

Expert System Applications in Chemistry

ACS SYMPOSIUM SERIES **408**

Expert System Applications in Chemistry

Bruce A. Hohne, EDITOR
Rohm and Haas Company

Thomas H. Pierce, EDITOR
Rohm and Haas Company

Developed from a symposium sponsored
by the Division of Computers in Chemistry
at the 196th National Meeting
of the American Chemical Society,
Los Angeles, California,
September 25–30, 1988



American Chemical Society, Washington, DC 1989



Library of Congress Cataloging-in-Publication Data

Expert system applications in chemistry

Bruce A. Hohne, editor, Thomas H. Pierce, editor

p. cm.—(ACS Symposium Series, 0097-6156; 365).

Developed from a symposium sponsored by the Division of Computers in Chemistry at the 196th National Meeting of the American Chemical Society, Los Angeles, California, September 25-30, 1988.

Includes bibliographical references.

ISBN 0-8412-1681-9

1. Chemistry—Data processing—Congresses. 2. Expert systems—Congresses.

I. Hohne, Bruce A., 1954- . 1. II. Pierce, Thomas H., 1952- . 2. III. American Chemical Society. Division of Computers in Chemistry. IV. American Chemical Society. Meeting (196th: 1988: Los Angeles, Calif.). V. Series

QD39.3.E46E96 1989
540'.285'633—
dc20

89-35271

CIP

Copyright © 1989

American Chemical Society

All Rights Reserved. The appearance of the code at the bottom of the first page of each chapter in this volume indicates the copyright owner's consent that reprographic copies of the chapter may be made for personal or internal use or for the personal or internal use of specific clients. This consent is given on the condition, however, that the copier pay the stated per-copy fee through the Copyright Clearance Center, Inc., 27 Congress Street, Salem, MA 01970, for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. This consent does not extend to copying or transmission by any means—graphic or electronic—for any other purpose, such as for general distribution, for advertising or promotional purposes, for creating a new collective work, for resale, or for information storage and retrieval systems. The copying fee for each chapter is indicated in the code at the bottom of the first page of the chapter.

The citation of trade names and/or names of manufacturers in this publication is not to be construed as an endorsement or as approval by ACS of the commercial products or services referenced herein; nor should the mere reference herein to any drawing, specification, chemical process, or other data be regarded as a license or as a conveyance of any right or permission to the holder, reader, or any other person or corporation, to manufacture, reproduce, use, or sell any patented invention or copyrighted work that may in any way be related thereto. Registered names, trademarks, etc., used in this publication, even without specific indication thereof, are not to be considered unprotected by law.

PRINTED IN THE UNITED STATES OF AMERICA

American Chemical Society
Library

1155 16th St., N.W.

Washington, D.C. 20036

In Expert System Applications in Chemistry; Hohne, B., et al.;
ACS Symposium Series; American Chemical Society: Washington, DC, 1989.

ACS Symposium Series

M. Joan Comstock, *Series Editor*

1989 ACS Books Advisory Board

Paul S. Anderson
Merck Sharp & Dohme Research
Laboratories

Alexis T. Bell
University of California—Berkeley

Harvey W. Blanch
University of California—Berkeley

Malcolm H. Chisholm
Indiana University

Alan Elzerman
Clemson University

John W. Finley
Nabisco Brands, Inc.

Natalie Foster
Lehigh University

Marye Anne Fox
The University of Texas—Austin

G. Wayne Ivie
U.S. Department of Agriculture,
Agricultural Research Service

Mary A. Kaiser
E. I. du Pont de Nemours and
Company

Michael R. Ladisch
Purdue University

John L. Massingill
Dow Chemical Company

Daniel M. Quinn
University of Iowa

James C. Randall
Exxon Chemical Company

Elsa Reichmanis
AT&T Bell Laboratories

C. M. Roland
U.S. Naval Research Laboratory

Stephen A. Szabo
Conoco Inc.

Wendy A. Warr
Imperial Chemical Industries

Robert A. Weiss
University of Connecticut

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

THIS IS THE SECOND TIME we have organized a symposium on artificial intelligence (AI) and prepared a book based on the symposium. Since our first effort (*Artificial Intelligence Applications in Chemistry*, ACS Symposium Series 306), the field has changed dramatically, in large part because of the advent of commercially available PC-based expert system software. Expert system development has moved out of the hands of the AI gurus and into the hands of practicing chemists. This is a positive step for the field of chemistry. There no longer appears to be a danger of AI being dismissed as a fad or held back through lack of interest from the chemical community. Now, the issues have become cost effectiveness, intellectual property protection, and application selection.

The main thrust of this book is to present examples of how expert systems can solve chemical problems. To make the book more useful to novices in the field of artificial intelligence, we have written a brief introductory chapter explaining expert systems. The glossary at the end of the introduction should be of help to novices and experts alike. We have also included highlights of a panel discussion held at the symposium. The panel was posed the question, "Can knowledge bases be made generally available in a useful format?" Unfortunately, more issues were raised than questions answered.

We wish to thank the authors who contributed their time and ideas to the symposium and the book. In addition, we would like to thank the staff of the ACS Books Department, both for their advice and for providing us the opportunity to publish this book. Finally, we acknowledge the encouragement and support we received from our management at Rohm and Haas Company.

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

THOMAS H. PIERCE
Rohm and Haas Company
Bristol, PA 19007

July 5, 1989

Chapter 1

Introduction

Expert System Applications in Chemistry

Bruce A. Hohne¹ and Thomas H. Pierce²

¹Rohm and Haas Company, Spring House, PA 19477

²Rohm and Haas Company, Bristol, PA 19007

This symposium series volume is the second of its kind (1). Both symposia were organized with several purposes in mind. The first, and most general, is simply to expose the chemical community to expert systems. Expert systems should be of interest to people in a wide variety of fields of chemistry. The second is to educate chemists in the capabilities of expert systems; both what they can and can not do for them. Finally, by presenting a variety of applications, it was hoped that attendees would generate further new ideas for expert system applications. The second symposium presented the additional opportunity to review the progress of some of the work described in the first symposium.

The first symposium, presented at the fall meeting of the American Chemical Society in 1985, was open to any type of artificial intelligence application. The second symposium was restricted to expert systems. This is only a minor restriction because most practical artificial intelligence applications to date have used expert systems in one form or another. Below is a brief overview of expert systems. More information on expert systems for chemistry can be found in reference 2. General expert system information can be found in references 3 and 4. Because artificial intelligence has developed its own vocabulary, a short glossary is also included.

Expert Systems

Expert systems are programs that attempt to solve problems in a way similar to how a human expert would solve them. They incorporate "rules of thumb" that experts in the field have developed through years of experience. The problems attacked are not necessarily procedural, they are often vague, complex, and can contain incomplete or inexact information.

Expert systems contain three basic pieces: a knowledge base, an inference engine, and a user interface. The knowledge base contains the information which the program uses to reach decisions. A key

0097-6156/89/0408-0002\$06.00/0

© 1989 American Chemical Society

difference between expert systems and classical computer programs is the fact that the knowledge is separated from the program. The inference engine is the program that manipulates the knowledge base to reach these decisions. The user interface allows the program and chemist to communicate with each other in an effective manner. These three pieces of an expert system are described below.

Knowledge Base. The knowledge base of an expert system is a resource of information about a specific domain, or problem area. The knowledge base is the most important part of the expert system, and the most difficult to construct. The completeness and accuracy of the knowledge base will determine how well the expert system will perform solving problems. The scope of problems which can be solved is completely determined by the scope of the knowledge base.

Knowledge must be encoded in such a way that it can be a) entered into the computer; b) manipulated by the inference engine; c) understood by the expert. There are several common ways to encode the expert's knowledge, each with its own advantages and disadvantages. The encoding scheme must match the underlying structure of the knowledge.

The two most common methods of encoding knowledge are production rules and frames. Sometimes both methods are used. Production rules take the form: IF x is true THEN y is true. For example:

IF the pH is less than 7
THEN the solution is acidic.

The left hand side (x) may contain any number of clauses combined by Boolean algebra operators. In all but the simplest expert systems it is possible to include the expert's confidence in the conclusion. A frame describes hierarchical dependencies between objects. In a frame, the upper level (or parent) object passes attributes to the objects beneath it in the hierarchy (children). In other words, children inherit attributes from their parents. For example:

```
(FRAME - Alcohol
  (SLOT - Reactions
    (VALUE - "Oxidized by permanganate")
    (VALUE - "..."))
  (SLOT - Infrared Absorbances
    (VALUE - 3600 cm-1 - 3100 cm-1)))

(FRAME - Ethanol
  (SLOT - Chemical Class
    (VALUE - Alcohol)))
```

Inference Engine. The inference engine is the central program which manipulates the rules and facts in the knowledge base to reach conclusions. The structure of the inference engine depends strongly upon the type of knowledge base which the expert system incor-

porates. Inherent in all of the program structures, however, is a basic set of functions which expert systems perform. These functions are described below using an example knowledge base which incorporates production rules.

The inference engine may approach the problem from either the top or the bottom, beginning with either facts or conclusions. If a user begins with several hypotheses and wants to determine which, if any, are correct, then the program should examine all the facts using a goal-directed (also called goal-driven) approach. However, if the user begins with a series of facts which are known to be true and wants to determine what conclusions can be reached, the program should use a data-directed (also called data-driven) approach.

A goal-directed expert system begins with a limited set of possible hypotheses and attempts to prove the validity of each one. This type of expert system uses a reverse-chaining (also called backward-chaining) algorithm. The knowledge base is searched to find a rule which concludes the initial hypothesis. The IF-clauses from this rule then becomes the hypotheses for the next level of the search. The process continues until all of the remaining IF-clauses are known to be true (hypothesis is true) or until no more rules apply (hypothesis is false). This approach starts at the bottom (conclusions) and works its way to the top (facts). An example of this mechanism is illustrated by an HPLC trouble-shooting problem:

A scientist is experiencing problems with an HPLC. One possible problem is a broken pump. The expert system searches its rules for one that concludes the pump is broken. It may find a rule like:

```
IF  "there is no solvent flow"  
    "all tubing is properly connected"  
    "there are no obstructions"  
    "the solvent program module is functioning"  
THEN "the pump is broken".
```

The next step would be to search for rules which conclude "there is no solvent flow". The process would continue until the conclusion "the pump is broken" is either proved or disproved. In this example, the chemist would be asked to input information whenever an IF clause was an observable symptom of an HPLC problem.

Data-directed expert systems begin with a list of the facts known to be true, and see what conclusions can be drawn from those facts. This type of expert system uses a forward-chaining mechanism. Each rule in the knowledge base is tested to see if all of its IF clauses are contained in the list of known facts. When such a rule is found, the system adds the THEN-clauses from the rule to the list of known facts. All the rules in the knowledge base are scanned repetitively until no new facts can be concluded. An example of using forward-chaining is illustrated by a structure elucidation problem based on an IR spectrum:

A scientist wants to identify a compound for which the peak locations in an IR spectra are known. The list of known facts is the location and intensity of the peaks in the spectrum. The program searches the knowledge base for a rule which has these peak locations in its IF-clause. For example, assume there are peaks at 3300, 1410, 1300, and 1050 cm^{-1} . That would cause the following rule to fire:

IF "there is a large peak between 3000-3500 cm^{-1} "

THEN "there is evidence for an alcohol".

This would place the fact "there is evidence for an alcohol" on the list of known facts. On the second pass through the knowledge base the following rule could fire:

IF "there is evidence for an alcohol"

"there is a peak between 1460-1400 cm^{-1} "

"there is a peak between 1350-1260 cm^{-1} "

"there is a peak between 1080-1010 cm^{-1} "

THEN "there is evidence for primary alcohol".

The system continues to search its knowledge base for rules which will identify probable structures of the compound.

The two examples above show that the structure of a problem determines what type of search to use. In the first example, there is only a limited number of components which could be broken, and a large number of possible symptoms. A data-driven approach to this problem would be to ask the scientist for all the symptoms, some of which may require disassembling the HPLC to determine. The problem, therefore, demands a goal-directed approach.

In the second example, the scientist began with a set of facts and was looking for all possible conclusions. The goal-directed approach to this problem would be to attempt to prove whether each one of the millions of compounds reported in the chemical literature could be responsible for the observed spectrum. Again, the structure of the problem determines the structure of the expert system.

The performance of an expert system may be increased by using heuristic rules to eliminate solutions which may be possible, but are unlikely given the constraints of the problem. This is how human experts solve complicated problems. This added knowledge is called meta-knowledge. In knowledge bases which use production rules, meta-knowledge takes the form of meta-rules. These rules instruct the expert system how to choose which rule to use when more than one set is relevant. Using organic synthesis as an example, a meta-rule might take the form:

IF: "multiple reactions have the same product"

THEN: "use the one with the highest product yield".

User Interface. The user interface can be separated into two parts; the part used by the expert to build the knowledge base, and the part used by the novice to solve domain problems. The user interface must provide the expert with the following functions: a method to

enter knowledge, a method to modify rules, and a method to test and manage the knowledge base. The contents of the knowledge base must be easily expandable. When an expert adds knowledge, or rules, the program must check for contradictions with previous rules. It should also test these new rules against previously defined problems. When the system produces an incorrect answer, the expert must have ways to examine the knowledge base and make corrections to it.

The user interface must make it possible for the novice to quickly and effectively solve problems. The functions required for this include: prompting for information, explaining why the information is needed, and telling how a conclusion is reached. Explanations usually display the progression of rules used by the system to solve the problem. Some of the more sophisticated commercial expert systems take advantage of computer graphics to perform these functions.

Developing Your Own Expert System

The references contain a great deal of information about developing expert systems. Rather than repeat their advice, this section presents the issues from a chemist's point of view. There are five major issues which must be considered when developing an expert system:

1. Should you develop in-house expertise in expert systems, rely on outside consultants, or some combination of the two?
2. Should your organization be involved in expert system development? Be aware of the limitations of the technology.
3. Application selection, including the domain expert.
4. Select what computer platform(s) the system must be able to run on.
5. Select what software package to purchase.

The first two issues are political, and must be decided within your organization. The remaining issues have a political component, but are mostly technical issues.

Selecting an application to automate using an expert system is a detailed process, covered in the references. The entire process reduces to determining if the project is cost justifiable, and determining that expert systems are the best technology to meet your needs. Areas which should be rich with possible applications include: instrumentation (optimization, trouble-shooting, operation); chemical structure elucidation; computerized manuals (government regulations, corporate guidelines); product specific (applications specific to one vendor's product or product line); safety (risk assessment, safety reviews, emergency procedures); process control (optimization, trouble-shooting, control).

Selecting the hardware and software to run an application are interrelated, because some expert system software runs only on certain hardware. Hardware options vary from personal computers to mainframes. Software pricing runs from about \$100 to 500 times

that. With the power of today's expert system shells, there is almost no reason to develop your own shell / inference engine. Most applications can be prototyped on a PC, and then migrated to more powerful hardware, if and when necessary. The need to migrate applications favors shells which run on multiple computer platforms.

The Future Impact of Expert Systems in Chemistry

Expert systems offer the possibility of revolutionizing the practice of chemistry. There are, however, two major problems which must be overcome. The first, which is easy to overlook until you develop your first expert system, is the time required to extract the knowledge from the expert and encode it into an expert system. The second major problem is the amount of computation required to solve a given problem. This problem is compounded because the most useful applications tend to be the most complex.

The time required to build the knowledge base may be the most severe problem, when you consider both the amount of chemical information available today and the rate more is being discovered. Fortunately, small systems which do not contain a broad base of chemical information can still be very useful. Even so, unless there are major improvements in knowledge engineering, the rate at which knowledge bases can be compiled will be a limiting factor.

Several approaches are being investigated to solve the problem of execution speed. Work is proceeding on two hardware solutions: faster processors that are designed specifically for artificial intelligence and parallel architectures which work on multiple parts of the problem simultaneously. Software solutions attempt to reduce the number of computations using more efficient algorithms and better heuristics.

The possible benefits of expert systems in all fields are so great, that these problems will be solved, making expert systems more cost effective. The cost of "doing chemistry" has skyrocketed. This increase has been caused by increased costs of equipment and raw materials, and the increased cost of handling chemicals safely. Expert systems applied to the areas listed above, and in this volume, can help increase productivity and safety in the field of chemistry.

The future of chemical applications in chemistry will bring many exciting new applications. The time will come when the use of expert systems will be an integral part of the practice of chemistry. Expert systems will not replace chemists, they will rather be very useful assistants which handle the details and allow the chemist to concentrate on the more challenging problems.

Glossary

Artificial Intelligence (AI): The field of computer science which attempts to develop computer systems which solve problems traditionally thought to require "human" intelligence.

Backward-chaining: A technique in logical inference using IF-THEN

rules. The strategy is to prove a conclusion (THEN clause) by proving all the IF clauses in the rule. These IF clauses become the conclusions in the next level of the search. The search moves backwards in this fashion, using the knowledge base to prove or disprove the initial hypothesis. For example, the reactants in a chemical reaction may be the products of another reaction, thereby allowing the scientist to trace the reaction back to simple starting materials.

Certainty factor: The confidence the scientist has in a given piece of information. Usually a value in a range of numbers (e.g. 0 to 1 or -1 to 1). They may or may not have a statistical basis. They may apply to the confidence the expert has in the conclusion in a rule, or to the input supplied by the end-user during a problem solving session.

Confidence factor: see Certainty Factor

Data-driven: see Forward-chaining

Domain: The area of study, or expertise, covered by an expert system. In this case, a field of chemistry.

Domain expert: The person whose expertise is encoded into the expert system. This person's expertise, or knowledge, is the basis for the knowledge base.

Explanation facility: The part of an expert system's user interface which describes how the program reached a particular conclusion. This ability is a key advantage to expert systems, and is an important part of end-user acceptance of the system.

Forward-chaining: A technique in logical inference using IF-THEN rules. All possible conclusions are drawn from a list of known facts. Because each new conclusion (THEN-clause) can be used as a fact (IF-clause) to derive other conclusions, the program iterates until no new facts can be concluded.

Frames: An hierarchical method of representing knowledge in an expert system. Using the hierarchy, information is either explicitly stated or inherited from a higher level. Rules are also used to connect facts throughout the frames. Expert systems using frames will search hierarchies to associate facts with conclusions.

Goal-driven: see Backward-Chaining

Heuristic: An expert's "rule of thumb" used in solving a problem. Heuristics provide directions which can speed up the time a program takes to solve a problem. Using heuristics does not guarantee that the best solution will be found. This limitation is acceptable in cases where more possible solutions exist than can be explored. It can also refer to self-learning.

Inference engine: The part of an expert system which performs the logic calculations using the knowledge base and user's input. The engine determines which rules are appropriate and under which conditions to apply each rule.

Knowledge base: The part of an expert system which contains the information about the problem area. Different methods for

representing this information are used by expert systems. The two most common methods are currently rules and frames.

Knowledge engineering: The study of techniques to acquire the domain expert's knowledge, and encode it into the expert system.

LISP: A programming language used to write AI systems. LISP is an acronym for LISt Processor.

Production rule: Another name for IF-THEN rules used in representing information in an expert system's knowledge base.

Production system: Another name for an expert system.

PROLOG: A programming language commonly used to code AI application programs. PROLOG is an acronym for PROgramming in LOGic. It is used predominantly in Europe.

Reverse-chaining: see Backward-chaining.

Shell: An expert system shell is a package of programs including an inference engine, user interface, and knowledge base maintenance tools. It is, in essence, an expert system with no knowledge base. The expert begins here, and then creates their own knowledge base.

References

1. Pierce, T.H., Hohne, B.A., Eds.; Artificial Intelligence Applications in Chemistry, American Chemical Society Symposium Series Vol. 306; American Chemical Society: Washington, DC, 1986.
2. Hohne, B.A., Pierce, T.H., Bright, M.A.; in Encyclopedia of Microcomputers Volume 1; Marcel Dekker, Inc.: New York, 1987, p 342-368.
3. Weiss, S.M., Kulikowski, C.A.; A Practical Guide to Designing Expert Systems; Rowman & Allanheld: New Jersey, 1984.
4. Hayes-Roth, F., Waterman, D.A., Lenat, D.B., Eds.; Building Expert Systems; Addison-Wesley: Massachusetts, 1983.

RECEIVED June 9, 1989

Chapter 2

Panel Discussion

Can Knowledge Bases Be Made Generally Available in a Useful Format?

Bruce A. Hohne¹, Joe Karnicky², Hugh B. Woodruff³, David Edelson^{4,5},
and Thomas H. Pierce⁶

¹Rohm and Haas Company, Spring House, PA 19477

²Varian Associates, Palo Alto, CA 94303

³Merck Sharp and Dohme, Rahway, NJ 07065

⁴Department of Chemical Engineering, Florida A&M University/Florida State
University, College of Engineering, Tallahassee, FL 32316-2175
Tallahassee, FL 32306-4019

⁵Supercomputer Computations Research Institute, Florida State University,
Tallahassee, FL 32306-4052

⁶Rohm and Haas Company, Bristol, PA 19007

A panel discussion was held to review the problems involved in sharing Knowledge bases. This discussion examines the advantages of sharing expert system rules, and the likelihood of knowledge bases being shared. The viewpoints are given by industrial representatives, a university professor, and chemical instrument vendors.

Expert systems are valued almost completely by the knowledge contained in their knowledge bases. One way for expert systems to become more popular and useful is to share the knowledge bases. The sharing of the knowledge bases would allow developers other than the original knowledge-base creator to "improve or extend" the knowledge base into new areas. Sharing the knowledge base would enable expert systems to be built on previously designed expert systems. This might extend the utility of expert systems into more complex chemical applications. For the most part, this sharing has not yet happened. The reasons that knowledge bases are not divulged or shared are reviewed by the panel.

Bruce Hohne

Can knowledge bases be shared by the Chemical Community? Reasons for sharing knowledge are twofold. Knowledge is POWER and Knowledge is EXPENSIVE. Having knowledge about a process or technology gives people the power to control, change, and improve the them. Obtaining this knowledge base is, however, expensive, so developers must be able to recoup their expenses.

The first question one must ask is; Why would anyone want to share a knowledge base? A knowledge base is a collection of information which offers the potential to solve specific problems. Sharing this knowledge base would 'solve' that problem for other

0097-6156/89/0408-0010\$06.00/0

© 1989 American Chemical Society

people. Reviewing another's knowledge base would allow researchers to improve the knowledge base. This could make the knowledge base more 'knowledgeable' than the original expert, benefiting both the developer and customer. Also, the knowledge upon which a knowledge base is built will evolve over time and this evolution might be accelerated if the knowledge bases could be tested and modified by chemists other than the developer.

There is a global problem associated with sharing knowledge bases; it is a technology issue. Knowledge bases can be stored using a variety of different types of representations, such as IF-THEN-ELSE rules or logic frames. Within each type of representation different vendors have different syntaxes. This technology problem is slowly going away as more expert system shells are sold. These packages are becoming more standardized in their representations and syntaxes, and often have common interfaces.

The other problems involved with sharing knowledge bases depend in part on the source of the work. There are three sources of knowledge bases: vendors, academic, and industry. Each has a different goal, therefore, a different set of problems.

Vendors develop expert systems to sell. They must, therefore, be able to make a profit on their work. Knowledge bases are very expensive to develop, so it is very important to the vendor to protect their investment. One approach which may be used to both market and cost justify these systems is to include them as a "value added" feature with an instrument.

Academic develops expert systems as part of a research effort. They are usually asked only to show "proof of concept". Knowledge bases at this stage of development are not polished enough to be shareable. If future development is not publishable, there will be no incentive to continue the work.

Industry develops expert systems for in-house use. The cost is justified by increased productivity or increased competitive-edge. Many times the knowledge encoded is a trade secret, making sharing impossible.

Joe Karnicky

At the Varian Research Center, expert systems are being developed for analytical chemistry method design, instrument calibration and operation, and flexible manufacturing. Sharing knowledge bases has not happened yet and might not be possible because of significant economic and technical obstacles.

There are considerable economic obstacles to the sharing of knowledge bases. The development of large expert systems that solve difficult problems is expensive, and the construction of the knowledge base is the most expensive part. Often, the knowledge must be developed and systemized before it can be encoded. Figure 1 shows the development costs for a selection of large expert systems. Corporations recover these development costs through improved in-house operations, through sale of the expert system, or by considering

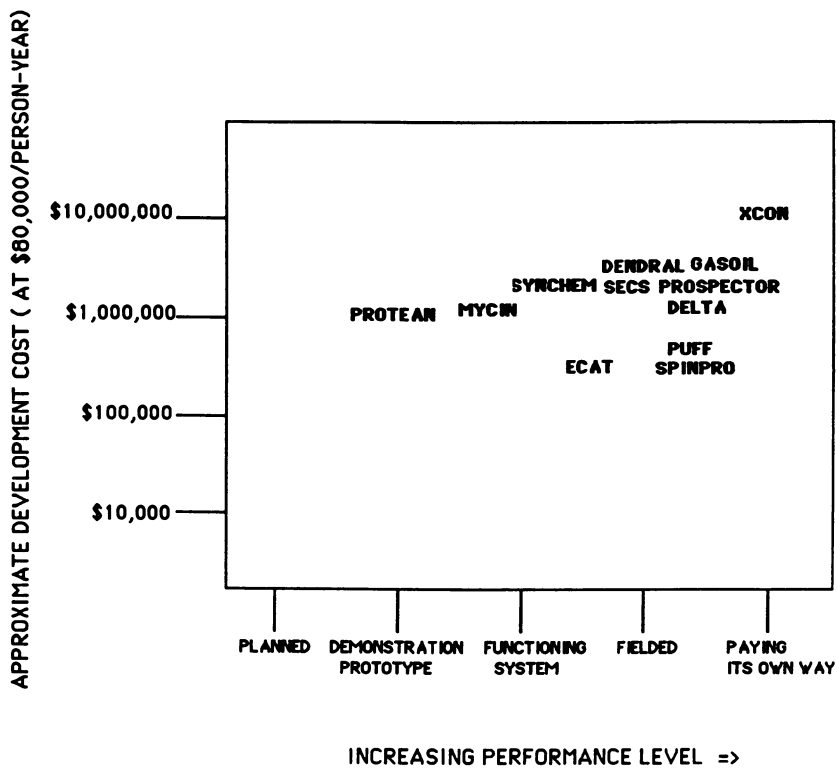


Figure 1. Development costs of some large expert systems.

the expert system as a value-added feature in a product. In any case, the knowledge base is kept confidential and treated as a trade secret. Patent and copyright laws for software are sufficiently uncertain that a corporation is reluctant to make its knowledge bases generally available, even under license.

There are also significant technical obstacles today, especially if one is interested in sharing knowledge bases in the sense of re-using knowledge bases, possible combining knowledge bases from more than one expert system program to get enhanced performance. Because of differences in the syntax and semantics of knowledge base representations, knowledge bases cannot easily be shared between inference engines. While standardization of knowledge representation is improving, this problem currently makes sharing of knowledge bases difficult. A second technical problem is the fact that in most expert systems being written today, the individual pieces of knowledge (e.g. rules) are not truisms in themselves, but rather, are only valid in the context of related rules and of implicit procedural semantics. Figure 2 shows three rules from the Varian ECAT program for High Performance Liquid Chromatography (HPLC) method design. Because, in general, the context and hidden assumptions for two expert systems will differ, a simple extraction of the knowledge base from one to the other is not possible.

