World Scientific Series in Automation – Vol. 4

INTELLIGENT MODELING, DIAGNOSIS AND CONTROL OF MANUFACTURING PROCESSES

Edited by

B-T B Chu and S-S Chen

World Scientific

World Scientific Series in Automation – Vol. 4

INTELLIGENT MODELING, DIAGNOSIS AND CONTROL OF MANUFACTURING PROCESSES

Edited by

B-T B Chu and S-S Chen

Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC 28223, USA

World Scientific
Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 9128

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 73 Lynton Mead, Totteridge, London N20 8DH

Library of Congress Cataloging-in-Publication Data

Chu, Bei-Tseng Bill.
Intelligent modeling, diagnosis, and control of manufacturing processes / by Bei-Tseng Bill Chu and Su-Shing Chen.
p. cm. -- (Series in automation : vol. 4)
Includes bibliographical references.
ISBN 9810208170
1. Manufacturing processes -- Mathematical models. 2. Process control -- Mathematical models. I. Chen, Su-Shing. II. Title. III. Series.
TS183.C52 1992
670.42'01'5118--dc20 92–19673
CIP

Printed in Singapore by JBW Printers & Binders Pte. Ltd.

INTELLIGENT MODELING, DIAGNOSIS AND CONTROL OF MANUFACTURING PROCESSES

SERIES IN AUTOMATION

Editor-in-Charge: S. S. Chen

Published

Vol. 1: Autonomous Mobile Robots – Vehicles with Cognitive Control
A. Meystel

Vol. 2: Image Understanding in Unstructured Environments
edited by S. S. Chen

Vol. 4: Intelligent Modeling, Diagnosis and Control of Manufacturing Processes
edited by B.-T. Chu and S. S. Chen

Forthcoming

Vol. 3: Image Analysis and Its Applications
Xu Jian-Hua

PREFACE

This book will be of interest to practitioners as well as researchers in the area of manufacturing process modeling, diagnosis and control. Over the past few years, we, the editors, have been working on modeling and diagnosis of semiconductor manufacturing processes. Through interaction with people from different research areas, we have come to realize the many limitations of commonly used approaches and that great opportunities exist to combine these approaches and achieve *Intelligent Modeling, Diagnosis, and Control of Manufacturing Processes.*

In our view, an intelligent manufacturing process control system must have a rich model of its operating environment, be able to automatically diagnose problems, and be able to plan for corrective actions. Based on continuous functions, classical control/statistical theories do not take advantage of the rich body of qualitative knowledge/experience available in diagnosing and ultimately controlling manufacturing processes. On the other hand, working primarily on the representation of qualitative experience and knowledge, people working in Artificial Intelligence (AI) and Model-based Reasoning have for the most part ignored integrating quantitative and qualitative knowledge.

This volume contains a collection of papers that present what we believe are promising techniques for representing qualitative and quantitative knowledge as well as integrating both types of knowledge in problem solving. Papers in this volume can be roughly divided into three groups.

The first four papers represent popular AI approaches to diagnostic reasoning. The paper by Punch, Goel, and Sticklen introduces a diagnostic reasoning model based on generic tasks, a popular approach within the AI community. The paper by Suzuki and Iwamasa demonstrates how "first principle" knowledge can be qualitatively captured and effectively utilized in process diagnostic applications. McDowell and Davis address the issue of compilation of "first principles" into a more efficiently usuable form for diagnostic reasoning. The paper by Peng and Reggia introduces their well known causal network model for diagnostic reasoning based on formal probability theory; they also demonstrate how distributed processing can be used to form diagnostic hypotheses quickly.

The next two papers demonstrate the representation of quantitative process knowledge using statistical techniques. Extending a well known statistical technique, evolutionary operations, Sachs et al. present a new adaptive control scheme that hypothesizes the structure of a process model. The paper by Lin and Spanos provides a detailed case of building a quantitative process model and its subsequent application in process diagnosis and control.

