IFAC

International Federation of Automatic Control

COMPUTER SOFTWARE STRUCTURES INTEGRATING AI/KBS SYSTEMS IN PROCESS CONTROL

IFAC Workshop
Lund, Sweden, 10 - 12 August 1994

Edited by

K.-E. ARZEN

PERGAMON

UK	Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK
USA	Elsevier Science Inc., 660 White Plains Road, Tarrytown, New York 10591-5153, USA
JAPAN	Elsevier Science Japan, Tsunashima Building Annex, 3-20-12 Yushima, Bunkyo-ku, Tokyo 113, Japan

First edition 1994

Library of Congress Cataloging in Publication Data

Computer software structures integrating AI/KBS systems in process control: IFAC workshop, Lund, Sweden, 10-12 August 1994/edited by K.-E. Årzén. -- 1st ed.
p. cm.
"IFAC Workshop on Computer Software Structures Integrating AI/KBS Systems in Process Control. Sponsored by the International Federation of Automatic Control (IFAC), Technical Committee on Applications (APCOM)" --P. (iv).
Includes index.
1. Process control--Data processing--Congresses. 2. Computer software--Congresses. I. Årzén, K.-E. II. International Federation of Automatic Control. Technical Committee on Applications. III. IFAC Workshop on Computer Software Structures Integrating AI/KBS Systems in Process control (1994: Lund, Sweden)
TS156.8.C66 1994 629 8'9--dc20 94-39392

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-08-042360 4

This volume was reproduced by means of the photo-offset process using the manuscripts supplied by the authors of the different papers. The manuscripts have been typed using different typewriters and typefaces. The lay-out, figures and tables of some papers did not agree completely with the standard requirements: consequently the reproduction does not display complete uniformity. To ensure rapid publication this discrepancy could not be changed: nor could the English be checked completely. Therefore, the readers are asked to excuse any deficiencies of this publication which may be due to the above mentioned reasons.

The Editor

Printed in Great Britain

IFAC WORKSHOP ON COMPUTER SOFTWARE STRUCTURES INTEGRATING AI/KBS SYSTEMS IN PROCESS CONTROL

Sponsored by
International Federation of Automatic Control (IFAC)
- Technical Committee on Applications (APCOM)

Co-sponsored by
IFAC Technical Committee on Computers (COMPUTS)
IEEE Control Systems Society
SAIS Swedish Artificial Intelligence Society

Organized by
Department of Automatic Control, Lund Institute of Technology
on behalf of the Swedish IFAC Committee

Financially Supported by
Swedish Research Council for Engineering Sciences (TFR)
Lund Institute of Technology
ABB, Asea-Brown Boveri AB
Gensym Corporation

International Program Committee
K.-E. Årzén (S) (Chairman)
K.J. Åström (S)
D. Birdwell (USA)
H. Bersini (B)
A. Van Cauwenberghe (B)
M. Fjeld (N)
R. Fjellheim (N)
C.C. Hang (SGP)
K. Hangos (H)
E. Hollnagel (DK)
Y. Ishida (J)
B. Kuipers (USA)
R. Leitch (UK)
M. Lind (DK)
T. McAvoy (USA)
J. Morris (UK)
L. Mõtus (EST)
G. Pang (CAN)
M. Rodd (UK)
G. Suski (USA)
J.C. Taunton (UK)
T. Terano (J)
A. Titli (F)
V. Venkatasubramanian (USA)
H.B. Verbruggen (NL)
C. Weisang (D)
E. Woods (N)
H.-J. Zimmermann (D)

National Organizing Committee
K.-E. Årzén (Chairman)
E. Dagnegård
T. Hägglund
C. Rytoft
B. Wittenmark

CONTENTS

PLENARY PAPERS I

SOFTWARE INTEGRATION

KNOWLEDGE REPRESENTATION/MODELLING

POSTER SESSION I

OTHER TECHNIQUES

PLENARY PAPERS II

CONTROL

OPERATOR SUPPORT

POSTER SESSION II

APPLICATIONS

MONITORING AND DIAGNOSIS

Towards Integrated Process Supervision: Current Status and Future Directions

Venkat Venkatasubramanian
Laboratory for Intelligent Process Systems, School of Chemical Engineering,
Purdue University, W. Lafayette, IN 47907, USA.

Abstract Process supervision deals with tasks that are executed to operate a process plant safely and economically. These tasks can be classified as data acquisition, regulatory control, monitoring, data reconciliation, fault diagnosis, supervisory control, scheduling and planning. While these operational tasks may be intrinsically different from each other, they are, however, closely related and can not be treated in isolation. Hence, there exists a clear need for an integrated framework so that the operational decision-making can be made more comprehensively and effectively. While such an integrated approach is very compelling and desirable, achieving it is no simple task as there are many challenges in realizing integration. In this paper, we review these challenges and indentify the underlying issues which need to be addressed for achieving an integrated approach to process supervision. We discuss the role of artificial intelligence in this context and how it provides a problem-solving platform for integration. We also survey the current status of automated approaches to operations and conclude with some thoughts on future directions.

Key Words. Artificial intelligence; Failure detection; Integrated plant control; Monitoring

1. INTRODUCTION

Integrated process supervision is the overall, coordinated, management of different operational tasks in a process plant. These operational tasks can be hierarchically categorized as data acquisition, regulatory control, monitoring, fault diagnosis, supervisory control, scheduling and planning. The lower level of the hierarchy involves layers that deal with tasks such as data acquisition and regulatory control. At the intermediate level, one has layers for the tasks of monitoring, data reconciliation, diagnosis, and supervisory control. At a higher level, one has the layers that perform plant-wide optimization, scheduling and planning of process operations. At the lower level, the perspective is local in character, like that of a regulatory controller which is limited to implementing a functional relationship (e.g. the control law) between the manipulated and controlled variables. The intermediate level is concerned with coordination between units, unit optimization and monitoring of production and operating constraints. It also performs fault diagnosis and suggests recovery from these malfunctions. At the higher level, the perspective is more global in character, like that of planning of process operations.