Hugh Woodruff

Actually, at least two questions confront us:

- 1) Can knowledge bases be made available? (Yes, although frequently with some difficulty.)
- 2) Should knowledge bases be made available? (a debatable issue)

Why should one make knowledge bases available? The idealistic answer is for the betterment of science - to help researchers solve problems. I have been involved for over 10 years with several collaborators in the development of PAIRS (Program for the Analysis of IR Spectra), functioning as the IR "expert" during rules development and modification, and I can attest that there is a tremendous investment in the knowledge base. However, from the outset, our intent was to make the knowledge base accessible. Just making it accessible is not enough. In the early versions of PAIRS, it would be quite difficult for a user to understand which rules were being utilized during an interpretation and what was the logic behind those rules. More recent enhancements to PAIRS have incorporated an explanation facility that makes the results much easier to understand. Because the knowledge base is separated from the program, users can customize the system to meet their own needs. For example, some European researchers have modified PAIRS to create a system called PAR_S, Program for the Analysis of Raman Spectra. The ability to customize and improve knowledge bases are good reasons for them to be shared.

```

(cmp37
  (descr (to avoid denaturing the protein by reverse phase organics.)
  (type rule)
  (text (if the sample is a protein and the user wants to recover biological
    activity then prefer ion-exchange mode over
    hydrophobic-interaction mode by .35))
  (pform (if (and (actual (analyte-chem-class proteins))
    (bioactivity-desired))
    then (prefer (separation-mode ion-exchange)
      (separation-mode hydrophobic-interaction)
      .35))))

(ask1.5
  (descr "no distinction is made between analyte subclasses and actual
    analyte subclasses - rule applies to both chemical and functional
    classes")
  (type rule)
  (text "if an actual analyte class has any subclasses known to
    the system, then ask the user about analyte subclasses")
  (pform (if (and (actual-analyte-class $class)
    (subclass-of $class $anysubclass))
    then (to-be-run
      (asks-analyte-subclass $class $subclass))))))

(cmpk5
  - (descr "cleanup and extension of cmp22.5 and cmp24- see cmpk5b")
  (type rule)
  (text "if the separation-mode is reverse phase and there are any
    acids entered with pka's <= 11 and no bases
    entered then design for the acids")
  (pform (if (and (separation-mode reverse-phase)
    (smallest pka $analyte $pka)
    (<= $pka 11)
    (generated-every plkb-of)
    (unknown (plkb-of $any $plkb)))
    then (design-for acids))))

```

Figure 2. Rule examples from the ECAT expert system for high performance liquid chromatography.

What are the dangers in making knowledge bases available? The greatest danger is that a system becomes unsupported. As the developers of PARS no doubt discovered, it is not an easy task to change the rules in a system as large as PAIRS. Yet they were able to succeed. To date, over 100 versions of PAIRS have been distributed by us or by the Quantum Chemistry Program Exchange. Potentially, each of these versions has been customized and distributed further. Thus, no standard knowledge base remains and it would be virtually impossible for us to try to support the system we initially created. Many system developers will not want to get involved in the legal issues, security concerns, or support problems that come with sharing a knowledge base.

Another issue that slows the sharing of knowledge bases is disseminating the information. Are knowledge bases publishable? Not many have been published, so the mechanism for sharing the information contained in a knowledge base does not yet exist. When a significant modification is made to a knowledge-based system, authors, editors, and reviewers are confronted with whether these modifications are research and thus warrant publication or are simply ongoing system maintenance.

There are no easy answers to any of the issues mentioned, but we must continue to address the concerns and not simply avoid them.

David Edelson

The issues involved with sharing knowledge bases are not new. The sharing of simple tabulated numerical data has not yet completely succeeded. Thermodynamic tables and other physical data which are available only in book form have not made the transition to electronic files. These simple knowledge bases would be very useful to scientists if one could easily access them from computers. Today, scientists still hand enter the data into computers for use in their computer programs. Until there are mechanisms for sharing numerical data, knowledge base sharing will be even more difficult.

Tom Pierce

A basic problem with sharing knowledge bases is the separation of trade secret information from public domain information. Today, if a knowledge base is to be distributed, then the knowledge must already be known to everyone, or else it will become a trade secret. There are some knowledge bases which are already public domain. A knowledge base developed from Government Regulations or chemical safety procedures would be an example. For government regulations the knowledge is already defined in the Federal Register, and it need only be encoded in an expert system shell. Test requirements for the Environmental Protection Agency or the requirements for overseas trading licenses are other examples of potential knowledge bases which are shareable and already in the public domain.

Many other knowledge bases will not be shareable. Those knowledge bases which are product specific, such as a color-matching paint mixing knowledge base, will probably remain as trade secrets. Another common use of knowledge bases is in corporate-specific decision-making. These knowledge bases are so specific to a situation that there is no benefit to sharing the knowledge base.

Many expert systems which have been developed are very large software systems. A large system does not lend itself to maintenance or modifications. With the recent improvement of expert system shells, the opportunity to develop smaller, modular expert system, with highly specific knowledge bases has become possible. A small system could be treated as a black box program. Expert systems could be developed and used in the same way as mathematical libraries on computers are today. These small modular expert systems would be easily shareable. Modular expert systems libraries could be used as building blocks to develop problem-specific expert systems.

Questions and Comments from the Audience

Expert system technology is still developing. It is too early to standardize it, as has been done with FORTRAN. However, the shells which are being developed are beginning to become a little more standardized.

Expert systems which make product recommendations, for only one vendor's product, would make a very useful electronic catalog.

Is copyright protection enough to protect a knowledge base? Are knowledge bases significantly different from paper media? Until these questions are answered, vendors will be reluctant to sell knowledge bases.

Because the applications of the technology are new, many issues must be resolved, such as: The cost of distributing a knowledge base may require a license fee to the expert system shell developer. This is very different from FORTRAN programs which can be distributed as object code, with no fees to the FORTRAN compiler developer.

Small expert systems with intimate knowledge of a specific problem are much cheaper to develop, but they may be too useful to the developer to share them.

Conclusions

There are, as of yet, no final answers. For today, there are more reasons for knowledge base developers to keep their knowledge base than to distribute it. Hence, most currently developed expert system will not be distributed for economic or trade secret reasons.

The group of expert systems most likely to be distributed are those which are incorporated into other products, such as instruments, or expert system which clarify government regulations.

The current costs of developing knowledge bases are too high to encourage distribution. Before this can change, expert system tools must improve to accept different syntax knowledge bases, and more knowledge must be collected into heuristic knowledge bases in the literature. The introduction of inexpensive, PC based, expert system shells may provide some of the tools necessary to decrease the costs of expert systems.

RECEIVED June 9, 1989

Chapter 3

Predicting Inorganic Reactions

The Development of an Expert System

James P. Birk

Department of Chemistry, Arizona State University, Tempe, AZ 85287-1604

An expert system has been developed, using PROLOG, to predict products of inorganic chemical reactions. The expert system was designed to model predictive approaches used by a number of chemists, using the simplest rules possible. After the user inputs chemical formulas of one or two reactants, the formulas are parsed to determine the component elements and to recognize any significant polyatomic groups. Oxidation numbers are then assigned to the elements. This information, in addition to a limited database, is used to place the reactants into classifications which will fire rules for various kinds of chemical reactions. The expert system can make predictions for combination reactions, decomposition reactions, single displacement reactions, double displacement reactions (including oxidation-reduction, acid-base, and precipitation reactions), and complex ion formation reactions.

Computer use in the educational process has become commonplace, ranging from drill-and practice, tutorials, and simulations to word processing and spreadsheets. Software for computer-assisted instruction (CAI) has undergone a substantial change in character over the past 20 years. Starting out as a calculation-aid, software has evolved into drill-and-practice and tutorial aids, along with a number of other more specialized types of instructional aid (1). However, most software does not offer the student an opportunity to study a subject in the truly individualized manner that would be consistent with the one-on-one situation existing when a student sits in front of a computer terminal (2). Most present software is computer-controlled rather than student-controlled. The computer generally guides the student through the subject matter. Although the student may be able to make selections from a menu, the computer is not very responsive to the student's needs.

0097-6156/89/0408-0020\$06.00/0
© 1989 American Chemical Society

Intelligent Tutoring Systems

Now, with the increasing availability of faster processors and more memory, the nature of instructional computing is on the verge of substantial change. The opportunity to place the student in charge of a computer-assisted learning experience is offered by the application of expert systems and artificial intelligence developments to instructional computing (3). In these systems, the usual question and answer format typical of CAI tutorial programs will no longer prevail. The program will no longer drag the student along in its wake. Rather, the program will become a tool and advisor for the student, providing a learning environment with much less predetermined structure than is now common, with the student rather than the computer in control. Students will interact with the program, asking questions, asking for help or advice, asking for examples. The program will respond intelligently to student queries, providing answers to questions, explaining answers, and explaining how answers were determined. The program may also ask questions to guide the student and to test the student, if so requested, but then will also be able to explain why the student's answer was either correct or incorrect. Such questions will be formulated at a level appropriate to the student's previous performance. Systems that have these capabilities are called intelligent tutoring systems.

A full intelligent tutoring system is made up of several parts (3). The knowledge domain is a database of facts that provide the basis for knowledge in the particular subject. The expert system consists of a set of rules that makes use of the database to provide expertise in the subject area (4). The user interface provides for communication between the program and the student in such a way that the student can ask questions of the program. A goal in many intelligent tutoring systems is natural language communication, in which interaction can take place in English sentences. The teaching model provides the pedagogical approaches, explanations, tutoring, questions, situations, etc. that would generally be provided by a teacher, after judging the progress of the student. The student model maintains a past history of student use and responses. This information is used by the teaching model to select appropriate challenges and explanations for each student.

Expert Systems

Central to an intelligent tutoring program is an expert system, which is a program that consists of a compendium of knowledge (facts and rules) about a particular topic. This expert system should have enough knowledge to allow the computer to answer questions or make predictions in a given knowledge area with the accuracy of a human expert (5).

The development of an expert system requires a understanding of the functioning and thinking patterns of a human expert. Such an understanding often eludes even that expert. To be an expert, a person must be so familiar with a knowledge area that predictions or solutions to problems seem to flow from that person with little visible effort. This is in fact a characteristic of college

instructors that undoubtedly dismays many students. By virtue of such intimate and long-standing familiarity with a subject, an expert often does not consciously understand the process by which answers or solutions are generated. However, to convert the computer into an expert, such analyses of thinking processes must be made in considerable detail. Thus, development of an expert system provides insight into the methodology used in the subject area. Preferably, an instructional expert system should model the approaches used by experts to reach conclusions that are not only accurate but also pedagogically sound. In this way, when the expert system is inverted, to be able to explain its own processing of a question, the explanations will match those that would be given by an astute teacher.

Descriptive Inorganic Chemistry

PIRExS. There has been widespread interest in increasing the coverage of descriptive inorganic chemistry in the general chemistry curriculum (6-13). The manner in which this is to be accomplished effectively has not yet been clearly delineated, however. With support from Project SERAPHIM, an attempt is underway to develop computer software which will help teach descriptive inorganic chemistry. We will describe here work involved in the development of an intelligent tutoring system to teach descriptive inorganic reaction chemistry. The initial efforts on this project have resulted in an expert system called PIRExS (Predicting Inorganic Reactivity Expert System), written in PROLOG for the IBM PC and other compatible microcomputers. This expert system can make predictions of the products of inorganic reactions, as well as explain the manner in which those predictions were made. Thus, this system comprises portions of the intelligent tutoring system that should ultimately result from this work. A large number of different types of reactions can be handled successfully by this expert system. The program will make predictions with a success rate of about 95% on the types of reactions shown in Table I. The success rate was determined by selecting about 1000 reactions used in general chemistry and inorganic chemistry textbooks. The program was considered to be successful if it could reproduce the reaction as written in the text, even though the program would usually come up with more than one possible reaction.

Expert System Development. The development of this expert system was initiated by gathering a group of eminent inorganic chemical educators to discuss the manner in which experts make predictions. Two possibilities were considered: Either an expert calls upon a large, perhaps subconscious, database of known reactions and predicts by analogy with known results, or an expert understands the basic principles and rules governing chemical reactivity and applies these results, again perhaps subconsciously, to a given situation. The real mode of operation of an expert may well be a combination of these approaches. Several conclusions arose from this meeting. A database approach was considered unsatisfactory since the possible combinations of the more than 100,000 known inorganic chemicals would require a massive compilation of known reactions. This

approach would also provide little pedagogical assistance since no student would willingly accept the task of memorizing such large numbers of reactions. Further, the absence of a reaction from the database would not allow a student to distinguish between two possibilities: Either the reaction does not take place, or the reaction was overlooked or omitted for some reason by the database creator.

Table I. Types of Reactions Predicted by PIREXS

disproportionation of elements
disproportionation of simple ions
disproportionation of compounds
disproportionation of oxo ions
internal redox of ionic compounds
thermal decomposition reactions of compounds: hydrides, hydrates, metal hydroxides, ammonium salts, peroxides, oxides, halides, thiosulfates, metal oxo salts, protonated metal oxo salts
binary combination reactions of elements
redox reactions of elements, ions, and ionic compounds in acidic solution
displacement of H ₂ from water, steam, or acids by elements
oxidation of a compound by molecular oxygen
redox reactions between gaseous nonmetal oxides
displacement of an element in a compound by another element
displacement of an element in a compound by carbon monoxide
oxide displacement reactions between oxides and oxo salts
oxide displacement reactions between oxides and oxoacids
ionic precipitation reactions
acid-base reactions between oxides
hydrolysis of various ionic species
hydrolysis of covalent compounds, especially halides
acid-base neutralization reactions
complex ion substitution reactions
covalent oxidative-addition reactions
reactions of concentrated sulfuric acid

A rule-based system is clearly superior from a pedagogical viewpoint. Such a system would be more open-ended since it could use periodic relations to extrapolate to related reactions. A rule-based system also has the capability of being inverted to be able to explain its answers, a necessity if the expert system is to be developed into an intelligent tutoring system. However, a rule-based system is also expected to have limitations in the number of predictions it can make, since rules must be formulated to cover many different types of reactions.

Finally, although our group of experts could make predictions with ease, verbalizing the exact approach and rules used to make the predictions was not an easy task. As explained earlier, this is consistent with the nature of an expert, so it is not a surprising result. The answers developed by the group also made it clear that numerous approaches could lead to successful predictions. In spite

of this variability in approach, a common thread appeared. Virtually all predictions were made by first categorizing the reactants into one or more reactant classes, then examining those reaction types that would be appropriate for those reactant classes.

Expert System Design. The expert system design was based on the general approaches used by our group of experts. After formulas of two reactants (sometimes only one reactant) are input, the formulas are parsed to identify component elements and recognizable polyatomic groups, then oxidation numbers and ionic charges are assigned, and finally the reactant species are checked as possible examples of a variety of reactant classes. This information is then used to trigger rules for reaction types appropriate to those reactant classes. Use was made of PROLOG's backtracking capabilities to search for all possible reactions for the reactants that were input.

Database. The expert system consults a database to get the information required to assign a reactant to the various possible reactant classes. Surprisingly little information is needed in this database either to assign reactant classes or to make predictions of chemical reactivity. The information that was placed in the database is listed in Table II.

Reactant Classes. In order to make predictions of chemical reactivity, it is necessary to check whether a reactant is a member of about 30 different reactant classes. These reactant classes are listed in Table III. In addition, for ease in programming, groups of reactant classes were defined. These groups contain as members those reactant classes that will undergo similar reactions. The reactant class groups are listed in Table IV.

The definitions of reactant classes were usually fairly simple, involving mostly information about the composition of the substance. For example, the definition of an oxoacid can be given by the following statement, which paraphrases the PROLOG predicate:

A substance is an oxoacid if
the substance contains three components,
and the substance contains hydrogen and oxygen,
and the substance contains another nonmetal,
and the substance is uncharged.

Another example is provided by the definition of a ionic compound:

A substance is an ionic compound if
the substance contains two or more components,
and the one component is a metal,
and another component is a nonmetal,
and the substance is uncharged.

Some of the common definitions were found not to be operable, since they require some knowledge of reactivity in order to be able to classify the reactant. For example, the usual definition of a Brönstead acid indicates that an acid is a substance which will donate a proton to another substance. However, this means that we must know whether the proton donation process occurs before we can decide whether the substance is an acid. Since we must know whether

Table II. Database for PIRExS

General information:**Elemental Information:**

name, symbol, atomic number, group number, period number,
metallic character, electronegativity

Ionic Radius:

values for all the simple monatomic ions of the elements

Reduction Potential:

E° values for conversions between adjacent and non-adjacent
oxidation states

Synonyms:

preferred chemical formulas and alternative formulas that might
be input by a user or be created by the program

Some acid and base dissociation constants

Boiling points of nonmetal oxides

Information Specific to the Current Reactants:**Compound:**

formula and charge

Components:

elements and their numbers

number of polyatomic groups containing the element

Oxidation Numbers:

oxidation numbers of all the component elements

Count:

total number of different elements in the reactant

Polyatomic Group:

formula and charge of recognizable polyatomic groups

Acid:

formula, charge, and pK_a of any acid and its conjugate

Base:

formula, charge, and pK_b of any base and its conjugate

Strongest Acid:

formula, charge, and pK_a of the strongest acid present

Strongest Base:

formula, charge, and pK_b of the strongest base present

Complex Ion:

identity of the metal and its oxidation number

identity of the ligands, their number and charge

Free Ligand:

formula and charge of any species that is a potential ligand

Table III. Reactant Classes and Examples

None	Element (Cu)
Water (H ₂ O)	Ammonia (NH ₃)
Ammonium Hydroxide (NH ₄ OH)	NonMetal Oxide (NO ₂)
Metal Oxide (CuO)	Covalent Compound (CO ₂)
Ternary Covalent Compound (CH ₂ F ₂)	Ionic Oxo Salt (Na ₂ CO ₃)
Polyatomic Anion Salt (KNCS)	Oxoacid (H ₂ SO ₄)
Strong Oxoacid (HClO ₄)	Ionic Compound (Na ₂ S)
Binary Ionic Salt (Na ₂ O)	Acid (H ₂ S)
Compound (NaCl)	Ammonium Salt (NH ₄ Cl)
Protonated NonMetal Salt (NaHS)	Protonated Oxoanion Salt (NaHSO ₄)
Basic Compound (Na ₂ S)	Metal Hydroxide (NaOH)
Oxoanion Salt (Na ₂ CO ₃)	Ion (Na ⁺)
Simple Ion (Cl ⁻)	Cation (Cu ²⁺)
Anion (HS ⁻)	Basic Anion (S ²⁻)
Oxoanion (CO ₃ ²⁻)	Oxocation (VO ²⁺)
Protonated Oxoanion (HSO ₄ ⁻)	

Table IV. Groups of Reactant Classes

Binary Salt:	Oxide Species:
binary ionic salt	nonmetal oxide
binary acid	metal oxide
Ionic Species:	metal hydroxide
ion	Redox Species
ionic compound	ionic compound
binary acid	ion
oxoacid	oxoacid
ammonium salt	element
Oxo Ion:	Polyatomic Salt
oxoanion	polyatomic anion salt
oxocation	oxoanion salt
Base:	protonated nonmetal salt
basic compound	ammonium salt
basic anion	Acid:
ammonia	acid
ammonium hydroxide	nonmetal oxide

a substance is an acid before we can decide whether a prediction of a proton donation reaction is a likely possibility, this definition is circular in terms of the desired use. To circumvent this possibility, we use a definition in terms of composition:

A substance is an acid if
the substance contains two or more components,
and the substance contains hydrogen
and the substance contains another nonmetal,
or the substance is the hydrogen ion.

This definition will be true for a number of substances that we generally do not consider to be acids, however. For example, methane, hydroxide ion and amide ion would be acids by this definition. Of course, a knowledge of the acid strengths would provide a criterion for eliminating substances such as these. Since we also wanted information about acid strengths in order to make predictions about the most likely reactions, it was necessary to refine this definition. In essence, rules had to be defined for the strength of each different class of acid (and base), so that definitions actually had to be developed for all these classes as well. Although trends in acid strength of binary acids can be predicted within a given period or group of the periodic table, there does not appear to be any reasonably simple way to predict values of the acid dissociation constants for these species. In this case, reference is made to a database of values. For oxoacids, it is possible to make order-of-magnitude predictions of the acid dissociation constant values, based on the oxidation number of the central atom, the number of oxygen atoms, and the number of hydrogen atoms (14). It is also possible to predict values of the acid dissociation constants of aquometal ions, based on the charge, radius and electronegativity of the metal ion (15-16).

Bases can be defined in a similar manner. Bases include basic compounds (uncharged species containing two or more components, including a metal and a nonmetal), basic anions (negatively charged species containing one or more components, including at least one nonmetal), ammonia, and ammonium hydroxide. As with acids, this definition will recognize substances such as NaCl or Br⁻, which are not very basic. Thus, it is also necessary to develop a series of rules or database tables (the same as those for binary acids, with $pK_b = 14 - pK_a$) to establish the magnitude of the base strength. This provides criteria to eliminate potential bases which are too weak to be considered meaningfully.

Thus, the simple definitions of acids and bases that are most often used had to be expanded into quite a large number of definitions and rules to be useful for predicting reactions between acids and bases, a reaction class which is usually considered to be fairly simple. Similar results were found throughout this project. Material which is considered to be fairly simple and straightforward when we try to teach it to our students, often is found to be much more complex when we try to teach it to the computer, since we cannot expect the "student" to make the major intellectual leaps that we seem to expect of our human students. This has proved helpful in understanding some of the problems students have in learning material the way it is commonly taught. Writing a set of rules and testing them with a variety of examples is a useful way of defining the limitations of various instructional approaches.

Reaction Classes. Once the reactants can be identified as members of various reactant classes, it is possible to make a predetermination of the reaction classes that should be examined.

This allows us to eliminate reaction classes which are not possible. For example, if the reactants consist of two elements, there is no need to examine acid-base reactions, or reactions between elements and water, or decomposition reactions, etc.

Reactions with One Reactant. If there is only one reactant, there are a limited number of possible reaction types. In aqueous solution, a disproportionation reaction is possible for an element, an oxo ion (Table IV), a simple ion, or a compound. A compound can also undergo a decomposition reaction of various types, as listed in Table I. An ionic compound can undergo an internal redox reaction, in which the oxidant and reductant are chemically combined in a single reactant. An ionic species (Table IV) or a polyatomic salt (or polyatomic anion) can undergo hydrolysis in aqueous solution. All other reaction classes are considered only if there are two reactants.

Reactions with Two Reactants. Numerous reaction types (see Table I) are possible in the case of two reactants. In some cases, several reaction classes need to be considered for two reactants. Consider the case of FeF_3 and $\text{Cr}(\text{HCO}_3)_2$, which may be an improbable set of reactants but which illustrates nicely the operation of the expert system.

Since these are ionic compounds and, therefore, redox species (Table IV), the expert system will check for redox reactions, using reduction potentials for each combination of component elements from the two reactants to decide whether any redox will occur. Iron(III) will oxidize chromium(II) in this system.

Parsing for acids and bases identifies the iron(III) ion as an acid with pK_a of 5.01, the fluoride ion as a base with pK_b of 10.85, the chromium(II) ion as an acid with pK_a of 11.4, and bicarbonate ion as an acid with pK_a of 7.3 and as a base with pK_b of 11.2. Based on these results, predictions are made of the extent of hydrolysis of each of these ions. Picking the strongest acid (smallest pK_a) and strongest base (smallest pK_b) from these species, the expert system predicts that aqueous iron(III) ion will react with aqueous fluoride ion to form iron(III) hydroxide and hydrofluoric acid with a pK of 1.86.

Since the reactants are both ionic species, the expert system will check for aqueous precipitation reactions. In this case, using solubility rules, the expert system predicts logically, but incorrectly, that iron(III) carbonate will precipitate from the solution. In fact, if a precipitate is formed, it is more likely to be iron(III) hydroxide, since iron(III) ion and carbonate ion are both rather susceptible to hydrolysis. Rules have not yet been established to predict when hydrolysis of the components will override precipitate formation.

Parsing for complex ions indicates that the solution will contain hexaquaairon(III) ion and hexaquachromium(II) ion as complex ions, while the fluoride ion and water are possible ligands. The presence of a complex ion and a free ligand indicates that complex ion substitution reactions are possible. Since both chromium(II) ion and fluoride ion are hard, fluoride ion should substitute in a stepwise manner for the bound water molecules on the

hexaaquachromium(II) ion, forming a series of fluorochromium(II) complexes.

Not all examples will generate this many possible reactions, but in all cases, the reactants are examined to see if they fall into reactant classes for which reactions of all the types listed in Table I are possible. The reactant classes and their possible reactions are summarized in Table V.

Rules for Reactivity Prediction. Rules for predicting reactivity of given chemical species were developed as simply as possible, in order that they would be appropriate for later pedagogical use. The use of reduction potentials for the prediction of redox reactions has already been mentioned, as has the use of pK_a and pK_b values to satisfy the rule that the strongest acid will react with the strongest base to undergo acid-base neutralization reactions.

Some of the reactivity rules make use only of the reactant classifications described earlier. An example is the nonaqueous reaction between oxides:

oxide reaction if

one reactant is a metal oxide,
and the other reactant is a nonmetal oxide,
then a metal oxoanion salt is formed.

oxide reaction if

one reactant is a metal hydroxide,
and the other reactant is a nonmetal oxide,
then a metal oxoanion salt is formed.

oxide reaction if

one reactant is a metal oxide,
and the other reactant is water,
then a metal hydroxide is formed.

oxide reaction if

one reactant is a nonmetal oxide,
and the other reactant is water,
then an oxoacid is formed.

oxide reaction if

both reactants are nonmetal oxides,
and the nonmetals have oxidation states,
one higher and one lower than the initial states,
and the reduction potentials to these new oxidation states
combine to give a positive value,
then two new nonmetal oxides are formed by redox.

Many rules make use of periodicity, as exemplified by the reaction of elements with water. The rules for this class of reactions can be summarized as follows:

vigorous reaction with cold water if

the element is a metal,
and the metal is in group 1 or group 2,
and the metal is in the third period and higher,
then a metal hydroxide and molecular hydrogen are formed.

mild reaction with cold water if

the element is a metal,
and the metal is beryllium, or magnesium,
or in group 3, or a lanthanide,
then a metal hydroxide and molecular hydrogen are formed.

no reaction with cold water if
 the element is any other metal,
 or the element is a nonmetal other than a halogen.

oxidation of water if
 the element is fluorine,
 then hydrogen fluoride and molecular oxygen are formed.

disproportionation if
 the element is a halogen,
 then hydrohalic acid and hypohalous acid are formed.

reaction with steam if
 the element is a metal,
 and the reduction potential from the lowest oxidation
 state is less than -0.414 V,
 then a metal oxide and molecular hydrogen are formed.

reaction with acid if
 the element is a metal,
 and the reduction potential from the lowest oxidation
 state is less than 0.0 V,
 then a metal ion and molecular hydrogen are formed.

