16

A Framework for Knowledge–Based Diagnosis in Process Operations

P.R. Prasad & J.F. Davis*
Department of Chemical Engineering
and Laboratory for AI Research
140 W. 19th Avenue
The Ohio State University
Columbus, Ohio 43210.
e–mail : davis@kcgl1.eng.ohio–state.edu

Abstract

Process control computers can be extended to include automated diagnosis through the integrated use of Artificial Intelligence techniques. Diagnosis, a complex reasoning activity, is first characterized and then decomposed into its constituent information–processing tasks (IPTs). Each IPT is described in terms of input, output, knowledge representation and inferencing strategy. Examples drawn from practical implementations of knowledge–based systems are used to illustrate each IPT. How these IPTs interact and are integrated to form a framework for constructing knowledge–based diagnostic systems is described. It is shown how this task–based approach provides a natural basis for pulling together a variety of technologies, such as neural networks, statistical methods, conventional numerical methods and knowledge–based systems, into a comprehensive system for automated diagnosis.

1. INTRODUCTION

In response to demands for increased production levels and more stringent product quality specifications, the intensity and complexity of process operations have been rising steadily. To alleviate the operating requirements associated with these demands, plants have increasingly relied upon automatic control systems. It is well–known that control systems — traditional analog devices or the more recent distributed control systems (DCSs) — are effective for automatically making local process changes within some range to establish or maintain operating conditions in the face of well–defined disturbances.

Even with highly sophisticated DCSs, it is, however, clear that processes are subject to equipment failures and unexpected changes in conditions that often result in off–spec

*To whom all correspondence should be addressed

product, reduced production or unsafe situations. Further, these equipment failures and process abnormalities, if left uncorrected, can induce additional failures in related equipment. These process and equipment "malfunctions" lead to significantly higher operating costs and/or reduced profits.

Substantial benefits, with respect to improved plant operation, can thus be obtained by expanding the scope of process control computers to aid operators in diagnosing equipment failures and process abnormalities. The advantages of advisory systems are especially apparent if designed for early fault detection and identification.

Conventional control largely deals with control actions in the form of manipulated parameter settings (such as control valve position) that are computed directly using a specified control algorithm, like PI, PID or IMC [1]. The control algorithm is typically executed with the "diagnostic" goals of the controller essentially fixed. Fault detection and isolation (FDI) is an extension of control that leads to "control actions" in response to faults that can be characterized with a high degree of certainty. Furthermore, on–line FDI is addressed with analytical redundancy where data from a plant are compared to expected values generated by a mathematical model [32, 33].

Advances in artificial intelligence (AI) now offer the potential for new approaches to extending control systems where the system adapts its "diagnostic" objectives in response to changing conditions. In addition, these techniques provide the capability of reasoning in uncertain and unstructured environments in which adequate mathematical models may not exist. For diagnosis these knowledge–driven techniques involve the interpretation of sensor readings and other process observations, detection of abnormal operating conditions, generation and testing of malfunction hypotheses that can explain the observed symptoms and finally resolution of any interactions between hypotheses. Fundamentally, diagnosis is viewed as a decision–making activity that is not numeric in nature. While the governing elements are symbolic, numeric computations still play an important role of providing certain kinds of information for making decisions and drawing diagnostic conclusions.

As the role of process control computers expands beyond the numeric–algorithmic activities of conventional control and into the reasoning level activities associated with diagnosis, there is a need for structured symbolic decision–making methods. With the advent of AI technologies, such as knowledge–based systems (KBS), neural networks and fuzzy logic, significant advances have been made in extending the capabilities of plant control systems with automated diagnosis.

AI–based techniques have been applied throughout the process plant control infrastructure — from the low–end "execution level" to the high–end "supervision and planning level" [2]. The execution level includes the use of techniques such as "fuzzy control" or "neural control" for closed–loop control. An example is the fuzzy control of autoclave–cured composites reported by Wu and Joseph [3]. Fuzzy logic is used to express and manipulate ill–defined qualitative terms like "large", "small", "very small", etc. in a well–defined mathematical way to mimic the human operator's manual control strategy. Qualitative rules are used to express how the control signal should be chosen in different situations. "Neural control" refers to the use of neural networks to develop process models which are then used to implement robust, model–predictive controllers [4]. The high–end "supervisory level", on the other hand, seeks to extend the range of conventional control algorithms through the use of KBSs for tuning controllers, performing fault diagnosis and on–line reconfiguration of control systems. Tzouanas, et al. [5] have reported using a KBS to support the deployment of multivariable control systems in cases of controller saturation, sensor failure and reconfiguration of SISO control loops. Other examples have been reported by Basila, et al. [6] and Astrom, et al. [7]. KBSs have also been used to determine the best control system configuration and/or select the best control algorithm given the operating constraints. Birky, et al. [8] have used a KBS to assist in the design of control configurations for a distillation column. A review of other KBSs for design assistance is given in James [9].

The focus of this chapter is knowledge–based diagnosis as a supervisory level extension to the execution level techniques represented by conventional numerical control and the more recent intelligent control techniques. Our approach to knowledge–based diagnosis is grounded in the generic task theory originally proposed by Chandrasekaran [10,11]. The aim of this theory is to identify information–processing tasks as "building blocks" of reasoning strategies which are both generic and widely useful. Furthermore, it is recognized that complex reasoning activities rely upon different methods and even different technologies. These are reflected explicitly as mechanisms for accomplishing the individual tasks. An automated system for the diagnostic activity is thus computationally described as the integration of a small set of well–defined tasks. The integration of the tasks to form a framework, thus provides a natural basis for pulling a variety of technologies together in a comprehensive system for automated diagnosis.

In the following, we present a characterization of diagnosis, followed by a decomposition of the activity into its constituent tasks or sub–problems. How these tasks interact and are integrated to form a framework for constructing knowledge–based diagnostic systems is then described. This is followed by detailed descriptions of the tasks and the use of different techniques — AI–based and traditional — to accomplish their respective problem–solving goals. Examples drawn from KBSs implemented for industrial process operations are used to illustrate the integration of the various problem–solving techniques in forming the complete diagnostic framework.

2. CHARACTERIZATION OF DIAGNOSIS

Fault diagnosis can be broadly characterized as a separate reasoning activity which sequentially follows abnormality detection. As a reasoning activity that is *triggered* by detection, fault diagnosis can be more specifically characterized as the activity of mapping from symptoms to a conclusion comprised of one or more malfunction hypotheses. These malfunction hypotheses explain the symptoms in sufficient detail to take corrective action. In process operations, the symptoms include both abnormal and normal performance conditions as indicated by various process sensors, alarms, operator observations and laboratory analyses.

A variety of approaches to building diagnostic KBSs have been recently developed. MODEX2 [12] is an approach that integrates behavioral knowledge organized explicitly for diagnosis with more fundamental simulation knowledge that allows for system behaviors to be generated during run–time. Petti, et al. [13] advocated the use of numeric plant models to arrive at diagnostic conclusions. Similarly, Grantham and Ungar [14] have demonstrated the adaptation of models to account for new operating states in the diagnosis of novel faults. Finch and Kramer [15] have described a strategy for diagnostic focus, where knowledge about the functionality of process equipment is used to form diagnostically useful abstractions. Calandranis, et al. [16] have described a KBS approach representing diagnostic knowledge in tables. As an alternative to knowledge–based approaches, neural networks have been used by Venkatasubramanian, et al. [17] for both fault detection and diagnosis. Kramer and Leonard [18] have critiqued the use of backpropagation neural networks for this purpose.

While the common objective of all these systems is diagnosis, varying levels of emphasis have been placed on different aspects of problem solving. Some focus on alarm analysis and the rapid generation of corrective actions to avert safety problems [19]. These systems are designed to respond to critical abnormal behaviors. Time available for suggesting corrective actions is typically short and the root cause of the observed behavior may not be known or identified. On the other hand, there are some diagnostic systems where the emphasis is on resolving symptomatic data as they appear in time [16,20,21]. Due to the temporal aspect associated with the data, truth maintenance and generation of consistent malfunction hypotheses are important in spite of conditions such as out of order alarms and inverse response. The majority of reported systems consider symptomatic data in the form of a 'snap–shot' within a window of time. Successive snap–shots of data are used to perform real–time diagnosis.