The overall problem of integrated process supervision involves several subproblem areas that are related to each other and can not really be treated as individual problems in isolation. For example, low-level events such as controller failure or some other equipment malfunction, can have a significant impact on the higher-level plans by calling for the revision of planned schedules. Likewise, higher level decisions have a serious impact on lower level activities such as supervisory and regulatory control. In the case of data reconciliation, traditionally one does not consider parameter drifts and structural faults as part of the problem. However, an integrated view is necessary for reconciliation of measured data in the presence of process faults. Thus, while these operational tasks may be intrinsically different from each other, they are, however, closely related to each other and can not be treated as isolated tasks. Hence, we need an approach wherein all these different tasks can be integrated into a single unified framework and so that the operational decision-making can be made more comprehensively and more effectively.

Over the years, a variety of tools and techniques have been developed to address these tasks. They include process modeling and simulation techniques, large scale linear and nonlinear optimization methods, advanced model-based and knowledge-based process control techniques, model-based data reconciliation methods, and statistical, neural net-based and knowledge-based fault detection and diagnosis methods. While no single tool or technique can solve the entire process supervision problem by itself, the proliferation of disparate tools imposes barriers to task integration by fragmenting system implementation as well as the solution process. Such fragmentation also impedes the understanding of the results and complicates their communication and implementation. This is one of the key challenges towards integration.

While an integrated approach is very compelling and desirable, achieving it is no simple task as there are many challenges in realizing integration. The intent of this survey paper is not so much to provide the answers but to try to identify the key challenges faced, the related fundamental issues, and review the current status and the emerging trends. In this perspective, we will also examine the role of artificial intelligence in integration. Since the scope of this exercise is very broad, we will mainly focus our discussion on the integration of low-level and intermediate-level tasks, namely, the tasks of regulatory control, data reconciliation, monitoring, diagnosis, and supervisory control in this paper.

2. PROBLEM SOLVING PARADIGMS IN PROCESS SUPERVISION

The common problem-solving paradigms that underlie integrated process supervision can be categorized as pattern recognition and classification, symbolic reasoning, and optimization. Many of the taks in process supervision can be handled in different ways. For example, process fault detection can be treated as a statistical classification problem where one tests a measurement against a null hypothesis that the process is normal. Alternatively, by considering a fault to be a parameter disturbance, fault diagnosis can be treated as a parameter estimation problem [Isermann, 1984]. Yet another view is to treat fault diagnosis as a classification of measurement data into fault groups using neural networks. In addition, we also have qualitative methods for fault diagnosis that use causal models of the process to search for the source of abnormality. This is a symbolic reasoning problem. Thus, the use of quite different solution methodologies for the same problem poses serious challenges towards integration. Since this is a central issue in integration, a better understanding of these problem solving paradigms in terms of their domain of application, types of problems that can be solved using these techniques, advantages and disadvantages is essential. To this end, a brief overview of these various problem-solving paradigms is provided in this section.

2.1. Pattern classification approach:

Syntactic pattern recognition is concerned with classifying symbolic information into a given set of classes. The classification task may be guided by a set of rules or grammar that defines the membership relationships or mapping between the patterns and the classes. Alternatively, one could specify this guiding information by a causal model (e.g., in the case of diagnosis) or in general by a set of constraints. Statistical pattern classification, on the other hand, is concerned with classifying numeric information into a given set of classes. Many problems in process operations can be categorized into one of these classification tasks. For example, reasoning about the cause of an abnormality in a process behavior can be considered as a syntactic or statistical pattern classification problem:

• Classifying sensor measurements into one of the fault classes. This is considered as a syntactic classification problem when the reasoning is based on causal models. It is statistical classification, when the numeric values of the measurements are used.

• Classifying temporal trends of sensor measurements into one of the known classes. Time series information of the sensors can be used directly for statistical classification or an abstracted syntactic representation of the measurements for symbolic reasoning purposes.

• Data reconciliation can be posed as a statistical classification problem where one tests a measurement against a null hypothesis to detect any gross sensor faults. In the absence of any gross errors, the data is then rectified to reduce the effect of random noise.

• In modeling for control, composite models can be developed by using classification. For example, choosing the proper model to use can be decided based on the operating regime the process is in and this can be solved as a pattern classification problem.

2.2. Symbolic reasoning approach:

In symbolic reasoning, one often addresses three different kinds of reasoning. They are abductive, inductive and default reasonings. Abduction is the generation of a hypothetical explanation (or cause) for what has been observed. Unlike simple logical deduction, we can get more than one answer in abductive reasoning. Since there is no general way to decide between alternatives, the best one can do is to find a hypothesis that is most probable. Thus, abduction can be thought of as reasoning where we weigh the evidences in the presence of uncertainty. Searching for the cause of an abnormality in a process system is thus an abductive reasoning. In MODEX2 [Venkatasubramanian and Rich, 1988], a model based expert system for fault diagnosis, abductive reasoning is used to generate hypotheses for the sources of faults. In addition, abduction also provides explanations of how the cause could have resulted in the abnormality observed. Such a facility is useful in providing decision support to plant operators. Use of proper knowledge representation technique matters a great deal in determining the computational effort. Model based reasoning allows for efficient bottom-up abduction by suggesting proper rules to try out. Efficiency of such bottom-up search in abduction is considerable [Charniak and McDermott, 1984].

Early work in learning concentrated on systems for pattern classification and game playing. Inductive learning is the classification of a set of experiences into categories or concepts. Inductive learning is performed when one generalizes or specializes a concept definition learned so that it includes all experiences that belong to the concept and exclude those that do not. The clear definition of a concept or category is rarely simple because of the great variety of experiences and uncertainty (noisy data or observations). For this reason, one prefers an adaptive learning scheme. An example of such an adaptive learning scheme is failure-driven learning. Failure-driven learning is refining a concept from failures of expectations as one accumulates related experiences. The failure of heuristic judgement in detecting a source of malfunction in fault diagnosis can trigger a change in the knowledge (or rule) that resulted in the judgement [Rich and Venkatasubramanian, 1989]. Experiences with abnormalities in a plant can be used to generate rules that relate a set of observations with specific causes. One can refine this experiential knowledge over time by generalizing to successful cases not covered and specializing when exceptions are noticed.

