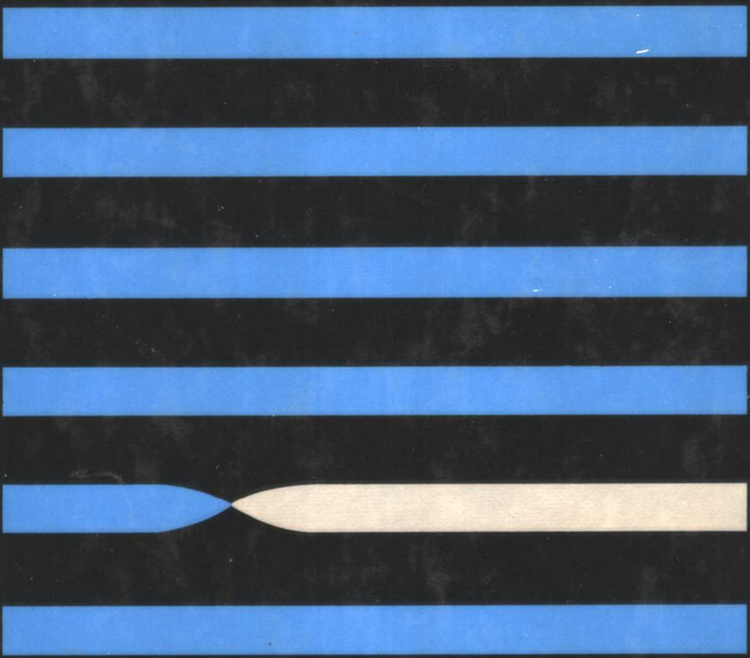
E. T. Keravnou L. Johnson

# COMPETENT EXPERT SYSTEMS



## A CASE STUDY IN FAULT DIAGNOSIS

# Competent
Expert
Systems

# Competent Expert Systems

## A case study in fault diagnosis

ET Keravnou & L Johnson

**McGraw-Hill Book Company**
New York St. Louis San Francisco Montreal Toronto

**Library of Congress Cataloging-in-Publication Data**

Keravnou, E.T.
Competent expert systems.

Bibliography: p.
Includes index.
1. Expert systems (Computer science) 2. Automatic checkout equipment. I. Johnson, L. II. Title.
QA76.76.E95K44 1986 006.3'3 86-20969
ISBN 0-07-034168-0

First published in Great Britain in 1986 by Kogan Page Ltd.,
120 Pentonville Road, London NI 9JN.

First American edition published by McGraw-Hill in 1986.

1234567890 DOC/DOC 8932109876

Printed and bound by R.R. Donnelley & Sons Company.

# Contents

# Preface

Since our first book (Johnson and Keravnou, 1985) we have developed our own expert system to illustrate the theoretical and practical notions we have been developing. This book is a report on that work. The ideas have been jointly developed (with the qualification necessary by the acknowledgements below), but the programming effort was undertaken almost entirely by Keravnou. We feel that the fact that these ideas are implemented gives added depth and value to the book. We report the work *we have done* not the work we would like to do. In this way, we hope the book will be of value to both academic and industrial research centres and as useful case study material for those training in research.

## Outline

The book is divided into four parts. Part I discusses the methodological and theoretical framework of the practical work reported in Parts III and IV. Part II provides a perspective on representative artificial intelligence approaches to fault diagnosis and verification, and acts as a backcloth to Part III. It may be read independently of the other parts. Part III explains how we have employed our advocated methodology to reconstruct an existing fault finding system, CRIB. Part IV explains how case specific information (findings) can be employed intelligently in the context of diagnostic systems specifically and problem-solving in general. Once more we employ our methodology to build a knowledge-based system that reasons from the general findings knowledge.

### PART I THE METHODOLOGICAL FRAMEWORK

After the introductory chapter we argue, in Chapters 2 and 3, that an expert system must explicitly capture aspects of a competence model of the relevant expertise. This would not only enhance the system's problem-solving capabilities by facilitating knowledge revisions but also enhance the human-computer system by yielding a strong 'cognitive coupling' between the system and the user. To this end, we have investigated as the central tenet of a methodology for building expert systems, the explication of domain strategies, knowledge structure and dialogue structure. These three aspects are interdependent and should be treated as such. The methodology, therefore, puts a particular accent on the purpose of knowledge elicitation. We conclude by indicating the advantages that accrue from the methodology.

