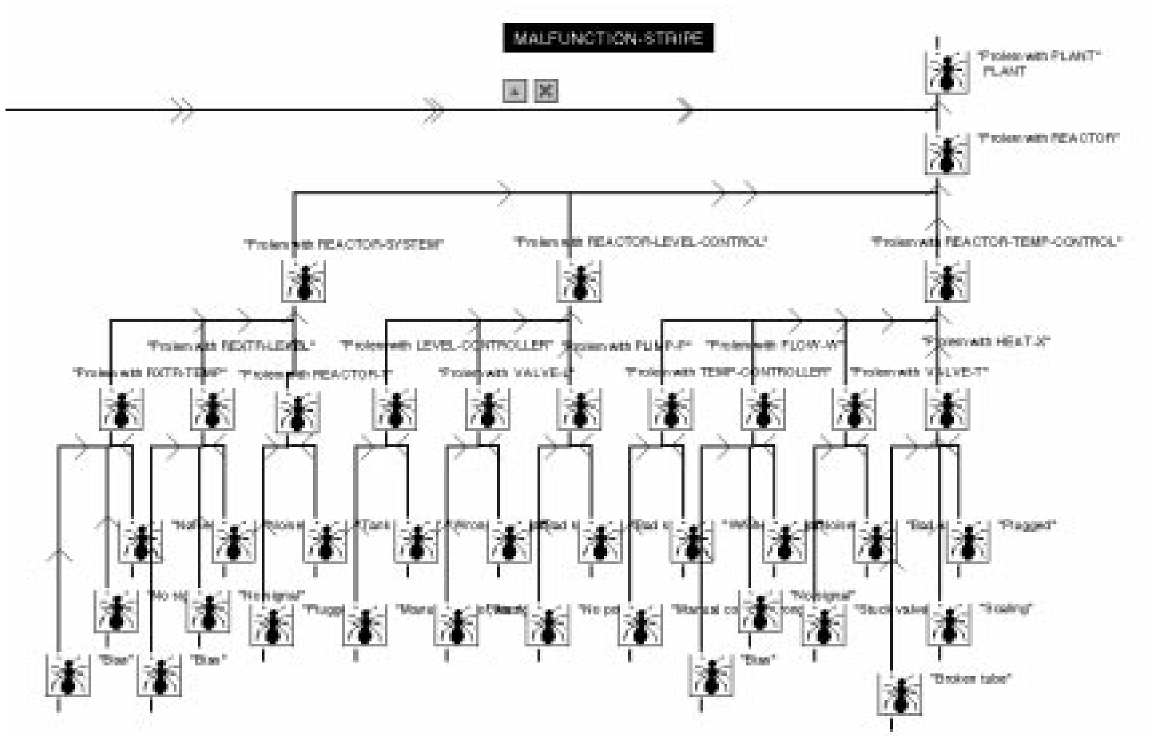

Figure 5.10 Kramer Case Study-reactor system malfunctions

6. DISCUSSION AND CONCLUSIONS

Blackboard systems are a viable way to attack the problem of fault diagnosis in complex chemical plants. The components of a blackboard system have been discussed. The blackboard framework, the knowledge experts, and the process model must be understood and integrated to develop a working system. The process model is the basis for blackboard systems. Accurate process representations must be easily developed.

The problem of model development and automation has been addressed. Human development of goal–trees and malfunction–trees for complex systems can be difficult, inaccurate, and time consuming. Real systems may have thousands of interrelated goals, malfunctions, and units. Element by element incremental development by a human can take months and still contain many errors. A method for developing process models quickly and effectively from process flow sheet information has been proposed. This method relies on combining low–level qualitative models of unit operations with developer information of the plant hierarchy to create an overall representation of a chemical process. This method greatly speeds the modeling of complex systems and helps to ensure that human error is minimized. The physical representation is used along human input to develop the logical dependencies of the process. The goal-tree and malfunction–tree models are created from the logical representation and the low–level models of unit operations from the equipment library.

A fairly simple case study was presented. In this example, a flow process involving a exothermic stirred tank reactor was analyzed. The development tools allow for easy creation of the different models. The goal and malfunction representations capture the basic information needed for diagnosis.

7. FUTURE WORK

There are many different problems associated with blackboard-based diagnosis which must still be addressed. These problems can be associated with the overall blackboard system, the knowledge experts, and the blackboard model structure.

Blackboards are often based on heuristic methods for arriving at conclusions. The scheduler and information aggregator are often implemented as rule-based systems. Because of this, there are often few guarantees that a result will be found or that the result produced is correct. Developing bounds on the solution space of each knowledge expert would help this problem. Using mathematical formulations for scheduling actions and combining information may lead to performance guarantees.

The current blackboard application only looks at historical information to ascertain what faults may be present. Using predictive experts which could propose expected trends in the process could help diagnose when there is not enough information available for normal experts to find a root cause. Prediction could also be helpful for control. A control input may be optimal for tracking, but a different input may be needed for fault diagnosis or fault management. This could be considered an identification problem. Tradeoff between optimal control and fault identification become evident. The capabilities of the blackboard system could become very beneficial.