The final group of four papers offers several approaches to integrate qualitative and quantitative process knowledge and use them in diagnosis and control. The paper by Chang and Spanos shows how a statistical model (described by Lin and Spanos) can be utilized in diagnostic reasoning by employing the Dempster-Shafer framework, a framework often used in the AI community to work with qualitative knowledge. Chu describes how a continuous system model can be transformed into causal relations. The paper by Chen presents a model of process control utilizing neural networks and fuzzy logic. Irani, Cheng, Fayyad, and Qian present a machine learning algorithm that learns decision trees from quantitative data; they also offer some insightful comparisons between the classic statistical approach and the machine learning approach.

It is our hope that this book will provide an introduction to the representation of qualitative and quantitative process knowledge for people who may not be familiar with these issues. In particular we hope that it will spark further research activities in the integration of these types of knowledge for the intelligent modeling, diagnosis and control of manufacturing processes.

B. B. C.
S. S. C.
Charlotte, NC

CONTENTS

CHAPTER 1

Manufacturing Diagnosis and Control: A Task-Specific Approach*

William F. Punch III
Ashok K. Goel
Jon Sticklen†

1 Introduction and Overview

Manufacturing technology generally does not receive sufficient attention at engineering research laboratories in the U.S. As a result, some American manufacturing industries are beginning to lag behind that of her economic competitors. With the growing international competition, however, is an emerging national awareness that one of the keys to continued economic prosperity is the ability to develop novel and better methods of manufacturing.

Our research on manufacturing is based on the notion of *task-specific problem-solving* or *generic tasks* [10, 9, 8, 7]. Our research goal is to show the efficacy of the task-specific approach (TSA) by constructing functioning systems for manufacturing diagnosis and control. These systems, in turn, are expected to yield generic problem-solving architectures that would be useful for addressing a large class of manufacturing diagnosis and control problems. The goal of this chapter is twofold. First, to review how the task-specific

---

* *This chapter combines and expands papers presented at the AAAI-90 Workshop on Intelligent Diagnostic and Control Systems for Manufacturing and the 1990 AAAI-SIGMAN Workshop on Planning and Control.*

†Punch and Sticklen are with the AI/KBS Lab., Computer Science Department, Michigan State University, East Lansing, MI 48824. They can be contacted at punch@cps.msu.edu and sticklen@cps.msu.edu, respectively. Goel is with the AI Group, College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332, and can be contacted at goel@cc.gatech.edu.

approach can and has made a contribution to the domain of manufacturing, especially in the areas of diagnosis and control. Second, to examine how various task-specific architectures can be integrated into comprehensive systems that more fully address problem-solving in the manufacturing and other domains.

2 Task-specific Problem-Solving

The intuition behind task-specific problem-solving is simple yet powerful: cognition has a number of primitive types of generic *tasks* and problem-solving methods. These generic tasks are encoded using control structures and knowledge representations that are tuned to the job of each task. For example, the problem-solving of diagnosis may require significantly different strategies and knowledge than say design, though each may share some aspects. This viewpoint leads to a twofold research agenda: First, examination of real-world problem-solving domains to discern new generic tasks, and second creation of appropriate computational strategies and representations that can encode those tasks. The promise of the task-specific approach is that once a sufficient number of tasks are identified and their problem-solving requirements understood, then more complex problem-solving methods can be realized by combining tasks. That is, more complex problem-solving can simply emerge from an appropriate sequencing of task-specific problem-solvers.

The task-specific approach makes a number of commitments:

Cooperative Problem-Solving: The view taken of problem-solving is that of a group of processes, each dedicated to solving specific kinds of problems, operating in concert such that more complicated behavior emerges. Each process has its own control and its own knowledge representation that is specifically suited for the task it solves. This view is opposed to a number of other AI paradigms, notably the logic and rule paradigms, which promote a uniform low-level representation of either control or domain knowledge in which all problem-solving activities can be encoded. The point to be emphasized is *not* that such uniform architectures are incapable of solving complex problems, but instead that they concentrate on problem-solving at a single level (of representation, of control) which may not easily lend itself to different kinds of problems.