In Chapter 4 we analyse the diagnostic task at a domain independent, high level of abstraction. More specifically, we indicate how Peirce's three stages of inquiry – abduction, deduction and induction – are reflected in the workings of a diagnostic task. We suggest that the errors of a novice diagnostician can be analysed with advantage from this perspective and that the inference nature of a diagnostic inquiry should be directly reflected in the focusing aspect of the inquiry.

## PART II OTHER WORK

In Chapter 5 we consider – with one exception (D-algorithm) – only illustrative examples of the artificial intelligence approach to fault diagnosis and verification. These systems form the backcloth against which our fault finder can be compared and contrasted. Most of these attempts constitute exercises in 'automated reasoning' rather than exercises in capturing the human problem-solving heuristics and strategies.

## PART III RECONSTRUCTING CRIB: A DEMONSTRATION OF THE METHODOLOGY

In Chapter 6 we encode the understanding of the competence underlying the particular domain (as given in the Deemen reports by the original CRIB team) in terms of the conceptual tools discussed in Chapter 3.

In Chapter 7 we give the representation structures for the competence model. We attach more importance to the generic representation structure that makes explicit the aspects of human reasoning discussed in Chapter 3. When *this structure is instantiated for our domain of expertise*, it completely captures and makes explicit the conceptual view of the reasoning knowledge. The discussion on the factual knowledge representation structures is specific to our particular domain.

In Chapter 8 we discuss how instances of the reasoning and factual knowledge representation structures given in Chapter 7, are implemented in a computer system. This system constitutes a 'soft-wired' simulation model for the particular expertise.

In Chapter 9 we illustrate our reconstruction through a sample consultation, and proceed to discuss the strategic explanation facility in more detail.

In Chapter 10 we present various extensions to our system framework. These range from domain independent extensions to our task structure (analytical, representational, operational extensions) to extensions specific to our CRIB reconstruction. The latter cover the incorporation of a mixed initiative type of dialogue, the incorporation of a learning mechanism, and the storage of the domain knowledge on secondary medium during fault investigations. Through this discussion we hope to illustrate that a mixed initiative type of dialogue and a 'learning' mechanism can be easily incorporated in an expert system, if the particular domain knowledge structure and the reasoning knowledge are made explicit in the system.

PART IV INTELLIGENT HANDLING OF DATA (FINDINGS)

In Chapter 11 we describe how the general findings (symptoms, signs, historical data, test results, etc) knowledge should be organized for use in the context of a competent automated diagnostician.

In Chapter 12 we discuss the operation of the central task, DECIDE-STATUS, in more detail and show how its subtasks or strategies co-operate via a so called implications network. The structure of an implications network and the propagation of truth values along it is discussed. An account of the operation of the various strategies follows. These operations are relevant to any intelligent problem-solving activity; they are an aspect of commonsense reasoning.

In Chapter 13 we summarize and present our conclusions.

## Acknowledgements

Chapters 2 and 3 are developed from the paper *The Need for Competence Models in the Design of Expert Systems* (Johnson, 1985a). Part IV is a development of *Organizing a Findings (Data) Base for use in a Competent Automated Diagnostician* (Johnson, 1985b). We thank the editors for permission to use this material. Particular acknowledgements are made by references throughout the text, but we would like to make a more general acknowledgement of the influence of the published work of Clancey, Chandrasekaran, and their co-workers. We hope that this work is a complement to theirs. Our methodological framework is drawn from the research done in the field of education, principally by Ogborn and Bliss. Their work on the HELP project (Bliss, J. and Ogborn, 1977 and 1979) entailed the elicitation of physicists' actual methods of working and their ways of organizing their knowledge for discovering physical knowledge – these are problems analogous to those which confront the knowledge engineer. This was the first time that Systemic Grammar Networks were used as a device for analysing qualitative data.

The work of G.A.S. Pask has had subtle but valuable influence. F.H. George generously provided material which formed the basis of the reconstruction of CRIB (Part III of the book). We believe his work should be more widely known. We have benefited from discussing CRIB with T.R. Addis and C.M. Elstob and from the published account by Hartley (1984). We have also benefited from discussions with S. Murdoch, L. Bottaci and N.E. Johnson about the overall aims and objectives of our work and the particular ideas contained in this book.