One frequently makes default assumptions on the values of various quantities that are manipulated, with the intention of allowing specific reasons for other values to override the current values (e.g. since the outlet is blocked, the flow is now zero), or of rejecting the default if it leads to an inconsistency (e.g. since the outlet of the tank is blocked, there cannot be a decrease in tank level). A fundamental feature of default reasoning is that it is nonmonotonic. In traditional logic, once a fact is deduced, it is considered to remain true for the rest of the reasoning. This is what one means by monotonic. However, as new evidence arises, often one needs to revise the deduced facts to maintain logical consistency. Let us consider our previous argument where we deduced that the tank level cannot decrease (since the outlet of the tank is blocked). After this deduction, if we get new evidence that the tank has a large leak, we will have to retract the conclusion that the tank level cannot decrease. Such a reasoning where retraction of deductions is allowed is nonmonotonic. Default reasoning or nonmonotonic reasoning is an invaluable tool in dealing with situations where all the information is not available at a time or if one has to reason about many, probably inconsistent, cases simultaneously. Reiter [1987] has shown how default logic can be used for reasoning about multiple faults or causes for an abnormality. Reasoning with assumptions explicitly is a related concept [Kavuri and Venkatasubramanian, 1992).

2.3. Optimization approach:

Optimization problems in process operations such as model identification fall under the continuous case, while problems such as allocation of plant resources requiring discrete decisions are combinatorial optimization problems. For example, plant-wide scheduling and optimization in the continuous case and assignment and allocation of plant resources in cases which require the sharing of manufacturing resources between different products are examples of optimization problems in planning. Other examples are:

- Management of inventories and maintenance planning.
- Online estimation of process model, for optimization and model-based process control, data reconciliation, parameter estimation for fault diagnosis.
- Online prediction of the performance of an operating plant.
- Online optimization of control profiles in batch and continuous operations.

Most of the planning problems which are discrete optimization problems are usually solved off-line and hence one can try to solve really large problems. In contrast, most of the continuous optimization problems have to be solved on-line and hence computational effort becomes an important consideration here. Other concerns include convergence problems in multi-dimensional search spaces and local minima problems in continuous nonlinear optimization problems.

3. INTEGRATED PROCESS SUPERVISION: CHALLENGES AND THE ROLE OF AI

Though an integrated framework is very attractive in terms of the benefits it can provide, there are a number of conceptual and implementational challenges that have to be overcome before an industry-wide following of this approach takes place. This section discusses the key requirements and the role of AI in addressing these challenges.

i. *It is necessary to reason about process operations without assuming accurate models.*

In most cases, plant behavior is not accurately known. Even rigorous models are not adequate to predict plant behavior with satisfying accuracy. Furthermore, configuration of plants change during their lifetime. Process operating conditions may vary with the demands for different products produced in the plant. All of these force the operators to make their operating decisions with approximate models of process behavior. AI provides us with techniques for developing qualitative and approximate models, doing inexact reasoning, etc. to cope with situations such as this.

ii. *It is necessary to reason with incomplete and/or uncertain information about the process.*

Operators often face situations where they receive conflicting information about the status of the process or the various process units. This could be due to faulty sensors, for example. Also, they often deal with situations when all the information needed about the process may not be available. Thus, operators are forced to reason and make assessments about the process with incomplete and/or uncertain data. Realizing these operational constraints in practice and having a means to handle incompleteness and uncertainty is essential to the decision making process. Again, artificial intelligence techniques play a useful role in handling this requirement of an automated system.

iii. *It is necessary to understand, and hence represent, process behavior at different levels of detail depending on the nature of the task.*

The amount of information that is available to the operator is often sufficient to understand the essentials of the behavior of a process. However, the voluminous data results in an information clutter and the operator is now faced with the task of gleaning the important features he needs from this vast amount of data. Information from process measurements, perhaps over an entire month, needs to be organized so that he can get a more global picture of a section or the overall plant easily. Given the large size of plants and different information requirements of tasks, it is necessary to reason with knowledge at different levels of detail. Reasoning with knowledge at different levels of detail is a difficult task as one has to carefully ensure the consistency of the information at different levels of detail. Given the information clutter, it is inevitable that the operator have some way to look at the required information in a compact way. For example in a process plant, there may be as many as 1500 process variables observed every two seconds for behavior during a selected period [Bailey, 1984]. The trends are displayed on monitors and there can be two, four or eight process variables displayed per screen at any one time. This dictates the need for a hierarchic organization from process subsystems to loop clusters down to single loops. This also emphasizes the need for an automated framework for extracting important qualitative features of process behavior from raw sensor data. Powerful knowledge representation and pattern classification techniques of AI are indispensable for this problem.

iv. *It is necessary to make assumptions about a process when modeling or describing it. One has to ensure the validity and consistency of these assumptions.*

When a process unit is described by a model, the model is constructed based on some assumptions, mostly assumptions of normal behavior. However, in diagnostic applications, these assumptions may be violated. In order to avoid inconsistencies, it is necessary to explicitly consider and change the model and the assumptions during the reasoning process. What is needed is a representation of the process model that can represent the process behavior for a given set of assumptions. It is necessary to explicitly define the underlying assumptions, have a scheme to verify the consistency of these assumptions and choose the process model based on these assumptions.