Rules for some other reactions make use of information from a database. For example, sodium sulfite will react with silicon dioxide at high temperatures to form sodium silicate and sulfur dioxide. This example follows the general rule:

nonmetal oxide displacement if
 one reactant is a metal oxoanion salt,
 and the other reactant is a nonmetal oxide,
 and the displaced nonmetal oxide is more volatile
 than the reactant nonmetal oxide.

The relative volatility is determined from the boiling points of the nonmetal oxides, which are stored in a database.

Properties such as charge or oxidation state, which are determined by a formula parser for the current reactants, may be a determining factor in other rules, such as the hydrolysis of halides:

halide compound hydrolysis if
 the compound is a nonmetal halide,
 and the compound is not a carbon tetrahalide,
 and the compound is not a nitrogen trihalide,
 and the compound is not sulfur hexafluoride,
 and the second reactant is water,
 then a nonmetal oxide and hydrogen halide are formed.

halide compound hydrolysis if
 the compound is a metal halide,
 and the second reactant is water,
 and the metal has an oxidation state of $+4$ or greater,
 then a metal oxide and hydrogen halide are formed,
 or a metal oxoion and hydrogen halide are formed.

The oxometal formula (oxide or oxoion) is determined by the common form of that metal in the appropriate oxidation state in aqueous solution. The latter is determined by another set of rather comprehensive rules.

Table V. Reactants and Possible Reactions

Reactant 1	Reactant 2	Reaction Class
none	none	no reaction
element	none	disproportionation
simple ion	none	disproportionation
compound	none	disproportionation
ionic compound	none	internal redox
compound	none	thermal decomposition
oxo ion	none	disproportionation
element	element	binary combination
redox species	redox species	ionic redox
element	water	hydrogen displacement
compound	oxygen	oxidation by O ₂
oxide species	oxides species	oxide acid-base
nonmetal oxide	nonmetal oxide	oxide redox
covalent compound	O ₂ or halogen	oxidation of covalent oxide or halide
nonmetal oxide	halogen	oxidative addition of halogen
nonmetal halide	halogen	oxidative addition of halogen
element	compound	displacement
carbon	metal hydroxide	displacement
carbon	oxoanion salt	displacement
carbon monoxide	compound	displacement
ionic oxo salt	nonmetal oxide	oxide displacement
oxoacid	nonmetal oxide	oxide displacement
conc. H ₂ SO ₄	element, binary ionic salt, ionic oxo salt, poly- atomic anion salt	reactions of hot, conc. sulfuric acid (acid-base or redox)
water	covalent compound	hydrolysis
ionic species	none	hydrolysis
ionic species	water	hydrolysis
ionic species	ionic species	hydrolysis
acid species	base species	acid-base neutralization
ionic species	ionic species	precipitation
complex ion	free ligand	complex ion substitution

The Explanation Facility

In order to convert an expert system to an intelligent tutoring system, the expert system must be able to explain its answers. This explanation facility has been added to the PIRExS program by means of amendments to the rules and by addition of disk files that contain all the rules expressed in natural language (English). Every rule defined in the program was labelled internally by a tracking variable. When the program attempts to execute a rule, this tracking variable is added to the database identified by a number that is defined within the rule. This tracking variable has the form:

track(number,character)

where the number is assigned systematically for all the rules and the character can be either T or F, depending on whether the rule was found to be true or false. Consider, for example, a rule numbered 9580. On entry to this rule, track(9580,F) is added to the database. If the rule is found to be false, then the database is left as is. However, if the rule is found to be true, track(9580,F) is removed from the database and track(9580,T) is added to the database. In addition, at appropriate places relative to other program output, calls are made to a routine that checks the database for any tracking variables, looks up the corresponding messages in the disk files, prints these messages, and removes the tracking variables from the database. In this way, appropriate explanatory messages are printed along with the reactivity predictions. To aid in reviewing the program output, each page of output is saved to the database and can be recovered by a page-up/page-down function. Thus, an entire set of explanatory messages can be reviewed or printed out as needed to understand the entire process of making predictions for a given set of reactants.

Pedagogical Uses

Although PIRExS is not an intelligent tutor, it can still be used to teach students about inorganic chemical reactivity. It does not tutor a student, but it can explain the logic and rules used to make reactivity predictions. It remains for a teacher to create an environment in which the program can be used as a tool to teach inorganic reactivity. The most obvious approach is to provide the students with reactivity problems to solve, allowing the use of PIRExS as a resource. Synthesis pathways provide good problems (17). For example, students could be asked to prepare chromium(III) sulfide, starting with potassium chromate. Another possible use is the discovery of reactivity rules. For example, students could be asked to develop a set of rules that would predict the stability with respect to thermal decomposition of carbonates. In such an application, the instructor would want to eliminate the explanation facility, since it would provide the answer with little student effort or imagination. Another feature of PIRExS provides this sort of instructor flexibility. A disk file, containing keywords for explanations as well as for each of the reaction classes, is read into the database during program start-up. If an instructor uses an editor to remove or comment out any of these keywords, student

access to that feature of the program is blocked. Thus, program capabilities can be adapted to the current needs of the students or the instructor.

Acknowledgments

The author is grateful for support in the form of a fellowship from Project SERAPHIM, funded through the Directorate for Science and Engineering Education of the National Science Foundation and the Fund for Improvement of Post-Secondary Education of the Department of Education. The author is further grateful to John and Betty Moore and the Department of Chemistry, Eastern Michigan University for their hospitality during a sabbatical leave, during which much of this work was completed.

Literature Cited

1. Moore, J. W. Acad. Comp. 1987, 2 (3), 18-21, 45-49.
2. Duckworth, E. L. Acad. Comp. 1987, 2 (3), 6-10, 38-43, 62-63.
3. Klopfer, L. E. J. Comp. Math. Sci. Teaching 1986, 5 (4), 16-32.
4. Floyd, M. Turbo Technix 1988, Mar./Apr., 64-66.
5. Myers, W. IEEE Expert 1986, Spring, 100-109.
6. Basolo, F. J. Chem. Educ. 1980, 57, 45-46.
7. Basolo, F. J. Chem. Educ. 1980, 57, 761-762.
8. Hudson, M. J. Chem. Educ. 1980, 57, 770-772.
9. Basolo, F.; Parry, R. W. J. Chem. Educ. 1980, 57, 772-777.
10. Cassen, T.; DuBois, T. D. J. Chem. Educ. 1982, 59, 377-379.
11. Beach, D. H. J. Chem. Educ. 1984, 61, 520-521.
12. Basolo, F. J. Chem. Educ. 1984, 61, 520-521.
13. Zuckerman, J. J. J. Chem. Educ. 1986, 63, 829-833.
14. Ricci, J. E. J. Amer. Chem. Soc. 1948, 70, 109.
15. Wulfsberg, G. Principles of Descriptive Inorganic Chemistry; Brooks/Cole Publishing Co.: Belmont, CA, 1987; p. 27-31.
16. Baes, C. F. Jr.; Mesmer, R. E. The Hydrolysis of Cations; Wiley-Interscience: New York, 1976; Chapter 18.
17. Kauffman, G. B. J. Chem. Educ. 1987, 64, 252-254.

RECEIVED June 9, 1989

Chapter 4

CH₂ESS

A Chemistry-Based Expert System Shell

Scott T. Stauffer¹, Raymond E. Dessy¹, and John W. Roach²

¹Department of Chemistry, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061

²Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061

CH₂ESS (CH₂EMISTRY-based EXPERT SYSTEM SHELL) is a tool for building chemically oriented expert systems under development at Virginia Tech. The development strategies, the advantages and limitations of the approaches employed, and progress-to-date are described.

The modern industrial analytical facility requires a laboratory information management system to meet the fiscal and legal imperatives it faces. Although today's LIMS packages are really only first generation data management packages, the future is readily apparent. Second generation LIMS will include such functions as project planning, statistics and modelling packages, and advanced data base constructs centered around a standardized scientific query language. The associated data base will become one of the most important assets that a company owns. To use it effectively requires an Expert System that can perform rudimentary reasoning in the chemical arena. An example would be the need to develop analytical procedures for new compounds. The problems and approaches associated with this type of development are described below. The goal of the project has been to explore what facilities are required, and how they should interact.

CH₂ESS DESCRIPTION

The purpose of CH₂ESS, or CH₂EMISTRY-based EXPERT SYSTEM SHELL, is to provide chemists with a tool that they can use to construct their own expert systems for problems

0097-6156/89/0408-0034\$06.00/0
© 1989 American Chemical Society

they are experiencing. There are six major components required of such a tool:

- Automated knowledge acquisition
- Molecular structure representation and recognition
- Reaction libraries (allows reasoning on reaction mechanisms)
- Parameter estimation and calculation using Linear Free Energy relationships
- General Chemical knowledge (atoms, bonds, reactants, products, conditions, ...)
- GUESS, a General pUrpose Expert System Shell

This article demonstrates how CHESS incorporates these components, and presents the capabilities of an automated knowledge acquisition system. The authors are using such a shell to construct an expert system for methods development in analytical chemistry. The domain chosen deals with derivitization reactions that result in products that can be spectroscopically detected. Compounds selected for the system include pharmaceuticals and vitamin-like compounds. While other components of CHESS are discussed, their details are to be published elsewhere.

This project is unique in that it is one of the first attempts to develop an expert system shell specifically designed for chemistry. While other researchers have implemented expert systems, all have either developed their own shell or utilized a commercially available shell. CHESS will assist in the development of an expert system in the chemistry arena, allowing researchers to focus more on the problem being solved instead of computer science oriented details.

Lets begin by looking more closely at the most important component of CHESS; its ability to automatically acquire knowledge. This component enables persons not familiar with expert systems or programming to serve as both the expert and the system builder.

KEY - Knowledge Extraction utility

The ability to automatically acquire, check, and process knowledge is of the utmost importance. KEY provides the expert with a way by which individual pieces of knowledge can be extracted and then fused together into a working expert system without requiring the services of a knowledge engineer.

The expert, a senior research chemist for example, defines a problem domain and injects some of his knowledge about that particular area into the KEY system. KEY analyzes this knowledge and provides the expert with automated techniques to isolate and eliminate inconsistencies and ambiguities. Once these are removed, the knowledge is converted into rules. These rules are the power that drives the inference engine. The result of

combining the two is the creation of a small expert system. An end user, a lab technician for example, will employ these rules and the inference engine to help answer their questions.

In order to illustrate how KEY works, let's look at a specific example. In the 1820's there were 50 or so elements that were known. While it was recognized that some elements were related, nobody had yet succeeded in ordering the elements in any sort of a periodic table. German scientist Johann Dobereiner made what is perhaps the first step towards such a classification. He pointed out that certain elements could be gathered into clusters of three based on similar chemical properties. The clusters (e.g., Cl-Br-I) became known as Dobereiner's triads.

Attempting to classify such subsets of elements demonstrates how KEY operates. Imagine that scientists of the day gathered all the information they had discovered about the elements. KEY could organize such information, creating a valuable reference source. Information pertaining to a new sample of an element could then be fed into KEY, and the system would determine which of the elements already present in its knowledge base most closely resemble the new sample. A virtually identical match would indicate the discovery of another form of an existing element - an allotrope. Moderately strong matches would indicate not only the possibility of a newly discovered element, but also indicate which elements shared similar properties.

Let's utilize some of our current knowledge of the elements to demonstrate the KEY system. In order to classify a subset of elements, a list of characteristics is needed. There are two ways of obtaining these characteristics. If the user was unsure what characteristics to utilize, KEY has the ability to assist in uncovering them by randomly taking groups of three elements and asking the user to provide a characteristic that distinguishes one from the other two. Another way is to provide KEY with a list. Ionization energy, atomic volume, acidity of representative oxides, elemental state, atomic weight, most stable oxidation state, atomic number, and electronegativity are 8 common characteristics we would employ today. Using the elements and their characteristics as labels, a two dimensional table or spread sheet can be built. A semiquantitative rating is placed at each element and trait intersection. The value represents the degree to which the elements possesses the trait. Figure 1 represents such knowledge. This information must be analyzed for any inconsistencies.

There are many different ways to analyze such grids. Most of them are based on statistical methods that perform a distance calculation between two elements or traits. Factor analysis attempts to account for as much of the variation as possible with as few factors as possible.


```

Cl = Chlorine
Br = Bromine
I = Iodine
Ca = Calcium
Sr = Strontium
Ba = Barium
Li = Lithium
Na = Sodium
S = Sulfur
Se = Selenium
Te = Tellurium

Cl Br I Ca Sr Ba Li Na S Se Te trait pairs
5 5 4 2 1 1 1 1 4 4 3 Hi-ionization-e = 5, lo-ionization-e = 1
5 5 4 2 2 1 2 1 4 4 4 hi-electronegat = 5, lo-electronegat = 1
2 2 2 3 3 4 1 2 1 1 2 hi-atomic-volum = 5, lo-atomic-volum = 1
2 3 5 2 3 5 1 1 2 3 5 low-atomic-weig = 5, hi-atomic-weigh = 1
1 1 1 5 5 5 5 5 1 1 2 oxide-acidity = 5, oxide-basisity = 1
5 3 1 1 1 1 1 1 1 1 1 gaseous-elmtl-s = 5, solid-elmtl-st = 1
2 3 5 2 4 5 1 1 2 3 5 high-atomic-num = 5, low-atomic-numb = 1
1 1 1 2 2 2 1 1 5 4 4 hi-oxidation-st = 5, lo-oxidation-st = 1

```

Figure 1. Example grid using 11 elements and 8 traits

However, it is a mathematically rigorous procedure. Conversely, correlation analysis is much easier to perform. It simply subtracts one object from another. Objects that have a minimal difference are assumed to be similar.

Several analysis methods are available to users of KEY, and others can be easily added. The present studies rely heavily on cluster analysis because it is relatively simple to implement, yet performs reasonably in eliminating problems in the grid. This method groups sets of objects into clusters, then determines similarity based on the distance between the clusters. Using correlation or cluster analysis, the user can analyze the grid in order to discover similarities and inconsistencies that may exist. The numbers generated by cluster analysis shown in Figure 2 represent a percentage similarity between the two elements.

Observe that KEY has automatically picked out the higher percent matches, based on a threshold level that the user can adjust. This is the information that scientists of the 1800's would have found most useful in their quest to order the elements. All but one of the similarities that are reported deal with elements that are neighbors in the same group. More careful thought reveals that Iodine and Tellurium are in the same row of neighboring groups.

Imagine that a new element has just been discovered and one would like to decide where it should go in the Periodic Table. The first step in analyzing our new element (lets call it Potassium) is to fill in all the appropriate values for each of the 8 characteristics and re-run our analysis (Figure 3). Notice the higher percent similarity between our new element and elements 4 and 8, with Ca = 85 and Na = 85. A threshold level of 85% prevented the similarities Sr = 79, Li = 79, and Ba = 75 from being shown. Similarities below the cutoff value are assumed not to be essential in analyzing the grid.

Ca and Na show the highest of the matches with our new element. Should our new element go at the intersection of these two elements? Today, we in fact know that it does. This highlights a strength of using such a procedure. Often in the analysis stage, the experts that are using the system will uncover new interrelationships. Or they may discover a new way to look at a problem.

Note that the correlation analysis has not produced any inconsistencies in the grid, only that certain of the elements were not totally distinguished one from another. Ordinarily, this type of analysis process is an iterative one, where a new trait is supplied to more totally distinguish between two elements, or perhaps where two traits are combined into one (for instance, atomic number and atomic weight are related). What is left is a non-ambiguous grid.

NOTE - Cluster analysis has determined that several of the elements in the grid have similar values for at least 85 percent of the traits currently present in the grid. The similar elements are listed below:

A 94% match : elements 7 (Lithium) and 8 (Sodium).
 A 91% match : elements 9 (Sulfur) and 10 (Selenium).
 A 88% match : elements 1 (Chlorine) and 2 (Bromine).
 A 88% match : elements 4 (Calcium) and 5 (Strontium).
 A 85% match : elements 5 (Strontium) and 6 (Barium).
 A 85% match : elements 3 (Iodine) and 11 (Tellurium).

Please hit RETURN to continue -

Figure 2. Results from a Cluster Analysis

NOTE - Cluster analysis has determined that several of the elements in the grid have similar values for at least 85 percent of the traits currently present in the grid. The similar elements are listed below:

A 94% match : elements 7 (Lithium) and 8 (Sodium).
 A 91% match : elements 9 (Sulfur) and 10 (Selenium).
 A 88% match : elements 1 (Chlorine) and 2 (Bromine).
 A 88% match : elements 4 (Calcium) and 5 (Strontium).
 A 85% match : elements 8 (Sodium) and 12 (Potassium).
 A 85% match : elements 5 (Strontium) and 6 (Barium).
 A 85% match : elements 4 (Calcium) and 12 (Potassium).
 A 85% match : elements 3 (Iodine) and 11 (Tellurium).

Please hit RETURN to continue -

Figure 3. Results from the second Cluster Analysis

In our example, not all of the traits are equally important or distinguishing. The next step in the procedure is to rate each of them on a scale of 1 to 5 (Figure 4), where 5 connotes a high degree of importance. After all of the inconsistencies have been removed and the relative uniqueness of the traits has been established, KEY can generate rules.

Figure 5 shows several representative rules generated by the system automatically. Each rule has a probability associated with it. These are called certainty factors (CFs). CFs are a measure of how certain the system can be of that particular rule. Here, CFs take on values between -100 and 100. Another scale frequently used is 0 to 100. The CFs are a product of three terms: the first is the grid value, the second is the importance of the trait, and the last is dependant on which CF scale the inference engine utilizes.

These rules are ready to be run by an inference engine. The inference engine is a separate entity from the KEY system. In the present work, a rudimentary inference engine was written in Prolog to test the knowledge base that KEY generated. Each answer the inference engine provides is a result of the combination of certainty factors from several rules.

This is the point where the end user, a lab technician for example, would first come in contact with the system. Perhaps a sample has been submitted that is slightly different from others being tested, and established procedures are not adequate. The analyst would answer questions posed by the system. In the example shown (Figure 6), the analyst describes the unknown element as having a high ionization energy, a high electronegativity, a low atomic volume, etc. The inference engine produces a list of solutions to the problem (Figure 7), and correctly identifies the unknown as Chlorine.

The methodology that lies behind this type of approach is called Personal Construct Theory, an area originally developed by Kelly (1). This theory, when applied to automated knowledge acquisition, provides the user with a method by which knowledge can be systematically extracted and inserted into a knowledge base. Boose's work at Boeing Computer Center (2) has produced numerous knowledge based systems utilizing this technology.

Classification and diagnosis problems have been treated successfully by this approach. Problems that require planning are not easily implemented because the user cannot initially specify all of the goals.

This approach allows a competent analyst to build an expert system for methods development without becoming a programming expert. It is the heart of CHESS. The other components in CHESS are described below.

```

Cl = Chlorine
Br = Bromine
I = Iodine
Ca = Calcium
Sr = Strontium
Ba = Barium
Li = Lithium
Na = Sodium
S = Sulfur
Se = Selenium
Te = Tellurium
K = Potassium

Cl Br I Ca Sr Ba Li Na S Se Te K K (rating) trait pairs
5 5 4 2 1 1 1 1 4 4 3 1 (5) Hi-ionization-e = 5, lo-ionization-e = 1
5 5 4 2 2 1 2 1 4 4 4 1 (4) hi-electronegat = 5, lo-electronegat = 1
2 2 2 3 3 4 1 2 1 1 2 5 (4) hi-atomic-volum = 5, lo-atomic-volum = 1
2 3 5 2 3 5 1 1 2 3 5 2 (4) low-atomic-weigh = 5, hi-atomic-weigh = 1
1 1 1 5 5 5 5 1 1 2 5 (3) oxide-acidity = 5, oxide-basidity = 1
5 3 1 1 1 1 1 1 1 1 1 1 (2) gaseous-elmtl-s = 5, solid-elmtl-st = 1
2 3 5 2 4 5 1 1 2 3 5 2 (4) high-atomic-num = 5, low-atomic-numb = 1
1 1 1 2 2 2 1 1 5 4 4 1 (3) hi-oxidation-st = 5, lo-oxidation-st = 1

```

Figure 4. A complete Grid with all elements and trait ratings

```

( (RULE 1 (element Chlorine 100)) if (ionization-energy hi-ionization-energy))
( (RULE 2 (element Chlorine -100)) if (ionization-energy lo-ionization-energy))
( (RULE 3 (element Bromine 100)) if (ionization-energy hi-ionization-energy))
( (RULE 4 (element Bromine -100)) if (ionization-energy lo-ionization-energy))

( (RULE 27 (element Iodine 40)) if (electronegativity hi-electronegativity))
( (RULE 28 (element Iodine -40)) if (electronegativity lo-electronegativity))

( (RULE 59 (element Sulfur 80)) if (atomic-volume hi-atomic-volume))
( (RULE 60 (element Sulfur -80)) if (atomic-volume lo-atomic-volume))

( (RULE 85 (element Chlorine 60)) if (acid-base-of-oxide oxide-acidity))
( (RULE 86 (element Chlorine -60)) if (acid-base-of-oxide oxide-basisity))

( (RULE 105 (element Tellurium 30)) if (acid-base-of-oxide oxide-acidity))
( (RULE 106 (element Tellurium -30)) if (acid-base-of-oxide oxide-basisity))

```

Figure 5. Typical rules generated by KEY

WHICH OF THE FOLLOWING IS TRUE ABOUT ionization_energy

A) hi_ionization_energy

B) lo_ionization_energy

Please type in your response as A/B -> A

You chose A, which was hi_ionization_energy

WHICH OF THE FOLLOWING IS TRUE ABOUT electronegativity

A) hi_electronegativity

B) lo_electronegativity

Please type in your response as A/B -> A

You chose A, which was hi_electronegativity

WHICH OF THE FOLLOWING IS TRUE ABOUT atomic_volume

A) hi_atomic_volume

B) lo_atomic_volume

Please type in your response as A/B -> B

You chose B, which was lo_atomic_volume

Figure 6. Questions asked by the inference engine running the rules generated by KEY

Chlorine WITH A PROBABILITY OF 78.85

Bromine WITH A PROBABILITY OF 68.97

Sulfur WITH A PROBABILITY OF 52.29

Selenium WITH A PROBABILITY OF 43.09

.

.

.

Strontium WITH A PROBABILITY OF -46.91

Barium WITH A PROBABILITY OF -64.50

Figure 7. Reported Results

Additional Tools in CHESS

The internal representation of molecules is accomplished using the technique developed by Wipke and Dyott (3), and later used by Molecular Design Limited (MDL) in several of their commercial programs. An MDL program, MACCS, is used to graphically input the molecular structure of the compound of interest, then save that structure into a file (molfile). The importation of this file provides CHESS with information such as the number and type of atoms and bonds, as well as stereochemical information.

These files serve several other functions. Each reaction stored in CHESS needs to know structures of products and reactants. This is accomplished using reference numbers that point to the molecular reaction files of CHESS. Other knowledge must be placed into these reaction libraries. All of the information must be represented in such a way as to permit CHESS to reason on the stored reaction mechanisms.

There are several different levels of reaction libraries, separated according to the level of reasoning that needs to take place. The first level of the library contains specific methods of analysis. If a match occurs here, the system would be acting simply as a methods reference book. The information needed at this level includes such parameters as the procedure, reactants and their concentrations, expected products, reaction conditions, procedural notes and a literature reference where the method was obtained.

Another level involves an analytical procedure which almost matches the users requirements. Perhaps the analyte molecule has a slight structural difference from the stored procedure, requiring a change in the pH of the medium during the analysis. Such slight differences in structure or reaction conditions require that CHESS have the ability to reason. Parameter estimation and calculation can be performed using Linear Free Energy Relationships (LFERs) and other types of additive relationships to predict properties.

For example, when the system needs to know the pKa of a molecule, there are several methods that can be used. The fastest and easiest way to determine the pKa is to look it up in tables of pKa values stored within the CHESS system. If no value is found there, it can be calculated using LFER's such as the Hammett equation. Hammett rho and sigma values are stored in tables inside the CHESS system. Situations that deviate from linear additivity are noted in an exceptions table to prevent erroneous calculations. If a calculation is not possible, the user would be asked to determine the value experimentally.

Other additive relationships in spectroscopy allows CHESS to predict spectral characteristics of a molecule. These allow calculation of UV lambda-max and e-max values using the methods developed by Woodward and Fieser (4).

Base values for parent molecules and increments for substituents are stored in tables. CHESS utilizes the molfile representations to determine base values and increments for calculations.

GUESS

GUESS, or the General pURpose Expert System Shell (5), is a domain independent expert system shell developed in the Computer Science department at Virginia Tech. It provides a variety of powerful tools to help construct, modify, and maintain an expert system. GUESS utilizes and merges ideas from several different shells. Its characteristics include diverse data representations and control structures, natural language capabilities, built in debugging aids, data base construction facilities, strong pattern matching capabilities and language generality based on Prolog. GUESS manages all of the modules found in CHESS.

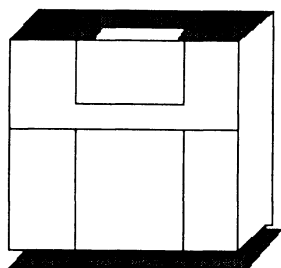
The major factor behind choosing GUESS is that all of the source code is available. New functions can be easily written and added to the system. Few of the commercially available shells currently offer their users source code that would allow this kind of on-sight modification.

A common theme running through all of the modules mentioned thus far is that CHESS has some general background knowledge of chemistry. The ability to represent and manipulate a molecule must be based on knowledge about atoms and bonds. Similarly, the ability to reason on reaction mechanisms must be based on knowledge about reactants, products, and the appropriate reaction conditions.

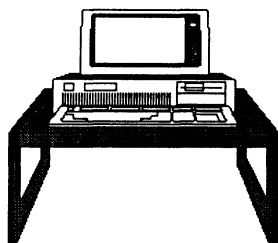
For instance, in higher levels of the reaction library, general guidelines for developing a method are provided. These general guidelines are based on converting a certain functional group to a spectroscopically detectible product. This requires the system to determine what functional groups are present in the molecule of interest. Prolog is the most appropriate language for implementing this type of knowledge because of the ease with which it implements relentless searching.

IMPLEMENTATION

A VAX environment was chosen for the development of the CHESS system for a number of reasons (Figure 8). First is the belief that small computers, such as PC's, do not offer the resources necessary to construct moderate to large scale expert system. Once developed, however, some systems may be ported to smaller computers. Another important reason is that MACCS is VAX based. GUESS was also originally written in a VAX environment.



VS.



<u>Name</u>	<u>Language</u>
GUESS	Prolog
Molfiles	MACCS
Chemical Knowledge	GUESS
LFER's	C
Reaction Library	GUESS
KEY Inference Engine	C
	GUESS (Prolog)

<u>Name</u>	<u>Language</u>
KEY Inference Engine	C
	Prolog

Figure 8. CHESS Architecture

A PC AT was chosen as the initial development platform for the KEY system. This environment was chosen partially because of the development tools that were available. This choice also allowed the KEY system to be a separate entity from CHESS, to allow developers to investigate ideas in a more personal environment. The finished system can be ported to the VAX environment for its inclusion into the CHESS system.

