Artificial Intelligence Applications in Chemistry

EDITED BY

Thomas H. Pierce

Bruce A. Hohne

ACS SYMPOSIUM SERIES 306

Artificial Intelligence Applications in Chemistry

Thomas H. Pierce, EDITOR
Rohm and Haas Company

Bruce A. Hohne, EDITOR
Rohm and Haas Company

Developed from a symposium sponsored by
the Division of Computers in Chemistry
at the 190th Meeting
of the American Chemical Society,
Chicago, Illinois, September 8–13, 1985

American Chemical Society, Washington, DC 1986

Library of Congress Cataloging-in-Publication Data

Artificial intelligence applications in chemistry.
(ACS symposium series, ISSN 0097-6156; 306)

"Developed from a symposium sponsored by the Division of Computers in Chemistry at the 190th meeting of the American Chemical Society, Chicago, Illinois, September 8-13, 1985."

Includes bibliographies and indexes.

1. Chemistry—Data processing—Congresses.
2. Artificial intelligence—Congresses.

I. Pierce, Thomas H., 1952- . II. Hohne, Bruce A., 1954- . III. American Chemical Society. Division of Computers in Chemistry. IV. American Chemical Society. Meeting (190th: 1985: Chicago, Ill.) V. Series.

QD39.3.E46A78 1986 542'.8 86-3315
ISBN 0-8412-0966-9

PRINTED IN THE UNITED STATES OF AMERICA

ACS Symposium Series

M. Joan Comstock, *Series Editor*

FOREWORD

The ACS SYMPOSIUM SERIES was founded in 1974 to provide a medium for publishing symposia quickly in book form. The format of the Series parallels that of the continuing ADVANCES IN CHEMISTRY SERIES except that, in order to save time, the papers are not typeset but are reproduced as they are submitted by the authors in camera-ready form. Papers are reviewed under the supervision of the Editors with the assistance of the Series Advisory Board and are selected to maintain the integrity of the symposia; however, verbatim reproductions of previously published papers are not accepted. Both reviews and reports of research are acceptable, because symposia may embrace both types of presentation.

PREFACE

ARTIFICIAL INTELLIGENCE (AI) is not a new field, as AI dates back to the beginnings of computer science. It is not even new to the field of chemistry, as the DENDRAL project dates back to the early 1960s. AI is, however, just beginning to emerge from the ivory towers of academia. To many people it is still just a buzz word associated with no real applications. Because AI work involves people from multiple disciplines, the work is difficult to locate and the application is sometimes difficult to understand.

We decided that now would be a good time for an AI book for several reasons: (1) enough applications can now be presented to expose newcomers to many of the possibilities that AI has to offer, (2) showing what everyone else is doing with AI should generate new interest in the field, and (3) we felt an overview was needed to collect the different areas of AI applications to help people who are starting to apply AI techniques to their disciplines. The final and possibly most important reason is our personal interest in the field.

Chemistry is an ideal field for applications in AI. Chemists have been using computers for years in their day-to-day work and are quite willing to accept the aid of a computer. In addition, the DENDRAL project, throughout its long history, has graduated many chemists already trained in AI. It is not surprising that chemistry is one of the leading areas for AI applications. Scientists have been developing the theories of chemistry for centuries, but the standard approach taken by a chemist to solve a problem is heuristic; past experience and rules of thumb are used. AI offers a method to combine theory with these rules. These systems will not replace chemists, as is commonly thought; but rather, these programs will assist chemists in performing their daily work.

Computer applications developed from theoretical chemistry tend to be algorithmic and numerical by nature. AI applications tend to be heuristic and symbolic by nature. Multilevel expert systems combine these techniques to use the heuristic power of expert systems to direct numerical calculations. They can also use the results of numerical calculations in their symbolic processing. The problems faced by chemists today are so complex that most require the added power of the multilevel approach to solve them.

Defining exactly which applications constitute AI is difficult in any field. The problem in chemistry is even worse because chemical applications that use AI methods often use numerical calculations. Some applications that are strictly numerical accomplish tasks similar to AI programs. The key feature used to limit the scope of this book was symbolic processing. The work presented includes expert systems, natural language applications, and manipulation of chemical structures.

