Putting Expert Systems into Practice

Robert G. Bowerman, B.Sc.
and
David E. Glover, Ph.D.

Putting Expert Systems into Practice

Robert G. Bowerman, B.Sc.
and
David E. Glover, Ph.D.


VAN NOSTRAND REINHOLD COMPANY
New York

Library of Congress Catalog Card Number 87-13296
ISBN 0-442-20842-1

Printed in the United States of America

Van Nostrand Reinhold Company Inc.
115 Fifth Avenue
New York, New York 10003

Van Nostrand Reinhold Company Limited
Molly Millars Lane
Wokingham, Berkshire RG11 2PY, England

Van Nostrand Reinhold
480 La Trobe Street
Melbourne, Victoria 3000, Australia

Macmillan of Canada
Division of Canada Publishing Corporation
164 Commander Boulevard
Agincourt, Ontario M1S 3C7, Canada

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Bowerman, Robert G., 1956-
Putting expert systems into practice

Bibliography: p.
Includes index.
1. Expert systems (Computer science) 2 Management information systems. 3 Computer integrated manufacturing systems. I Glover, David E II. Title
QA76.76.E95B68 1988 006.3'3 87-13296
ISBN 0-442-20842-1

Foreword

The commercialization of artificial intelligence (AI) and expert systems tech' 'ogies has been filled with misconceptions, overstated claims, and not a ..ttle marketing hype. Academicians tend to speak a different language from the systems professionals. Product marketing departments tend to employ a third language, and in the case of AI have often misunderstood and not connected with the computer professionals, the academicians, or the end users. It is only within the past few years that systems professionals in companies as dissimilar as American Express, Coopers & Lybrand, DEC, and DuPont have gained enough experience with both the promises and shortcomings of applied expert systems technology to begin to see through the smoke and behind the mirrors. To the newcomer, AI is often presented as a single entity or magical product; rather it is simply a collection of software engineering tools that make some difficult programming tasks easier and other seemingly impossible ones approachable. The major distinction between conventional systems and AI-based systems is that they address more abstract symbolic problems and are able to cope with uncertain or conflicting data to infer useful solutions.

From the point of view of the typical system developer or manager it's important to recognize that the introduction of expert systems is not a new phenomena in the history of software engineering. Just as high-level languages and relational databases emerged from computer science research to become part of the system developer's standard tool kit, so are expert systems just now entering the mainstream of software engineering. The passage of time and experience should make the use of tools such as inference engines, object-oriented programming, and frame based reasoning no more unusual than data dictionaries or transaction processing.

A few points of practical advice directed to the system development professional. These points have grown out of my experience in coordinating the AUTHORIZER'S ASSISTANT expert system for Travel Related Services division of American Express, and are also echoed well within this book.

- Don't use expert systems tools for their own sake. The technology is very seductive, however, business needs should always drive the selection of expert systems as an appropriate technology.
- Never underestimate how good people are at handling what seem like trivial problems. Real "common sense" is not so common.
- Be pragmatic. Pick small and well-defined problems to start with.
- Choose applications that are important to the business and interesting to the expert—without a dedicated champion, a long-term project will never succeed.
- Be willing to accept the experimental nature of projects. Not all will succeed, but something important will be learned from each one.
- Don't underestimate the difficulties of systems integration. Begin planning for connectivity as soon as the data and communications requirements are clear.
- Expect to find frustrations and not a lot of practical experience in moving from the prototype to the integrated delivery system. This caution is especially true if an expert system is to be joined with existing applications in a conventional mainframe environment. (Also be cautious with beware of expert systems targeted to run on mainframes—adequate response times are by no means guaranteed for non-trivial expert systems).
- Don't discard the skills of a lifetime of software project management! Expert systems frequently require incremental prototyping as part of the development cycle in order to understand the problem and its solution, but that doesn't mean that schedules and deliverables are cast aside.