A majority of the CHESS inference engine involves the storing and manipulation of facts. This prompted the selection of Prolog, specifically HC Prolog developed at Virginia Tech (6), as the implementation language.

KEY is written in the "C" programming language because of all of the mathematical and pointer manipulations that need to take place. An excellent symbolic debugger for PC-based "C" is available. However, the finished system is easily ported back to the VAX environment.

"C" as a language is better defined than Prolog, hence transporting programs written in "C" between the PC and VAX environments is much easier (Figure 8). To this date, no syntax errors have been encountered. Implementation of functions that occur in PC-based "C" but not in VAX "C" (e.g. high level string manipulation) are easily built from available lower level functions. The transfer of Prolog programs has not gone nearly as smoothly. There are many syntactical differences between the VAX HC prolog and that for the PC.

CONCLUSION

CHESS has developed into a much more formidable problem than was originally envisaged. Its logical extension toward a facility that can suggest an analytical procedure for a new molecule for which no model recipe is available has proven extremely troublesome. Many workers in the field have discovered that expert systems should first be used to solve the simpler problems that represent the most frustrating component of professional life. In that area CHESS has had some success. In particular the involvement of personal construct theory in the development of an automated knowledge acquisition tool has proven value. The concepts and approaches developed in the effort will hopefully be of value to the developers of the second and third generation LIMS packages required by the modern industrial laboratory.

Acknowledgment

The authors wish to thank Dr. Keith Caserta of the Procter & Gamble Company for the support of this research.

American Chemical Society
Library

1155 16th St., N.W.

Washington, D.C. 20036

LITERATURE CITED

1. Kelly, G. The Psychology of Personal Constructs; Norton: New York, 1955.
2. Boose, J.H. Expertise Transfer for Expert System Design; Elsevier: New York, 1986.
3. Wipke, W.T.; Dyott, T.M. J. Am. Chem. Soc. 1974, 96, 4825.
4. Silverstein, R.M.; Bassler, G.C.; Morrill, T.C. Spectrometric Identification of Organic Compounds, 3rd. ed.; John Wiley & Sons: New York, 1974; p 241.
5. Lee, N.S. Masters Thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, 1985.
6. Roach, J.; Fowler, G. Proceedings 18th Hawaii International Conference on Systems Sciences, 1985, p 88.

RECEIVED July 7, 1989

Chapter 5

Large-Scale Scientific and Engineering Computation

Design and Implementation of a Reactive Flow Simulator

David Edelson^{1,2}, R. C. Lacher^{2,3}, C. E. Martin²⁻⁴, S. Varma³, and H. Wang³

¹Department of Chemical Engineering, Florida A&M University/Florida State University, College of Engineering, Tallahassee, FL 32316-2175

²Supercomputer Computations Research Institute, Florida State University, Tallahassee, FL 32306-4052

³Department of Computer Science, Florida State University, Tallahassee, FL 32306-4019

⁴Florida State University Computing Center, 2035 East Paul Dirac Drive, Tallahassee, FL 32306-3042

In a previous paper (1) we discussed the concept of knowledge based systems acting as scientific assistants to numerical investigation in the hardware context of local workstation and remote mainframe and made a case for the feasibility of such an assistant for the scientist and engineer investigating various reactive flow problems. The organization of a project to implement a knowledge based reactive flow solver was described. We present an update on the progress of the project along with some details of the prototype implementation.

Intelligent Scientific Computing Systems

The computing environment for large scientific or engineering calculations is gradually becoming one of distributed processing, with each of several phases of a computation being performed on a different machine particularly suited for that operation. Thus numeric-intensive tasks are run on a supercomputer (which may further subdivide the job for vector or parallel processing); graphic output is displayed on a high resolution color workstation; software development and prototyping is done on a machine optimized for FORTRAN; database searches on a file server; AI on a LISP machine etc. The ideal envisions the user controlling the entire scenario from one position, connected by transparent networks to all the other components, some of which could be located remotely from his own institution (indeed some of the locations could even be unknown to him). He interacts with a supervisory program, which monitors the progress of his problem, provides him with needed information, and requests input through a series of windows. The routine decisions are made by machine experts, leaving the user to concentrate on the science of his problem and relieving him of the need to manage (and the expertise to do so) the computation. In the case of many complex problems, this latter can amount to better than 90% of the investigator's time, and restricts the participants in the discipline to the small number willing to learn the needed aspects of applied mathematics and computer science in addition to their own field. It has become a very inefficient and expensive way of solving problems.

NOTE: This chapter is part 2 in a series.

0097-6156/89/0408-0049\$06.00/0

© 1989 American Chemical Society

Fortunately, the technology is rapidly becoming available to allow great strides to be made in improving this situation.

During the last two years, the authors and their colleagues have been engaged in a prototype design of an expert system for achieving this environment, and implementing it for a particular knowledge domain. The project is entitled KNODE (for Knowledgeable NODE) to convey the overall structural concept of a hierarchy of machines connected as nodes on a network, each with its own expert system to manage its particular function in the scheme. The application selected for the pilot implementation is the field of reactive flow. The choice was guided by several factors: the expertise and experience of one of the authors in this field (2); the maturity of the field, in the sense that many problems have been reduced to well-defined chemical, physical, and mathematical formulations; the existence of a large body of mathematical and computational methods for solution of the problems; a large database of physical and chemical parameters needed to describe the systems; and the relevance to a wide range of important scientific and engineering problems.

This paper will review the design concepts which have emerged from this study, and will present a report on the progress in implementing some of the modules of the system. Completion of the project is expected to take a number of years, as work on various components is driven by the interests and aptitudes of members joining and leaving the research group. (This would also provide material for an interesting study in project management and personnel dynamics, but must await a later phase of the study.)

Overview of KNODE

The outermost structure for our KNODE is illustrated in Figure 1, which shows the major functional components and flow of data. Scientific data, in the form of numbers; specifications of chemistry, physics, or geometry; algorithms; code; or remote host output, generally flows clockwise around the diagram as indicated by solid arrows. Control flow is generally outward from the System Supervisor module. Data in the form of knowledge may flow in various directions as indicated by shaded arrows.

This basic functional and interface structure derives from the natural division of labor and consequent communication steps that would exist if the computation were being handled by a team of human experts: scientist/user - applied mathematician - scientific programmer - job controller - graphics specialist. The scientist/user is of course leader of the research team, which will learn from him as time progresses. Ultimately, the KNODE will also be capable of learning from its scientist/user in order to increase efficiency of scientific advancement, so that the KNODE will act as a sort of investigative assistant to the scientist/user. While the basic computational tasks of KNODE can be inferred from Figure 1, comments on some of the processes implied by that figure are in order.

The System Supervisor. The system supervisor is the administration component of KNODE. It must define and control access to the other modules of the system as well as manage most inter-module communication. A command interface will serve as the user's primary access to KNODE. The supervisor will translate user requests into appropriate instructions to be passed on to the other modules. The software modules of the system will be independent programs which communicate with each other by passing information through the supervisor and through information stored in intermediate files.

Scientific Language Interface. The SLI is a domain-specific interface designed to allow the intuitive expression of complex scientific problems in a language most natural for the investigator. The SLI is composed of an interactive display editor and an input language derived from the problem domain. In the

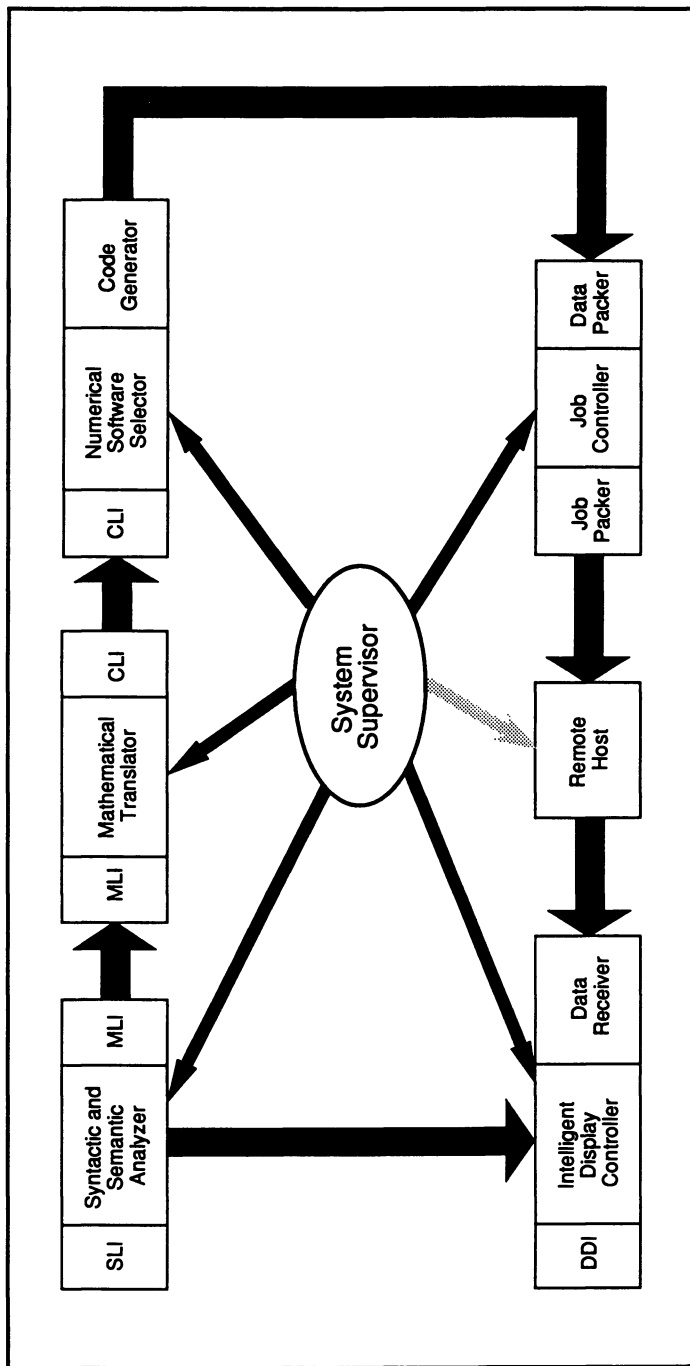


Figure 1. Principal modules and data pathways of KNODE.

KNODE project this will be the language of chemistry and fluid dynamics. As with many areas of scientific endeavor there is not always complete agreement among researchers regarding standards of notation. In the areas where there is more than one form of notation in wide use to express the same information, the SLI will, wherever possible, be devised to accept any form. Where notational conflicts arise, the most widely used notation will be adopted.

The information which describes a problem in reactive flow is a chemical mechanism, *i.e.* a sequence of chemical reaction steps which converts the starting material to the products; a geometry in which this occurs; and a specification of forces or flows from which the motions of the reacting media can be derived. Thus there are a number of disparate types of information to be provided, which indicates a decomposition into a number of separate input modules each of which is designed for a particular type of information. Here we need a chemical language editor and analyzer for the mechanism; a graphics system for specifying the geometry; and a method of mapping material and energy sources and sinks on to the graphic representation. The chemical language editor, in its ultimate refinement, could require as input nothing more than a list of starting compounds and conditions, then retrieving from a database all possible reactions. Since this utopia is not presently achievable, one could envision providing a list of reaction types, and generating the mechanism from this information. Limited implementation of this idea, particularly for free-radical reactions where the number of chemical equations and species can grow exponentially large, have been reported (3). At a lower level, and particularly for speculative mechanisms, the input of individual reactions is required. Our latest version of such an input interface/editor is presented in the next section.

Accessory to each reaction step is a requirement for a rate constant or expression. Here again a search of the literature is the ultimate refinement, returning the best available value. This possibility, although talked about for many years, is only now beginning to be realized (4). Should a reliable value not be found in the literature, a transfer to a quantum calculation, RRKM or other estimate, could be attempted; again these methods are also in their infancy. Finally, the user might be interrogated for his own estimate in an Arrhenius or other form, or even asked to consider the possibility that the reaction given does not occur under the specified conditions.

Similarly, the transport model could be specified generically as batch, constant volume or pressure, CSTR, plug flow, or other standard reactor design, causing a diagram to immediately appear on the screen with prompts for needed parameters and boundary conditions. At a lower level, a graphics editor/interpreter would be used to create the flowchart and superimpose the numerical values.

If energy sources and sinks are involved in the problem, the formulation of the energy balance becomes another function of the SLI. Additional databases are required as well: heat capacities, heats of formation, free energies, entropies. In some problems the proper choice of equations of state, deviations from ideality, etc. become important. This information is better organized than the kinetic data, largely through the efforts of the chemical engineers (5). Some simulation programs incorporate elaborate computational procedures for these thermodynamic quantities, but these modules are rarely available separately. A notable exception is the JANAF tabulation of thermodynamic data, which has been available in computer-readable format for several years.

Mathematical Language Interface. The inputs are translated into the relevant mathematical form, generally differential equations (with possible algebraic components or constraints) for the reactive flow problem (6). The chemical mechanism would become a set of mass-action equations as a default, or a user-supplied expression. Convection or advection translate to first-order partial

derivatives, and diffusion to second order. The required boundary conditions would be defined. Finally the resulting set of equations would be characterized as predominantly parabolic, elliptic, hyperbolic, or mixed, and a strategy selected for their solution.

A key aspect of the KNODE project is the creation of a module for the selection of the most appropriate available software to solve these systems. In making selections of software solvers for the problems discussed above, several commercially available libraries will be used. Among these are the IMSL (7), NAG (8), and PORT (9) libraries. Special-purpose packages, such as DSS/2 (10) and GRD1 (11) will also be considered. In addition, more specialized codes are being researched for inclusion in the domain of solvers to be considered. An expert system will be used to make *a priori* selections of solvers for problems, monitor the progress of the solution, and switch from one solver to another, as the needs arise. We feel that such a process of intelligent software selection, coupled with judicious monitoring and package switching, would provide optimal solution accuracy, while minimizing computer time. As better, adaptive or knowledge-based, mathematical software becomes available, this expert will be kept up to date on their capabilities. Various knowledge representation schemes are being considered for the software expertise. Production rules and frames are being researched, as well as relational databases. Criteria for deciding on a knowledge representation scheme include ease of adding to or modifying its knowledge base, modularity, and search efficiency. In addition to the basic capability for solving the differential-equation problem, the possibility of including one or more corollary computations, such as sensitivity or identifiability analysis (12), statistical data handling or parameter extraction, mass and energy balance monitoring, or economic optimization could be envisioned as future options.

Computational Language Interface. Once the most appropriate numerical method has been chosen, it is the function of this module to search its list of applicable software packages resident in libraries on machines to which KNODE has access, and find the best one for the present purpose. This machine-expert software librarian would then further have to supply information to enable the CLI to code interfaces to the solver subroutines, manage the data space, etc. The output of this module is a complete FORTRAN (or other language) program, with proper library calls, I/O management, and problem dependent subroutines for solving the problem together with needed input data files and provision for returning output data files. The assumption is that the problem will be run in batch mode on a remote host; this is the generally accepted procedure today, but should future developments alter this the CLI must be prepared to adapt to a new environment. In the event the remote host is a supercomputer, the CLI must also make provision for optimizing the code for the particular machine-dependent compilers. This may be performed by a pre-compiler resident on the supercomputer itself, by a knowledgeable optimizer on the user's workstation, or a combination of these.

Job Control. This module must receive a program, including FORTRAN (or other language) main programs, library interfaces, data files etc., package a complete job for a designated remote host, ship the job, monitor the job while it is out of control of the local host, retrieve the job when (and if) it is complete, and send returned data off to the appropriate elements of KNODE. The job controller must be capable of reacting appropriately to various abnormalities that may occur and over which it has no control, for example a faulty remote host, job runtime exceeding prescribed limits or exhausting other resources in some way, or a full file system on the local host. It must have access to all information, some confidential, necessary to operate the scientist/user's accounts on remote hosts. And it must be capable of operating without the

benefit of query to the scientist/user. A variety of remote hosts, at truly remote locations, will be accessed.

Scientific Visualization Module. For reactive flow problems (and with many others) the output is generally in the form of space and time dependences of the computed variables. The amount of data can be tremendous, and it is the function of the Display module to organize and present it to the user in the most comprehensible form, generally graphic in nature. These displays can range from simple X-Y plots to time-sequences (movies) of multidimensional displays including the effective use of color, shading, line texture etc. The type of display is expected to be automatically selected by interaction with the SLI, with user direction and override if necessary. The display may be presented to the user's workstation, but since KNODE knows what other facilities are available, it may choose to use a high-level graphics workstation, hard copy printers or plotters, videotape or optical disk recorders for the most effective presentation.

A reaction digraph display will also be available as an aid in understanding the overall structure of complex reactions. This screen type illustrates the global course of the reacting system as a directed graph with nodes for the individual species taking part in the reaction and directed edges for chemical change pathways, with nodes and edges labeled appropriately. Design of such a display depends on the details of the input chemical system (to determine an appropriate digraph) and on the topology of the resulting digraph (to determine an appropriate planar immersion. It is largely independent of the numerical results, although some results may be inserted on the display as data, and numerical input may yield information about the isolation of subsystems, which may then be disconnected into separate displays. The only successful displays of this type known to the authors have been created through painstaking human effort. Methods of implementing an automatic reaction digraph display technology will be researched and implemented in the final versions of KNODE. Clearly, this technology will be knowledge-based. Although static displays are useful up to a point, the more natural display for a dynamic (*i.e.* time-varying) system is a representation which similarly varies in time. The logical outgrowth of a sequence of static graphs, each of which shows relationships at a particular value of time, is a movie which displays the full time evolution of the system. Movies have been made in the past by the painstaking process of time-lapse photography of CRT displays of single time frames (13) but the availability of fast graphic workstations as well as moderately priced video recording facilities now presents the possibility of making this type of display available almost immediately at the user's desk. While many of the hardware, software, and communication details remain to be worked out, and the economic factors are still appreciable, the design of these features into the KNODE system remains one of the goals of the project.

Phased Implementation

Scientific Language Interface. The development of the Scientific Language Interface for the chemical reaction language has been subdivided into two tasks, a "chemical language editor" (ChLE) and a "chemical language interpreter" (ChLI). Previous implementations for this purpose have combined both these functions in a single program but the performance enhancements we (14) wished to achieve at this time indicated that the programming languages and techniques, and even the machines used for these two functions could be different, and it was not realistic to constrain their implementation to the same environment.

The ChLE has been designed to take full advantage of the display power of the newest workstations. In contrast to the earlier methods of entering

chemical notation, with its hierarchy of subscripts and superscripts being simulated by escape or other control sequences, the present concept is that of a WYSIWYG editor which automatically requests input for the various character fields and properly displays the results. Thus a chemical element is limited to two character positions, with the first automatically capitalized; the screen cursor is positioned at each sub/superscript location requesting input; syntax checking is performed immediately and illegal characters are refused. Provision is made for overriding the default notation specification to allow arbitrary text instead of chemical formulas. Predefined symbols may be used for labor-saving in the entry of lengthy chemical notation; substitution is immediate and error free. Entries are not limited in length; the equation merely scrolls off to the left of the screen, but can be recalled by the use of the cursor control keys.

The ChLE operates in two modes: "input mode" or "edit mode". The input mode has been described above, with automatic sequential field positioning as input is entered. In edit mode, the cursor can be moved to any field for additions or changes. The diagonal keys on the numeric keypad (1,3,5,7) are programmed to move the cursor to the pre /post sub /superscript position; the vertical and horizontal arrows retain their normal function. Thus any field can be edited while the screen maintains an immediate update.

Once a reaction line is complete, it is passed (as an ASCII stream, with suitable embedded control sequences) to ChLI. Semantic checks are performed here (*e.g.* chemical and charge balance) and the reaction is decomposed into molecular species as well as atoms. An error causes a return to the editor with a request for correction. A molecule and atom list is displayed, and any atom or molecule may be selected (by a mouse, for example) and used as a token in future input. The Supervisor monitors the ChLE-ChLI interaction and controls the window manager of the user's workstation accordingly (see Figure 2).

Once a reaction has been satisfactorily entered and interpreted, it remains for a rate expression to be assigned. The default is the mass-action formulation; however for special purposes (in chemical engineering, for example, where the reactions are more likely to be lumped stoichiometric expressions rather than fundamental chemistry) (15) an alternative may be used. In the present implementation, a FORTRAN statement is used; it is conceivable that alternate mathematical expressions could be used, which would require another stage of interpretation (*e.g.* MACSYMA) for conversion to code.

At this point, the parameters in the rate expression need to be assigned. This could be done by searching a database, independent calculation, manual entry, or deferred to a later stage of the simulation. ChLI keeps track of all the unassigned parameters and will remind the user to satisfy the required input.

When the reaction mechanism is completely entered, ChLI has all the information needed to create a symbolic matrix to pass along to MLI, as well as a molecule and atom list for mass balance calculation, graphic or printed identification, etc. In particular, it can create a text-processor (*e.g.* TROFF, \TeX) representation for hard copy output and documentation. The reaction matrix is also available to other modules, in particular the display module, for the creation of flowcharts and other designs for the most effective presentation of results.

Job Packer/Controller. The Job Packer/Controller is the component of the KNODE project that handles all communications between the KNODE system and the designated computational host. It is designed to package the program produced by the Code Generator, submit the batch job to the designated computational host, transfer all output files back to the return host, and

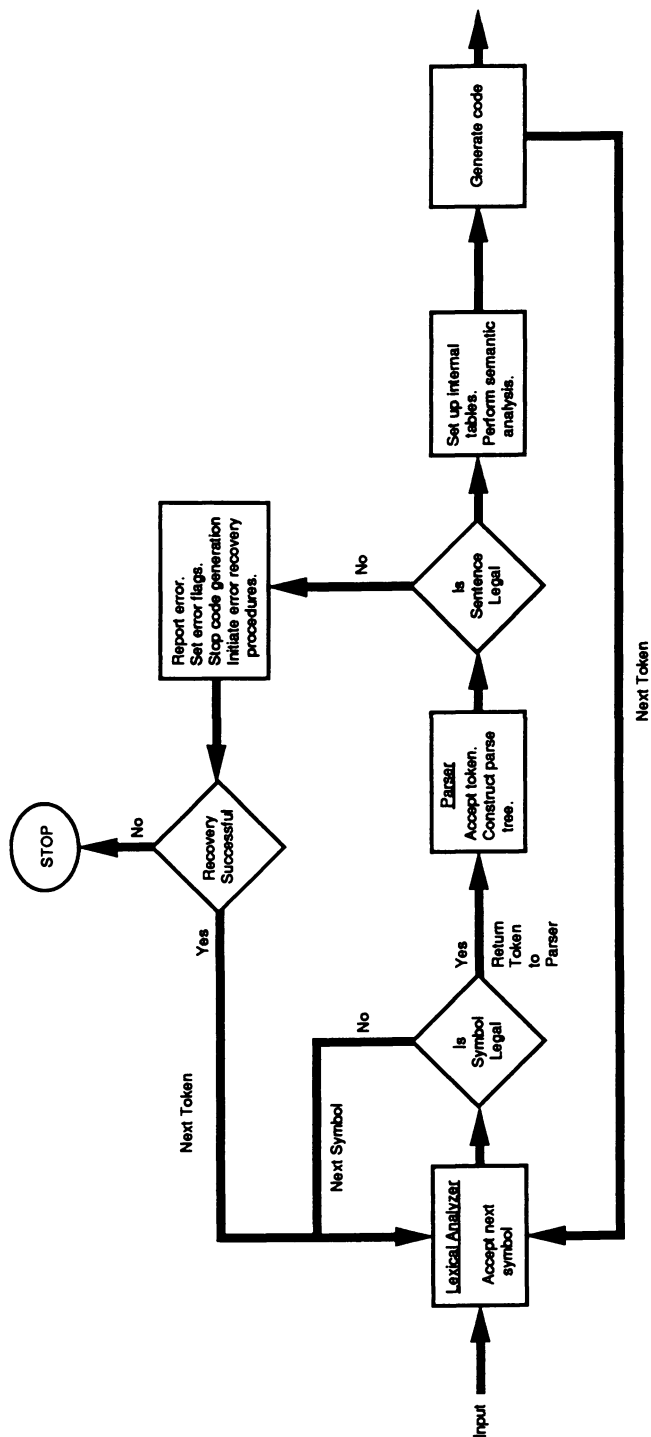


Figure 2. Syntactic and semantic analyzer function of the scientific language

notify the user that the job has completed. In general, the program produced by the Code Generator will be submitted to the computational host with the scientist having no prior knowledge about its operating system. Of course, this will require the Job Controller to be 'intelligent', i.e. it must have some type of knowledge-based expert system or database that will determine the necessary job commands to submit the program and retrieve the output. Figure 3 shows a functional flowchart of this module, while Figure 4 represents a database structure which provides the needed information on remote hosts and libraries.

The System Supervisor must provide the Job Controller with information about the program, including the computational host ID, the number and names of all the input and output files used by the program, and the names of the libraries or pools called by the program. The Job Controller will also have access to all the information necessary to operate the scientist/user's accounts on each computational host. Using the information obtained from the System Supervisor, the Job Controller will be able to decide which job commands are necessary to attach specified input and/or output files and to attach specified software libraries. The Job Controller will also be able to choose host parameters that are machine specific.

Once the program has been packaged with the correct job commands to compile and execute the code, the job will be submitted to the computational host. Extensive error checking is performed before the job is shipped to assure successful execution on the remote host, since the queueing delay may be the largest part of the elapsed wall-clock time required for the problem. A demon process will then be generated to monitor the job, which will allow other jobs to run concurrently. Once the output files are transferred back to the return host properly, the demon process will notify the System Supervisor and then halt. (Real-time interactive operating systems for supercomputers are under development, and the phantom arrow from the Supervisor to the remote host, Figure 1, show the future possibility of a monitor channel through this mechanism, but for the present it must be assumed that this does not exist.)

Eventually, the Job Controller will be able to package and submit programs from any host to any computational host, transfer all output files back to the return host, and notify the System Supervisor of the termination of the process. With these capabilities the Job Controller can be used not only with the KNODE project, but with other knowledge based systems with similar implementation or as a stand-alone asynchronous Job Controller.

There are a few notable communication problems that could be encountered between the computational host and the Job Controller. One problem that occurs regularly is a faulty computational host, that could abort the batch job abnormally. If this occurs, the System Supervisor must be notified, given the corresponding error code, in order to take the proper action. Another problem can be encountered when a communications link between the computational host and the return host is down. This will cause the output files to be 'LOST' until the link is back up and running properly.

We must also mention some problems that may occur when the output files are being transferred back to the return host. If, for instance, the return host has a full file system, the output files cannot be saved immediately, but the output files cannot return to the computational host either. The output files must reside on some type of scratch space until the System Supervisor can be notified.