The book is divided into five sections. The first chapter is outside this structure and is an overview of the technology of expert systems.

The book's first section, on expert systems, is a collection of expert-system applications. Expert systems can simplistically be thought of as computerized clones of an expert in a particular specialty. Various schemes are used to capture the expert's knowledge of the specialty in a manner that the computer can use to solve problems in that field. Expert-systems technology is the most heavily commercialized area in AI as shown by the wide variety of applications that use this technology. These applications help show the breadth of problems to which AI has been applied. Much of the work from other sections of the book also uses expert-system techniques in some manner.

The second section, on computer algebra, details chemical applications whose emphasis is on the mathematical nature of chemistry. As chemical theories become increasingly complex, the mathematical equations have become more difficult to apply. Symbolic processing simplifies the construction of mathematical descriptions of chemical phenomena and helps chemists apply numerical techniques to simulate chemical systems. Not only does computer algebra help with complex equations, but the techniques can also help students learn how to manipulate mathematical structures.

The third section, on handling molecular structures, presents the interface between algebra and chemical reactions. The storage of molecular representations in a computer gives the chemist the ability to manipulate abstract molecular structures, functional groups, and substructures. The rules that govern the changes in the molecular representations vary with each approach. Molecules can be described as connected graphs, and the theorems of graph theory can be used to define their similarity. Another approach uses heuristic rules for chemical substructures to define and display molecules.

The fourth section, on organic synthesis, discusses methods to construct complex organic syntheses using simple one-step reactions. Many groups have used the computer to search for synthetic pathways for chemical synthesis in the past. Each approach must deal with the problem of multiple possible pathways for each step in the reaction. The chapters in this section apply AI techniques to select "good" paths in the synthesis.

The final section, on analytical chemistry, is a combination of structure-elucidation techniques and instrumental optimizations. Instrumental analysis can be broken into several steps: method development, instrumental optimization, data collection, and data analysis. The trend today in analytical instrumentation is computerization. Data collection and analysis are the main reasons for this. The chapters in this section cover all aspects of the process except data collection. Organic structure elucidation is really an extension of data analysis. These packages use spectroscopic data to determine what structural fragments are present and then try to determine how those fragments are connected. Different people have used both individual spectroscopic techniques and combinations of techniques to solve this very difficult problem. This area holds great promise for future work in AI.

We gratefully acknowledge the efforts of all the authors who contributed their time and ideas to the symposium from which this book was developed. We also thank the staff of the ACS Books Department for their helpful advice. Finally, we acknowledge the encouragement and support we received from our management at Rohm and Haas Company.

THOMAS H. PIERCE
Rohm and Haas Company
Spring House, PA 19477

BRUCE A. HOHNE
Rohm and Haas Company
Spring House, PA 19477

January 13, 1986

CONTENTS

1

Artificial Intelligence: The Technology of Expert Systems

Dennis H. Smith

Biotechnology Research and Development, IntelliGenetics, Inc., Mountain View, CA 94040

Expert systems represent a branch of artificial intelligence that has received enormous publicity in the last two to three years. Many companies have been formed to produce computer software for what is predicted to be a substantial market. This paper describes what is meant by the term expert system and the kinds of problems that currently appear amenable to solution by such systems. The physical sciences and engineering disciplines are areas for application that are receiving considerable attention. The reasons for this and several examples of recent applications are discussed. The synergism of scientists and engineers with machines supporting expert systems has important implications for the conduct of chemical research in the future; some of these implications are described.

Expert systems represent a sub-discipline of artificial intelligence (AI). Before beginning a detailed discussion of such systems, I want to outline my paper so that the focus and objectives are clear. The structure of the paper is simple. I will:

- Describe the technology of expert systems
- Discuss some areas of application related to chemistry
- Illustrate these areas with some examples

Although the structure of the paper is simple, my goal is more complex. It is simply stated, but harder to realize: I want to demystify the technology of applied artificial intelligence and expert systems.

The word mystify means "to involve in mystery, to make difficult to understand, to puzzle, to bewilder." Therefore, I will try to remove some of the mystery, to make things easier to understand, to clarify what the technology is and what it can (and cannot) do.