We argue there are distinct differences between advisories for safety–related and root cause diagnoses with resulting differences in the respective knowledge–based frameworks. Root cause problems share the following characteristics [22]:

(1) The aim of diagnosis is to identify the root cause malfunctions that affect production and product quality. Early detection of problems before they develop into critical behaviors and prevention of continued adverse economic operating conditions are thus the important motivations. Root cause diagnosis is contrasted with rapid responses to abnormal behaviors that jeopardize the safety of the plant.
(2) Root cause diagnosis, therefore, implies stabilized malfunctioning operation such that there is usually sufficient time during diagnosis for performing detailed tests. We use the term 'pseudo steady state' in recognition of the fact that some of the symptoms used during diagnosis may not be truly at steady state, i.e. variables could be increasing, decreasing or oscillating.
(3) The reasoning process is usually more deliberative, and the search through the space of malfunction hypotheses is much more systematic and thorough. Additional tests are used to resolve hypotheses in detail.
(4) The corrective action that follows root cause diagnosis is usually to fix the primary cause(s), with the intention of preventing continued deterioration or recurrence. This is contrasted with rapid response actions which immediately counteract the effects of a critical abnormality. The aim of root cause diagnosis is to maintain the long term operating objectives of the plant rather than avert immediate short–term crises.

We, therefore, recognize that different "real–time" situations exist and that root cause diagnosis is one that is associated with behaviors that have relatively longer time constants. While "real–time" is a constraint, there is typically sufficient time to investigate hypotheses in some detail and to request additional tests and/or collect additional data to resolve a root cause.

Root cause diagnosis encompasses not only equipment failures but also other causes related to changes in operating conditions [23]. The goal of fault diagnosis is the identification of hardware malfunctions including breakdown and deterioration. Such malfunctions are usually associated with unit operations (e.g. leaks, blockages of pipes, mechanical malfunctions) or with control loop components (e.g. sensors, actuators) [24]. Fault diagnosis also involves the identification of deviating operating parameters. Examples include reduced heat transfer coefficients, low activity of catalysts and contaminated fermenters [24]. Although closely related, the distinction between these types of malfunctions is important as the abnormal operating parameters provide a starting point for determining corrective actions to be implemented, if no hardware malfunction is identified.

3. ANALYSIS OF THE DIAGNOSTIC ACTIVITY

Consider a simplified chemical plant that consists of a feed pre–heater, a reactor and a distillation column, as shown in Fig. 3.1. Now consider a scenario where the flow rate

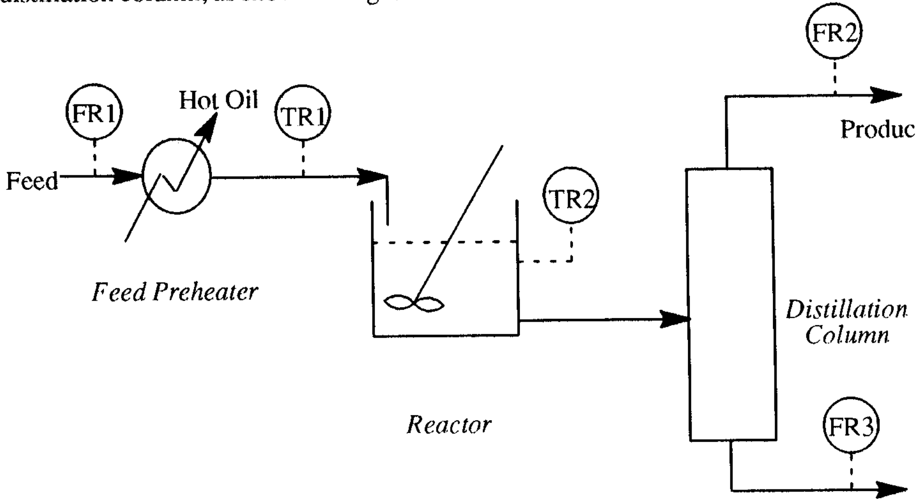

Figure 3.1 : A Simplified Chemical Plant Flowsheet

of hot oil through the heat exchanger is unexpectedly reduced due to a faulty valve. This low flow rate then causes insufficient heat to be transferred to the raw material in the feed preheater, which in turn results in a lower than normal temperature of feed to the reactor. The low feed temperature causes a lower reaction rate, which then results in a smaller amount of product. Let us now analyze how this diagnostic conclusion is reached using the symptoms provided by the sensors on the unit. It should be pointed out that though there are several sensors in the plant, only the product flow rate (FR2) and product quality (from laboratory reports) are tracked continuously.

Diagnosis is initiated by a decrease in product flow rate indicated by FR2. Using the flow rates measured by FR1, FR2 and FR3, it is found that the material balance around the plant, and hence around the distillation column, closes. Also, the product quality is observed to be within specifications. These observations lead to the conclusion that there is nothing wrong with the operation of the distillation column. The focus of problem–solving attention is then shifted to the reaction and feed pre–heat system. The observation that the temperature at the outlet of the feed preheater (TR1), is lower than normal narrows the possible malfunctions to the heat exchanger segment of the process. Further consideration of the heat exchanger identifies the low flow rate of hot oil through the heat exchanger, which in turn leads to the discovery of the faulty valve. "Faulty valve" is a malfunction hypothesis which explains the observed plant behavior and is of sufficient detail for corrective action (such as 'fix the valve') to be taken.

An analysis of this simple, but characteristically typical diagnosis reveals several distinct sub–problems:

(1) We note a distinct progression of hypotheses examined from general to specific. The first hypothesis considered was a malfunctioning distillation column. On ruling this out, the other two major systems, reaction and feed pre–heat were then considered. Once a problem with the feed preheater was established, the objective became one of finding the specific fault in the feed preheater. To this end, the components of the feed preheater

were investigated and a fault in the valve was detected. Thus, the goal achieved is that of generating malfunction hypotheses for evaluation. Only those systems that are identified to be malfunctioning are explored in further detail by examining sub-systems or components. The generation of malfunction hypotheses for evaluation, therefore, involves a search of the hypothesis space with different hypotheses pursued under different conditions.

(2) For each malfunction hypothesis examined there is a need to establish with some degree of certainty if it is true or false. For example, the normality of product quality and the material balance closure are used as symptomatic features to rule out any malfunction in the distillation column; while low temperature is used to establish that there is a malfunction associated with the feed preheater. Thus, each hypothesis is associated with a set of features which support or reject it. The evaluation of each hypothesis is carried out locally (at each hypothesis) by comparing features associated with the hypothesis with the observed symptoms. The basic mechanism for this is structured pattern matching.

(3) The symptoms used in problem solving are not always direct sensor readings. For example, the sensors do not directly provide information about the fact that the product flow rate is low or that the temperature is low. Instead, the sensors only indicate numeric values which must be interpreted as low, high or normal for use in diagnosis. Similarly, the material balance closure is not indicated directly on any instrument. Rather, it is calculated using several sensor readings. This task of providing qualitative interpretations of numeric sensor data is yet another sub-problem.

The above analysis demonstrates that the activity of diagnosis can be decomposed into several problem-solving tasks that are called *Information Processing Tasks* (IPTs). Each IPT has a specific objective or goal that is defined by a diagnostic sub-problem. For example, the goal of the first sub-problem is to systematically generate malfunction hypotheses for consideration. An IPT defines the mechanism for accomplishing the goal. Each sub-problem described above, can be similarly characterized in terms of its goals, input, output and problem-solving mechanism. The overall objective of diagnosis is achieved through the integrated efforts of the IPTs, each one performing its designated function using specific kinds of knowledge organized in a specific way.

The decomposition of a complex reasoning activity into its constituent IPTs was first proposed by Chandrasekaran [10, 11]. He postulated that a complex activity can be computationally and generically described as the integration of a small set of well-defined IPTs. These IPTs, thus, form the primitive building blocks for complex activities like diagnosis.

The advantages of the task-oriented view are many [25, 26, 27]. First, it encourages analysis of a problem at the level of IPTs, thus greatly facilitating the implementation phase. Secondly, the concept of activities and primitive IPTs leads to modular systems, consisting of software modules corresponding to each known task. With this concept, building a KBS for a new application involves first decomposing the activity into component tasks and then inserting the application knowledge into each of the IPT software modules as specifically required.