The work of E.T. Keravnou was sponsored by GEC/Marconi Research (Chelmsford, Essex), under the direction of Mr P. Sizer and she would like to thank him and the establishment sincerely.

ET Keravnou
L Johnson
*London*
*February 1986*

# PART 1

# The Methodological Framework

In Part I we discuss the methodological and theoretical framework of the practical work reported in Parts III and IV. The aim of, and motivation for, our research are set out in the introductory chapter. In Chapters 2 and 3 we discuss our methodology for building expert systems. Chapter 4 explains the theoretical paradigm we use to capture completely, at a high level of abstraction, the workings of any diagnostic task (fault diagnosis is the field in which we undertook practical work).

CHAPTER 1

# Introduction

## Aim of the research

The aim of our research was to develop a methodology for building expert systems that exhibit the following characteristics:

1. The processes are based on symbol manipulation.
2. A large set of these symbols are structured in a way that enables them to be interpretable as world knowledge.
3. The system must have a set of symbols and operations that are interpretable as a representation of specific knowledge and expertise.
4. The system should be capable of reconstructing inference paths, this reconstruction forming the basis for explanation and justification facilities for the system.
5. The system should perform at expert levels and should do so in such a way that the human-computer interaction conforms to the user's needs.

## Motivation for the research

The early expert systems such as DENDRAL (Lindsay *et al*, 1980); MACSYMA (Martin and Fateman, 1971); MYCIN (Shortliffe, 1976); PIP (Pauker *et al*, 1976); PROSPECTOR (Duda *et al*, 1979); INTERNIST-I (Pople, 1975) etc, were directly coded into a dialect of LISP. The majority of these systems were rule-based, ie their knowledge was uniformly represented in terms of pattern-action/conclusion associations (Waterman and Hayes-Roth, 1978; Davis and King, 1977). A few, such as PIP and INTERNIST-I, were frame-based (Minsky, 1975) or network-based such as CASNET (Weiss, 1974) (CASNET was in fact coded in FORTRAN). This 'vintage' era in the expert systems technology was followed by three streams of research: research into developing powerful symbolic programming environments, research into knowledge representation, and the development of knowledge representation languages and the construction of shells.

Programming environments, knowledge representation languages and shells are collectively referred to as tools (see Hayes-Roth *et al*, 1983; Harmon and King, 1985). The first of these provide intelligent editors and various knowledge engineering constructs in addition to the constructs provided by an ordinary symbolic manipulation language. Knowledge representation languages embed on one or more knowledge representation schemes. (ROSIE (Fain *et al*, 1982); RLL (Greiner and Lenat, 1980); HEARSAY-III (Erman *et al*, 1981); OPS5 (Forgy and McDermott, 1977); OWL (Szolovits *et al*, 1977); KRL (Bobrow and Winograd, 1977); and KRYPTON (Brachman *et al*, 1983) are notable examples of knowledge representation languages.) A shell or a skeletal system is a generalization of an expert system, made by deleting the domain specific knowledge from the knowledge-base and adding the facilities necessary for instantiating the knowledge-base for some other domain. In other words, a shell explicates the framework of the corresponding expert system (eg EMYCIN (VanMelle, 1979) derived from MYCIN, KAS (Reboh, 1983) derived from PROSPECTOR, EXPERT (Weiss and Kulikowski, 1979 and 1981) derived from CASNET). A further generalization has been made in the system, AGE (Nii and Aiello, 1979) which is a sort of super shell that provides, at least in theory, a choice of system frameworks.

The current trend in building expert systems is to choose a tool to be employed in the construction of the system. While programming environments do not constrain the designer in any way, knowledge representation languages and (especially) shells, do. Once a language or a shell is selected, the knowledge engineer must help the expert to 'structure his/her knowledge' (Hayes-Roth *et al*, 1983, p.129). In practice this means to '... initiate the process of teaching the expert to formulate his or her thoughts into [the chosen tool].' (Harmon and King, 1985, p.202.)

Current expert systems, both domain-crafted and those produced through the use of a shell, exhibit a number of drawbacks: they are inflexible in their problem-solving capabilities (eg some systems can only deal with the most common problem cases), they cannot converse in an