As an example, consider the problem of controlling a process. The controller configuration, parameters and the control law are determined by the mathematical models of the process and the controller. The success of the control scheme crucially depends upon whether the assumptions that underlie these models are still valid. For example, models assume that the sensors provide accurate information. In the case of a gross fault in the sensor, the controller action not only becomes ineffective but may even cause adverse process behavior. Similarly in a hierarchical model for process operations, the decisions made at a higher level can have significant impact on the lower level implementations and thus their assumptions are crucial. Failing to detect the violation of an assumption can result in a gross disruption of the operations. AI provides us with the framework for treating assumptions explicitly, thereby making the automated system readily alert to assumptions violation.

v. *It is necessary to integrate tasks and solution approaches. This requires integrating different problem-solving paradigms, knowledge representation schemes, and search techniques.*

To effectively provide an integrated framework, one needs to carefully address the knowledge representation and search issues. It is necessary to represent structural, functional and behavioral information about the process. We can think of these as three complementary sources of information each organized hierarchically. One needs to address how the three hierarchies are built and how they interact. One of the key functions of such a knowledge representation is to let one examine the process at any preferred level of detail in any desired hierarchy. For example, for the task of process fault diagnosis, one is concerned with structural information within the individual units and the overall connectivity of the process. For planning tasks, one may take a higher-level perspective on the process plant, lumping many units together as a larger, abstract, input-output module. The different tasks may employ different problem-solving paradigms which, in turn, would call for different representation and search strategies. All of these need to integrated to offer a complete perspective of process operations. Due to the character of the issues involved in here, artificial intelligence plays a crucial role.

vi. *It is necessary to keep the role of an operator primary and active, not secondary and passive, in the operating environment that is managed with the assistance of on-line intelligent systems.*

While it might be acceptable to delegate all the control to computers when we are dealing with regulatory control problems, it might be more risky to do so when it comes to supervisory and higher-level decision-making. This is due to the character of the problems and issues involved as well as due to the limitations of current intelligent systems. In addition, one has other concerns such as the liability and legal aspects of this problem. Thus, it is important to have the operators actively involved in the decision-making process and make the on-line intelligent systems play an advisory role. This is also necessary to keep the operators' skills sharp, as otherwise their skills could deteriorate over time due to their increased dependence on the advisory systems as a crutch. There is a delicate balance that has to be achieved here. Since the operator's role would be primary, this creates special demands on the design of the advisory systems, such as:

- simple, operator-friendly user-interfaces
- emphasis on visual, graphical display of information for ease of understanding
- structured, guided access to data and knowledge about process status and behavior
- explanatory capabilities to offer insights into the systems reasoning and recommendations

Thus, the design of such systems should be operator-centered, with his or her needs and capabilities in mind. Such a perspective places considerable emphasis on man-machine interaction issues and the nature of the user-interface, which are important requirements that will benefit from artificial intelligence techniques.

One can see from this discussion that the use of AI techniques to face these challenges is not only desirable, but also necessary.

4. CURRENT STATUS OF AUTOMATION IN PROCESS SUPERVISION: A BRIEF REVIEW

The main focus of this paper is to address issues in integrated process supervision for the low-level and intermediate-level tasks. As mentioned before, these tasks are: regulatory control, process monitoring, fault diagnosis, data reconciliation and supervisory control. In this section, the current status of automation of these tasks are briefly reviewed.

4.1. Process Monitoring

Process monitoring refers to the task of identifying the state of the system from sensor data. Process trend analysis and prediction are important components of process monitoring. Knowing the current process trends, the causes that drive them, and the possible future evolution of these trends are essential for supervisory decision making. The central issues here are representing and reasoning with temporal evolution of process trends, multi-scale data, sensor noise and data uncertainties, and cause and effect models of process trends. Recent research in this area has shown some promise for integrated supervision applications. Stephanopoulos recognized the importance of process trend representation for higher-level process integration early on and developed a formal framework. This framework handles temporal data, reasonable discontinuous and continuous functions, and the abstraction of semi-quantitative and qualitative trends (Cheung and Stephanopoulos, 1989).

Venkatasubramanian and co-workers (Janusz and Venkatasubramanian, 1991; Rengaswamy and Venkatasubramanian, 1992) developed a similar approach in their qualitative representation scheme. The fundamental element in their representation scheme is the primitive. They use a finite difference method to calculate the first and second derivatives of the process trend changes and based on these values the primitives are identified. For noisy data, neural networks are used to extract the primitives as they are noise-tolerant. The ability to handle noise is incorporated in two stages. At a lower level, the neural net-based pattern classification approach is used to identify the fundamental features of the trends. At a higher level, the syntactic information is abstracted and represented in a hierarchical fashion with an error correcting code smoothing out the errors made at the lower level. Such syntactic approaches are suitable for hierarchical representation of the trend information and for developing error correcting codes, which eliminate the effects of high noise and outliers.

Multilevel abstraction of important events in a process trend is possible through scale-space filtering through the use of a bank of filters each sensitive to certain localized region in the time-frequency domain (Marr and Hidreth, 1980). An example of such a filter that has been extensively used in image processing is $\nabla^2 G$, where G is a Gaussian distribution (Marr and Hidreth, 1980). Another important recent development in this area is the emergence of wavelet-based frameworks. The recent work of Bakshi and Stephanopoulos (1992) using wavelet networks for representing trends shows considerable promise.

4.2. Fault detection and diagnosis

Fault detection and diagnosis is concerned with the detection of abnormal behavior and the identification of their causal origins. Over the recent years, there has been considerable progress towards the automation of fault detection and diagnosis. A general description of fault diagnosis would include the following kinds of abnormalities:

Gross parameter changes in a model
In any modeling, there are processes occurring below the selected level of detail. These unmodeled processes are lumped as parameters. "Parameter faults" arise when there is a disturbance entering the process from the environment through one or more parameters. An example of such a fault is a change in the concentration of the reactant in a reactor feed or the change in the activity of a catalyst.

Structural changes
Structural changes refer to changes in the model itself. They occur due to hard failures in equipment. Structural faults result in a change in the information-flow between various variables. This corresponds to dropping the appropriate model equations and restructuring the other equations to describe the current situation in the process. An example of a structural fault would be a controller failure which would imply that the manipulated variable is no longer functionally dependent on the controlled variable.