4. IPTS FOR CHEMICAL PROCESS PLANT DIAGNOSIS

While the analysis of the diagnostic scenario described in the previous section brings out three IPTs, our research in the chemical process diagnosis has led to the identification of six [23]. Table 1 gives an overview of these different IPTs. The interaction of these

IPTs to form a complete framework for diagnosis is illustrated in Fig. 4.1. In the following sections we describe the characteristics of each task and discuss how they are coordinated with one another to form the development framework for a comprehensive diagnostic system.

Table 1 : Description of IPTs identified in the process plant domain

Name of the IPT	Description of the task
Hierarchical Classification (HC)	Given a set of symptomatic features systematically generate feasible malfunction hypotheses and efficiently eliminate infeasible hypotheses
Structured Pattern Matching (SPM)	Given a malfunction hypothesis (generated by HC above) and a set of symptomatic features relating to the hypothesis, establish or reject it with a certain degree of certainty
Qualitative Interpretation(QI)	Map from numeric data generated by sensor into diagnostically useful interpretations
Sensor Validation	Identify sensor errors and provide correct values
Hypothesis Assembly	Generate the best explanation for sets of product quality changes in terms of deviations in operating parameters
Diagnostically Focused Simulation (DFS)	Resolve causally related malfunction hypotheses using simulation

4.1 The Core Diagnostic Tasks

4.1.1 Hierarchical Classification (HC)

HC is the heart of the diagnostic activity. The objective of this IPT is to map symptomatic data into one or more malfunction hypotheses defined in sufficient detail that they are recognized as root causes. Given the potentially large number of possible malfunctions, HC offers a mechanism for efficiently searching through the space of malfunction hypotheses and robustly arriving at the correct diagnostic conclusion. HC addresses this combinatorial issue by compiling diagnostic knowledge such that diagnosis remains highly focused and the space of likely malfunctions is narrowed rapidly. The compiled knowledge is expressed as a hierarchy of malfunction hypotheses organized from general to detailed. An example of this hierarchical arrangement of hypotheses for a fluidized catalytic cracking unit (FCC) is shown in Fig. 4.2 [22].

At the top level, general malfunction hypotheses reflect a general decomposition of the plant in terms of the various functional systems/subsystems or general fault categories. As shown for the FCC, there are three major functional systems — feed, reactor–regenerator and separation — and one fault category — catalyst problems — that form the hypotheses (nodes) of the hierarchy at the first level of decomposition. This decomposition strategy of identifying functional sub–systems and fault categories in increasing detail, is continued at each level in the hierarchy.

For the functional decomposition an arc indicates a relation described as "is a sub–system of". For the fault decomposition the arcs are interpreted as "is caused by". At the lower levels, the nodes reflect more specific malfunction hypotheses, which take the