There are a variety of other Job Controller system problems that could occur if the other KNODE modules are not running properly. If a file has been created improperly it is possible for the Job Controller to read the information

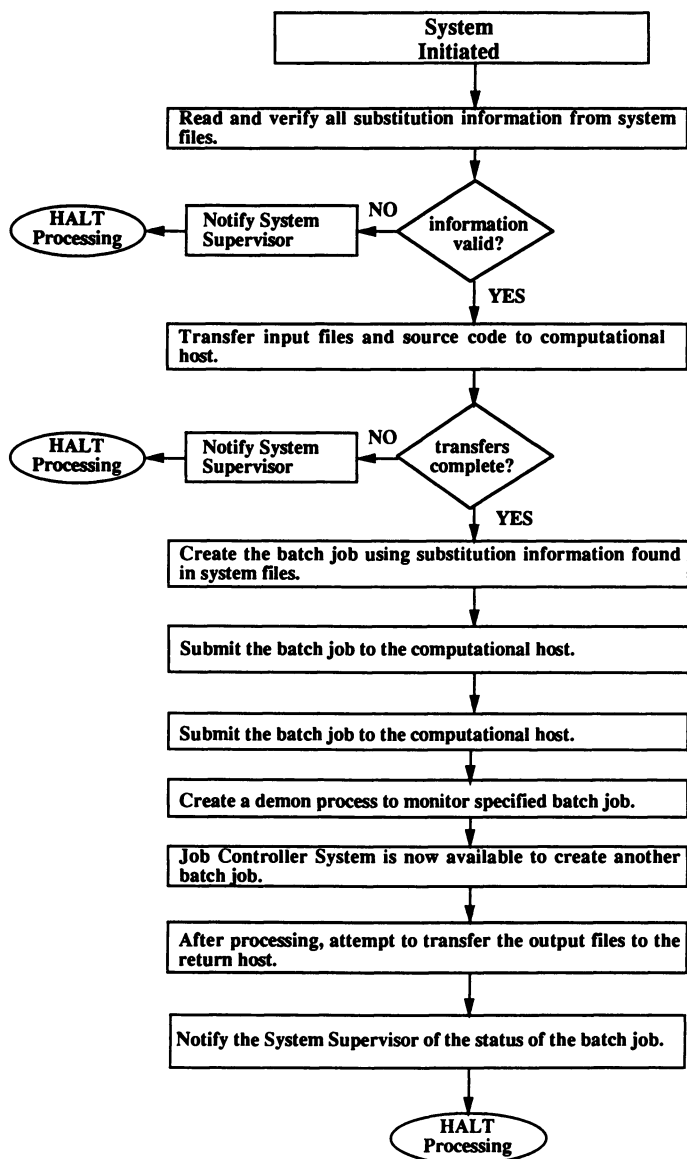


Figure 3. Control flow of the job control module.

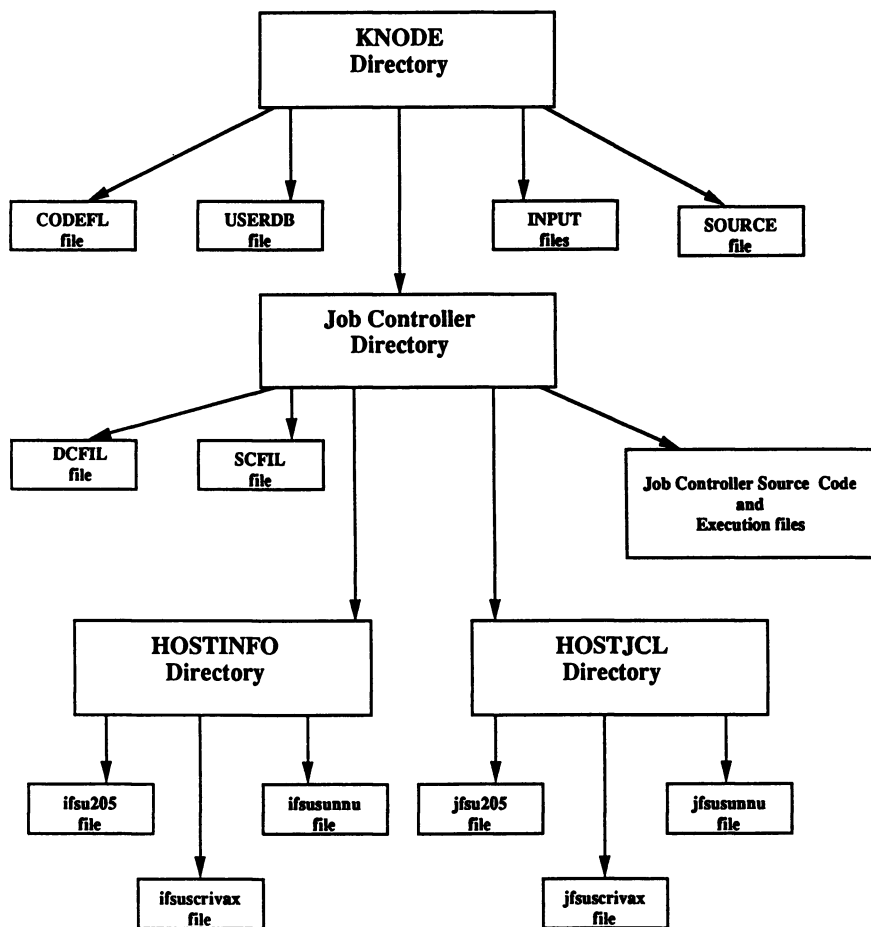


Figure 4. KNOBE file hierarchy as viewed from the job control module.

and suddenly abort with only a general system error message. This can be avoided with carefully designed editing and error checking modules to verify all the data files. Another problem can arise if the System Supervisor has failed to monitor the amount of resources remaining on the computational host. If a job has exceeded its prescribed limits or exhausted other resources in some way, it is possible that the program will not compile or execute due to the lack of funds. In this case, the System Supervisor can again be notified, with the corresponding error code, and proper action taken.

The pilot implementation of the Job Controller is being done on the Florida State University computer network. The KNODE workstations are Sun 3/60's running UNIX. These are networked to a VAX-780 also under UNIX with remote login, execute etc. This latter is in turn connected to the main University ethernet backbone, through which it accesses the Supercomputer front-end machines (a local area VAXcluster with an 8700, a 780 and numerous microVAX's and VAXstations) running VMS. Jobs are sent over a CDC network to a file server running CDC/NOS and to the supercomputers (CY205/ETA10) running CDC/VSOS. This configuration offers sufficient network/operating system diversity to present an environment in which most of the features (and difficulties) the Controller is expected to cope with are found.

Conclusions

This paper is a progress report of the initial phase of an effort which is expected to take several years for completion. Given the present rapid rate of evolution of new concepts in the field of expert systems and artificial intelligence, the development of distributed systems and networks, and continued introduction of new machine features, the design outlined in this paper could be modified during the course of development. Nevertheless, we feel that the basic modular concept will be adhered to, and that modifications will be in the nature of implementation details.

Acknowledgments

This research was supported in part by the Florida High Technology and Industry Council through an Applied Research Grant and by Florida State University through time granted on its CYBER 205 supercomputer. R.C.L. also acknowledges research support from the U.S. Office of Naval Research.

Literature Cited

1. Edelson, D.; et al *MATH/CHEM/COMP 1987*, Elsevier Science Publishers: Amsterdam, 1988; p 13.
2. Edelson, D; Rabitz, H. *Oscillations and Traveling Waves in Chemical Systems*, R. J. Field and M Burger, Ed.; John Wiley & Sons: New York, NY, 1985; ch 6.
3. Indritz, D. F.; Maday, M.; Sheinson, R.; Shaub, W.; *Construction of Large Reaction Mechanisms*; NRL Report 8498, 1981.
4. Herron, J. T.; Cvetanovic, R. J. *9th International Symposium on Gas Kinetics*, 1986.
5. Chase, M. W., Jr.; Davies, C. A.; Downey, J. R., Jr.; Frurip, D. J.; McDonald, R. A.; Syverud, A. N. *J. Phys. Chem. Ref. Data* 1985, 14, Suppl. 1.
6. Rosner, D. E. *Transport Processes in Chemically Reacting Flow Systems*; Butterworths: Boston MA, 1986.

7. *International Mathematics and Statistics Libraries, Inc.*, Sixth Floor-NBC Building, 7500 Bellaire Boulevard, Houston TX.
8. *Numerical Algorithms Group (USA) Inc.*, 1250 Grace Court, Downers Grove, IL.
9. Fox, P. A.; Hall, A. D.; Schryer, N. L. *The PORT Mathematical Subroutine Library* 1981, 4, 104.
10. Schiesser, W. *DSS/2 - An Introduction to the Numerical Methods of Lines Integration of Partial Differential Equations*, Lehigh Univ.: Bethlehem, PA, 1976.
11. Dissinger, G. R. *GRD1 - A New Implicit Integration Code for the Numerical Solution of Partial Differential Equations on Either Fixed or Adaptive Spatial Grids*, Doctoral Dissertation, Lehigh Univ.: Bethlehem, PA, 1983.
12. Rabitz, H.; Kramer, M.; Dacol, D. *Ann. Revs. Phys. Chem.* 1983, 34, 419.
13. Edelson, D.; Chambers, J. M. *Presentation to ACS Meeting*, 1975.
14. Edelson, D. In *Artificial Intelligence Applications in Chemistry*; B. Hohne and T. Pierce, Ed.; American Chemical Society Symposium Series 306: Washington, DC, 1986; p 119.
15. Berty, J. M. *Chemical Engineering Progress* 1988, p 61.

RECEIVED July 21, 1989

Chapter 6

New Directions in the SYNGEN Program for Synthesis Design

James B. Hendrickson, Zmira Bernstein, Todd M. Miller, Camden Parks,
and A. Glenn Toczko

Department of Chemistry, Brandeis University, Waltham, MA 02254-9110

We describe here the SYNGEN program for generating the shortest, sequential-construction routes to any target structure from a catalog of starting materials. The system of abstracting structures and reactions to digital generalizations for fast computer manipulation is described, as well as its utilization to generate and mechanistically test all possible sequences of construction reactions. Various new ways are described for the operator to make selections from the output. A program under development is described for interfacing SYNGEN with external reaction databases, in order to seek literature precedent for the generated reactions. Finally, a program is also described which proceeds from all starting materials in a *forward* direction to synthesize close analogs of a target using refunctionalization reactions as well as constructions.

A complete synthesis tree of all possible sequences to a synthetic target would be enormous, far larger than generally appreciated. An illustration of such a tree is shown as Figure 1, with reactions as lines (direction: left to right), the compounds as points, increasing generally in complexity from C_1 starting materials at the left, through many intermediates to the target molecule at the far right. Yields for each level back from the target are shown below. While the generation of such a tree is conceptually simple, its vast size dictates the paramount importance of creating stringent selection criteria to generate only the *optimal* routes. In the SYNGEN program we elected to seek only the shortest, most efficient routes. This focus resulted in a twofold protocol¹: first, only the skeleton is considered and only convergent skeletal assemblies from real, available starting material skeletons are accepted. These are derived by cutting the target skeleton all ways into two pieces, and each of these into two again, creating

0097-6156/89/0408-0062\$06.00/0
© 1989 American Chemical Society

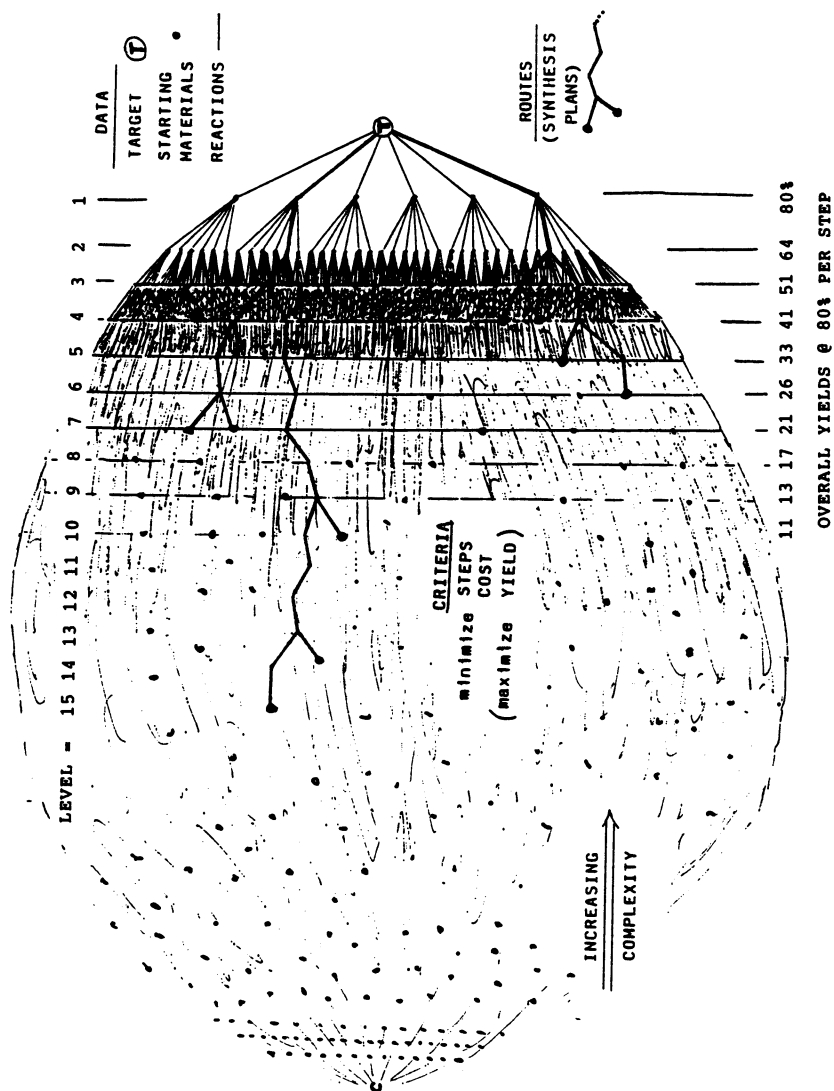


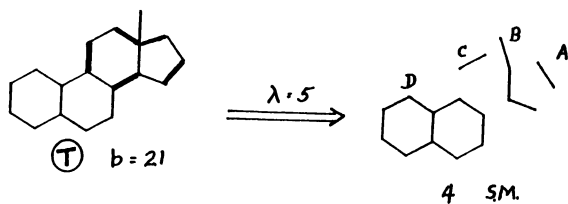
Figure 1. The Synthesis Tree

ordered convergent bondsets; only those with all four starting skeletons found in a catalog are then accepted. Second, the program generates the necessary functionality on those skeletons to afford the shortest routes, those with sequential constructions only through the ordered bonds of the bondset. Such routes are regarded as "ideal" syntheses¹. This proceeds in a retro direction from the target functionality for each bondset and ends with functionality defined on the four starting skeletons; the program accepts only routes with all four real starting materials found in the catalog. The overall procedure contains three stringent selection criteria which very much reduce the number of possible routes: (a) only convergent assembly modes from four starting pieces (two successive skeletal cuts); (b) only the shortest routes of successive construction reactions only; and (c) only those which derive from four real starting materials available in a catalog (our catalog has about 6000 starting materials). It can be calculated, for example, that the first criterion, the convergent skeletal dissection, reduces the possible assembly modes for the C₁₈ steroid skeleton of estrone from 42 million down to less than 900.

The first phase of selection, the skeletal *bondsets*, is described in Figure 2 which shows how dissection of the bondset bonds (k in number) directly defines the starting material (SM) skeletons. By itself a bondset does not indicate an order of constructions for the skeleton. While there are $b!/k!(b-k)!$ possible bondsets of k bonds, each one can be ordered in $k!$ ways, and, while the total number possible is very large, only a few are convergent. One such ordered bondset is shown for the C₁₈ steroid skeleton, with its corresponding *assembly plan*, the sequence of linking the k bonds together from SM to target. Such plans are in fact generalized synthetic routes describing independent families of sequences, taken from the synthesis tree, all of which are constructing the same set of k skeletal bonds. Furthermore, this convergent protocol affords a maximum of $k = 6$. Indeed, the simplest gross overview description of any synthesis is simply its ordered bondset, which affords its assembly plan (Figure 2).

The program does not use a database library of known reactions but simply generates all possible construction reactions in a generalized form, using a simple but rigorous and numerical description of molecules and reactions¹⁻³. The form derives from a synthetically fundamental definition of four kinds of attachment on any skeletal carbon: H for hydrogen (or electropositive element); R for r-bond to another carbon; P for p-bond to carbon; and Z for bond (r- or p-) to electronegative heteroatom. The numbers of each kind of attachment are then h, r, p, z , respectively, and add up to four. This is summarized in Figure 3, which also indicates that, if its skeleton is known and numbered, any molecule can be described by a digital zp-list, generalizing its functionality, over the numbered skeletal atoms. This yields a specific, though abstracted, full description of any molecule in a linear bit list suited to computer manipulation. The skeleton itself is digitally

BONDSET = Set of λ skeletal bonds constructed



Ordered Bondset:

Assembly Plan:

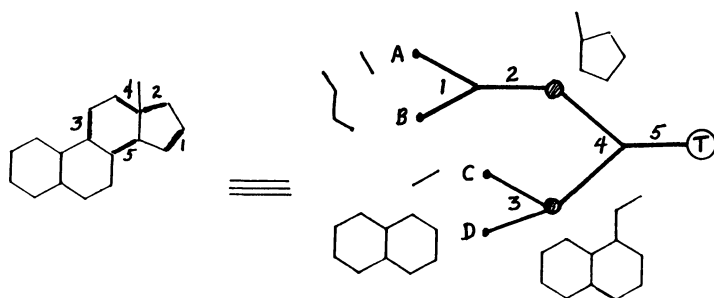


Figure 2. Bondsets and Assembly Plans

described, and uniquely numbered, by a simple manipulation of its connectivity, or adjacency, matrix⁴; this linear bit-list description is also used to derive the bondsets from the target skeleton. Figure 3 shows an example, which also illustrates that the oxidation state (x) of each atom, as well as that of the whole molecule (Rx), can be simply and accurately computed.

A *unit reaction* is the unit exchange of one kind of attachment for another and may be described at each carbon by two letters: the first the bond made, the second the bond broken. Hence the 16 possible unit reactions depict all possible changes at any carbon atom, listed at the bottom of Figure 3. Furthermore, these digital descriptors allow the calculation of the *reaction distance* (N_R) between any two compounds, i.e., the minimum number of unit reactions required to convert one into the other⁵.

The Basic SYNGEN Program

The construction reactions (RH, RZ, RP) are central to our procedure and a generalized form of any construction reaction is shown in Figure 4. Three carbons on each side, labeled a, b, c out from the bond formed, virtually always bear all the functionality that changes in a unit construction. All possible constructions across any designated bondset may thus be generated from the unit reaction changes, shown at the bottom; the various constructions will change attachments on 2-6 atoms and the only constructions used are those which are isohypsic³ ($RDx = 0$). A construction reaction can be seen as a combination of two half-reactions, one nucleophilic ($Dx = +1$) and the other electrophilic ($Dx = -1$), and generally these may be treated independently. Any generalized kind of construction will also be characterized by Dzp on each involved carbon and so can be described by a Dzp-list over the strand of six atoms spanning the bond formed. Hence the addition of the digital Dzp-list for any particular construction to the zp-list of that strand of atoms in a product molecule will generate the new zp-list for the substrate molecule in the retrosynthetic direction. Equally, subtraction of a construction Dzp-list from the zp-list of the substrate creates the product in the forward direction. This is shown in Figure 5 in which the Dzp-list for the Michael construction is applied to atom strands across the designated construction bond to generate the substrate from a particular product retrosynthetically. Either the descriptor list of Figure 3 (RH, RZ, etc.) or the Dzp-list generators illustrated in Figure 4 serve to summarize the *net structural change* in a given reaction.

With this digital formalism developed it is now an easy matter to generate all possible synthetic sequences of constructions only, from the previously selected convergent bondsets. These bondsets are created by cutting the target skeleton into two parts (cutting no more than two bonds) and the two parts into two more each. Thus each optimal bondset defines up to six bonds that will be constructed and the order of their construction. Starting

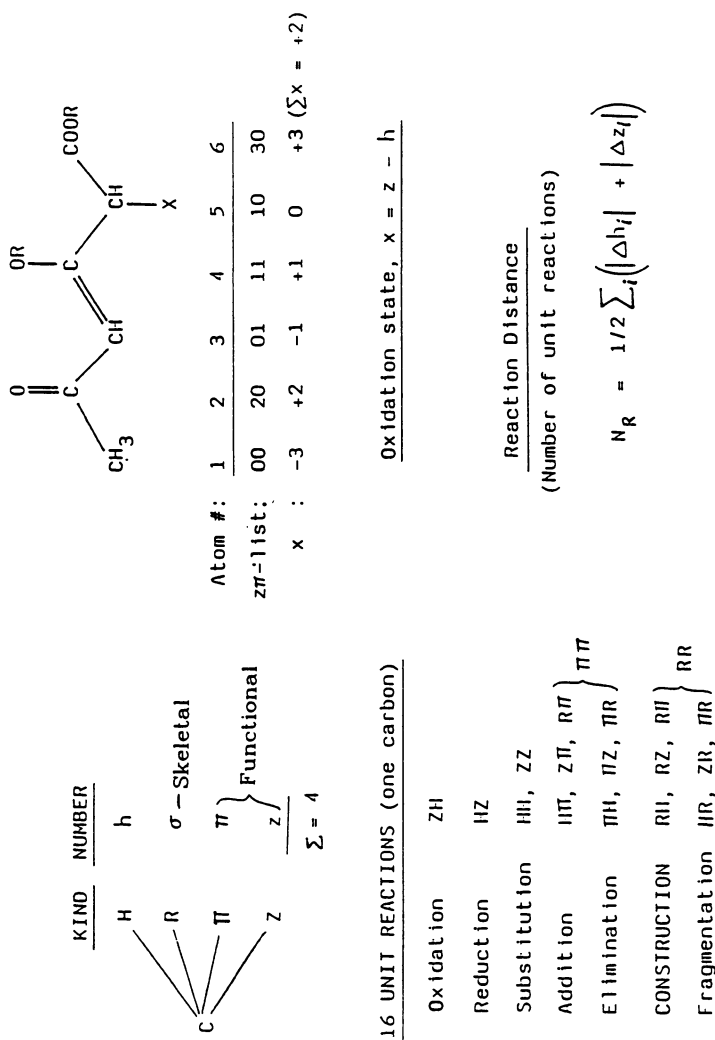


Figure 3. Characterization of Structures and Reactions

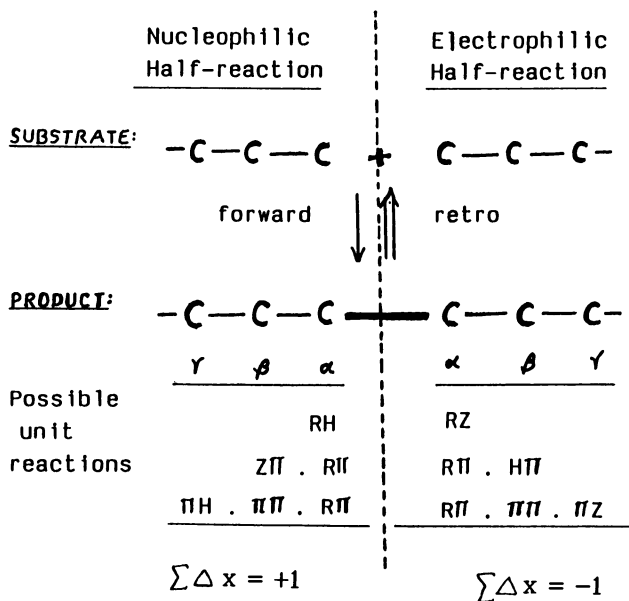


Figure 4. Generalized Form of Construction Reactions

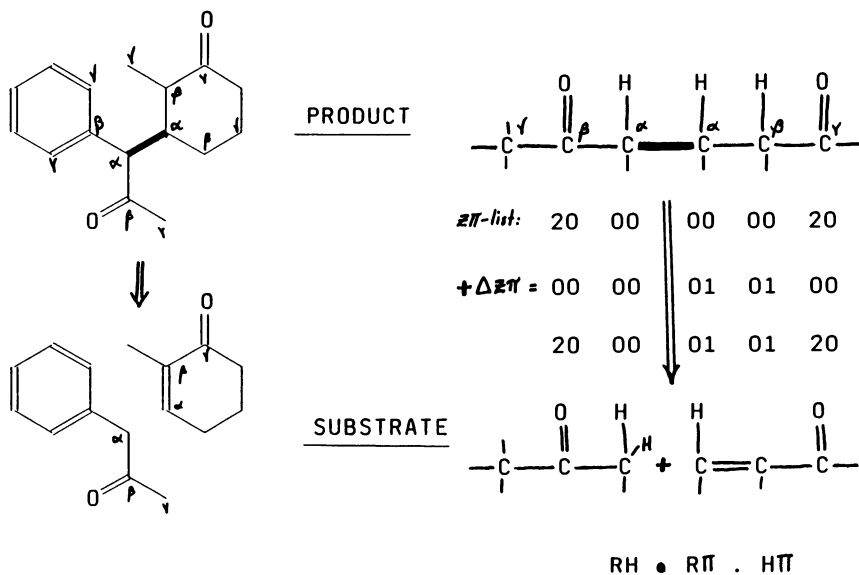


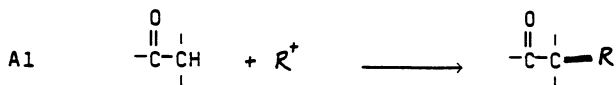
Figure 5. Example of Reaction Generation

retrosynthetically, the target functionality is defined as zp-lists for all strands of up to three atoms out from each end of the final constructed bond. To each such product zp-list is added the Dzp-list for each construction half-reaction to generate substrate zp-lists; only generated zp-lists of $(z+p) \setminus (4-r)$ are viable. Then nucleophilic half-reactions for each side are paired only with electrophilic ones on the other to define full isohypsic constructions and the substrates for these (as zp-lists). These substrate zp-lists now become the products for repeating the operation with the next bond defined in the bondset order. When all the bondset bonds have been sequentially treated in this way, there will result the zp-lists of the four generated starting materials and these can now be looked up in the catalog. If all four are found to be available compounds a successful synthetic sequence has been found and is recorded.

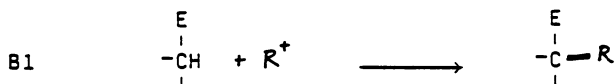
When this procedure is applied, using the nine possible pairs of three nucleophilic and three electrophilic half-reactions (Figure 4), the results obtained show many viable sequences but reveal two shortcomings. First, some common one-step constructions do not appear and, second, many of the generated reactions are chemically non-viable. The constructions which do not appear are those in which the actual "one-step" reaction is in fact two successive unit reactions, a construction and a refunctionalization. Thus the Wittig reaction is a construction followed by an elimination, while the common Grignard half-reaction consists of a prior reduction of RfX to RfMgBr followed by construction.