Quantitative models can be vastly improved. Linear horizon-based estimation can benefit from a better problem formulation. Using multiple linear models to represent a nonlinear and uncertain system can help produce accurate results. The nonlinear formulation of the horizon-based estimation scheme is dependent upon efficient solution methods. Faster algorithms can make the nonlinear horizon–based estimation problem tractable. Other types of nonlinear observers could also be developed.

As far as model development, some issues are still unanswered. The robustness of models is a definite concern. The following items are related to model robustness:

- Slight modifications to the P&ID may cause significant changes in the PRM. Adding a new valve or pipe could restructure the process completely

- Completely recreating models may be difficult or impossible because of user influence. Standard rules for model development could be developed

- Changes in operating conditions can also change the structure of the models. At different conditions, the importance of parts of the process may change

- In some cases, a high level goal can be met even though low level goals are being violated. In the same way, there may not be evidence of a high level malfunction if two low level malfunctions interact and cancel the results of each other.

These different problems still need to be dealt with in order for more useful models to be developed. One approach which may be useful is to keep multiple representations in the database. With tools for automating model development, models for many operating points can be created. Knowing when to switch between models could be a problem similar to gain scheduling.

Loop structures also lead to problems in both cyclic and acyclic loops. It is difficult to create a useful representation of a cyclic process. Usually, the cycle must be broken at some point. When this is done, useful information about causal relationships are lost. Acyclic graphs with loops have the advantage of no feedback signal. This is still a difficult situation to deal with. A general method for dealing with loop structures is needed.

These items are interesting areas that still need development. The experience gained from working with blackboard systems, knowledge experts, and process models will be useful as new aspects of fault diagnosis are approached.

LIST OF REFERENCES

LIST OF REFERENCES

[1] K. Arzen. Using multi-view objects for structuring plant databases. *Intelligent Systems Engineering*, 1993.

[2] A. Crespo, V. Botti, F. Barber, D. Gallardo, and E. Onaindia. A temporal blackboard for real-time process control. *Engineering Applications of Artificial Intelligence*, 7(3):255–266, 1994.

[3] R. S. Englemore, A. J. Morgan, and H. P. Nii. *Blackboard Systems*, chapter Hearsay-II, pages 25–29. Addison Wesley, 1988.

[4] L. D. Erman, F. Heyes-Roth, V. R. Lesser, and D. R. Reddy. *Blackboard Systems*, chapter The Hearsay-II Speech-Understanding System: Integrating Knowledge to resolve Uncertainty, pages 31–86. Addison Wesley, 1988.

[5] P. M. Frank. Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy- A Survey and Some New Results. IFAC/IMAC/IFORS, Pergamon Press, July 1989.

[6] M. Iri, K. Aoki, E. O'Shima, and H. Matsyama. An algorithm for diagnosis of system failures in the chemical process. *Computers and Chemical Engineering*, 3:489–493, 1979.

[7] R. Iserman. Model based fault detection and diagnosis methods. *Proceedings of American Control Conference*, pages 1605–1609, June 1995.

[8] N. Iyer and A. Farell. Design of a stable adaptive nonlinear observer for an exothermic stirred tank reactor. *Computers in Chemical Engineering*, 1996.

[9] P. R. Kersten and A. C. Kak. A tutorial on lisp object-oriented programming for blackboard computation (sloving the radar tracking problem). *International Journal of Intelligent Systems*, 8:617–669, 1993.

[10] M. A. Kramer and Jr. B. L. Palowich. A rule-based approach to fault diagnosis using the signed directed graph. *AIChE Journal*, 33(7):1067–1078, 1987.

[11] M. J. Liebman, T. F. Edgar, and L. S. Lasdon. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers in Chemical Engineering*, 16(10):963–987, 1992.

[12] Dave Mandelkern. Ingredients for real-time expert systems. *ISA*, pages 1475–1481, 1993.

[13] R.C McFarlane, R.C. Reineman, J. E. Bartee, and C. Georgakis. Dynamic Simulator for a Model IV Fluid Catalytic Cracking Unit. *Computers Chem. Engng*, 17(3):275 – 300, 1993.

[14] D. Mylaraswamy. *DKIT: A Blackboard-based, Distributed, Multi-Expert Environment for Abnormal Situation Management.* PhD thesis, Purdue University, 1996.

[15] D. S. Nam, C. Han, C. Jeong, and E. S. Yoon. Automatic construction of extended symptom-fault associations from the signed digraph. *Computers in Chemical Engineering*, 20:5605–5610, 1996.

[16] H. P. Nii. Blackboard application systems, blackboard systems from a knowledge engineering perspective. *AI Magazine*, pages 82–106, August 1986.

[17] I. Nimmo. Adequately address abnormal operations. *Chemical Engineering Progress*, pages 36–45, September 1995.

[18] A. Y. Shumesky. Observer-based on-line parameter estimation in nonlinear systems. In *13th Triennial World Congress*. IFAC, 1996.

[19] M. L. Tyler, K. Asano, and M. Morari. Application of moving horizon estimation based fault detection to cold tandem steel mill. Preprint.