The good news for the pioneering companies is that expert systems techniques have already been used successfully to increase productivity and profitability in a wide variety of environments. The AUTHORIZER'S ASSISTANT, for example, leads to a higher percentage of charges being authorized while at the same time reducing the number of overdue collections. It has condensed the six inch thick Authorization Manual, together with the expertise of the best and most experienced authorizers, to a set of 800 rules, resulting in a less stressful job for the operator (previously they had to study 16 screens of data in just 90 seconds), together with improved consistency—24 hours a day, 7 days a

week, around the world. It also has provided American Express with a permanent retention of the company's expertise, a competitive differentiation, and a proof of concept for other expert systems project at American Express. This book, which highlights many of the pitfalls found by early leaders together with pragmatic ways to navigate around them, should be a useful starting point for the MIS or CIM system professional approaching the challenges of expert system development and wishing to build systems which follow in the footsteps of the success stories.

Ted Markowitz
Director—Technology Strategy
American Express Company
Chairman of SMART-F$
(Society for the Management of AI Resources
and Technology—Financial Services).

Preface

This book is directed to those executives, project managers, and software engineers who have read about expert systems and now want to put the theory and technology into practice. Although expert systems have received a great deal of coverage in the trade press, at the time of this writing the total number of shell products sold is well under 20,000. We feel that a reasonable amount of practical guidance is all that is needed to move expert systems shells to a market position similar to database managers, where each major vendor has sold in excess of 100,000 copies of its product into mainstream corporate environments.

Expert systems are likely to be the next popular productivity tool in business and industrial environments, following the approach of database managers, spreadsheet calculators, word processors, and business graphics. We feel the guidance presented in this book will provide a catalyst to help an emergent trend reach fruition, that is, to put expert systems theory and technology into practice. Industry observers have noted that relational database management systems (RDBMS) moved from development into the mainstream around 1983. Three years later, even small corporations had taken advantage of RDBMS technology to replace various manual systems. Expert systems now show signs of following a similar pattern.

Putting Expert Systems into Practice discusses strategies and tactics for creating robust production applications. It addresses those general principles for success that will not become outdated as new products are released and application areas expand. Strategies and guidelines are emphasized rather than extensive documentation of existing implementations. The discussion includes a broad analysis of leading and emergent shell tools, tool sets, environments, and symbolic programming languages within the context of examining generic expert system development tool features. This *feature-oriented* approach has two benefits: (1) the knowledge presented will not age so quickly as the emerging

field continues to develop rapidly, and (2) the reader will acquire greater facility in analyzing new offerings. The book emphasizes that system designers and implementers must base their creations on sound, stable software engineering and knowledge engineering principles.

The discussion throughout the text concentrates on creating production application in two mainstream corporate environments: Management Information Systems (MIS) for business and Computer-Integrated Manufacturing for industry. Care has been taken to present the material in a terminology and style accessible to readers from traditional commercial and industrial computing backgrounds. Detailed academic programming issues are omitted in favor of strategic design and planning issues.

After an introduction to establish the pragmatic approach, the sequence of chapters follows the expert systems development life cycle, phase by phase, from conception to completion. Chapter 1 introduces the basic knowledge necessary for understanding expert systems, including the artificial intelligence terminology used in expert systems work, the various dimensions involved in implementing expert systems, and an overview of the benefits and limitations of the current technology. Chapter 2 expands on the limitation theme through a common-sense, philosophical discussion of why there is no real "intelligence" in expert systems. Chapters 3 through 9 follow the stages in developing expert systems. Chapter 3 covers *application selection* presenting the general criteria for reviewing the feasibility and cost-effectiveness of candidate applications. The key element is recognizing types of knowledge suitable for expert systems applications. *Application quantification*, the demands for knowledge-representation schemes, inferencing mechanisms, user interfaces, and knowledge-acquisition aids, are addressed in Chapter 4. The requirements of the application, including such traditional issues as machine requirements, size of systems, pricing, and availability must be identified and quantified so appropriate tools can be selected. Chapter 5 addresses the fundamentals of expert system *shell tool set selection.* It analyzes the feature requirements examined in Chapter 4 for nearly 30 expert systems shell building tools, to illustrate the process of selecting the most appropriate tools for the job at hand.