We discerned three kinds of refunctionalization unit reactions which can usefully couple with a construction in a nonstop, or one-step, procedure: prior reduction to carbanion nucleophiles, elimination following construction, or various tautomerizations before or after construction. The overall net structural changes for these composite constructions were then added to our list of construction half-reactions and, after some further subdivision of common types into chemically recognizable subheadings, we had expanded the list from the six of Figure 4 to 24 half-reactions: 16 nucleophiles and 8 electrophiles which combine to afford 100 full construction reactions. (Note that the total should be $16 \times 8 = 128$, but three half-reactions are limited to forming double bonds across the constructed bond.) These 24 half-reactions are sampled in Figure 6, taken from the "help screen" of SYNGEN. They are briefly labeled with two-character descriptors. The first character is a letter for nucleophiles or a number for electrophiles, which generally depicting the minimum required substrate functionality $(z + p)$ on the α -carbon. The second character generally depict the span, or strand length of changing carbons = 1-3.

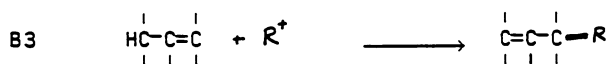
This expansion of possible construction half-reactions to incorporate composite cases now afforded all expected constructions, but the number of chemically non-viable reactions which were also produced remained large. To prune these down to



Enolate (CO-stabilized carbanion)



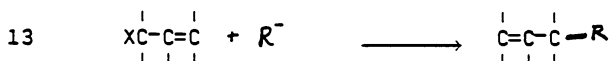
Hetero-stabilized carbanion



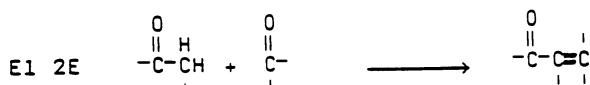
Allylic Pi-Nucleophile



Conjugate addition



Allylic alkylation



E1 A1 or B1 + elimination of H

2E Carbonyl addition/elimination

Figure 6. Samples of SYNGEN Half-Reactions

more reasonable chemistry required an overlay of mechanism testing for chemical viability. It is apparent that any such tests may eliminate possibly interesting new chemistry, but without them the output is excessive. We proposed to solve this problem by encoding the tests as modules attached to each half-reaction generator so that they may be easily altered independently, allowing the nature and proportion of "non-viables" to be variable.

The mechanism tests are of two kinds: "require" and "reject"; the former tests for the presence of required activation and regioselectivity on the atoms near the construction bond; the latter searches for the presence of incompatible functional groups and interfering side reactions. To do this we can make quick numerical checks of the values of *h*, *r*, *p*, or *z* on the proximal atoms, but we quickly recognized that the merging of all electronegative heteroatoms as "z" was too severe a generalization for mechanistic tests. Hence we defined a subset of "z" to indicate the *mechanistic function* of the heteroatom as electron-withdrawing, electron-donating or leaving group. A digital checklist string of single bits for the relevant *h*, *r*, *p*, *z*, and *z-function* values for each atom on (and adjacent to) a reactive a,b,c-strand is established first for strands at each end of a constructed bond, defined by the bondset. Then, for each half-reaction to be generated, two test lists (bit strings) are applied, each by a single AND operation over all the atoms at once, one list for "require" followed by one for "reject". A zero result for the first AND operation disallows generation of the reaction, as does a non-zero result for the second.

These simple mechanism tests are very fast (just two AND operations) and also allow an important expansion of our primary definition of the skeleton. Up to now the "skeleton" of the molecules (target, intermediates, starting materials) was considered to be only a carbon frame. The mechanism tests allow for incorporation of N, O, S atoms into the skeletal frame since they may be regarded simply as special carbons, using the same construction half-reactions but applying different mechanistic tests for activation and rejection. Hence the checklists and test lists above are now expanded to include the nature of the skeletal atoms N, O, S as well as *h*, *r*, *p*, *z*, and *z-function* for all skeletal atoms on and next to the reactive strand. Target skeletons may now be seen as including these heteroatoms and the catalog is defined the same way.

With all the above operations incorporated, the program yields results which are in general chemically realistic. In practice the program has now been adapted from previous DEC 11/23 and microVAX I versions to a form (~50,000 lines of FORTRAN) which runs on a microVAX 3500 or full VAX computer. The target molecule is drawn onto the screen in a rapid and facile drawing program, using a mouse- or thumbwheels- directed cursor, in a fluid, sequential-bonds input and using the keyboard for letter-string additions (-OR, -COOH, etc.) where desired. The structures may be drawn in

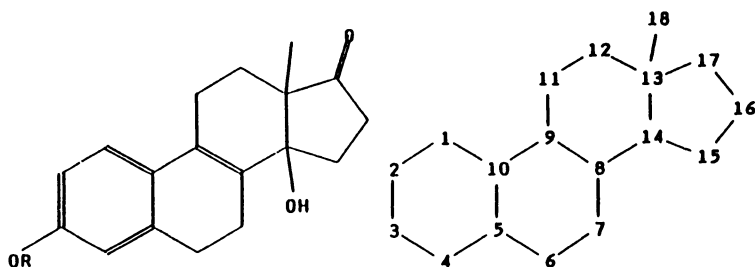
any crude form which maintains atom connectivity since normalization to size and correct bond lengths and angles takes place automatically on processing.

Once the target structure is drawn, the program proceeds without operator input, as described above, first to identify ordered convergent bondsets from the skeleton and the available starting skeletons catalog, then to proceed through sequential construction reaction generation retrosynthetically back through each bondset (through two levels of cuts) from target functionality to the functionality required for each of the starting materials. These are then sought in the catalog of fully functionalized starting materials, accepting only routes generating real compounds. The final set of results, which requires 2-3 minutes computer time for the illustrated steroid example, is then summarized as in Figure 7, showing the target as drawn and its atom numbering. The summary below it shows for each of the two levels of cuts the numbers of successful bondsets, starting materials, intermediates and reactions. In the example of Figure 7 four first-level cuts produced seven starting materials directly and 26 intermediates, which combine in 61 reactions. The 26 intermediates are then all constructed at second level via 34 different bondset combinations utilizing 161 starting materials and 393 reactions.

Operator Selections from the Output

When the program is used on a variety of different targets, the number of reaction sequences obtained is highly variable, depending of course on the structural articulation of the target and the availability of suitable starting materials. Considering the stringency of the selection criteria summarized at the outset, we were in fact surprised at the generally large numbers of sequences commonly produced by SYNGEN. This led to a consideration of flexible modes suitable for examining either large or small outputs, i.e., various ways in which the operator can select subsets of the output which accord with his own practical interests. Successive subsets of the total output can be selected, and the number of consequent reactions displayed, by making manual selections (delete or retain) from the displayed entries in each category: bondsets, starting materials, intermediates or reactions at either cut level (summary below Figure 7). This is shown in Figure 8, with the selection menu, for deletion of two of the four first-level bondsets of the summary; "bondsets" at the top of the menu is now marked D for delete. "VIEW SEL" will subsequently offer only those entries in any category (top of menu) relevant to the retained bondsets. In this way the operator can examine successively smaller chosen subsets of the output.

Figure 9 shows a typical display screen of reaction entries, here at first level for the two bondsets selected in Figure 8. Each entry shows two molecules to be joined at two bondset bonds (annellation) to form the target (Figure 7). These bonds are marked with one dot or two to show the order of reactions. The top line



LEVEL	B'SETS	ST. MAT.	INTERMED.	REACTIONS
1	4	7	26	61
2	34	169	3	393

Figure 7. SYNGEN Screen: Output Summary

TARGET	BONDSETS-D	ST. MAT.	INTERMED	REACTIONS	
B:1	(1)	B:2 (1)	B:3 (1)	B:4 (1)	LEVEL 1
					VIEW ALL
D	C	E	F	H	VIEW BEL
					NXT PAGE
# RXNS: 4	# RXNS: 8	# RXNS: 19	# RXNS: 30		DELETE <
					RETAIN
					(SCREEN)
					PLOT-PLT
					PLOT-PRT
					STOP

Figure 8. SYNGEN Screen: Bondset Selection

shows the four reactions of bondset #1, the rest show the first eight reactions of bondset #4. The two pairs of half-reaction symbols for the two successive constructions appear below each entry.

A second kind of choice is also available for starting material and reaction categories, sampled in Figure 10. In the former category the operator can specify a maximum cost per mole of starting materials, sequences in which one starting material is found at the first-level cut, sequences using the same intermediate skeleton for each half at level one, starting materials requiring some prior refunctionalization (see below), and other choices of this kind which focus on particular features for efficiency. In the category of reactions, the operator may choose either to retain or delete reactions with certain features (Figure 10). Here reactions with these features are flagged during generation for operator selections afterwards. Certain elements of regioselectivity, not serious enough for absolute rejection by the mechanism tests above, can here be examined or deleted by the operator. The incompatibility of offstrand functional groups is similarly flagged for examination, and in the case of annulations (two constructions creating a ring) whether the conditions necessary for each construction are incompatible with both proceeding in one step.

Finally, the computer applies the half-reaction generators quite mechanically and will often produce several variants for one construction which to the chemist appear conceptually equivalent, such as alkylation by a b-halo- or an a,b-unsaturated carbonyl electrophile. Several such kinds of *equivalent reactions* may be removed from the total which the operator must finally look over. This allows him to see more clearly the different synthetic "ideas" produced, without the distraction of excessive minor variations. Examples of these minor variations are clearly apparent in the reactions display of Figure 9, e.g., 4-40 and 4-42, or 4-40 and 4-44.

These kinds of choices, sampled in Figure 10, allow for more facile scanning of the total output and more are being developed as their need appears in looking at a number of targets. The SYNGEN output-viewing menu currently provides about twenty such choices, which can be called singly or in combinations, to delete or to retain. With each one the *number* of resultant reaction entries is shown before the actual entries themselves are displayed.

Searching for Literature Precedents.

When the operator is faced with dozens of reactions, e.g., Figure 9, he will ultimately want to know more about the practical details and will need literature precedents. The reactions were generated from abstract descriptions of net structural change, and not from a database library of reactions, and so no literature background exists in SYNGEN. However, such databases do exist elsewhere and we have set out to provide the operator with a routine allowing him to access these databases to discover the nearest relevant precedents for any specified reaction generated by SYNGEN.

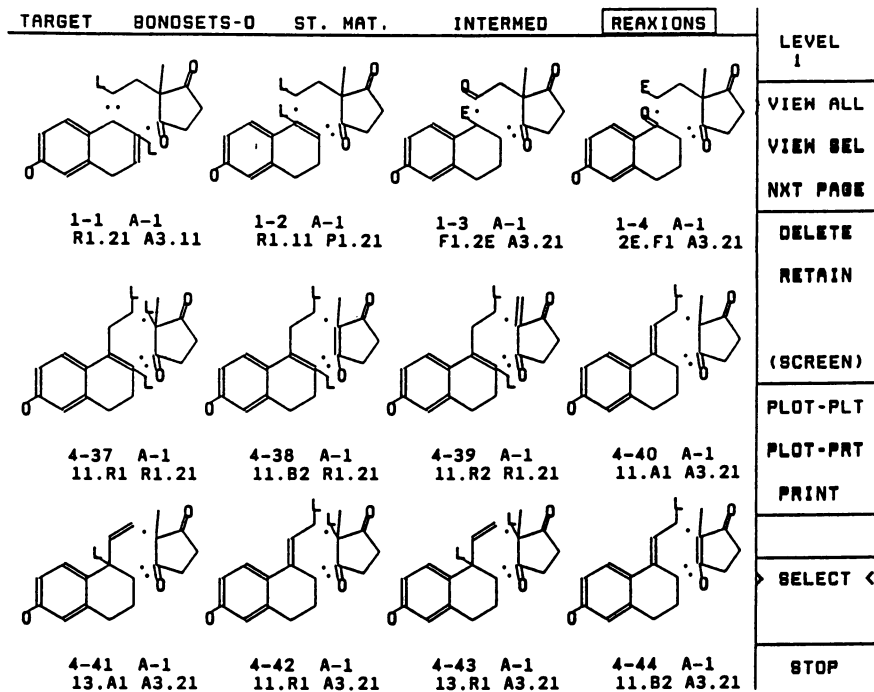


Figure 9. SYNGEN Screen: Sample Page of Reactions

1. Prior selection from BONDSETS, .S.M., INTERMEDIATES, REACTIONS
2. Further selection from ~20 categories flagged when generated

Examples:

<u>STARTING MATERIALS *</u>	<u>REACTIONS</u>
COST	UNCERTAIN REGIOCONTROL
FOUND AT FIRST CUT	BALDWIN'S RULES
REFUNC. REQUIRED	COMPATIBILITY: 2 REACTIONS
EQUIVALENT HALVES	COMPATIBILITY: OFFSTRAND FG
	EQUIVALENT REACTIONS

* Starting materials catalog ~6000 compounds

Figure 10. Examples of Operator Selection Choices

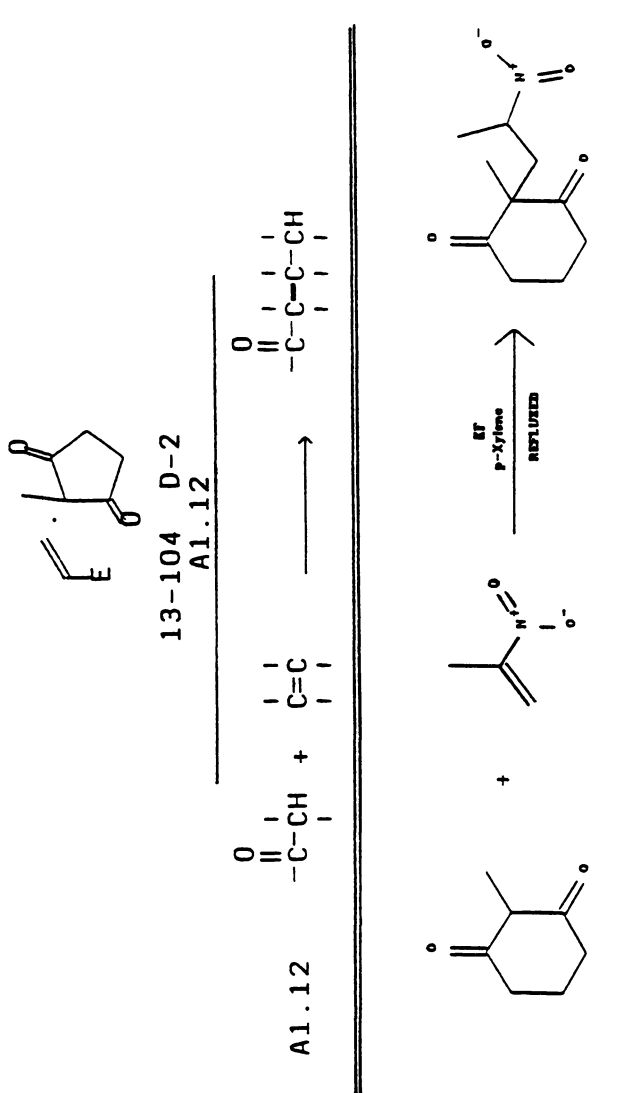
In order to accomplish this, a program was written which creates a subset from the REACCS⁷ or SYNLIB⁸ databases containing only construction reactions. Thus the database is searched first for the formation of a C-C bond and then the reactive strands (a, b, c) out from each end of the bond are identified, the functionality translated to zp-lists, and the net structural change identified as one of the 24 half-reactions on each side. This subset of constructions can then be searched from SYNGEN on command for a particular reaction, to find matches in the reaction library. The full entries from REACCS or SYNLIB can then be displayed with their attendant descriptions of laboratory detail and literature reference. When the number of matches is too large for convenience, several modes of prioritizing are being instituted, such as matches of other functionality, cyclization or not, substructure matching, year and yield in the reference, etc. An example of such a match is shown in Figure 11.

In our initial examination of the current literature file from REACCS there are 29,700 reactions (in 23,100 entries) and 16,500 are constructions, the rest being refunctionalizations (or a few more complex multiple constructions). To date we have found 5600 construction matches, i.e., about 34%. The rest are written in ways that require more translation and as this proceeds, the proportion matched will rise, but there are still many which are too specific, or involve hidden refunctionalization steps, to be useful general precedents (also a nontrivial number are incorrectly entered). Even so, the literature base available to the SYNGEN user is already substantial, and our search procedure makes this readily available.

The SYNLIB database gives a lower proportion of matches. Of the 46,000 reactions in the SYNLIB database we found 23,000 construction reactions. However, since the database contains no effective atom mapping from starting material to product, and since many starting materials are not encoded as substrates (appearing only as text above the conversion arrows) we were able to classify only 3900 (17%). Nevertheless, a considerable library of precedent is also available here. In the future we hope to expand our reference interface to allow access to other REACCS databases (although the Theilheimer reaction set has very few construction reactions) and to ORAC⁹. It was recently reported¹⁰ that there is very little overlap among the references in REACCS, SYNLIB, and ORAC. It may be noted that this overlay of our descriptive system onto these reaction databases may be expanded to catalog, sort and search quickly reactions of any kind other than constructions as well¹¹.

Refunctionalization Reactions: The FORWARD Program

One of the strictest constraints placed on the SYNGEN protocol is that requiring an "ideal" synthesis, i.e., a direct sequence of constructions only with no refunctionalization reactions to "repair" functional groups. Certainly in real life such syntheses are very rare, though for SYNGEN they constitute a desirable goal since they should be the shortest sequences possible. An overview



E. V. Dehmlov, S. S. Dehmlov, See comments
T. Yanami, N. Kato, A. Yoshikoechi, J. Chem. Soc. Chem. Commun., p. 726, 1975

E. V. Dehmlov, S. S. Dehmlov, "Phase Transfer Catalysis", Second Ed., Verlag, Deerfield Beach, p. 163, 1983.
 See litref(2) for details.

Figure 11. Sample Match of SYNGEN to REAGCS

of convergent synthesis plans⁶ is shown in Figure 12, the circles representing compounds (solid = starting materials; open = intermediates; T = target). In the basic version of SYNGEN above the only routes allowed are those with two levels of consecutive constructions from four real starting materials directly to target, i.e., B→C. Refunctionalizations to repair the functionality between constructions are common in actual syntheses but constitute a longer sequence, hence disallowed in SYNGEN. However, the generalizations of the $\alpha\pi$ -list descriptors in SYNGEN will sometimes hide some refunctionalizations which would become necessary when reduced to practice, e.g., protecting groups added (and removed) or alteration of functional groups for more practical reactivity.

In order to relieve this constraint, we considered expanding SYNGEN to incorporate refunctionalizations (ΔFG) at either end of the sequence (Figure 12, A→B and C→D). This will create longer sequences but might offer more realistic chemistry or locate promising routes not generated so far by SYNGEN. Thus the functionality generated by the program for a certain starting material might be somewhat different from any actually available for that skeleton in the starting material catalog. We can calculate for each actual starting material of a desired skeleton the number of unit reactions (N_R ; Figure 3) separating it from the generated compound, essentially the number of steps required to convert it. If the skeleton is large enough to warrant the extra steps we can accept the route, in effect specifying refunctionalization of an available starting material before its use in the generated construction sequence (we currently allow one step, $N_R = 1$, for $SM \geq C_5$ and two steps $\geq C_8$). The output will then flag these routes for separate examination (Figure 10), and the display will show both the real starting material and its altered functional groups required for the construction sequence. In Figure 12 this expansion affords routes described by A→B→C, and is incorporated in the present SYNGEN program.

Defining refunctionalizations at the end of the construction sequence is harder. Here, the constructions build the target skeleton but with incorrect functional groups (T' in Figure 12), requiring some refunctionalization to arrive at the actual target, T. The problem is clearly seen in the steroid chosen as target in Figure 7. This is actually an intermediate in the Torgov-Smith synthesis of estrone¹². It has the skeleton of estrone but extra functional groups which must be removed in the last steps of the synthesis; this intermediate is actually made by an "ideal" synthesis, found by SYNGEN. When estrone itself is entered as target this route cannot be found since it involves a final refunctionalization, and those routes which are found are few and not so practical.

Here we propose to create the bondsets and starting skeletons as before and then, in a forward direction, to combine pairwise all the catalog starting materials of these skeletons. This will use the same half-reactions in the forward direction, creating the intermediates and ultimately the target skeleton bearing that functionality (T') which is a natural result of the construction

requirements from the four starting materials of each combination. The intermediates, T', must then be refunctionalized to target, T. This expansion affords routes described by B→C→D in Figure 11. The two procedures may be summarized as (1) for SYNGEN and (2) for the FORWARD program, where C = construction sequence and Δ = refunctionalizations.

- | | | | |
|----|---------------------------|--------------|--------------------------------|
| 1. | T ;Skeleton | SM skeletons | T ;Skeleton
↓(catalog) |
| 2. | T FG | SM FG
C | SM Δ
T' FG ← all SM FG
C |
| | (1) <u>SYNGEN Program</u> | | (2) <u>FORWARD Program</u> |

The central problem here is one of explosive combinatorics. Not only must all starting materials be pairwise combined but each symmetrical permutation of each starting molecule must be separately utilized! However, the reaction distance calculation provides a potent tool for deleting unrealistic combinations. Thus we can calculate the distance N_R for the carbons of each starting material permutation to their state in the target; some are bound to require far too much functional repair. A certain amount of N_R is required for each construction and the excess allowed can be strictly limited. Also, as first level intermediates are formed, their distances (N_R) to target can be computed and those that diverge (N_R increases from starting material) can be eliminated. Finally, only those routes that generate functionalized targets, T', which are judy one or two steps from the true target, T, would be accepted.

At present the development of this program shows that the number of routes so generated without restrictions is enormous, as expected, but that the pruning possible with the reaction distance calculation is also very great and actually brings the output down to a manageable level. Taking the C_{18} steroid of Figure 7 as an example again, the unrestricted forward generation produced millions of routes. With the above restrictions incorporated, to generate only estrone-skeleton targets (T') one step or less from the Figure 7 target (T), the program produced 30 such products (and the target itself), and the attendant sets of synthetic pathways to each. Allowing two steps of refunctionalization afforded 386 different subtargets (T')! Examples of the subtarget structures produced are shown in Figure 13 with the functional groups indicated by z-values. The FORWARD program dramatically shows how many synthetic routes can be generated; there are nearly 11,000 routes just from the first-level intermediates to these 417 targets. We are currently exploring ways of parsing this output (as in SYNGEN), but at least it is within reasonable computing limits, despite the combinatorics, as a result of applying the restrictions of reaction distance.

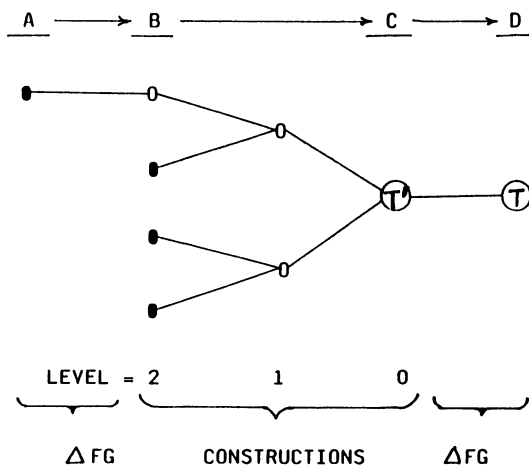


Figure 12. Generalized Plans: SYNGEN and FORWARD Programs

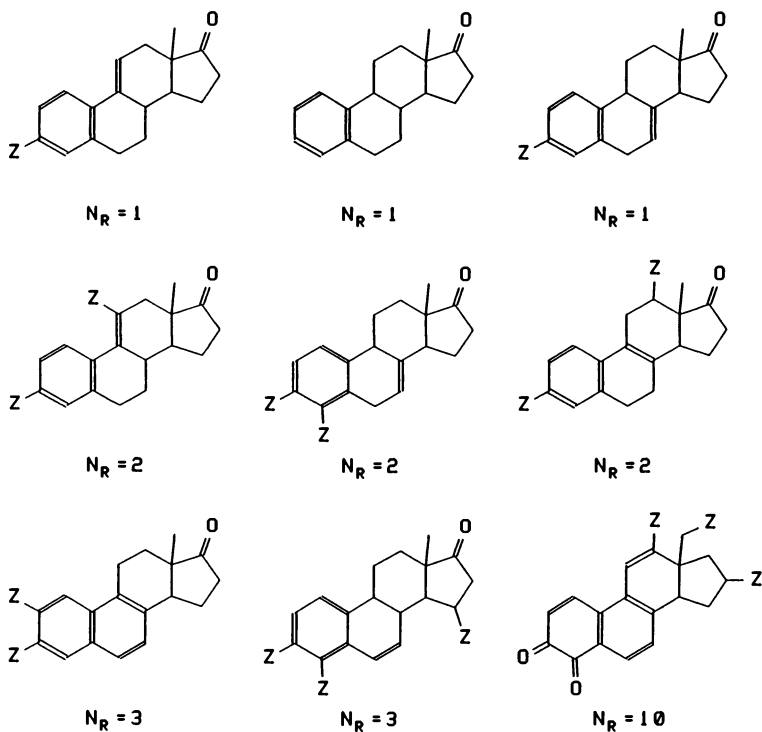


Figure 13. Sample Output from FORWARD Program

Conclusion

All efforts to produce a subset of synthetic routes to a target from the synthesis tree must be judged on how the subset is chosen: both on how good are the routes selected and on whether it rejects some that might be good. The whole tree itself is of course too big to reproduce. The subset used in SYNGEN will produce all convergent routes from four starting materials found in a catalog. At present these are then heavily pruned by applying mechanism tests, and then the generated output is subsequently pruned again by the operator. The mechanism tests serve primarily to suppress a large output volume of non-viable reactions, but they may also delete some with potentially interesting new chemistry. We plan to examine some flexible modes of reducing the severity of mechanism tests while strengthening the selection tools for the operator to deal with the increased output, so that more new chemistry may be discovered.

Acknowledgement

We are grateful for support of this project to the National Science Foundation and to the Eastman Kodak Company.

Literature Cited

1. Rationale and overall description: Hendrickson, J.B. Accts Chem. Res., 1986, 29, 274; also ref. 2.
2. Hendrickson, J.B.; Toczko, A.G. Pure Appl. Chem., 1988, 60, 1563; Hendrickson, J.B.; Grier, D.L.; Toczko, A.G. J. Am. Chem. Soc., 1985, 107, 5228.
3. Hendrickson, J.B. J. Am. Chem. Soc., 1971, 93, 6847; J. Chem. Ed., 1978, 55, 216.
4. Hendrickson, J.B.; Toczko, A.G. J. Chem. Inf. & Comp. Sci., 1983, 23, 171; Hendrickson, J.B.; Grier, D.L.; Toczko, A.G. Ibid., 1984, 24, 195.
5. Hendrickson, J.B.; Braun-Keller, E. J. Comp. Chem., 1980, 1, 323.
6. Hendrickson, J.B. J. Am. Chem. Soc., 1977, 99, 5439.
7. Reaction database from Molecular Design, Ltd., San Leandro, CA.
8. Reaction database from Distributed Chemical Graphics, Inc., Meadowbrook, PA.
9. Reaction database from ORAC Limited, Leeds, England.
10. Borkent, J.H.; Oukes, F.; and Noordik, J.H. J. Chem. Inf. & Comp. Sci., 1988, 28, 28.
11. Hendrickson, J.B. Ibid., 1979, 3, 129.
12. Ananchenko, S.N.; Torgov, I.V. Tetrahedron Lett., 1963, 1553; Smith, H. et al. Experientia, 1963, 19, 394; J. Chem. Soc., 1963, 5072.