Malfunctioning sensors and actuators
Gross errors usually occur in actuators and sensors. There could be a fixed failure, a constant bias (positive or negative) or an out-of-range failure. These are also important faults that need to be identified quickly in view of the fact that some of the instruments might provide feedback signals which are essential for the control of the plant.

The solution strategies for fault diagnosis range from purely qualitative to purely quantitative, with various combinations in between. There are different perspectives from which one can view the problem of fault diagnosis. One can look at the fault diagnosis problem from the perspective of the transformations the measurements go through before arriving at the final solution. Figure 1 shows these transformations.

Fig. 1. Transformations in a Fault Diagnostic System

Measurement space is the space of sensor measurements that is available to perform fault diagnosis. Feature space is the space of reduced set of features representative of the measurement space that is developed by transforming the measurement space using *a priori* knowledge about the process. A set of decision variables are developed from the feature space, and class space is the set of integers indexing each individual fault and an additional integer to represent the normal operation of the process and is the final interpretation delivered to the user by the diagnostic system. The transformation from feature space to decision space is performed by a search technique that tries to minimize the mismatch between the actual observations and the observations for different faulty modes, either in a qualitative or quantitative manner. For example, symbolic reasoning is done qualitatively where one tries to minimize the mismatch between the observation (sensor i is low, sensor j is high and so on) and the template for various faulty modes. In contrast, in parameter estimation methods the mismatch is a least squares norm and is minimized by searching in a parameter space that models the various faults. The transformation from the decision space to class space is effected using either thresholding, template matching or symbolic reasoning as the case may be. Hence, the two important components in a diagnostic system are the *a priori* knowledge and the search technique used.

One can view diagnostic systems from either of these two perspectives as well. A general classification of fault diagnostic systems can be done based on the three different solution methods used. They are: Knowledge-based, Analytical model-based, and Pattern recognition-based methods. Each of these diagnostic methods is a combination of a particular type of *a priori* knowledge and a search technique. Knowledge-based systems predominantly use qualitative models of the process in tandem with different search techniques. The quantitative model-based approaches rely on mathematically representing the inconsistencies between the actual and expected behavior as *residuals*. Pattern recognition is the task of assigning a pattern to one of k predetermined classes. The knowledge about these classes is usually obtained from process history data. Under each of these general solution philosophies there are many combinations of *a priori* knowledge and search techniques and is not possible to enumerate all these combinations. Hence, the attempt here is to provide a flavor for some of these methods.

Knowledge-based methods
In this subsection, knowledge-based techniques are illustrated with the aid of some typical approaches to fault diagnosis.

Hypothesize-and-test Using Causal Models:
There are many approaches to the design of model-based expert systems for chemical process fault diagnosis. One approach is the hypothesis/test strategy. A rigorous approach to hypothesis formulation is the method of O'Shima and coworkers (Iri et. al., 1979; Shiozaki et. al., 1985).

The basic premise of this approach is that, for a fault hypothesis to be considered viable, causal pathways must link the proposed fault origin with all observed abnormal measurements. A signed digraph is used as the representation of the influences between the process variables.

Another class of model-based reasoning methods used for diagnosis use agenda-based search techniques. An example of this approach is MODEX (Rich and Venkatasubramanian,1987), which is a system developed to reason from first-principles model-based knowledge. It extension, MODEX2 (Venkatasubramanian and Rich, 1988), is a two-tiered approach using a compiled knowledge-base at the top tier and a deep-level causal model at the second tier.

Finite State-Space Search of Fault Trees:
Fault trees provide a computational means for combining logic to analyze system faults. To perform consistent diagnosis from fault trees, the trees must comprehensively represent the process causal relationships. To ensure this consistency, causal models in the form of signed digraphs are developed. Causal fault trees are developed from these digraphs (Lapp and Powers, 1977). Fault trees determine causal pathways through which primal events (faults) can propagate through the system to cause the top event (some significant malfunction).

Once a fault-tree is synthesized, the information from it is stored in the form of cut sets. Cut sets exhaustively specify all possible paths in the digraph resulting from the fault. Kramer and Palowitch (1989) developed a method of deriving rules for diagnosing faults from signed digraphs. Ulerich and Powers (1987) used digraph models to include human operator action and failure models due to operator action. They also illustrate how real-time data can be used to infer events in a control loop.

Search in malfunction hypotheses hierarchy:
Knowledge within these systems is organized as a hierarchy of malfunction hypotheses, representing different levels of process abstraction (Ramesh, et. al., 1992). Each level of hierarchy represents an increasing level of process detail. Under the establish-refine strategy, a hypothesis under consideration is evaluated by examining the knowledge-groups associated with it. The search mechanism consists of exploring the hierarchy in a top-down fashion and a malfunction hypotheses is completely validated.

Analytical Model-based Approaches:
The analytical model-based approaches require knowledge about the process in terms of either input-output models or first principles quantitative models based on mass and energy balance equations. Here, the following techniques are used.

Observer-based Fault Detection and Isolation:
The main focus in observer-based fault detection and isolation is the generation of a set of residuals which detect and uniquely identify different faults. These residuals should be robust in that the decisions should not be corrupted in the face of unknown inputs. Unknown inputs here include unstructured uncertainties such as process and measurement noise and modeling uncertainties. The aim of observers is to come up with an error innovation sequence like

$$e(k+1) = Fe(k) - TKf(k) \qquad (1)$$

where F is the observer system matrix and T is the input transformation matrix, and K is the fault distribution matrix.
If no faults occur in the process, $f(k) = 0$ then

$$e(k+1) = Fe(k) \qquad (2)$$

If the absolute value of the eigenvalues of F are less than 1, then $e(k) \to 0$ as $k \to \infty$.
In the absence of any faults, this observer will track the process independent of the unknown inputs d(k) to the process. Hence these are known as an unknown input observers. The necessary and sufficiency condition for the existence of these kinds of observers are described in Frank and Wunnenberg (1989).