Use-Specific Knowledge: The representational form of knowledge and the process(es) that uses it cannot be separated. In general, this claim states that knowledge might have to be represented in different forms to be utilized in different contexts. Therefore, if knowledge representation is going to be matched to the control structure then the representation of similar concepts may occur in different forms. Bylander and Chandrasekaran [1987] call this the *interaction problem.* In these terms, the claim is that even uniform architecture advocates who pursue separation of knowledge and control for the purposes of using the same knowledge in multiple problem-solving perspectives *implicitly* take into account coding the knowledge to match the task at hand.

A Task-specific problem-solver can be described in the following manner [9]:

- The *function* of the problem-solver. What type of problem does it solve, what kind of goals can it achieve? What is the nature of the information that it takes as input, and produces as output?

- The *symbolic structures and processes used to encode knowledge.* What are the primitive terms in which the forms of knowledge needed for the task can be represented? How should knowledge be organized for that task?

- The *control strategy.* What control strategy can be applied to the knowledge to accomplish the function of the generic task?

The remaining sections of the paper will describe a view of Diagnosis and Control problem-solving based on a task-specific approach. In particular, we will show how this approach can significantly impact the domain of manufacturing.

3 Diagnosis

A common definition of the term *diagnosis* is: The mapping of signs and symptoms to malfunctions[3]. As applied to the domain of manufacturing, this means the discovery of malfunction (in equipment, in feed materials, in coordinating processes etc.) that affect the quality and quantity of the product yielded. This rough definition of the problem does not yield one particularly approach for its resolution as there are many views on how diagnosis can be

[3]Caution: There are other views of what "diagnosis" means, for example, the view that emphasizes the link between diagnosis and therapy. The sections on MDX2 (section 4.2) and KRITIK (section 4.4) discuss this in more detail.

accomplished. For example, heuristic/empirical/compiled approaches to diagnosis are based on a representation which pre-enumerates the malfunction categories and which reasons by searching for categories that best account for the observed signs and symptoms [15, 13, 46]. Other approaches emphasize model-based approaches that do not pre-enumerate the categories. Instead, these approaches determine malfunction based on representations using detailed models of the domain and reasoning methods like design models [19], malfunction/behavior modes [16] or simulation [17, 52, 33]. Along this spectrum from compiled to deep systems we have investigated a number approaches to diagnosis, both in isolation and integrated into more comprehensive systems.

We have also investigated issues in diagnostic data validation, validation based on a higher level of analysis than techniques which rely on strictly statistical techniques [12, 14]. In the following sections we will describe our work in compiled reasoning, model-based reasoning and data validation in the context of diagnosis.

3.1 Classificatory Diagnosis

One approach to diagnosis is *hierarchical classification* [29]. The hierarchical classification method finds the categories in a classification hierarchy that apply to the situation being analyzed. A significant portion of expert systems such as MYCIN [46] can be viewed as classification. In fact, Clancey [15] has specifically analyzed MYCIN and shown it to be a kind of classification problem-solving.

Diagnosis as a classification problem-solving task is a matching of the data of the problem against a set of *malfunctions* (i.e diseases, system failures etc). If the present data is classified as a known malfunction, then the diagnosis is completed. Note that this is a *compiled/associational* approach to diagnosis as it requires that the possible malfunctions of the particular domain be pre-enumerated. For example, given some data about a car engine problem, the hierarchical classification task is to find the categories (e.g. broken piston, faulty distributor) that best describe the data of the problem. The task-specific characteristics of hierarchical classification are as follows.

The classifier requires a pre-enumerated list of the categories. These categories must be organized into a hierarchy in which the *children* (i.e the subnodes) of a node represent subhypotheses of the *parent* (i.e the superior node). Figure 1 illustrates a fragment of a tree from a hierarchical classification system for the diagnosis of malfunctions in a Chemical Processing Plant.