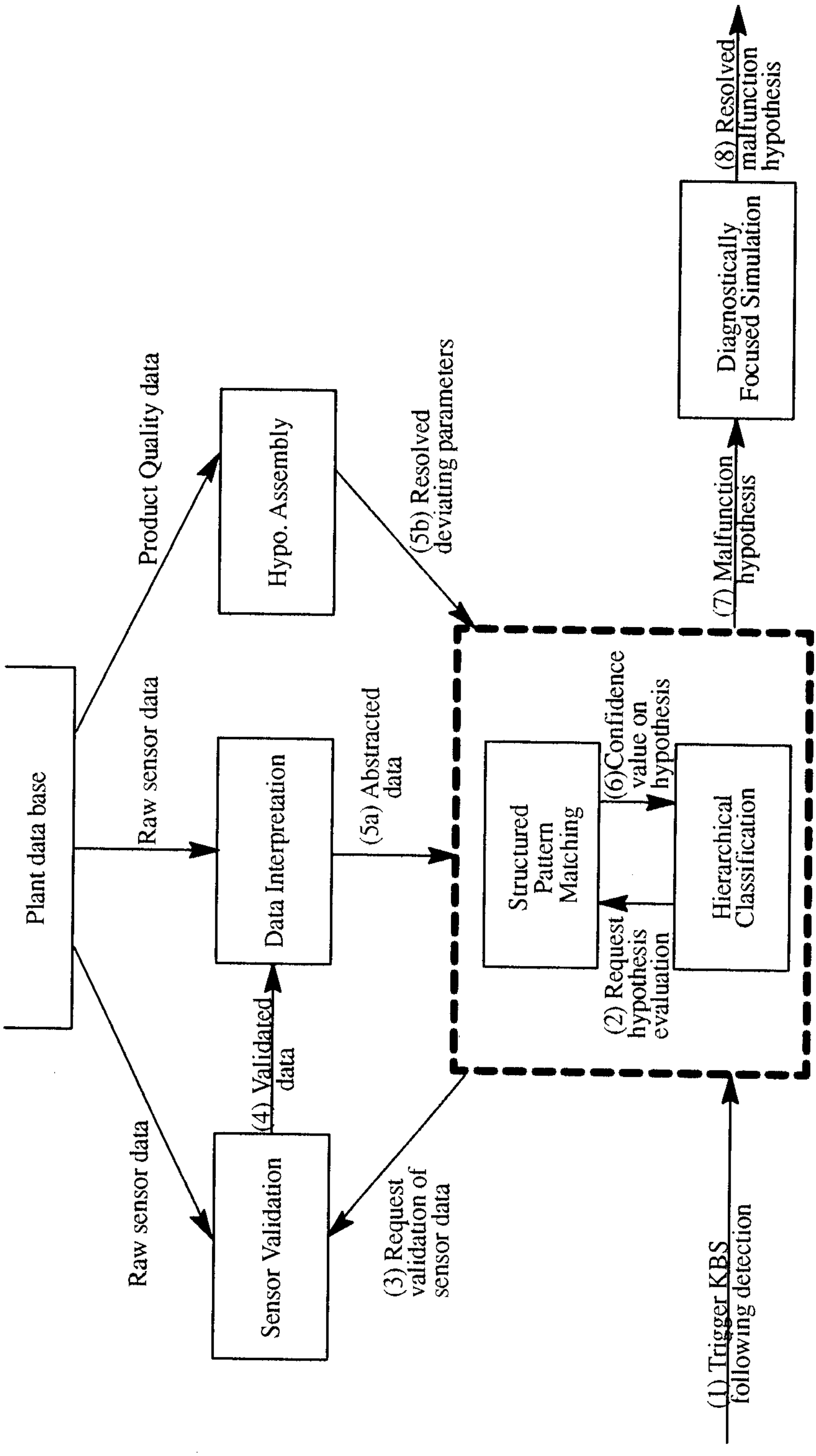

Fig. 4.1 : Components of Framework for KB Diagnosis

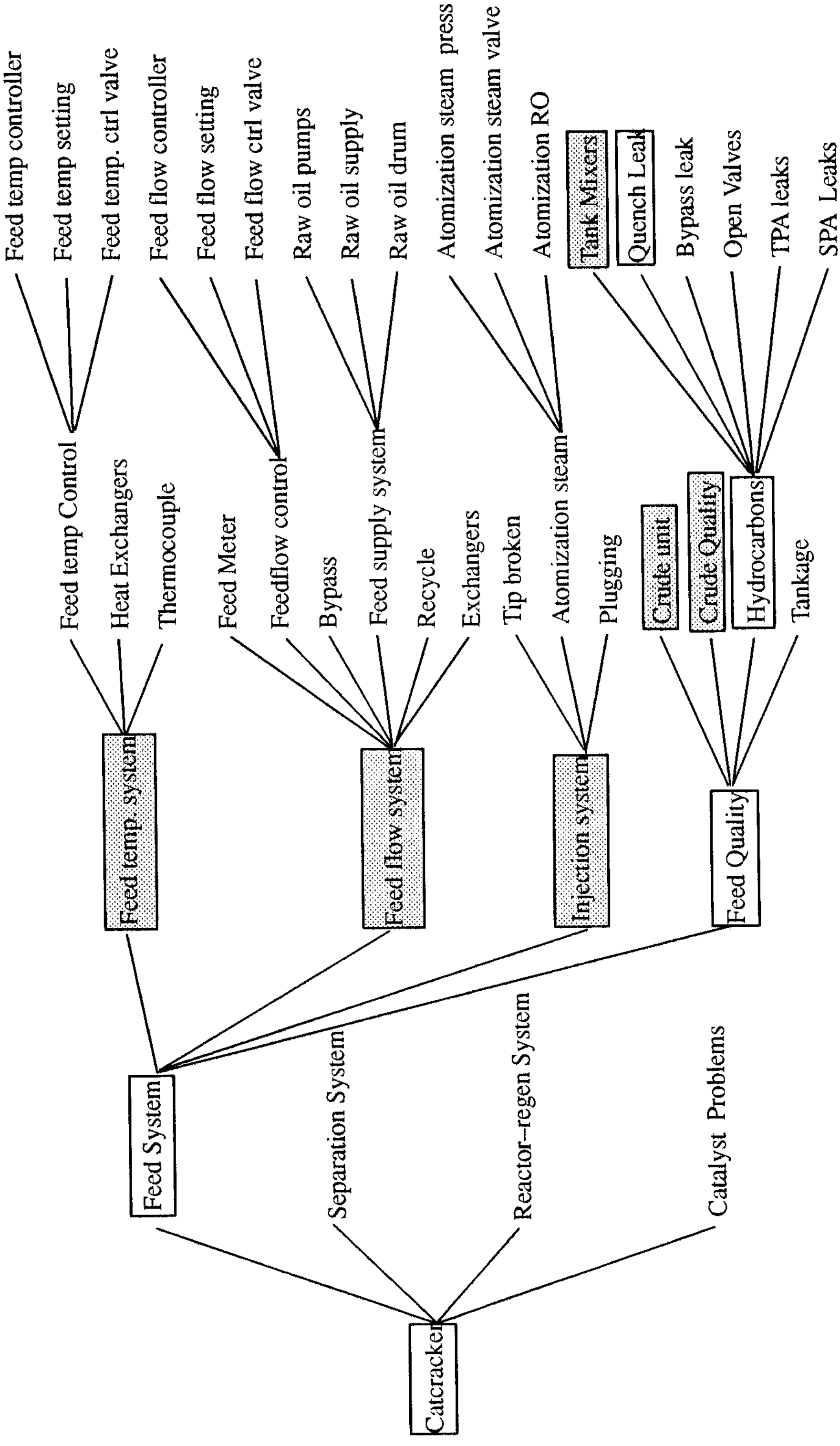

Fig. 4.2 : Hierarchy of malfunction hypotheses for Feed System of FCC

form of malfunctioning equipment items, specific modes of failure, improper operating parameter settings or improperly executed procedures. The hierarchy, thus captures various levels of process detail by a mixture of malfunction categories generated from functional, fault, mode of failure and structural decompositions of the process. As exemplified for the FCC Feed System in Fig. 4.2, these decomposition methods result in a problem solving structure particularly suited for diagnostic search.

Associated with this hierarchical knowledge organization is a problem solving strategy that begins at the top level of the hierarchy. For a given top level hypothesis, specific symptomatic information is used to evaluate the hypothesis (node) as either 'established' or 'rejected', with some measure of confidence. If the malfunction hypothesis is established, then the focus of problem solving shifts to its children. When a hypothesis is rejected, indicating that that segment of the plant is operating properly, all sub–hypotheses below are also rejected. This process of hypothesis evaluation is recursively applied at each level of the hierarchy until one or more tip–level hypotheses is established. This inferencing strategy is referred to as "establish–refine" [11, 23].

"Establish–refine", as applied to the hierarchy in Fig. 4.2, is illustrated by drawing boxes around the nodes. Hypotheses evaluated are drawn with boxes. Hypotheses rejected are indicated with shaded boxes, while an unshaded box represents a hypothesis that was established. All other hypotheses are pruned because a parent is rejected. The power of the establish–refine strategy in focusing the diagnostic search becomes readily evident by comparing the relatively small number of boxed nodes with the total number of nodes in the hierarchy.

If the establish–refine strategy is applied in a systematic manner and driven by an orderly consideration of hypotheses, we refer to the overall strategy as "malfunction–driven", i.e. the order the hypotheses appear at each level in the hierarchy drives the order in which hypotheses are considered. While the malfunction–driven, establish–refine strategy is the primary strategy used in HC, other problem solving strategies may be locally superimposed to augment efficiency in problem solving. Ramesh, et al. [11] have described several variations of establish–refine for the FCC diagnosis. The two most common variations are symptom–driven refinement and causally–dependent invocation. Symptom–driven refinement defines a situation in which the existence of a specific symptom is strongly indicative of a particular malfunction hypothesis and, as a result, systematic search is unnecessary. In terms of the malfunction hierarchy symptom–driven refinement results in the inference jumping across the hierarchy to a specific node. Causally–dependent invocation deals with causally–related hypotheses. Given the presence of certain symptoms, the establishment of one hypothesis results in the consideration of another hypothesis elsewhere in the hierarchy. The use of these strategies is required for complex operations when there exist multiple relationships among hypotheses and symptoms under various operating conditions.

4.1.2 Structured Pattern Matching (SPM)

SPM refers to the task of determining the established or rejected status of a malfunction hypothesis generated by HC. In this task, symptoms are matched against pre–defined patterns that reflect the local relations existing between specific symptomatic features and a given hypothesis. Compiling knowledge in this form eliminates the need for generating symptom patterns at run–time using some form of simulation. The basis for these patterns is, however, the input/output process behavior of that portion of the operation represented by the hypothesis.

Also, associated with each of these feature patterns is knowledge about the degree of certainty in the establishment or rejection of the hypothesis, given the presence or absence of symptoms. When symptomatic information matches a pattern of features, the appropriate confidence is assigned to the malfunction hypothesis. The confidence may itself be a simple pre-assigned value associated with each individual pattern or may involve more detailed computations using probability theory or other measures of uncertainty.

Matching symptoms to pre-defined feature patterns is a direct form of pattern matching. As an example, let us consider the evaluation of the hypothesis *Feed system* in the FCC hierarchy (Fig. 4.2). Table 2a lists the feature patterns associated with this hypothesis in a tabular form. As indicated, evaluation of this hypothesis is based on the values of the features *abnormality of feed data* and *upstream conditions*. *Feed system* is evaluated as 'established' if *upstream conditions* are 'not normal' and the *abnormality of feed data* is 'established' (Column 2 of Table 2a). *Abnormality of feed data* itself gets 'established' or 'rejected' based on the values of *state of pre-heat* and *material balance*, as indicated in Table 2b. For example, if the *state of pre-heat* is 'normal' and the *material balance* is 'normal' then *abnormality of feed data* is 'rejected' (Column 5 of Table 2b).

The '?' appearing in Table 2b indicates 'unknown'. The degree of certainty is reflected in the use of the following qualitative values: 'established', 'very likely', 'likely', 'unknown', 'unlikely', 'very unlikely', or 'rejected'. Each hypothesis (node) in the malfunction hierarchy can be associated with one or more such pattern matchers and the matchers themselves can be organized hierarchically. Fig. 4.3 illustrates the hierarchy of feature patterns used in this example.

Table 2a: Pattern matching table for 'Feed system'

Evaluation of 'Feed system'	Established	Very likely	Rejected
Abnormality of feed data	Established	Very likely	Rejected
Upstream conditions	Not normal	Not normal	Normal

Table 2b: Pattern matching for 'Abnormality of feed data'

Evaluation of 'Abnormality of feed data'	Established	Very likely	Likely	Rejected
State of pre-heat	Cold	?	?	Normal
Material balance	?	Very high – or very low	Not normal	Normal

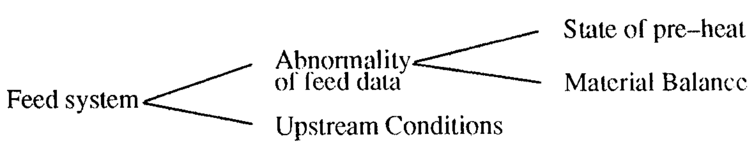

Fig. 4.3 : Hierarchy of feature patterns

As pointed out previously, the HC and SPM tasks act in very close conjunction. The generation of hypotheses is accomplished by the HC task, and the evaluation of the hypotheses is carried out by the SPM task. This very close integration between these two tasks is indicated by the heavy grid box in Fig. 4.1.

4.1.3 Diagnostically Focused Simulation (DFS):

Referring once again to the diagnostic framework shown in Figure 4.1, there is a task sequentially following HC/SPM which we refer to as DFS. The purpose for DFS grows from the recognition that HC and SPM tasks are unable by themselves to resolve multiple interacting malfunction hypotheses [28]. It is well–known that process malfunctions can interact via stream integration, control loops and sink–source relationships. These kinds of interactions can lead to scenarios such as an equipment malfunction in one part of the process causing an equipment item in a far removed part of the process to also malfunction.