Parity-Space Approach:
Parity equations are suitably arranged forms of the input-output model of the plant. The basic idea is to check the parity (consistency) of these input-output models of the plant by using the sensor outputs (measurements) and known process inputs. The idea of the approach is to rearrange the model structure so as to get the best isolation of the failures. Chow and Willsky (1984) proposed a procedure to generate parity equations from the state-space representation of a dynamic system. Gertler and Singer (1990) extended this to statistical isolability under noisy conditions by defining marginal sizes for fault alarms. All these methods are limited to failures that do not include gross process parameter drifts. However, they are an attractive alternative owing to their ability to determine, *a priori,* isolability of different faults. A general scheme for considering both direct and temporal redundancy in parity equation generation is provided by Chow and Wilsky (1986). In contrast, voting techniques are often used in systems that possess high degree of parallel hardware redundancy (Willsky, 1976).

Parameter Estimation:
Diagnosis of parameter drifts which are not measurable directly requires on-line parameter estimation methods. Accurate parametric models of the process are needed, usually in the continuous domain in the form of ordinary and partial differential

equations. Young (1981) and Isermann (1984) surveyed different parameter estimation techniques such as least squares, instrumental variables and estimation via discrete-time models.

Pattern Cassification Methods:
Pattern classification using process history data is usually performed using either statistical and non-statistical techniques. Bayes classifier is one of the more popular classifiers using the statistical properties of the input data. Neural networks have proved to be a popular non-statistical approach to pattern recognition.

Parametric and Non-parametric Classifiers:
Statistical pattern classification methods again can be roughly compartmentalized into two components: (i) a priori knowledge; assumption about the form of Probability Density Function (PDF) available, and (ii) search technique; the classifier design. Estimation of PDF (Fukunaga, 1972) can be classified as parametric and non-parametric estimation and the classifier can be also designed either in a parametric or non-parametric fashion.

Neural Networks as Classifiers:
A lot of interest has been shown in the recent literature in the application of neural networks for the task of pattern classification in fault diagnosis (Hoskins and Himmelblau, 1988; Kavuri and Venkatasubramanian, 1994). To understand neural networks better it helps to view them from a statistical pattern recognition perspective. Let us consider a standard two layer neural network. The first layer connecting the input to the hidden nodes tries to estimate the PDF for each class and the second layer connecting the hidden nodes and the output acts as a classifier. It is not surprising then that the network based classifiers are inferior to parametric classifiers when the density information for the class is available. When the assumption of a functional form for the density function could be made, parametric classifiers are a better choice. However, for a general classification problem, an *a priori* choice cannot be made for the functional form of the density. Moreover, the data available for the classes may not be enough to develop approximations to the density function. In such cases, non-parametric classifiers such as the network based classifiers are to be preferred.

As a general comparison of these different approaches, one can state that knowledge-based systems can be used where fundamental principles based approaches are more difficult or lacking, where there is an abundance of experience but not enough detail available to develop accurate quantitative models. However, they suffer from the resolution problems resulting from the ambiguity in qualitative reasoning. Parity space and observer-based approaches (analytical model-based methods) are shown to be equivalent in that they can be developed to generate the same residuals. Merits and demerits of one group carry to the other. These methods provide design schemes in which the effects of unknown disturbances can be minimized, isolability conditions ascertained, and sensitivity analysis performed in a consistent manner. The cost for obtaining these advantages are mainly modeling effort, computational effort, and the restrictions that one places on the class of acceptable models. Pattern classifiers are constructed solely based on process history data. The main advantages of classifiers are: their real-time performance, ease of knowledge acquisition, and applicability to a wide variety of systems. There are some limitations to methods which are based solely on process history data. It is the limitation of their generalization capability outside of the training data. This problem is alleviated by radial and ellipsoidal units by avoiding a decision in case there are no similar training patterns in that region. This allows the network to detect unfamiliar situations arising from novel faults. Besides its lack of ability to generalize to unfamiliar regions of measurement space, classifiers based solely on process history data also have difficulties in identifying multiple faults.

The review of the fault diagnosis approaches presented here does not adequately cover the considerable body of work that is available in the literature. This review is necessarily brief due to spatial constraints. For a more thorough review the reader is referred to Venkatasubramanian et. al. [1994].

4.3. Data Reconciliation
Data reconciliation can be viewed as a quantitative fault diagnosis problem with the focus on detecting sensor faults and sensor biases. Another important goal is to remove the measurement noise from process data to enhance the control performance. Data reconciliation usually consists of three parts: (i) identification of the biased parameter, (ii) estimation of the bias, and (iii) rectification of the sensor measurements.

Romagnoli and Stephanopoulos (1981) proposed a systematic method for identifying the source and location of gross errors in linear systems under steady-state conditions. There are three levels to their proposed strategy. (i) A structured search of the balance equations for measurements with gross errors. (ii) Sequential search of the balance equations that reduces the search further. (iii) Another level of sequential search that identifies the gross error. Mah and Tamhane (1982) proposed a statistical test on the residuals to identify gross error.

Crowe, et. al. (1983) proposed a matrix projection method for data reconciliation problems for the linear case. Valko and Vajda (1987) introduced the idea of decoupling the parameter estimation problem from the state variable estimation problem. Though the original problem is not naturally decomposable in this manner, the rationale for doing this is that one always has good initial values for the state variables, whereas, it is hard to provide good initial values for the parameters. Recently, Liebman et. al. (1992) proposed a nonlinear programming methodology for data reconciliation in nonlinear processes under transient conditions.

Most practical processes are nonlinear in nature and hence linear reconciliation methods might not be adequate for practical problems. Also, steady-state reconciliation methods might prove ineffective in handling transients. In this context, the nonlinear programming approach proposed by Liebman, et al. (1992) is quite promising. The issues that have to be addressed in this approach are, computational complexity in the case of large-scale nonlinear problems and nonconvexity problems.