I am going to discuss a special kind of computer software, but software nonetheless.

0097-6156/86/0306-0001$06.00/0

Everything I will describe could be built from the ground up using assembly language, BASIC or any other computer language. In the future, some expert systems will certainly be built using languages such as Fortran, C or PASCAL as opposed to LISP and PROLOG which are currently in vogue. So there is no mystery here. What is different, but is still not mysterious, is the approach taken by AI techniques toward solving symbolic, as opposed to numeric, problems. I discuss this difference in more detail, below. Most readers of this collection of papers will be scientists and engineers, engaged in research, business or both. They expect new technologies to have some substantial practical value to them in their work, or they will not buy and use them. So I will stress the practicality of the technology.

Where is the technology currently? Several descriptions of the marketplace have appeared over the last year. Annual growth rates for companies involved in marketing products based on AI exceed 300%, far outstripping other new computer-based applications, such as control and management of information networks, private telephone networks, automation of the home and factory. Of course, those are growth rates, not market sizes or dollar volumes. The technology will ultimately be successful only to the extent that it does useful work, by some measure. In this paper I illustrate some areas where useful work can be, and is being, done. There are many expert systems under development at major corporations, in the areas of chemistry, chemical engineering, molecular biology and so forth. Because many of these systems are still proprietary, the examples I will discuss are drawn from work that is in the public domain. However, the casual reader will easily be able to generalize from my examples to his or her own potential applications.

The Technology of Expert Systems

I am going to begin my discussion of the technology of expert systems with two provocative statements. The first is:

Knowledge engineering is the technology base of the "Second Computer Age"

It is possible to use knowledge, for example, objects, facts, data, rules, to manipulate knowledge, and to cast it in a form in which it can be used easily in computer programs, thereby creating systems that solve important problems.

The second statement is:

What's on the horizon is not just the Second Computer Age, it's the important one!

We are facing a second computer revolution while still in the midst of the first one! And it's probably the important revolution.

Characteristics and Values of Expert Systems. What leads me to make such bold and risky statements? The answer can be summarized as follows. First, knowledge is power. You can't solve problems using any technology unless you have some detailed knowledge about the problem and how to solve it. This fact seems so obvious that it is unnecessary to state it. Many systems will fail, however, because the builders will attempt to build such systems to solve ill-defined problems.

Second, processing of this knowledge will become a major, perhaps dominant part of the computer industry. Why? Simply because most of the world's problem solving activities involve symbolic reasoning, not calculation and data processing. We have constructed enormously powerful computers for performing calculations, our *number crunchers*. We devote huge machines with dozens of disk drives to database management systems. Our need for such methods of computing will not disappear in the future. However, when we have to fix our car, or determine why a processing plant has shut down, or plan an organic synthesis, we don't normally solve sets of differential equations or pose queries to a large database. We might use such numerical solutions or the results of such queries to help solve the problem, but we are mainly reasoning, not calculating.

How do we construct programs that aid us in reasoning as opposed to calculating? AI is the underlying science. It has several sub-disciplines, including, for example, robotics, machine vision, natural language understanding and expert systems, each of which will make a contribution to the second computer age. My focus is on expert systems.

Knowledge engineering is the technology behind construction of expert systems, or knowledge systems, or expert support systems. Such systems are designed to advise, inform and solve problems. They can perform at the level of experts, and in some cases exceed expert performance. They do so not because they are "smarter" but because they represent the collective expertise of the builders of the systems. They are more systematic and thorough. And they can be replicated and used throughout a laboratory, company or industry at low cost.

There are three major components to an expert system:

- the knowledge base of facts and heuristics
- the problem-solving and inference engine
- an appropriate human-machine interface

The contents of a knowledge base, the facts and rules, or *heuristics*, about a problem will be discussed shortly. The problem-solving and inference *engine* is the component of the system that allows rules and logic to be applied to facts in the knowledge base. For example, in rule-based expert systems, "IF-THEN" rules (production rules) in a knowledge base may be analyzed in two ways:

- in the forward, or data-driven direction, to solve problems by asserting new facts, or conditions, and examining the consequences, or conclusions
- in the backward, or goal-driven direction, to solve problems by hypothesizing conclusions and examining the conditions to determine if they are true.