RECEIVED June 9, 1989

Chapter 7

Drug Design Using a Protein Pseudoreceptor

F. A. Momany, R. Pitha, V. J. Klimkowski, and C. M. Venkatachalam

Polygen Corporation, 200 Fifth Avenue, Waltham, MA 02254

The design of new pharmaceuticals depends in many cases on a detailed knowledge of the receptor pocket into which the drug binds. Structural data on receptors is seldom available, so a program was developed to computationally generate an artificial protein receptor, denoted as a "pseudo-receptor". The method uses decision algorithms to pick the residues to mutate such that a close fit around the substrate of interest is achieved. Energy minimization and molecular dynamics calculations are used to optimize the fit of the pseudo-receptor about the substrate. An example problem is described using a hepta-peptide from the carboxyl terminal of CCK, as well as a benzodiazepine molecule which acts on the same receptor, as the substrates of interest.

The method of 'receptor fit' is an important concept for the working drug design scientist. This method recognizes that the fit between drug and receptor is one of steric recognition, electrostatic complementarity, and matching of the nonpolar regions of the drug to the receptor. However, few useful methods of drug design exist which report quantitative protein-ligand interaction energies, much less give a complete description of the three dimensional structures and flexibility of the receptor or ligand necessary for binding. Several qualitative and semiquantitative procedures such as those denoted QSAR(Quantitative Structure Activity Relationships), are used in drug design, but it is difficult to see how these linear methods, which correlate the binding energy of the

0097-6156/89/0408-0082\$06.00/0

© 1989 American Chemical Society

ligand with different physicochemical properties for different parts of the ligand, will work well for complex flexible biological ligands. To date, the only truly accurate description of binding sites are for those systems in which the X-ray diffraction structure of the protein/enzyme has been determined; there the method of binding of the substrate can be studied in great detail. Unfortunately, the explicit structure of receptors is in almost all cases unknown.

In this paper we develop a new method for finding the three-dimensional space that surrounds a substrate/ligand. This space, which is the chemical equivalent of the receptor, is represented as a protein structure, referred herein as a "pseudo-receptor". A variety of computational tools are used to create the pseudo-receptor. A molecular mechanics and dynamics program, CHARMM(1), is used to calculate the energy and conformational features of the pseudo-receptor. The program QUANTA(1) is used to define the preliminary protein sequence, secondary structure, graphically examine molecular interactions, interface with CHARMM, and model amino-acid mutations in the protein sequence.

In the study reported here, the protein pseudo-receptor is first constructed as a polyalanine chain, and folded to form a pocket into which the substrate is fitted. The polyalanine chain is next mutated residue by residue to redesign the side-chain interactions with the substrate. The electrostatic potential map on the substrate is used to show where the complementary charges on the protein must occur, and probe maps using hydrophobic probes show the nonpolar regions which must be included by adding hydrophobic residues in contact with specific regions of the substrate. The methodology is similar to that used to design artificial enzymes using synthetic precursors as a framework for adding functional groups. In the pseudo-receptor model, the framework is a polypeptide and the mutations add the binding stereochemistry and functionality. Future studies will show how to automate the selection of residues for ideal fits about the substrate. Here we will show how choices can be made for residue selections using computer graphics while employing mutations and energy searching to optimize interatomic interactions.

Materials and Methods

The initial conformation of a 100 residue polypeptide of polyalanine was obtained by folding the first 30 residues and last 30 residues as helices; the middle 40 residue were folded into a beta-sheet such that both helices were located on the same side of the sheet structure formed as shown in Figure 1. The preliminary structure was passed through limited energy minimization to remove serious contacts and appropriate alanine residues were mutated into glycine at the structural points where appropriate bends occurred and where glycine could remove some strain energy.

The substrates studied include the carboxy terminal seven residues of the gastrointestinal peptide hormone cholecystokinin, denoted CCK7, with sequence;

NH₂-Tyr-Met-Gly-Trp-Met-Asp-Phe-COOH,

and a benzodiazepine derivative; R-3-(3-indolyl-methyl)-5-phenyl-benzodiazepine(2), (see figure 2), which has been found to block peripheral CCK receptors. The low energy conformations of CCK7 were studied using the Boltzman Jump conformational search method described elsewhere (3) using ECEPP83 parameters (4), starting from low energy structures published previously (5). New very low energy structures of CCK7 were found, which when compared graphically to the low energy structure of the benzodiazepine derivative using least squares fitting procedures, showed considerable structural homology as shown in Figure 2. The CCK7 structure was next placed in a bath of 81 TIPS3 water molecules and again energy was minimized using the CHARMM parameters (version 21) to ascertain its stability in water. The resulting conformation of CCK7 changed very little from the calculated conformation shown in Figure 1.

The CCK7 structure was placed into the polyalanine molecule pocket such that the polar residues pointed outward toward the solvent, and the nonpolar residues pointed toward the interior, as shown in Figure 3. The mutation of the polyalanine was carried out using graphics to find those residues of alanine which pointed the side-chain methyl group toward some part of the substrate molecule. For example, if a side-chain pointed toward a phenyl group on the substrate, then that alanine was mutated using the protein contact rules (6,7)

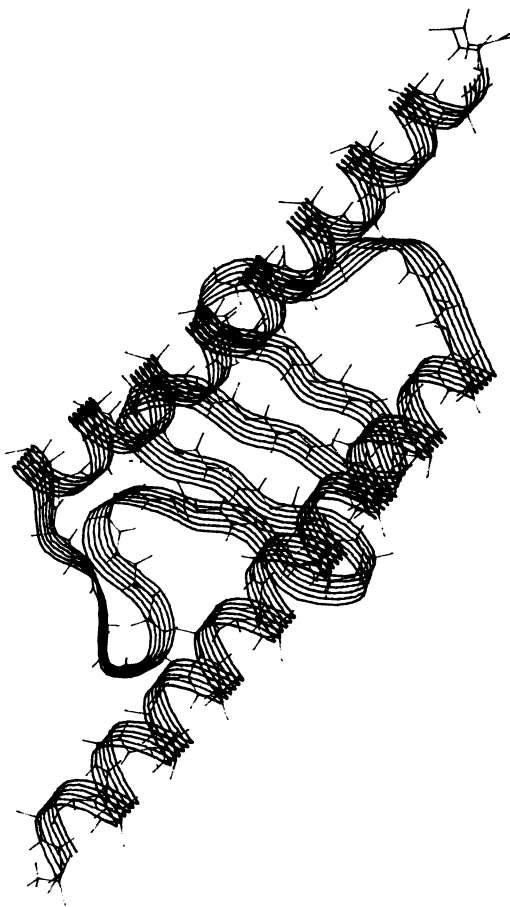


Figure 1. Ribbon structure of 100 residue polyalanine folded into a helix-beta-sheet-helix conformation to form an initial pocket for CCK7 binding.

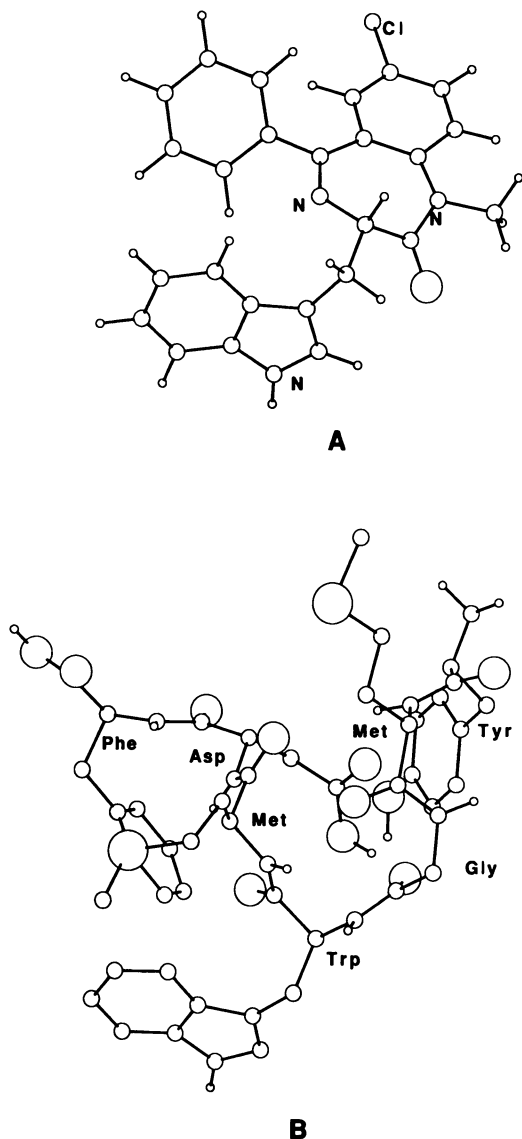


Figure 2. Stick drawing of low energy structure of (A) geometry optimized 3-Chloro-(3'-Indolinylmethyl)-5-Phenyl-1,4-benzodiazepine and (B) the low energy CCK7 conformation to show configurational similarities. Some hydrogens have been omitted from CCK7 for ease of viewing.

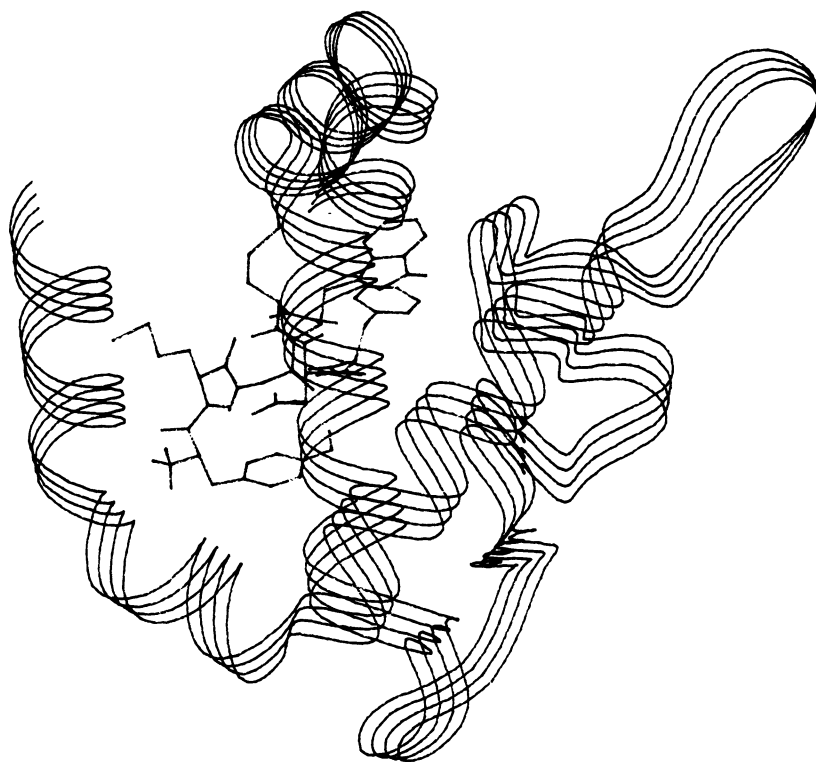


Figure 3. Structure of polyalanine after insertion of CCK7 into pocket, after bending helices to enclose the pocket, and after preliminary energy minimization.

to test for optimum interaction. In other words, if the contact rules suggested that a phenyl group should occur around that particular position of the substrate, then the mutation was to place a phenylalanine at that position on the polyalanine chain. Similarly, if a hydrogen bonding donor was required to interact with an acceptor from the substrate, then that type of mutation was made. At each point of mutation, several different side-chain conformations were examined and the best interacting configuration was retained. If a suitable side-chain conformation could not be found with a reasonable binding energy and without high energy, then that mutation was discarded and a secondary choice in residue was examined. The final mutated and energy minimized pseudo-receptor with the CCK7 located inside the pocket is shown in Figure 4. Taking the resulting complex, the surface of the substrate exposed to the solvent region is covered with water and energy minimized. Further, after removal of the CCK7 from the pseudo-receptor, the space previously occupied in the now empty protein pocket is filled with water molecules and energy minimized to obtain the energy required to remove the appropriate amount of water from the solvated receptor. Entropic contributions to the net free energy are calculated using the Einstein relationships.

Results

Packing around the CCK7 Trp residue includes a Trp at position 54 in the pseudo-receptor, Phe at position 15, Ile at 47, and Leu at 59. The region about the Phe residue of CCK7 includes Leu 59 and Gln 85. The Asp residue of CCK7 forms a hydrogen bond interaction with the backbone amide, and Phe 15 forms a pocket for the Tyr of CCK7. The packing is very compact with no holes or cavities inside the pseudo-receptor. The mutation effort was very efficient in finding close contacts between all binding residues and the energetics of the full flexible geometry optimization method allowed both substrate and pseudo-receptor to flex to fit compactly and form a tight binding interface.

The enthalpic and entropy contributions to the binding and stability of the pseudo-receptor and substrates will be described in more detail elsewhere. However, it is important to note that one must obtain the net free energy of the binding

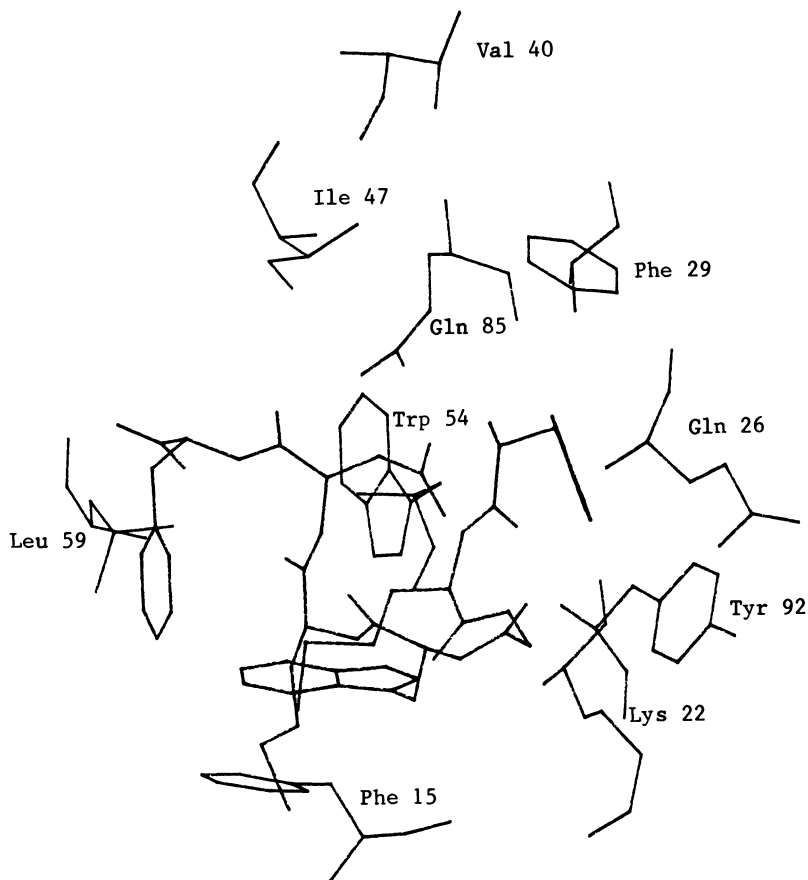


Figure 4. Configuration of sidechains of mutated pseudo-receptor with CCK7. Residues of the pseudo-receptor which have not been mutated have been removed for viewing clarity.

process in order to compare these calculated values to the experimental binding constants. This requires obtaining the internal enthalpy of the substrate, protein complex, and the solvation energy about the substrate and inside the receptor pocket in order to get the energy required to strip off the correct amount of water from both substrate and receptor. Further, one must examine the entropic contributions to the free energy by finding the internal fluctuations and translational and rotational motions of solvent, substrate and receptor when imbedded in solvent prior to and after binding of the substrate. These values are not easy to obtain and small net free energy differences between large enthalpy energies have to be precisely calculated. Several methods for obtaining these energies are being tested, including carrying out all calculations within a box of water and carrying out dynamic simulations at room temperature for extended periods of time to achieve an equilibrium situation for calculation of entropy terms.

The benzodiazepine structure was inserted into the pseudo-receptor pocket upon removal of the CCK7 and a very good fit was achieved without modification to the pseudo-receptor. The fit of the benzodiazepine is also stereospecific, binding only in one position and not fitting in any other configuration. Clearly, the method of optimizing the protein structure about a substrate is capable of finding a binding site which can then be further optimized by adding different types of substrates, either from sets of different agonists or from antagonists which are known to bind similarly, and cycling through the mutation and energetic minimization cycles until all the test data is fitted. We call this the learning cycle for the pseudo-receptor, and one can find optimal interactions for as large a data base as is deemed necessary. Once the final receptor pocket is found one can then use this to quickly screen further analogs of new structural types for binding.

Conclusions

An extension to the pseudo-receptor approach is to utilize the thermodynamic perturbation method (8) to calculate approximate values for the free energy of binding of various substrates to the pseudo-receptor.

A second consequence of the pseudo-receptor approach to drug design is to consider the pseudo-receptor as the drug molecule of interest. This protein could then be synthesized and used to experimentally test for binding to selected substrates and to subsequently be used as the drug itself. This problem would require additional structural modification to stabilize the protein structure in the form found to best bind the model substrate. Complete hydration models of the pseudo-receptor with substrate can be used to test for solvent stability in these studies.

Literature Cited

1. QUANTA and CHARMM are trademarks of Polygen Corporation, Waltham, MA.
2. Evans, B.E., Rittle, K.E., Bock, M.G., DiPardo, R.M., Freidinger, R.M., Whitter, W.L., Gould, N.P., Lundell, G.F., Homnick, C.F., Veber, D.F., Anderson, P.S., Chang, R.S.L., Lotti, V.J., Cerino, D.J., Chen, T.B., King, P.J., Kunkel, K.A., Springer, J.P., and Hirshfield, J., J. Med. Chem., 1987, 30, 1229.
3. Momany, F.A., Klimkowski, V.J., Pitha, R. and Venkatachalam, C.M., "Molecular Design using Supercomputers", Video Presentation, San Diego State University and San Diego Supercomputer Center, April 20, 1988.
4. Chuman, H., Momany, F.A., and Schafer, L., Int. J. Peptide Protein Res. 1984, 24, 233.
5. Pincus, M.R., Carty, R.P., Chen, J., Lubowsky, J., Avitable, M., Shaw, D., Scheraga, H.A., and Murphy, R.B., Proc. Natl. Acad. Sci. USA, 1987, 84, 4821.
6. Roberts, D.C. and Bohacek, R.S., Intl. J. Pept. Prot. Res., 1983, 21, 491.
7. Wako, H. and Scheraga, H. A., J. Protein Chem. 1982, 1, 5.
8. Mezei, M and Beveridge, D.L., Ann. N. Y. Acad. Sci., 1986, 482, 1.

RECEIVED June 9, 1989

Chapter 8

DOCENT

An Expert System To Aid the Molecular Modeler

Carl Trindle

Chemistry Department, University of Virginia, Charlottesville, VA 22901

The central construct of the expert system DOCENT which we are developing is its capacity to represent a macromolecule by "generalized cylinders", and to permit manipulation of the cylinders directly, instead of adjusting the molecular internal coordinates or space-fixed axes. The inverse problem, to recover reasonable values of the underdetermined atomic coordinates from the disposition of the generalized cylinders, is posed. This task is eased by a system which possesses some knowledge of stable chemical structures.

In reply to the question what unresolved difficulties are of the greatest urgency in artificial intelligence research, Aaron Sloman(1) replied: "A major unsolved problem is how to represent shape. I believe that when we know how to represent shapes, spatial structures, and spatial relationships many other areas of AI will benefit since spatial analogies and spatial modes of reasoning are so pervasive."

No discipline relies more thoroughly than chemistry on the power of visualization in three dimensions. Complex objects which stubbornly resist ordinary verbal and mathematical description can none the less be grasped visually. Chemists spend years developing the spatial intuition which informs their reasoning and speculation. Building on this ability, computer graphics now plays a key role in molecular modeling(2) and has changed that venerable practice drastically. Expert systems incorporating sophisticated computer graphics can ease the development of this conceptual power, and magnify it. We intend that DOCENT contribute to this end.

Molecular modeling software uses computer graphics as if the human user were an expert, to whom many subtle judgments must be deferred; the expert infers three dimensional information from the two-dimensional image, selects axes about which the molecule can be rotated, reorients fragments by distorting the molecule along soft internal modes of torsion and vibration, folds macromolecules to alter their topography, and recognizes promising features of the molecular surface which typify hospitable docking sites where molecules may fit together. The human expert can read the implications of displays which represent surfaces by shading or speckling, or color-code the

0097-6156/89/0408-0092\$06.00/0
© 1989 American Chemical Society

electrostatic potential. All of this is easy for the expert, difficult for the novice. It will be a service to both if the manipulation of graphic objects is simplified.

Our central idea is that the complexity of a display should be kept to a minimum. This is far from a new idea, nor is it at all controversial. Yet many graphics displays flaunt their power of resolution, crowding the screen with strokes denoting individual bonds and points denoting individual atoms. The atom and its bonds compose the natural focus of interest for chemists, but in macromolecules the detail is excessive and cannot be perceived. What is worse, the broad characteristics of the molecule cannot be seen, owing to the distraction and clutter of the fine-grained representation. Such displays are too full of detail to make plain the key "image events." These "image events" are features such as lines, contours, regions of homogeneous color or texture, simple shapes, and (parts of) surfaces. While a typical two-dimensional view of a scene might employ an array of 512x512 pixels, the number of significant features would be much smaller. What the novice user of a molecular modeling system needs is not the forest of lines, but assistance in recognizing chemically and geometrically more meaningful clusters.

Communication is eased by judicious simplification, even caricaturization. Caricaturization selects key features, exaggerates them, and conveys a vivid impression efficiently. Considerable progress in caricaturization or cartooning of macromolecules has been made since the hand sketches of Richardson (3) appeared. Software is available which shows the structure of the molecule by backbones or ribbons(4). Our aim is to permit the direct manipulation of the caricature. We propose to develop a display system which represents a macromolecule in a hierarchy, to design a user interface which permits the user to make high-level changes in a structure, and to couple the system to a computational module which reports the energetic changes accompanying the gross geometric changes and refines the atomic-scaled structure so best to accommodate the changes.

To proceed in this direction our system needs five main features:

1. it must incorporate a representation of the atomic-scale structure of the molecular system. Internally this would be a set of Cartesian coordinates or the equivalent set of internal coordinates. On the graphics screen, this would be at minimum a collection of lines representing individual bonds.
2. It must possess low level processes which act on the Cartesian coordinates to diagnose molecular features such as chains and rings.
3. It must incorporate three-dimensional graphic and internal high-level representations which are flexible and powerful enough to capture geometrically complex objects, and sufficient to caricature the molecular structure. These might include CPK spheres or Richardson's coils and ribbons which are already available in graphics displays, but might be more general.
4. It must permit operations directly on the high-level representation of the molecule, which is not simply a vehicle for display.
5. It must permit the recovery of a consistent set of atomic coordinates from a high-level representation, after that structure has been adjusted by the user.

Caricature of the Macromolecule by Model Matching

The highly influential effort by Brooks(5, 6) directed toward solving the AI problem of automatic interpretation of two dimensional images by model-matching is particularly helpful here. Brooks defines the elements of models by "generalized cones" (also known as generalized cylinders), which are primitive elements of volume defined by a planar cross section, a space-curve "spine", and a sweeping rule. A few examples are given in Figure 1. Form A has a circular cross section, a straight-line spine of specified length, and a sweeping rule which keeps the radius of the cross section constant. Form B differs only in that the cross section is a square, while form C has a sweeping rule which permits a uniformly decreasing radius of its circular cross section. Form D shows that the spine can take on a form which permits the representation of a random coil.

Such modeling need not be highly realistic to be useful. AI vision researchers (7, 8, 9) have noted the fact that complex shapes -- skeletons of animals -- can quickly be recognized from the few lines provided by pipe-cleaner models; the attenuation of these generalized cylinders is a useful caricature. This type of simplification is very familiar to chemists, since ball-and-stick molecular figures are constructed from generalized cylinders. The helix, other ribbons, and CPK space-filling models are other simple examples. Brooks' major contribution to our problem is his data structure's capacity to represent classes of objects. A range of generalized cylinders, from which many members of a class of objects can be constructed, is represented in the range of parameters associated with the cross sections and the sweeping rules in Brooks' program. Chemists will find this generality congenial, since we are accustomed to a language of forms, including such terms as alpha-helix, coil, beta-turn etc. which embrace classes of structures rather than any highly specific structure. These terms suggest generalized cylinders which lie outside the set of forms which Brooks defines.

Brooks achieves a representation of classes of objects by an unusual data structure; we will call this a "Brooks Data Structure" -- BDS. We can adapt the BDS to a LISP-coded specification of a macromolecule as follows: the macromolecule is to be described by a TREE, the ROOT of which is the coarsest characterization of the molecule. For a macrocycle, the ROOT would be the great ring; for a branched chain it would be the longest backbone. This ROOT would be a list, the basic data structure of LISP, with each element in the list a data set defining a generalized cylinder.

Sidechains or substituents would be specified by nodes lower in the TREE. The arc connecting the substituent node to the ROOT would possess PROPERTIES which would define the orientation of the side-chain relative to the segment of the ROOT to which it was attached.

Use of Generalized Cylinders in the Interpretation of a Scene

Figure 2 shows a digitized image of a scene, produced by scanning a photograph of a very realistic painting of a staircase. We find the scene very easy to interpret, even though much of the color and in-

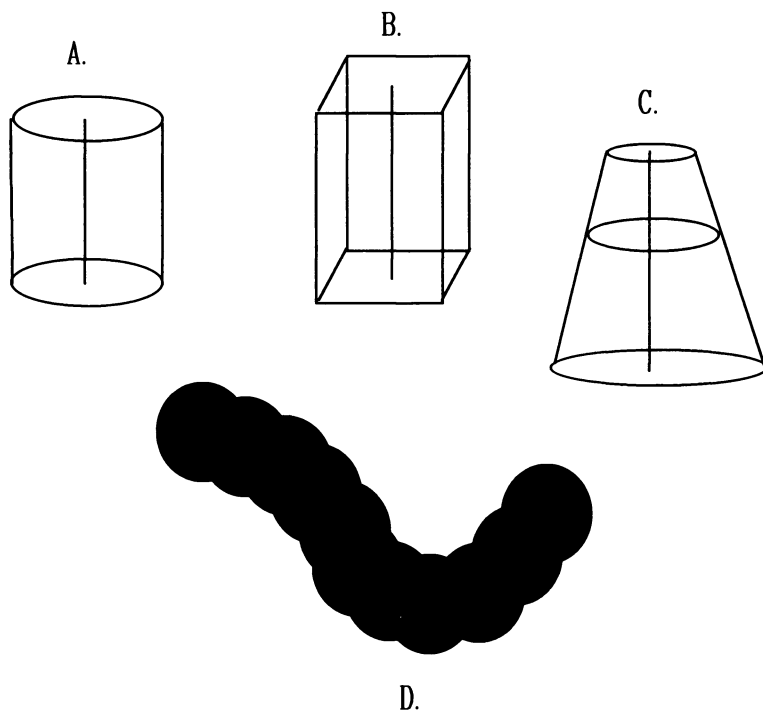


Figure 1. Examples of generalized cylinders, which are defined by a space curve spine, a cross sectional form, and a sweeping rule.

tensity information has been discarded. Our interpretation is assisted by the fact that we know something about the shape and connectivity of a staircase and its parts; the newel posts, bannister, risers and landing. All of these lend themselves to representation by generalized cylinders. The simplest subproblem is defined by the set of newel posts. They are essentially identical right cylinders of circular cross-section with a constant radius and common length. They



Figure 2. Views of "Stairway 1910, Richmond", by Lowell Nesbitt

are arranged in a highly constrained way; their bottom ends placed at Cartesian X_k, Y_k, Z_k , progress up the staircase in equal Z -steps: $Z_{k+1} = Z_k + H$, where H is the height of a riser. Their top ends define the spine of the generalized cylinder representing the bannister.