4.4. Supervisory Control

Supervisory control, typically, has a variety of functions. It includes model updating, controller tuning, reacting to equipment failures, "gain scheduling" to reflect changes in the disturbance variables, changes in the process, and so on. It might also include, for example, in batch plant automation, dealing with sequential controls and exception handling. It could also potentially include automatically making major changes in controller configurations or control algorithms to reflect process changes, online optimization, automated startups and shutdowns of continuous plants and scheduling. These are knowledge intensive tasks and knowledge-based methods have been proposed previously in the literature.

Kraus and Myron [1984] presented a self-tuning controller that uses pattern recognition techniques. Automatic controllers are tuned manually in usual practice. The control engineer perturbs the closed loop, observes the pattern of response, and compares this response to the desired pattern. Then, using his experience, he adjusts the control parameters. The pattern recognition based self-tuning PID algorithm monitors the closed-loop recovery following a set point or load disturbance and automatically calculates the P, I and D so as to minimize the process recovery time, subject to user-specifiable damping and overshoot constraints. Cooper and Lalonde (1990) have presented the idea of detecting naturally-occurring input excitation events based on the recent history of manipulated process input and also the calculation of model gains to develop a continuous gain schedule function for better control of nonlinear systems. The utility of knowledge-based expert systems in performing diagnostics and tuning control systems in real-time has been discussed by Arzen (1991). The formal integration of pattern recognition techniques in control systems to design "Intelligent Controllers" has been proposed by Stephanopoulos (1991).

5. BRINGING IT ALL TOGETHER: FUTURE DIRECTIONS IN INTEGRATED PROCESS SUPERVISION

As we reviewed in the last section, there has been considerable progress made in the last decade in developing efficient solution strategies for the various operational tasks. In this section, a perspective on how these paradigms, tools, and techniques in process operations might evolve and come together in the near future is presented. To this effect, first, one possible integration framework is discussed. The intent here is to discuss the nature and the extent of interaction that might occur between the various tasks in such a framework. Then, some perspectives on how the conceptualization and solution techniques for these various tasks themselves might develop is provided.

5.1. A Framework for Integrated Process Supervision:

One can approach the formulation of an integration framework from different viewpoints such as: (i) information flow, (ii) flow and management of data (iii) functional blocks view, and (iv) knowledge management. From the perspective of operational tasks, the most important facet of the integration framework is the functional blocks view and one such interpretation of the integration framework is provided in Fig. 2. The figure shows the structure and the interfacing of various process operational tasks and the information-flow dependence between modules that perform these tasks. The main functions of the monitoring system are to provide concise executive summaries to be presented to the operator, extract and abstract hierarchical trend explanation to be passed on to a diagnostic system and, detect and remove outliers. The fault diagnostic system houses different kinds of knowledge like rules, temporal patterns, causal models and pattern information. The diagnostic system assists the operator in identifying the root cause of the problems and also passes on this information to both supervisory control and data reconciliation modules. The data reconciliation module estimates the values of the parameters to be sent to the supervisory control module and also provides the regulatory control with the reconciled process data. Supervisory control module would have the complete information about the state of the process. The supervisory system would utilize the trend information and the diagnostic information to suggest any changes needed at the regulatory control level.

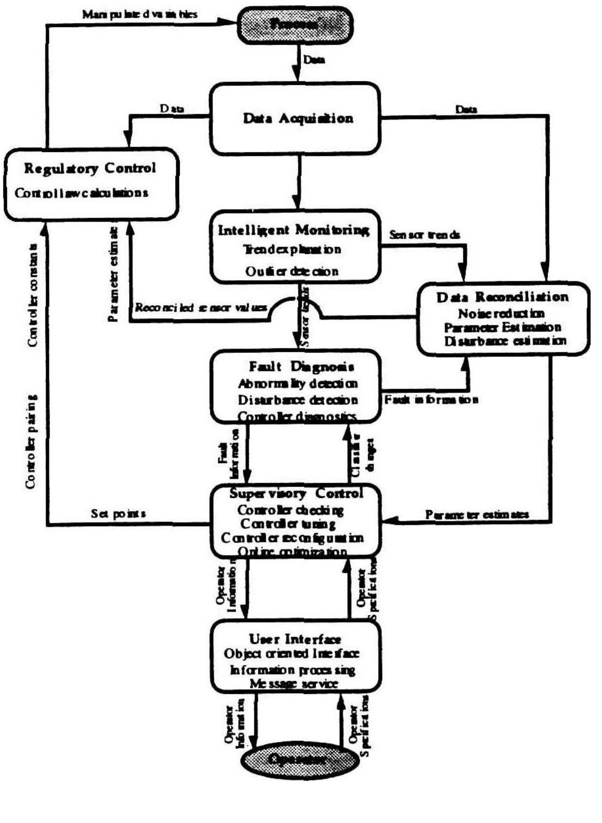

Fig. 2. A framework for Integrated Process Supervision

5.2. Process Monitoring

The important recent developments in the field of process monitoring have been the advances made in statistical process monitoring and syntactic pattern recognition, as noted earlier. The ultimate use of process monitoring is in diagnosis. The integration of these monitoring techniques with diagnosis is still not fully developed. One needs to see improvements in this regard in the future. Also, the recognition of operators as an important part of the operational decision-making has put an onus on the monitoring systems. The monitoring systems should be able to provide information to the operators in a way that they can understand them. These systems should also be able to interact with the operators, take suggestions from them and interpret these suggestions for the other operational tasks. Hence, one needs integration of Natural Language Processors (NLP) into typical monitoring systems in the future.