For the purposes of this paper, I will not describe the inference procedures further. I will also say very little about the human-machine interface. However, since expert systems are designed to be built by experts and used by experts and novices alike, the interface is of crucial importance. The examples discussed later illustrate how powerful interfaces are implemented through use of high resolution bit-mapped graphics, menu and "button" driven operations, a "mouse" as a pointing device, familiar icons to represent objects such as schematics, valves, tanks, and so forth.

The Knowledge Base. The knowledge base holds symbolic knowledge. To be sure, the knowledge base can also contain tables of numbers, ranges of numerical values, and some numerical procedures where appropriate. But the major content consists of facts and heuristics.

The facts in a knowledge base include descriptions of objects, their attributes and corresponding data values, in the area to which the expert system is to be applied. In a process control application, for example, the factual knowledge might include a description of a physical plant or a portion thereof, characteristics of individual components, values from sensor data, composition of feedstocks and so forth.

The heuristics, or rules, consist of the judgemental knowledge used to reason about the facts in order to solve a particular problem. Such knowledge is often based on experience, is used effectively by experts in solving problems and is often privately held. Knowledge engineering has been characterized as the process by which this knowledge is "mined and refined" by builders of expert systems. Again, using the motif of process control, such knowledge might include rules on how to decide when to schedule a plant or subsystem for routine maintenance, rules on how to adjust feedstocks based on current pricing, or rules on how to diagnose process failures and provide advice on corrective action.

Expert systems create value for groups of people, ranging from laboratory units to entire companies, in several ways, by:

- capturing, refining, packaging, distributing expertise; an "an expert at your fingertips";
- solving problems whose complexity exceeds human capabilities;
- solving problems where the required scope of knowledge exceeds any individual's;
- solving problems that require the knowledge and expertise of several fields (fusion);
- preserving the group's most perishable asset, the organizational memory;
- creating a competitive edge with a new technology.

The packaging of complex knowledge bases leads to powerful performance. This performance is possible due to the thoroughness of the machine and the synthesis of expertise from several experts. Similarly, if the knowledge base cuts across several disciplines, the *fusion* of such knowledge creates additional value. An obvious value of expert systems is what is referred to above as preserving the organizational memory. Many organizations will have to confront the loss of some of their most valuable experts over the next few years, whether through graduation, death, a new job, or retirement. Several companies are turning to expert systems in order to capture the problem-solving expertise of their most valuable people. This preserves the knowledge and makes it available in easily accessible ways to those who must assume the responsibilities of the departing experts.

Considering commercial applications of the technology, expert systems can create value through giving a company a competitive edge. This consideration means that the first companies to exploit this technology to build useful products will obviously be some steps ahead of those that do not.

Some Areas of Application. I next summarize some areas of application where expert systems exist or are being developed, usually by several laboratories. Some of these areas are covered in detail in other presentations as part of this symposium. I want to emphasize that this is a partial list primarily of scientific and engineering applications. A similar list could easily be generated for operations research, economics, law, and so forth. Some of the areas are outside strict definitions of the fields of chemistry and chemical engineering, but I have included them to illustrate the breadth of potential applications in related disciplines.

- Medical diagnosis and treatment
- Chemical synthesis and analysis
- Molecular biology and genetic engineering
- Manufacturing: planning and configuration
- Signal processing: several industries
- Equipment fault diagnosis: several industries
- Mineral exploration
- Intelligent CAD
- Instrumentation: set-up, monitoring, data analysis
- Process control: several industries

Many readers will have read about medical applications, the MYCIN and INTERNIST programs. There are many systems being developed to diagnose equipment failures. Layout and planning of manufacturing facilities are obvious applications. Chemistry and molecular biology systems were among the earliest examples of expert systems and are now embodied in commercial systems.

There is a suite of related applications involving signal processing. Whether the data are from images, oil well-logging devices, or military sensor systems, the problems are the same; vast amounts of data, only some of which are amenable to numerical analysis. Yet experts derive valid interpretations from the data. Systems have already been built to capture this expertise.

There are many diagnosis and/or advisory systems under development, applied to geology, nuclear reactors, software debugging and use, manufacturing and related financial services.