Knowledge organized using functional systems/subsystems, fault categories, etc. in HC is particularly effective for robustly identifying single malfunctions and independent multiple malfunctions. For these situations, no further problem solving is necessary. In the case of multiple, interacting malfunctions, however, resolution requires consideration of the process topology and reasoning along the structural paths that potentially link two or more malfunctions. Since HC does not explicitly reflect process structure, the objective of DFS then is to provide the means by which reasoning about structure can be brought to the diagnostic process. As a task, DFS is responsible for identifying the structural paths which could link multiple malfunction hypotheses, propagating causal effects along this path, and then establishing or rejecting interactions based on a comparison of the simulated process behavior with that observed.

The input information to the DFS task is provided by the diagnostic results from the HC and SPM tasks. At the conclusion of HC, there is a diagnostic assessment in the form of a hierarchy of established and rejected malfunction hypotheses as illustrated in Fig. 4.4.

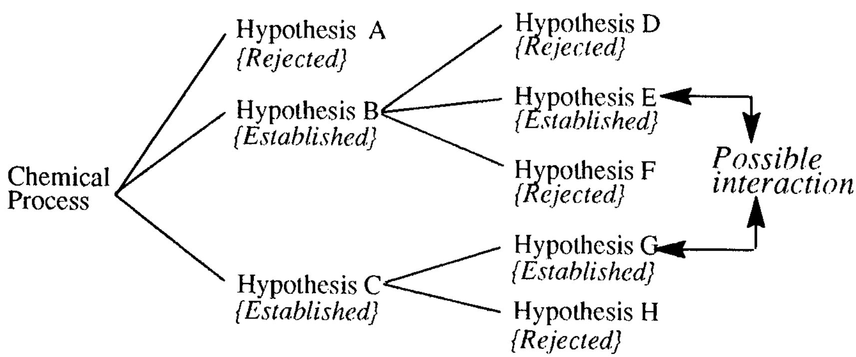

Fig. 4.4 : Possible interactions between hypotheses

If multiple malfunctions are identified, as in Fig. 4.4, then the possibility of their interaction must be resolved. *

DFS is used on an as–needed basis which is determined by the HC results. If DFS is needed then the task carries through several key problem–solving elements in order to resolve interacting malfunctions. Referring to Figure 4.4, the established tip–level hypotheses are each associated with specific malfunctioning modes, e.g. plugged valve,

*There are a number of these kinds of HC patterns that can result and are indicative of different kinds of interacting malfunctions leak in pipe, etc. To resolve a possible interaction, DFS first establishes a simulation agenda comprised of these malfunction modes that may be linked. The agenda lists the malfunctions that will be imposed on a model of the process and propagated to see if the simulation can re–create the behavior indicated by another malfunction hypothesis. Constraining the use of simulation only to the limited malfunction modes identified by HC is important in computationally solving the problem in a reasonable amount of time.

Efficiency is further enhanced by constructing the simulation model in detail only for the parts of the process that need to be simulated. As illustrated in Fig. 4.1, it is only necessary to run detailed (component–by–component) simulations for the process subsystems associated with the two hypotheses. All other systems in the plant indicated by rejected hypotheses are operating normally and can be simulated using broadly–defined, system–level models.

With the run time generation of the simulation agenda and appropriate model abstractions, the DFS task then checks to see that a causal path does indeed exist. If there is no path then there is no interaction. If a path does exist then a causal simulation is executed and the system behavior, resulting from the malfunction is determined. The simulated results are then compared with the observed behavior. If the simulated symptoms match the observed symptoms then there is an interaction. The direction of causal propagation establishes which of the interacting malfunction hypotheses is the root cause. If there is no match then the conclusion is that the malfunction hypotheses are independent. Figure 4.5 shows a flowchart for the DFS task problem–solving.

In one view, DFS acts as the interface between the compiled problem solving of HC/SPM and qualitative simulation. As an integrated approach, DFS brings together multiple sources of knowledge, a situation–specific interpretation of diagnostic results and a balance between the use of run–time simulation and compiled problem–solving in diagnosis.

4.2 The Auxiliary Tasks

4.2.1 Qualitative Interpretation of Numeric Data

Critical to the successful on–line implementation of knowledge–based systems (KBS) is the conversion of numeric plant data into diagnostically useful values. We refer to this task as "Qualitative Interpretation" (QI) [29]. While data interpretation and hypothesis evaluation are often not distinguished, we recognize strong differences in them by defining QI and HC as separate IPTs. Examples of QI include descriptions of process elements or variables as normal, abnormal, high, low etc., trends in state variables as increasing, decreasing, etc., classifications of patterns as cycling, pulsing, etc. and landmark identification (times corresponding to initiation of events).

QI performs two critical roles from the overall KBS perspective:

(1) it provides the interface between a digital process data acquisition system and the KBS, and
(2) it performs an important data reduction function.

Instead of having to deal with a large amount of temporal sensor data, QI generates useful symbolic abstractions which support efficient reasoning about interesting qualitative states of a process.

QI may be categorized into two classes: (a) context–free and (b) context–dependent. Context–free QI is the simplest form in that only sensor data associated with the primary process variable is required; there is no external context which must be considered in

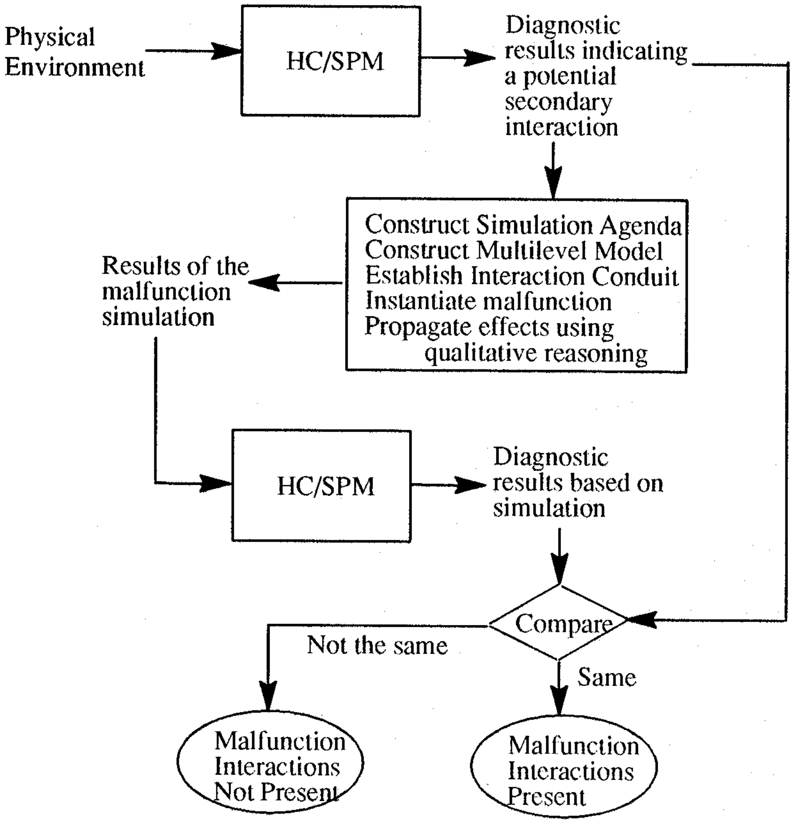

Fig. 4.5 : Evaluation of potential interactions

order to arrive at the appropriate QI. A single time series of data from a particular sensor is necessary and sufficient to draw a qualitative conclusion. Trend and landmark identification are examples. Context–dependent QI refers to those interpretations where consideration of additional information is required beyond the sensor data for the process variable of interest. An important example is normality identification. The additional information needed may correspond to time traces of other process variables, knowledge of the type of feedstock being run, condition of mechanical equipment, etc. Fig. 4.6 illustrates the context–dependent QI problem, where the normality of coolant flow can be judged only by considering all three variables — reactor temperature, coolant flow and cooling water supply temperature — simultaneously.