Chapter 6 covers the broader issues of *configuration selection,* outlining the needs for and disadvantages of using specialized programming environments, AI languages, support tools, and shell tools. Strategies and procedures for acquiring knowledge from experts, *knowledge acquisition,* is covered in Chapter 7. This stage is critical to the success of the project and usually requires intensive time and prototyping iterations. Before the new expert system can become a productive part of an enterprise, it must be integrated with a corporation's existing systems.

Issues of *systems integration* are addressed in Chapter 8. With increasing frequency integrated, rather than stand-alone, hardware/software systems are being required by MIS and CIM directors. *System startup* and *management,* or people roles, are the two themes of Chapter 9. People issues are central in moving the completed development into the daily production of a corporation, and this chapter discusses the roles of knowledge engineers, domain experts, consultants, managers, and knowledge-base administrators and support staff, which are at the heart of any project's success. Finally, Chapter 10 addresses some of the issues influencing future trends, in both the short and long terms, in expert systems and AI developments.

Although broad similarities between the structure and use of the expert systems technology and traditional software tools exist, various differences are critical. This distinction is most clearly illustrated by the expert system development life cycle, which may involve a series of prototypes and several iterations through the knowledge-acquisition loop. Some of the principles and strategies presented are common in standard software-engineering practices. We have tried to distinguish those development features and practices peculiar to expert systems developments, while maintaining an appropriate view of software development as a whole. This perspective is often manifested as additional knowledge needed to transform traditional systems analysis into *knowledge engineering.* Building successful applications is based on a clear understanding of the philosophical and practical distinctions between expert systems technology, traditional software systems, and the capabilities of real human experts.

Judging from the literature and our own observations, we believe too many corporations have restricted their expert systems technology to toy problem prototypes using simplistic shells on small computers. Corporations do want more robust and effective results, but may lack the knowledge of where to start or how to proceed. The limited features of small PC-based shells cannot be expected to produce applications with the capabilities of such large expert systems as PROSPECTOR, DENDRAL, XCON, or DRILLING ADVISOR. For many serious production applications, we argue, the developer may have to reject small PC shells and select something more capable running on a larger machine, such as a DEC VAX or IBM mainframe. Fortunately, this process is being facilitated by an increasing number of shell tools ported from AI LISP environments to traditional languages like C, so they can run on general-purpose hosts and be integrated into mainstream environments. What is next needed to generate real-world production-quality systems is knowing how to select the application and the appropriate tools, how to size and specify the configuration, and how to proceed with knowledge

capture, codification, testing, and system integration. *Putting Expert Systems into Practice* addresses these strategic issues and introduces computer executives, project managers, and software engineers to the principles required for a successfully fielded implementation.

We thank the numerous academicians and computer industry leaders who have inspired us and helped us complete this work. Special acknowledgment is due Maharishi Mahesh Yogi (the best knowledge engineer we ever met), John Hagelin, Bruce Lester, Robert Fertig, David Rosenbloom, Dave Nelson, Jeffrey Perrone, Jon Cooper, Ella Pappius, Donna Goldin, Steven Olmstead, Robert Baron, Ray Ford, Paula Baudoux, Alex Green, John Sorflaten, Su Bowerman, Tom Miller, Paul Brook, and our many friends among vendors.

Robert G. Bowerman and David E. Glover, Ph.D.
Heliosoft, Inc.
P.O. Box 1205
Fairfield, Iowa 52556
(515) 472-2185

Contents

Chapter 1
Understanding Expert Systems: Technology and Capabilities

SECTION HEADINGS

OVERVIEW

This introductory chapter of *Putting Expert Systems Into Practice* provides two functions. First, it introduces the key concepts and the discipline's terminology. Second, it presents an overview of the book's major thesis—the creation of robust, expert system production applications. Throughout, the discussion focuses on the process of recognizing strategies and tactics for getting expert systems up and running as integral components in an enterprises' business or manufacturing systems.

1-1. EXPERT SYSTEMS: FROM AI RESEARCH TO PRODUCTION APPLICATIONS

What is an expert system? An *expert system* is a system of software or combined software and hardware capable of competently executing a