There are several applications to scientific and engineering instrumentation which especially relevant to chemistry and chemical engineering. These include building into instruments expertise in instrument control and data interpretation, to attempt to minimize the amount of staff time required to perform routine analyses and to optimize the performance of a system. There are several efforts underway in process control, focused currently in the electrical power and chemical industries.

Before looking at some applications in more detail, let me briefly describe why the number and scope of applications is increasing so dramatically.

The Technology is Maturing Rapidly. The work that computers are being required to do is increasingly knowledge intensive. For example, instrument manufacturers are producing more powerful computer systems that are integral to their product lines. These systems are expected to perform more complex tasks all the time, i.e., to be in some sense "smarter". Two developments are proceeding in parallel with this requirement for "smarter" systems. The software technology for building expert systems is maturing rapidly. At the same time, workstations that support AI system development are making a strong entry into the computer market. For the first time, the hardware and software technology are at a point where development of systems can take place rapidly.

Beginning in 1970, programming languages such as LISP became available. Such languages made representation and manipulation of symbolic knowledge much simpler than use of conventional languages. Around 1975, programming *environments* became available. In the case of LISP, its interactive environment, INTERLISP, made system construction, organization and debugging much more efficient. In 1980, research work led to systems built on top of LISP that removed many of the requirements for programming, allowing system developers to focus on problem solving rather than writing code. Some of these research systems have now evolved to become commercial products that dramatically simplify development of expert systems. Such products, often referred to as *tools*, are specifically designed to aid in the construction of expert systems and are engineered to be usable by experts who may not be programmers.

Supporting evidence for the effects of these developments is found by examining the approximate system development time for some well known expert systems. Systems begun in the mid-1960's, DENDRAL and MACSYMA required of the order of 40-80 man-years to develop. Later systems of similar scope required less and less development time, of the order of several man years, as programming languages and system building tools matured. With current, commercially available tools, developers can expect to build a prototype of a system, with some assistance, in the order of one month. The prototype that results already performs at a significant level of expertise and may represent the core of a subsequent, much larger system (examples are shown below). Such development times were simply impossible to achieve with the limited tools that existed before mid-1984.

Developing Expert Systems. How has such rapid progress been achieved? The improvement in hardware and software technologies is obviously important. Another important factor is that people are becoming more experienced in actually building systems. There has emerged, from the construction of many systems designed for diverse applications, a strong model for the basic steps required in constructing an expert system. The four major steps are as follows:

- Select an appropriate application
- Prototype a "narrow vertical slice"
- Develop the full system
- Field the system, including maintenance and updates

First, one must select an appropriate application. There are applications that are so simple, that require so little expertise, that it is not worth the time and money to emulate human performance in a machine. At the other end of the spectrum, there are many problems whose methods of solution are poorly understood. For several reasons, these are not good candidates either. In between, there are many good candidates, and in the next section I summarize some of the rules for choosing them.

Second, a prototype of a final system is built. This prototype is specifically designed to have limited, but representative, functionality. During development of the prototype, many important issues are resolved, for example, the details of the knowledge representation, the man-machine interface, and the complexity of the rules required for high performance. *Rapid prototyping* is already creeping into the jargon of the community. The latest expert system building tools are sufficiently powerful that one can sit down and try various ideas on how to approach the problem, find out what seems logical and what doesn't, reconstruct the knowledge base into an entirely different form, step through execution of each rule and correct the rules interactively. This approach differs substantially from traditional methods of software engineering.

The third step, however, reminds us that we do have to pay attention to good software development practices if a generally used, and useful system is to result from the prototype. Development of a full system, based at least in part on the prototype, proceeds with detailed specifications as the system architects define and construct its final form.

The last step is just as crucial as its predecessors. The system must be tested in the field, and the usual requirements in the software industry for maintenance and updates pertain.

The primary differences, then, between development of expert systems and more traditional software engineering are found in steps one and two, above. First, the problems chosen will involve symbolic reasoning, and will require the transfer of expertise from experts to a knowledge base. Second, rapid prototyping, the "try it and see how it works, then fix it or throw it away" approach will play an important role in system development.