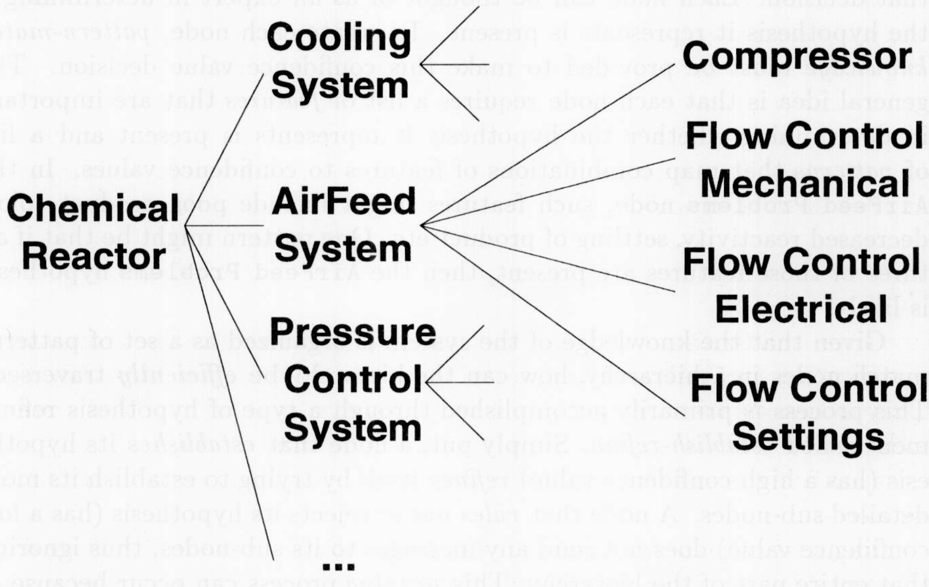

Figure 1: Fragment of Chemical Processing Plant classification tree

Note that as the hierarchy is traversed from the top down, the categories, or in this particular case *hypotheses* about the failure of the Chemical Plant, become more specific. Thus the children of the hypothesis `AirFeed Problems` can be broken into more specific hypotheses of `Compressor Malfunction` and various kinds of `Flow Control Malfunctions`.

Each node in the hierarchy is responsible for calculating the "degree of fit" or *confidence value* of the hypotheses that the node represents. For example, the node `AirFeed Problems` node is responsible for determining if there is an air feed malfunction and the degree of confidence it has in that decision. Each node can be thought of as an expert in determining if the hypothesis it represents is present. To create each node, *pattern-match knowledge* must be provided to make this confidence value decision. The general idea is that each node requires a list of *features* that are important in determining whether the hypothesis it represents is present and a list of *patterns* that map combinations of features to confidence values. In the `AirFeed Problems` node, such features might include poor product color, decreased reactivity, settling of product etc. One pattern might be that if all three of those features are present, then the `AirFeed Problems` hypothesis is likely.

Given that the knowledge of the system is organized as a set of pattern-match nodes in a hierarchy, how can the hierarchy be *efficiently* traversed? This process is primarily accomplished through a type of hypothesis refinement called *establish-refine*. Simply put, a node that *establishes* its hypothesis (has a high confidence value) *refines* itself by trying to establish its more detailed sub-nodes. A node that *rules out* or rejects its hypothesis (has a low confidence value) does not send any messages to its sub-nodes, thus ignoring that entire part of the hierarchy. This *pruning* process can occur because of the hierarchy's organization of nodes based on hypothesis-subhypothesis. For example, the subhypotheses of `AirFeed Problems` are simply more detailed hypotheses. If there is no evidence for `AirFeed Problems` (i.e., it is ruled out), then there is no point in examining more detailed hypothesis about failures of that subsystem.

The above strategy of hierarchical classification has been formally characterized and analyzed in [28, 11]. The analysis of hierarchical classification reaffirms the computational advantages of hierarchical organization of classificatory knowledge. While the general task of classification is computationally intractable, if and when classificatory knowledge can be organized in a hierarchy, the task can be solved tractably.