5.3. Fault Diagnosis

One of the important underlying points in all fault diagnosis methods is the inadequacy of a single method to handle all the requirements for a diagnostic classifier. Though all the methods are restricted, in the sense that they are only as good as the quality of information provided, still some methods might be better suited for a particular problem at hand than others. Hence, hybrid methods where different methods work in conjunction to perform collective problem-solving are an attractive alternative. For example, fault explanation through a causal chain is best done through the use of digraphs, whereas, fault isolation might be very difficult using digraphs due to the inherent ambiguities in qualitative models. Analytical model based methods are superior in this regard. Hence, hybrid methods might provide a general powerful problem-solving approach. Consider another example, where a pattern-based classifier and a trend-based classifier are used sequentially for improved search. The pattern based classifier localizes the search based purely on the spatial organization of various fault patterns. Once this is done, a trend based classifier can take the set of fault hypotheses to see if they can be further distinguished based on the temporal pattern they exhibit. One can hope to improve the resolution characteristics of an overall diagnostic framework by combining various approaches like these.

There has been some work on hybrid architectures. The two-tier approach by Venkatasubramanian and Rich (1988) using compiled and model based knowledge is an example of a hybrid approach. Frank (1990) advocates the use of knowledge-based methods to complement the existing analytical and algorithmical methods of fault detection. The combination of methods allows one to evaluate different kinds of knowledge in one single framework for better decision making. The resulting overall fault detection scheme would have a knowledge base consisting of both heuristic knowledge and analytical models, data base, inference engine and explanation component. These methods provide promising prospects for the solution of general diagnostic problems.

Dynamic simulation in diagnosis:

When one has access to dynamic models of a process, one should take advantage of such models in real-time diagnosis. However, this is not usually done due to the complexity of the models and the difficulties involved in integrating diverse approaches in a single framework. But using a framework such as the one in Figure 2, a hybrid system would be feasible.

The basic idea here is to use a signed digraph for doing fault diagnosis at a first level. The completeness for a diagnostic system using digraphs is usually quite good, whereas, the resolution might be poor. The digraph will come up with a malfunction hypotheses set which would also include the actual fault(s). A prioritizer will then order the faults for further validation. Under a single fault assumption, the faults can be simulated using on-line first principles model. The simulated data and actual data can be compared using trend modeling approaches for validation. Through a hybrid approach like this, one can hope for improved completeness and resolution in a diagnostic module.

5.4. Data Reconciliation

As mentioned before, most attempts in the past had restricted themselves to linear and/or small systems. With the recent progress in optimization and the emphasis on plant-wide control, people are attempting large-scale nonlinear optimization problems. From a purely computational point of view, if the only bottleneck is the "largeness" of these problems, it may be handled by faster computers,parallel computing and more efficient algorithms. However, there are other issues like local minima problems inherent in many nonlinear process situations which makes the solution to large-scale nonlinear optimization methods very difficult, particularly in real-time. Hence, one needs to think of new ways of formulating the problem and new solution strategies drawing from different fields that might mitigate this complexity. For example, instead of using purely gradient-based approaches, one can think of a combination of qualitative and quantitative approach. Diagnostic qualitative knowledge can be used to reduce the search space and provide good initial guesses thereby enhancing the performance of the data reconciliation module.

5.5. Supervisory control

While there has been a lot of work in regulatory control over the years, much less attention has been devoted to supervisory control issues (Garcia et. al., 1991). As noted earlier, supervisory control includes model updating, controller tuning, reacting to equipment failures, changes in controller configurations or control algorithms to reflect process changes, online optimization, automated startups and shutdowns and so on. The following outlines some of the important issues to be addressed in the context of controller performance.

Identification of out-of-tune controllers:

The simplest kind of supervisory control action one can think of is the monitoring of individual control loops. In doing this a test signal is sent periodically to perturb the closed loop system to a small degree. By using pattern matching techniques on the resultant output of the perturbed system, one can identify out-of-tune controllers or controllers with degraded performance. Once the problematic controllers are identified, they can be tuned using rules in a knowledge-based system or other techniques.

Designing controllers for various inputs:

Generally, there is no single perfect controller for all kinds of input disturbances. A controller designed for a step input in a particular variable might not give acceptable performance if there is a ramp input in that variable. Furthermore a control system cannot be designed to work well for disturbances in all the input variables. These issues might become crucial if the system is being operated under tight quality control requirements. This is another area where one might see future developments in supervisory control systems. A supervisory control system can adaptively toggle between various control laws based on the specification about the state of the plant. This specification about the state of the plant can be provided by a combination of sensor trends, fault hypotheses from the diagnostic system and the estimates from the data reconciliation module.

Update the model to redesign controller:

Other than the input disturbances, there can be some structural changes in the plant model itself. In such a situation the different kinds of controllers previously designed might no longer be valid. This could call for redesigning the controllers. This is another place where one might use the information from the diagnostic system to update the model. Once the model is updated, one can decide about the controllers that are affected by this change in the model. With this new information from the model, a redesign of the controllers can be performed.

Variable pairing selection:

The variable pairing for the controller depends to a great extent on the state of the system. Once there is a change in the state of the system, the original pairing might no longer be optimal. There might exist new pairings corresponding to the state of the system that might provide optimal control action. By using methods like Relative Gain Arrays (RGA) and Singular Value Decomposition (SVD) techniques in conjunction with knowledge-based systems, a supervisory control module can advise the operator of the different options available and suggest an optimal configuration.

Detuning and reconfiguration of controllers:

In case of process upsets one might want to continue production in the system with minimal impact. Having minimal impact on the system might call for reconfiguration of the controllers in the system. This can be done effectively if one has built-in redundancy in the control system in the design stage itself. To this end one might need to detune some controllers and bring into operation other controllers. Having bypass lines and rerouting streams might be another way of moving variability in the process to different locations. Detuning of controller might also be done in the case of unanticipated instability in the system.

User specified, online, interactive design of controllers:

Another kind of activity that the supervisory control module can do is the online interactive design of control parameters. In a typical plant, operating strategies change form time to time. Such changes might necessitate the redesign of control parameters. In some situations, for example, one might want a small settling time without worrying too much about the overshoot in the response. In contrast, in some other situation, one might want as small a