There are three critical aspects to the QI problem which point to the methods that are appropriate. First, we fundamentally view QI as a pattern recognition problem. It is this problem–solving mechanism that distinguishes QI from other IPTs. This characterization is motivated by the nearly universal presence and use of trend recorders and/or graphical displays in control rooms. Secondly, the dynamic nature of processes demands that the pattern recognition process associated with QI be adaptive. Sensor patterns are affected by production rates, quality targets, feed compositions, equipment condition, etc., all of which change frequently. A sensor pattern interpreted as "normal" one time may be correctly interpreted as "abnormal" at another time. Thirdly, the expertise for

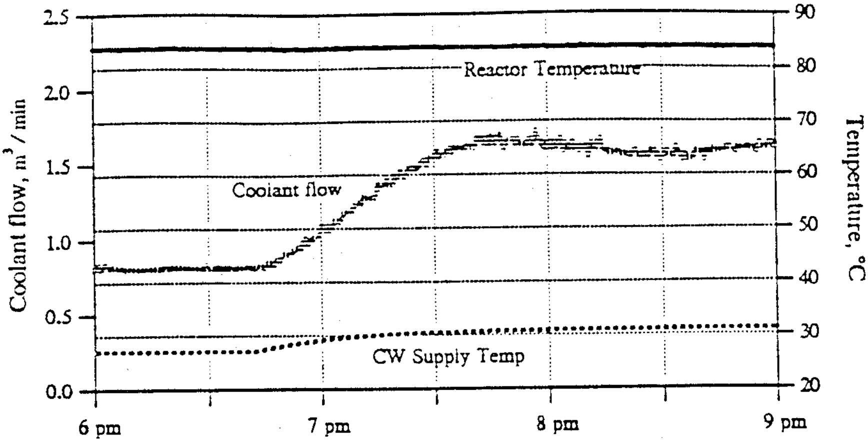

Fig. 4.6 : A typical sensor pattern as displayed on a strip chart recorder

performing QI (from experienced plant operators) exists or can be collected in the form of labeled sensor patterns of known interpretations. For the purposes of automating the QI process, the availability of these sensor patterns makes supervised learning an attractive mechanism for addressing the adaptivity requirement. In light of this characterization, the key attributes that any general purpose QI method should have are: (i) robust pattern recognition capability, (ii) ability to incorporate context or reference information for resolving context-dependent QI and (iii) adaptive characteristics which allow the method to learn from existing labeled sensor patterns.

The scope of the QI problem spans a wide spectrum in terms of complexity. The simplest forms of context-free QI (increasing, decreasing, etc.) do not necessarily require sophisticated pattern recognition methods. From a machine standpoint, context-free QI is probably most efficiently addressed using traditional signal analysis techniques. On the other hand, context dependent QI requires more sophisticated techniques especially in situations of variable context such as transient periods associated with start-up and shutdown, often changing process conditions or a variable environment. Statistical methods including limit checking, EWMA models, Shewhart charts as well as other statistical quality control (SQC) methods can have serious limitations under these variable, context-dependent circumstances.

In comparison to Bayesian approaches, backpropagation neural networks and fuzzy sets, classification of sensor patterns based on clustering or proximity measures has emerged from the various pattern recognition methods as the method of choice for the process QI problem [29]. As shown in Figure 4.7, this method utilizes the structure of the patterns to perform pattern recognition. The underlying assumption of the clustering approach is that patterns in a common pattern class exhibit similar features and that this similarity can be quantified using an appropriate proximity index. Using this concept, a given pattern is assigned to the class of patterns to which it is most similar.

With respect to the QI problem, clustering approaches have the advantage of providing the capability of dealing with limited and poorly distributed pattern data, a common situation with process operations. Rather than attempting to partition the entire data interpretation space using linear discriminants, or probability distribution functions, clustering-based methods simply identify the structure based on the available pattern

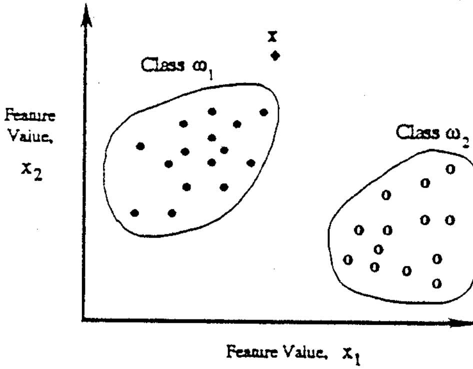

Fig. 4.7 : Clustering characteristics of a 2 class problem with 2 observable features

data. This also leads to the ability of the approach to classify novel patterns as "don't know," a very important property for QI integrated into a diagnostic approach.

4.2.2 Sensor Validation

Diagnosis in process plants makes extensive use of data provided by sensors. Like any process equipment, sensors are also susceptible to failures. However, unlike other equipment, sensors also provide the means of observing the state of the operation. From a diagnostic standpoint a faulty sensor can be a root cause or it can result in a faulty reading which can cause errors in the diagnostic conclusions. The aim of the sensor validation task is, therefore, to identify sensor errors as malfunctions and provide correct values so that a diagnosis corresponds to the true behavior of the operation.

Faults in sensors include readings that are:

(1) outside the sensor/process limits
(2) changing at a physically improbable rate
(3) stuck at some constant value, or
(4) biased.

The first three kinds of faults encompass gross mechanical/electrical operation of sensors and are relatively easy to detect by some form of limit checking. However, identification of sensor bias usually involves the use of other sensor data and process models.

Sensors fall into different categories based on how their information is used. Shum, et al. [31] have identified two broad categories of sensors used directly in the diagnostic activity: Type I sensors lead to direct control actions, either automatically through the process control system or through operator action. Sensors used to provide only state information are grouped together as Type II sensors. With the view that detection and diagnosis are two separate reasoning activities, there are also sensors that are monitored continuously for detecting any abnormalities in the plant and triggering diagnosis. A single sensor may be used for more than one purpose and may, therefore, fall into more than one categc ry.

The kind of sensor fault and the sensor category play important roles in deciding when validation is necessary, the rigor required and the means of resolving conflicts. Fig. 4.8

gives an overview of the distributed nature of the sensor validation problem as it is currently viewed for both detection and diagnosis.*

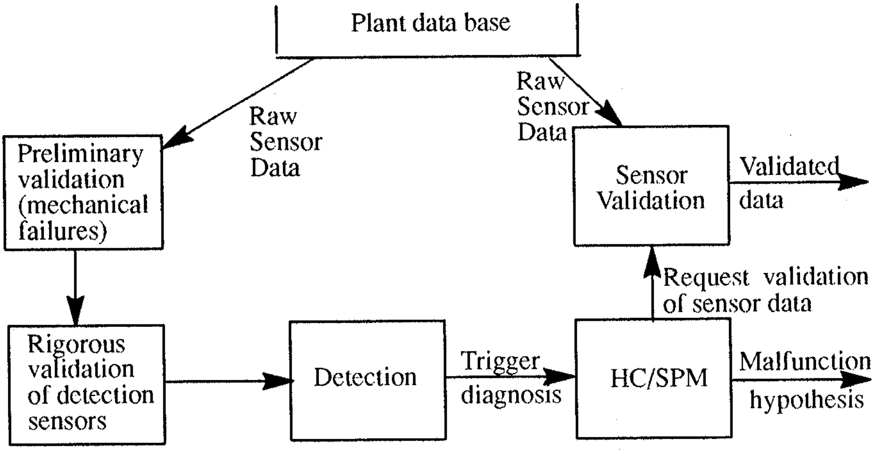

Fig. 4.8 : Distributed nature of sensor validation

In Fig. 4.8, preliminary validation of sensors includes identification of inoperative sensors. Preliminary validation is applied to all sensors and involves comparison of sensor readings to fixed or rate of change limits. Following this, rigorous validation of 'detection sensors' is performed to identify and reconcile bias. The present view is to use gross error detection techniques [31].

Once diagnosis has been triggered, sensor validation and HC become closely integrated tasks as shown in Fig. 4.1. In a classification hierarchy, Type I sensors appear explicitly as malfunction hypotheses (nodes in the hierarchy). This recognizes the fact that a Type I sensor can itself be a source malfunction. In other words, a Type I sensor failure leads to erroneous control actions, which in turn can lead to other undesirable operating conditions in the plant. For example, in the feed system hierarchy, shown in Fig. 4.2, the malfunction hypotheses "Feed meter" and "Thermocouple" appear explicitly. Type II sensors, on the other hand, are used only in the evaluation of a hypothesis. They do not appear explicitly as hypotheses in the hierarchy since they cannot be source malfunctions and since their failures do not lead to other operating problems in the plant. The validity of these Type II sensor readings is nevertheless important because they provide essential information for progressing the diagnosis.