The strategy of hierarchical classification is embodied in a domain-independent tool called CSRL [6] that allows domain experts, who need not be AI specialists, to program diagnostic systems. CSRL has been used in a number of manufacturing domains including diagnosis of chemical processes [47] and nuclear power plants [30]. It has also been used in other domains including internal medicine [49, 51, 43] and international politics [55].

3.2 Abductive Diagnosis

A second approach to diagnosis is that of *abduction.* Abduction as a diagnostic process was first applied by Pople in the INTERNIST system [40], followed by the set-covering model of Reggia [44]. We have been involved in work on a method known as *abductive assembly* [48, 39] first applied in the RED red blood cell analyzer[4]. This work has been extended in a number of ways including approaches to distributed abduction [51], concurrent abduction [27] and development of the domain-independent abductive assembly tool PEIRCE [37].

The process of abductive assembly is driven by the need to *explain* a set of data. In the context of diagnosis, abductive assembly is a method for explaining a set of symptoms or *findings* in terms of malfunction hypotheses present in the domain. We use the term "hypothesis" to refer to an object that might explain some of the findings constituting the abductive problem. So we say that a hypothesis "offers to explain" some finding or set of findings, such as "`AirFeed Problems Hypothesis` offers to explain the color change noticed in the product". An abductive system collects a set of hypothesis that together explain as many of the current findings under the mutual constraints imposed by the other hypotheses as possible. This set of hypotheses is called a *compound hypothesis* or *compound explanation*. Thus abductive assembly is a method for assembling explanatory hypotheses to explain domain findings under some constraints.

Given the goal of trying to explain the findings, we find that there are three general subgoals for achieving abductive assembly [37]:

1. Obtain a candidate set of hypotheses for possible inclusion in the compound explanation.

2. Explain the findings by constructing a compound explanation.

3. Critique the compound explanation.

[4] RED was not itself a diagnostic system, but it's approach has since been applied to a number of diagnostic domains [49].

Obtaining the candidate set of hypotheses for use by the abductive assembler can be done in a number of ways. In the original RED system [39, 48], a hierarchical classifier was used to reduce the list of potential hypotheses by using only those malfunction hypotheses it deemed plausible. That is, if a node is ruled-out then that hypotheses is not provided to the abductive assembler. This "heuristic filtering" greatly reduces the complexity of the assembly process and focuses its efforts in areas that appear plausible from another viewpoint. Besides plausibility information, other information must be provided to the abductive assembler via the hypotheses. This includes:

- Knowledge about what findings can be explained by this hypothesis and, if such knowledge is available, to what degree. This information can be generated based on the state of the problem being solved or pre-compiled.

- Knowledge about what other hypotheses this hypothesis might conflict with, again either generated or pre-compiled. For example, the hypothesis of `Hot Reactor` cannot simultaneously exist in the compound explanation with the hypothesis `Cold Reactor` as they are mutually incompatible.

- Knowledge about what other hypotheses are entailed or suggested by this hypothesis. For example, in a closed vessel the hypothesis `Raised Pressure` might entail that `Raised Temperature` also be used.

The details of explaining the findings using the assembly process are shown in Figure 2. Once hierarchical classification has generated the list of malfunction hypotheses, the abductor begins assembling an explanation. First, a finding is selected to be explained. From the hypothesis list, a set of hypotheses that offers to explain that finding are formed as the candidates for assembly. From this candidate list, a hypothesis is selected for inclusion into the final explanation set. The selection is based on each candidate member's plausibility, its compatibility with other members of the final explanation and other factors. The selected hypothesis is integrated into the final explanation and the selected finding is marked as explained, as well as any other findings that the hypothesis can explain. The process of selecting a finding, selecting an explanatory hypothesis, integrating the selected hypothesis into the final explanation and updating of the findings continues until either all the findings are explained or some condition prevents a complete explanation.

Since we engage in abductive problem-solving to find out what is true or believable about a situation, we want to know if the explanatory hypothesis