With sensors organized as hypotheses in HC (Type I sensors) or as SPM features (Type II sensors), the validation mechanism is driven by the HC search strategy. When a particular malfunction hypothesis is evaluated, the sensor readings used are requested from a data base. This request activates the validation procedure only for sensor readings associated with the malfunction hypothesis under consideration. The validation procedure makes use of a variety of relations to identify error and to generate alternate values. These relations reflect a variety of sources including redundant sensors, reliability history of sensors, analytical computations based on process models and other sensor data, empirical correlations, or qualitative process relationships. The data are either validated

*This figure includes only an illustrative portion of the knowledge–based diagnostic framework shown in detail in Fig. 2

or the hypothesis is marked as 'suspect' for possible re-consideration. The overall effect of various data which cannot be validated is to alter the HC search strategy.

4.2.3 Hypothesis Assembly

Process diagnosis relies not only on sensor data but also on product quality data as symptomatic information. Unlike sensor data which provide localized views of the operating state of the process, product quality deviations provide a broader, more abstract view. Directional and magnitude changes in a set of product quality attributes typically provide a basis for identifying a set of deviating operating parameters that account for the observed product quality deviations. While product quality can certainly deteriorate as a result of equipment malfunctions it can also deteriorate due to changing external conditions like weather or feedstock variations. In such cases, the identification of inappropriate operating parameter settings for the new conditions is useful as a starting point for further corrective action.

In either case, each product quality attribute may be suggestive of one or more operating parameter deviations. The IPT, therefore, becomes one of constructing the best explanation for product quality deviations in terms of operating parameter deviations. Because multiple parameter deviations are assembled into a global explanatory hypothesis, this IPT is referred to as 'Hypothesis Assembly' [22].

The primary task of identifying which malfunction correctly applies to the current situation, i.e. the HC task, is achieved by weighing the conclusions made from component pieces of evidence. In the case of sensor-related evidence, such conclusions relate observed symptoms directly to a malfunction. In the case of product quality related evidence, conclusions are first drawn about operating parameter deviations, which can then be used to draw conclusions about an equipment or inappropriate parameter setting malfunction.

Hypothesis assembly provides a qualitative way of considering a set of product quality deviations and constructing a plausible explanation in terms of operating parameter deviations. It is accomplished by considering relevant operating parameters which can partially explain observed magnitude and directional changes in the product quality attributes. Because assembly is carried out as an auxiliary task to HC, the complexity of considering all possible combinations of product quality attributes is reduced by focusing only on the operating parameters which affect a particular malfunction hypothesis. In other words, the distributed nature of the malfunction hierarchy is used to determine a small, relevant set of these parameters. The assembly mechanism is then called upon to determine what combination of parameters best accounts for the observed data. Even with the reduction in complexity afforded by the HC backbone, pre-enumeration of all possible product quality deviation patterns poses a combinatorial problem. The assembly module provides a way of reasoning about the observed deviations at run time.

For the FCC unit, let us consider a simple situation where HC/SPM invokes the hypothesis assembly task [22]. Assume that for some hypothesis being evaluated the observed symptoms are:

(a) high regenerator temperature,
(b) low conversion,
(c) high ratio of hydrogen to carbon (H/C) on spent catalyst, and
(d) high cokemake

Fig. 4.9 illustrates the operating parameters that account for the observed symptoms. 'Low reactor temperature', or 'low stripping steam rate' can singly explain both 'low

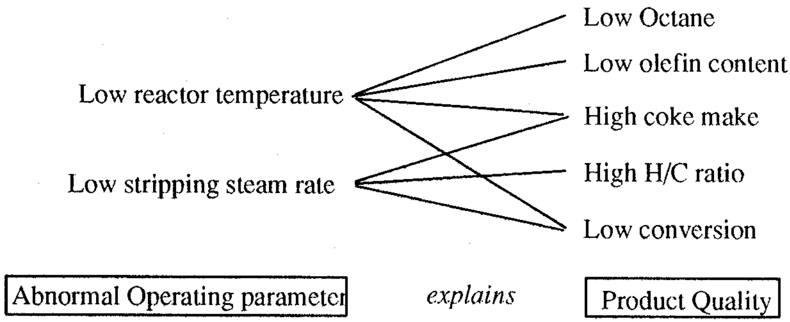

Fig. 4.9 : Explanatory relations for hypothesis assembly

conversion' and 'high coke make'. However, 'high H/C ratio' can be accounted for only by low stripping steam rate. The assembly process first selects those operating parameters that are indispensable or essential for the composite explanation. For this example, low stripping steam rate is selected first. What the composite explanation can account for is compared with what needs to be accounted for. Following this, it further checks the composite operating parameter explanation for any superfluous parts. This cycle is repeated until the list of product quality changes to be explained is exhausted. In this example, since low stripping steam rate can account for the other product quality data also, assembly terminates with low stripping steam rate as the only piece of evidence to be used in the evaluation of the hypothesis under consideration.

Additional complexity exists in the assembly process because changes in product quality attributes are the result of changes in several operating parameters, which may not necessarily be independent. The types of operating parameter interactions include those in which (a) one causes or implies another; (b) one is incompatible with another; and (c) one is an explanatory alternative to other parameters which can also potentially account for the same product quality change. As illustrated in the above example, knowledge required for hypothesis assembly includes an enumeration of all the relevant operating parameters, the product quality changes they can potentially account for, and the interactions, implications and incompatibilities among those parameters.

5. INTEGRATION OF IPTs FOR THE DIAGNOSTIC FRAMEWORK

The above task decomposition of the diagnostic activity provides a problem–solving framework which facilitates the problem analysis, knowledge acquisition, system development and maintenance of the knowledge base. Fig. 4.1 shows how the various tasks are integrated to achieve the overall diagnostic goal.

HC/SPM and DFS form the core diagnostic tasks in the overall framework. HC and SPM, two highly integrated tasks, control the course of diagnostic reasoning and direct the search towards a solution. The overall performance and efficiency of the system is largely dependent on them. Given the scope of diagnostic problem solving, HC and SPM comprise the expertise for making the diagnostic problem practically tractable. DFS is called upon as needed to achieve a higher level of diagnosis by resolving malfunction hypothesis interactions. It provides the means of appropriately constraining the use of simulation which can be computationally explosive in diagnosis. The qualitative interpretation, sensor validation and hypothesis assembly tasks augment the role of HC/SPM by providing information in forms useful for diagnosis.

The interaction between HC/SPM and the other auxiliary tasks essentially takes place at the level of individual hypotheses (nodes) in the classification hierarchy and in the context of establishing or rejecting those hypotheses. HC systematically generates malfunction hypotheses for evaluation. SPM matches the symptoms to the features of a malfunction hypothesis to evaluate it and provide a measure of confidence in that hypothesis. The symptoms themselves may not always be available in a diagnostically useful form from sensor readings; hence the data interpretation task infers diagnostically useful data from raw numeric sensor data. Before the data are interpreted and supplied to the HC/SPM task, sensor validation may be invoked to validate some or all of the sensor readings. In an analogous way, product quality data needs to be interpreted for use by HC/SPM. The task is quite different, however, than interpreting numeric sensor data. Hypothesis assembly is invoked to generate the composite evidence pattern that best explains all the data, while making sure that there are no superfluous evidence patterns in that composite.

6. CONCLUSIONS

This chapter has described a conceptual framework for knowledge–based diagnosis in chemical process plants. The framework consists of an integrated set of well–defined information processing tasks. Each of these tasks has its own distinct form of knowledge organization and problem–solving methodology. Some of the tasks are knowledge–based in nature, some are numeric, while others involve pattern recognition. The task viewpoint thus explicitly recognizes the diversity of problem solving found in the diagnostic activity and serves as the basis for integrating appropriate technologies.

From an implementation viewpoint, the conceptual framework forms the basis of an effective programming environment for building diagnostic KBSs for continuous processes. A task–based programming environment not only facilitates the building of system but also provides a high level of modularity and transparency to the user. Each task is embodied as a programming module which explicitly captures both the problem–solving methodology (inference strategy) and knowledge organization. This offers the builder of a KBS the advantage of using the framework, wherever it is applicable, without having to encode the underlying problem–solving strategy for each new application — i.e. only the domain specific knowledge needs to be encoded for any new application. Consequently, attention during implementation can be focused more on domain details and less on program development. For example, once the "establish–refine" method is programmed and a way of representing malfunction hypotheses as nodes in a hierarchy is available, the template can be used over and over again.

From a broad perspective, the structured framework takes full advantage of the diverse types of knowledge available in the domain for problem–solving power. Knowledge may be compiled, model–based, qualitative and/or quantitative. The task–based architecture provides a natural basis for integrating symbolic, neural reasoning and conventional numeric approaches into the diagnostic KBS. Furthermore, the framework offers the developer the potential to extend the scope of applicability by allowing the integration of auxiliary functions (such as sensor validation, qualitative interpretation and hypothesis assembly) into the main activity of diagnosis.

7. ACKNOWLEDGEMENTS

We wish to thank each past and present member of the AI in Chemical Engineering group at The Ohio State University for his or her input into the content of this paper. It represents a summary of many projects over a number of years.

REFERENCES

[1] C.E. Garcia and M. Morari, "Internal Model Control: 1. A unifying review and some new results", *Ind. Eng. Chem. Proc. Des. Dev.*, Vol. 21, pp. 308–323, 1982.

[2] K.–E. Arzen, "Knowledge–based control systems – aspects on the unification of conventional control systems and knowledge–based systems", *Proc. of American Control Conference*, pp. 2233–2238, 1989.

[3] H.–T. Wu and B. Joseph, "Knowledge–based control of autoclave curing of composites", *SAMPE Journal*, Vol. 26, No. 6, pp. 39–54, Nov./Dec. 1990.

[4] N. Bhat and T.J. McAvoy, "Use of neural nets for dynamic modeling and control of chemical process systems", *Comput. Chem. Eng.*, Vol. 14, No. 4/5, pp. 551–560, 1990.

[5] V.K. Tzouanas, C. Georgakis, W.L. Luyben and L.H. Ungar, "Expert multivariable control", *Comput. Chem. Eng.*, Vol. 12, pp. 1065–1074, 1988.

[6] M.R. Basila, G. Stefanek and A. Cinar, "A model–object based supervisory expert system for fault tolerant chemical reactor control", *Comput. Chem. Eng.*, Vol. 14, No. 4/5, pp. 551–560, 1990.

[7] K.J. Astrom, J.J. Anton and K.–E. Arzen, "Expert control", *Automatica*, Vol. 22, No.3, pp. 277–286, 1986.

[8] G.J. Birky, T.J. McAvoy and M. Modarres, "An expert system for distillation control design", *Comput. Chem. Eng.*, Vol. 12, No. 9/10, pp. 1045, 1988.

[9] J.R. James, "A survey of knowledge–based systems for computer–aided control system design", *Proc. American Control Conference*, pp. 2156, 1987.

[10] B. Chandrasekaran, "Expert Systems: Matching techniques to tasks", in *Artificial Intelligence in Business*, W. Reitman, Ed., Ablex Pub., 1984.

[11] B. Chandrasekaran, "Generic tasks in knowledge–based reasoning: High–level building blocks for expert system design", *IEEE Expert*, Vol. 1, pp. 23–30, 1986.

[12] V. Venkatasubramanian and S.H. Rich, "An object–oriented two–tier architecture for integrating compiled and deep–level knowledge for process diagnosis", *Comput. Chem. Eng.*, Vol. 12, No. 9/10, pp. 903–921, 1988.

[13] T.F. Petti, J. Klein and P.S. Dhurjati, "Diagnostic model processor: Using deep knowledge for process fault diagnosis", *AIChE Journal*, Vol. 36, No. 4, pp. 565–575, April 1990.

[14] S.D. Grantham and L.H. Ungar, "A qualitative physics approach to troubleshooting chemical plants", Presented at the 1989 *Annual AIChE meeting*, San Francisco, CA, 1989.

[15] F.E. Finch and M.A. Kramer, "Narrowing diagnostic focus using functional decomposition", *AIChE Journal*, Vol. 34, No. 1, pp. ,25–36, 1987.

[16] J. Calandranis, G. Stephanopoulos and S. Nunokawa, "DiAD–Kit/Boiler: On–line performance monitoring and diagnosis", *Chem. Eng. Progr.*, Vol. 86, No. 1, pp. 60–68, Jan. 1990.

[17] V. Venkatasubramanian, R. Vaidyanathan and Y. Yamamoto, "Process fault detection and diagnosis using neural networks — I. Steady–state processes", *Comput. Chem. Eng.*, Vol. 14, No. 7, pp. 699–712, 1990.

[18] M.A. Kramer and J.A. Leonard, "Diagnosis using backpropagation neural networks — analysis and criticism", *Comput. Chem. Eng.*, Vol. 14, No. 12, pp. 1323–38, 1990.

[19] D.D. Sharma, B. Chandrasekaran and D.W. Miller, "Dynamic procedure synthesis, execution and failure recovery", *Proc. of the First Int. Conf. on Applns. of AI to Eng. Problems*, Southampton, UK, April 1986.

[20] O.O. Oyeleye, F.E. Finch and M.A. Kramer, "Qualitative modeling and fault diagnosis of dynamic processes by MIDAS", *Chem. Eng. Comm.*, 96, pp. 205–28, 1989.

[21] F.E. Finch, O.O. Oyeleye and M.A. Kramer, "A robust event–oriented methodology for diagnosis of dynamic process systems", *Comput. Chem. Eng.*, Vol. 14, No. 12, pp. 1379–96, 1990.

[22] T.S. Ramesh, J.F. Davis and G.M. Schwenzer, "Knowledge–based diagnostic systems for continuous process operations based upon the task framework", *Comput. Chem. Eng.*, Vol. 16, No. 2, pp. 109–127, 1992.

[23] S.K. Shum, M.S. Gandikota, T.S. Ramesh, J.K. McDowell, D.R. Myers, J.R. Whiteley and J.F. Davis, "A knowledge–based system framework for diagnosis in process plants", *Presented at the Seventh Power Plant Dynamics, Control and Testing Symposium,* Knoxville, TN, May 15–17, 1989.

[24] G. Stephanopoulos, "Brief overview of AI and its role in process syustems engineering", *CACHE Monograph Series on AI in Process Systems Eng.*, Vol. I, G. Stephanopoulos & J.F. Davis, eds., 1990.

[25] B. Chandrasekaran, "Towards a functional architecture for intelligence base on generic information processing tasks", *Proc. of the Tenth Int. Joint Conf. on AI (IJCAI)*, Milan, Italy, August 1987.

[26] J.F. Davis, "A task–oriented framework for diagnostic and design expert systems", *Proc. of the Foundations on Computer Aided Process Operations (FOCAPO)*, Park City, UT, 695–700, July 1987.

[27] T.S. Ramesh, S.K. Shum and J.F. Davis, "A structured framework for efficient problem solving in diagnostic expert systems", *Comput. Chem. Eng.*, Vol. 12, No. 9/10, pp. 891–902, 1988.

[28] J.K. McDowell and J.F. Davis, "Diagnostically focused simulation: Managing qualitative simulation", *AIChE Journal*, Vol. 37, No. 4, pp. 569, 1991.

[29] J.R. Whiteley and J.F. Davis, "Knowledge–based interpretation of sensor patterns", Technical Report, AI in Chemical Eng. Group, The Ohio State University, Columbus, OH, June 1991.

[30] S.K.Shum, K.S. Kumar and J.F. Davis, "A knowledge–based system architecture for diagnosis and sensor validation in chemical process plants", Presented at the *AIChE Spring National Meeting*, Houston, TX, 1991.

[31] D.R. Rollins and J.F. Davis, "Gross error detection: Power functions, test statistics, confidence intervals and estimates for data reconciliation and process diagnosis", Presented at *AIChE Fall National Meeting*, Chicago, Nov. 1990.

[32] R. Isermann, "Process fault detection based on modeling and estimation methods", *Automatica*, Vol. 20, pp. 387–404, 1984.

[33] P.M. Frank, "Fault detection in dynamic systems using analytical and knowledge–based redundancy — A survey and some new results", *Automatica*, Vol. 26, No. 3, pp. 450–472, 1990.