The steps are also constrained; they are planar, arranged to fit the bottom cuts of the newel posts. Their exact shape depends on the shape of the wall, which we presume is smooth and vertical. It need not be flat; any curvature can be represented by a section of a generalized cylinder.

It is necessary to situate every cylinder in space. In the BDS cylinders are assigned to nodes, and two cylinders' relative disposition is stored in an arc connecting their respective nodes. (In LIISP both arcs and nodes are lists.) The arcs can establish a connection between a specified location on one cylinder (such as newel post j , end i) with another (such as bannister B , socket k .) Information on the nature of the connection, such as the relative orientation of the spines is included in the arc. In this case, the posts would also be constrained to be vertical in the laboratory (external) coordinates.

This very preliminary discussion makes plain that high-level (if rather crude) model matching guides our perception and quick interpretation of a familiar scene. A few simple models assembled in space give an overall impression of the arrangement of objects in the scene. It is easy to recognize and express powerful constraints on the location of the generalized cylinders representing the models. Many staircases can be built with the generalized cylinders at hand, and subject to the constraints imposed by demands of connectivity and smoothness, among others. But the space of alternatives is severely limited by our choice of the newel post as the unit manipulated, rather than individual fibers of wood.

Use of an ACRONYM-Style Modeler in Molecular Displays

An example of the simplification which can be gained as we ascend in scale is already well known in the physical chemistry of macromolecules as the "Kuhn statistical segment" picture of chain dynamics. A full description of the arrangements of molecular chains in space is relatively complicated if one works at the scale of single CC bonds, each of which is severely constrained in orientation by its immediate neighbors. However the correlations on the orientations of successive chain segments diminishes in importance as one takes as the segments the vectors between increasingly remote atoms along the molecular chain. A step sufficiently long that the correlations vanish is the most convenient Kuhn segment. The analysis of the effectively random flight of a chain of Kuhn segments(10) is then much simplified, at the cost of some detailed information on the position of some particular atoms. In our display we would represent any random coil by a few Kuhn segments, and our BDS would be very simple. However in order that transformations be made between the generalized-cylinder representation and a geometrically consistent array of

atoms, it is required that some of the atomic information be retained in the BDS. This would include at minimum the subchain length, *i. e.* the number of atoms in each Kuhn segment, and the identity of each atom in the chain so that model bond lengths and valence angles can be assigned (see next section).

Even in the atomic-scale graphical representation of macromolecules it is easy to recognize helical segments. In our display we represent any helical portion of a molecule by a single rod. Properties of the rod would include the number of turns and the number and nature of atoms or residues involved in the local helical structure. Other types of secondary structure lend themselves to representation by generalized cylinders, but have not been incorporated into our code, which is in an early stage of development.

Deducing Atomic-Scale Geometry from Generalized Cylinders

The generalized cylinder representation of a macromolecule is a much sparser description of the system than is the full set of Cartesian or internal coordinates which define the position of every atom. The generalized cylinder representation leaves the molecular structure seriously underdetermined. It is eventually necessary to recover an estimate of all the atomic coordinates for more detailed study. Here is where DOCENT takes on some of the features of an expert system, which must make plausible judgments without complete information. One aspect of this problem has already been addressed (11); given the atom labels and a molecular topology it is often possible to model the detailed molecular geometry from a limited number of bond lengths, valence angles, and torsion angles which persist in common molecular fragments. Generally these transferrable parameters define only a small subset of all interatomic distances. However the distance geometry methods developed by Crippen (12) permit the construction of a set of Cartesian coordinates in three dimensions, beginning with such a small data set and incorporating known constraints. But while the Cartesian coordinates developed by distance geometry are legal, representing a structure actually imbedded in three dimensions which obeys short-range constraints, they are not energetically optimum. Of course the plausible trial structure may be refined by molecular mechanics (13), at a cost.

The generalized cylinder representation has the decided advantage that it constrains a structure more thoroughly than the local bond properties can, by requiring specific distances between atoms remote from one another along a molecular chain. The chain is composed of $K \times N$ atoms, where K is the number of Kuhn segments and N is the number of atoms in each segment. B is the bond distance between adjacent atoms in the chain and L is the length of a Kuhn segment. But $L \neq B \times N$ generally, owing to folding. For the subset of atoms within a particular Kuhn segment, L defines the distance between the 1st and N th atom. There are many possible N -step paths of length L and distance geometry selects but one (at a time). Nonetheless each new constraint has an important impact, reducing the search space. Further constraints are found in the arcs of the BDS. The connection angle between the spines of attached cylinders dictates distances in the neighborhood of the connection. If a chain is made up of K Kuhn segments there can be $2K + 1$ such constraints.

Manipulation of a Generalized-Cylinder Representation

The representation of a staircase or a molecular chain in a BDS is very general, and embraces a family of recognizably similar systems. Operations on the BDS can effect transformations within this extensive set. Such operations fall into several distinct classes: rigid-body motions, cylinder deformations, non-conservative transformations, and creation-annihilation operations.

Rigid body motions preserve all the connections, but reorient individual cylinders. (Due to these connections it is generally not possible to move only a single cylinder; the entire structure can be altered.) Under such transformations on the scene discussed above, the staircase could perhaps be changed slightly in steepness, but if the parts are rigid the construction is almost completely defined. In contrast connection angles in a Kuhn-segment chain are flexible, so the Kuhn cylinders could be reoriented easily. The single cylinder representing a helical section could not be bent, but it could be reoriented; the connection between the helix-cylinder and an adjoining Kuhn-cylinder would be flexible.

Cylinder deformations alter the generalized-cylinder parameters, including spine length, the radius of the cross section, and also the specifiers of the sweeping rule. This would have major effects on the staircase, permitting variable steepness, landing width and shape, riser height, bannister shape and length, etc. For a chain this type of transformation would permit the unwinding or growth of helical portions and the extension or shrinking of the effective sphere containing the end-to-end vector of a coil.

Non-conservative transformations establish or delete connections. This would merely reduce the staircase to a shambles, but could effect chemical reactions such as closing or opening a ring.

Creation-annihilation transformations introduce or remove cylinders entirely.

In the user interface, provision must be made for the user to identify the cylinder or connection on which the operation is to act, and for the user to specify the desired changes. The click-select and menu specification which is characteristic of the Macintosh or of Microsoft Windows (TM) is suited to at least crude specification of these operations. With such direct manipulation of the cylinders the user would find it extremely easy to fold a molecule. For example, rather than painstakingly altering individual torsion angles as is required in software provided with the IRIS graphics station, the user could move an entire sidechain "as an object" in itself.

The Ribbon as a (Set of) Generalized Cylinder(s)

Carson (14) has presented algorithms for generating a ribbon drawing for proteins. Briefly one defines a local vector and "guide points" for each residue. The guidepoints situated along each vector are linked by a cubic spline curve, which assures continuous first and second derivatives of the composite curve. The smooth curves may be represented more simply by line segments, and "solid" models may be constructed by defining an elliptical cross section with major axis along the curve's binormal and minor axis along its normal. The relative magnitude of these axes are chosen to encode the secondary

structure: coils and turns are nearly circular in cross-section, while the ellipses become highly eccentric for the helical sections. It is evident that the "solid" models developed by Carson for purposes of vivid display are sequences of simple generalized cylinders, with elliptical cross sections and sweeping rules locally simple, but capable of encompassing very general chain structures. The smooth appearance of the ribbon diagram is desirable in most displays, but require a large number of local cylinders. We expect that a cruder representation, with a bare minimum of distinct cylinders, will be adequate to our purposes. Our priority is the direct manipulation of cylinders, which is easier if there are only a few.

Alternative Applications of Generalized Cylinders

We have devoted almost all our attention toward the manipulation of single molecules, to help the modeler fold large systems into interesting shapes. But the idea of large-scale representations as the top of a hierarchy of representations of ever-decreasing scale and ever-increasing detail is generally applicable to a variety of related problems and procedures. Brinkley, *et. al.* (15) have applied the "divide and conquer" strategem to the problem of generating and refining a large set of structures consistent with NMR data. They systematicall break the large problem of determining the volume accessible to atoms in a macromolecular system by generating partial solutions to smaller subproblems, and incorporating constraints arising from these partial solutions of subproblems in their efforts to solve other subproblems. In the spirit shared in this work, they reduce the number of effective atoms by "grouping locally constrained sets of atoms" and treat "the entire group as an abstract object before considering detailed [atomic] locations." These objects are modeled as simple solids or (as necessary) collections of simple solids. This is interpretable as a generalized cylinder representation applied to structural refinement.

Generalized cylinders lend themselves to the representation of a related but "inverse" problem. Rather than folding a single macromolecule into a form satisfying some conditions imposed by its own conformational preferences, one may begin with the assumed form of a substrate molecule and assemble a collection of generalized cylinders which fit and enfold the species. This can be considered a variant of the problem of automated assembly, to which Brooks has already applied his ACRONYM system (7). There is no energetic consideration here, apart from the requirement of geometric fit of rigid objects. But if that geometric fit defines the parameters of generalized cylinders, and if we can infer something of the atomic coordinates of chains from those parameters as we suggest above, the possibility exists that we can deduce characteristics of the primary structure which would produce the tertiary structure represented by the set of cylinders which take on the desired shape.

Conclusion

There are considerable advantages, of simplicity and power, to the large-scale and approximate representation of macromolecules by "generalized cylinders". Gross features of the structure are made easier to grasp visually. But more significant, major alterations to the

structure can be made easily by direct manipulation of the cylinders themselves, instead of the usual adjustment of a series of atomic-scale torsion coordinates or space-fixed axes. An appreciation of the energetic effects of gross structural changes requires that the "inverse problem", of recovery of reasonable values of the generally underdetermined atomic coordinates from the disposition of the cylinders, be solved. Distance geometry methods, extended by conditions on the large-scale structure imposed by the arrangement of the generalized cylinders and the persistent features of geometry of molecular fragments, promise reasonable first guesses, which may be refined by molecular mechanics calculations.

Literature Cited

1. Sloman, A. Artificial Intelligence, 1985, 25, 386
2. Kollman, P. in Ann. Rev. Phys. Chem., Strauss, H., Ed.; Annual Reviews Inc.: Palo Alto CA, 1987
3. Richardson, J. S., Adv. Protein Chem., 1981, 34, 167
4. Burrige, J. M.; Todd, S. J. P. J. Molec. Graphics, 1986, 4, 220
5. Brooks, R. A. Artificial Intelligence, 1981, 17, 285
6. Brooks, R. A. Int. Joint Confs Artificial Intelligence, 1981, 7, 619
7. Cohen, P. R.; Feigenbaum, E. A. The Handbook of Artificial Intelligence, Vol III; Kaufman: Los Altos CA, 1982
8. Marr, D. Vision; Freeman: San Francisco CA, 1982
9. Brady, M.; Bobrow, D. G. Artificial Intelligence (Special issue on computer vision) 1981, 17; Lowe, D. G. Artificial Intelligence, 1987, 31, 355
10. Cantor, C. R.; Schimmel, P. R. Biophysical Chemistry; Freeman: San Francisco CA, 1980; p1010ff
11. Trindle, C.; Givan, R. Stud. Phys. Theor. Chem., 1983, 28, 473; Trindle, C. In ACS Symp. Ser 306; Pierce, T.; Hohne, B. Eds.; American Chemical Society, Washington, DC, 1986; p159
12. Crippen, G. M. Distance Geometry and Conformational Calculations, Wiley: New York, 1981
13. Burkert, U.; Allinger, N. L. Molecular Mechanics; American Chemical Society, Washinton DC, 1982
14. Carson, M. J. Molec. Graphics, 1987, 5, p 103
15. Brinkley, J. F.; Altman, R. B.; Duncan, B. S.; Buchanan, B. G.; Jardetzky, O. J. Chem. Inf. Comput. Sci., 1988, 28, 194

RECEIVED June 9, 1989

Chapter 9

Designing an Expert System for Organic Synthesis

The Need for Strategic Planning

Peter Y. Johnson¹, Ilene Burnstein², John Crary³, Martha Evans²,
and Tunghwa Wang²

¹Department of Chemistry, Illinois Institute of Technology,
Chicago, IL 60077

²Department of Computer Science, Illinois Institute of Technology,
Chicago, IL 60077

³G. D. Searle & Company, Skokie, IL 60077

SYNLMA, an expert system for organic synthesis, with a theorem prover as its inference engine and NCI's XTCHEM as its user interface, uses a retrosynthetic approach to find reaction pathways and generate a problem-solving tree representing the alternative designs it has explored. Presently, the system is capable of handling compounds of the order of complexity of Darvon, Ibuprofen, and the bicyclic system, cocaine. The combinatorial explosion that results from the input of larger target molecules has convinced us of the need for strategic planning during the synthesis process. We have developed a three-stage approach to aid SYNLMA in the planning process. The first stage identifies abstracted potential starting materials or name reaction derived synthons using graph overlay techniques to compare them with complex substructures in the target molecule. The second stage involves using "PMCD" strategies to define graphical paths between the target and abstracted synthons or starting materials. The leaf nodes on this path represent "chemical islands" which are then connected by general reaction rules. The third stage defines the tree by supplying specific reaction rules.

SYNLMA is an expert system designed to produce reaction pathways for organic synthesis problems. Many groups have worked on the organic synthesis problem, in the main using conventional programming techniques (1-7). What makes the SYNLMA system unique is the partitioning of the system into independent units consisting of a

0097-6156/89/0408-0102\$06.75/0

© 1989 American Chemical Society

chemical knowledge base, a user interface and a reasoning component. Our choice of a theorem prover as the inferencing engine adds strength and flexibility to the system design.

Separation of the reasoning component from other components of the system has proved to have many advantages (8-10). It has allowed us to experiment with different representations of chemical knowledge without major changes in the overall system. The design has also allowed us to easily add or delete knowledge from our data base, and has given us the potential to interface with a wide variety of commercial data bases such as those provided by the Institute of Scientific Information and Chemical Abstracts.

Presently SYNLMA has been applied to the syntheses of small compounds of the order of complexity of Darvon (1, Fig. 4) (10), Ibuprofen (2, Fig. 3) (11), and the bicyclic compound cocaine (3, Fig. 6) using a data base of several hundred selected reaction rules and nearly fifty selected starting materials. Those rules and starting materials needed to duplicate the published syntheses of 1, 2, and 3 were included as a subset of our total reaction rule data base. Attempts to synthesize larger molecules, or interface with commercial data bases have resulted, for a number of reasons in a combinatorial explosion, clearly illustrating the need for strategic planning. We are now in the process of adding planning intelligence to the system so that it will be more efficient in developing reaction pathways. The goal is to have SYNLMA more closely model the thought processes of the synthetic chemist. Our flexible design is enabling us to experiment with a new three-stage planning strategy using the inferencing capabilities of the theorem prover, while keeping the chemical representation scheme and user interface intact.

Description of the Present SYNLMA System

To initiate a chemical synthesis, a user of SYNLMA first interacts with the front end of the system, a series of Pascal programs called XTSYN, based on the National Cancer Institute XTCHEM structure input package (12). XTSYN was developed by John Crary (11) on an IBM/AT with a 80287 math coprocessor and a Hercules graphics board. The user enters a graphical representation of the target molecule at the keyboard. The system converts this graphical representation into connect table format and stores it in a file. At the user's request XTSYN will convert the connect table representation into clause format which can also be stored in a file, which serves as input to SYNLMA. XTSYN also has the capability to perform the reverse process of converting clause representations into connect table form, which can then be used to generate a graphical display of a given molecule.

This capability is particularly useful during a run. Subgoal compounds generated as clauses and incorporated in the problem solving tree can be displayed on the screen in graphical format for user inspection. A complex data structure in the form of a doubly linked list is used by XTSYN to convert the molecule representations from one form to another (11).

In order to produce solutions to a given problem, the theorem prover must be provided with a theorem to be proved and a set of axioms all in clause form. In the case of SYNLMA, the target compound becomes the theorem to be proved. It is converted into a clause list which consists of individual clauses representing the chemical environment of each atom in the compound bonded to at least two other

atoms. Axioms containing chemical knowledge in the form of functional groups, reaction rules, and starting materials are also represented in clause format (10). The target is decomposed into simpler precursor compounds by pattern matching with the appropriate reaction rules chosen on the basis of functional group identification and priority information. These less complex compounds are then considered as subgoals by the system. The process of decomposition continues until the bottom level precursors are available compounds or other user defined constraints are satisfied. This backward reasoning process, called retro-synthetic analysis by chemists, is often called backward chaining by computer scientists.

The Multi-Layered Design Approach. The present system is built of several layers as shown in Fig. 1. The top layer of SYNLMA directs the synthetic process, calling the middle layer to perform one-step reactions. The bottom layer is a custom-made theorem prover modeled after "ITP," an interactive theorem prover. "ITP" itself is built upon a package of Pascal routines called Logic Machine Architecture (LMA), which implements a resolution based theorem prover. Inferencing rules, based on classical logic techniques, are continuously applied to existing clauses to generate new information. Both ITP and LMA were designed and implemented by the theorem proving group at Argonne National Laboratory (13-14).

The top layer of SYNLMA maintains a set of complex data structures which represent synthesis information found by the lower layers. One of these structures, the problem solving tree, is a representation of the pathways SYNLMA has generated for the synthesis of the target compound. The nodes of the tree represent molecules, the arcs represent reaction rule information. The root node is the target compound. The leaf nodes are the starting materials. Other data structures managed by the top layer are the molecular hash table which contains information for all of the molecules found so far in the development of the problem solving tree. Only one entry is made for a molecule in the molecular hash table. This entry contains two vectors that uniquely identify that molecule so that it is not processed more than once. A reaction rule hash table is kept for all the reactions referred to in the problem solving tree. Associated with each reaction are parameters indicating yield, experimental difficulty, cost, and safety. In addition to these data structures, a work list of all the molecules waiting to be processed is kept by the top layer in linked list form sorted in order of the values calculated for them by the evaluation function. The molecule with the lowest value is chosen to be processed first. These values are calculated by an evaluation function containing heuristic information which takes into account the complexity of the molecule, the functional groups it contains, and its position in the problem solving tree.

In order to derive precursor compounds, SYNLMA must search the reaction rule data base and match the goal compound with the product side of a reaction rule. This process begins at the top layer, which builds the problem solving tree. It calls the middle layer to add a new branch to the tree.

The middle layer is designed as a network of what we call environment pairs and serves as the interface between the chemical knowledge data base and the inference engine. Each pair consists of a) a call to the theorem prover with all the necessary information

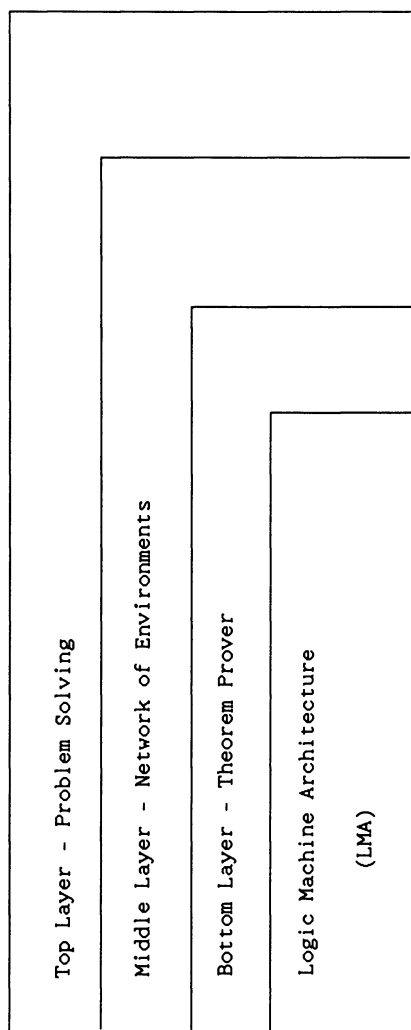


Figure 1: Layer Structure of SYNLMA

it needs such as clause lists, terminating conditions and strategies, and b) a utility program that extracts and stores information from the call to the theorem prover and also gathers information from the data base. Information found in one call to the theorem prover is made available for further calls. Fig. 2 shows the environment network of the middle layer, and its component environment pairs.

The first environment pair, 1A and 1B, is responsible for finding and storing functional group information. When the middle layer is called to add a new branch to the tree, the first step is a call from 1A to the bottom layer, the theorem prover, which generates functional group information for each goal compound. The second half of the pair, 1B, collects and stores this information. Once the functional group information is available, the next environment is called and 2A starts to search, by priority, a subset of the reaction rule files containing only those reactions that pertain to these functional groups. To make this search efficient, our initial approach has been to partition the reaction rule data base into subsets (chapters) ordered by unique functional group numbers. Priority for calling the functional group chapters has been set by the first author. As soon as a match is found between the product side of a reaction rule and the goal compound, the utility program in the pair, environment 2B, stores this information. There may be several rules whose product portion matches with the goal compound. All these potential reactions are stored for consideration by the third environment pair.

The third pair constructs new subgoal compounds from the reactant half of a reaction rule. It accomplishes this by substituting known atoms from the subgoal compound molecule for the variables in the reaction rule.

The fourth environmental pair processes the newly generated subgoals. The theorem prover is called to check each subgoal for chemical feasibility, presence or absence in the list of available starting compounds and to discover whether it has been previously generated by the system. This information guides the utility program so that it can insert each subgoal into the appropriate data structures maintained by the top layer.

During the course of the building of the problem solving tree, the chemist can extract subgoal molecules from the nodes of the tree. The subgoals, which are generated as clause lists can be passed to XTSYN which will display them on the screen in a graphical format and store them in XTCHEM connect table form for future display. Currently the search process terminates when any of the following conditions occur: 1) there is no more memory left in the computer, 2) the work list is empty; no more molecules to process, 3) the user-specified upper limits on the height and/or depth of the problem solving tree are exceeded, 4) the user-defined maximum number of solutions is reached.

Scope of the Present System

With the present design, a data base of fifty selected starting materials, and two hundred selected reaction rules SYNLMMA is currently able to generate synthetic trees, often in a very naive or inefficient manner, for molecules of the size and complexity of Darvon, Ibuprofen,

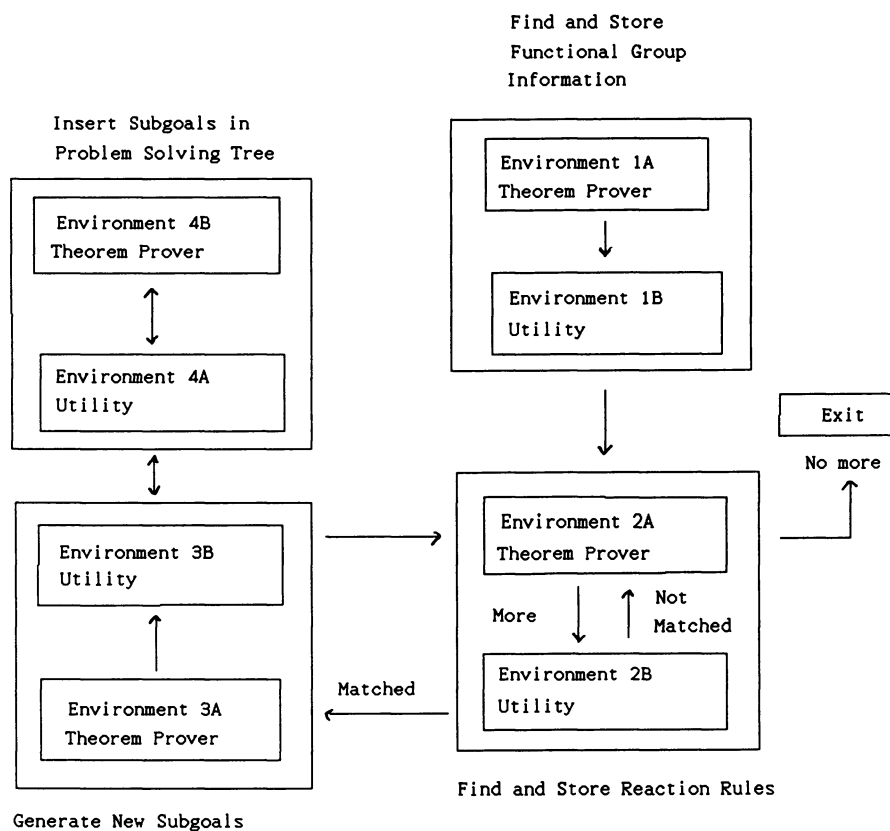


Figure 2: A Network of Environment Pairs

and cocaine. A pruned sample problem solving tree showing plausible routes to Ibuprofen is displayed in Fig. 3. The most serious potential problem is combinatorial explosion. These synthetic trees represent limiting cases for system resources on a VAX-11/750 running the UNIX operating system. It is clear that attempts to synthesize more complex molecules, or interface with data bases containing thousands of starting materials and reaction rules will result in a combinatorial explosion.

Errors in pruning also cause significant problems. Omitted pruned paths generally resulted from our not using reaction rule constraints or nonselective and/or non-intelligent use of the rules. This is one reason why none of SYNLMA's paths represent published syntheses of Ibuprofen (15) in spite of the fact that the requisite rules were in the data base. On the positive side, the synthetic paths to Ibuprofen discovered by SYNLMA are straightforward and would probably work as shown.

SYNLMA, in its present form is chemically unsophisticated. It does not have the reaction insights, information on structural limitations, and planning strategies that the expert can call into play during the course of solving a synthesis problem. For example, when planning the synthesis of a C₂₅ n-alkane which contains a long linear chain of repeating "(CH₂)_n" groups, a chemist, hoping to minimize the synthetic steps in his synthesis, would typically start the search for precursor synthons having approximately half the chain length containing appropriate bond making functional groups. One of the SYNLMA solutions to this problem was a step-wise synthesis of the entire chain, one methylene unit at a time, using a nonselective bond-making reaction such as a carbene insertion reaction. Clearly no knowledgeable chemist would take this approach! This same nonselective carbene reaction was used as part of the SYNLMA solution to suggested synthesis of Darvon as shown in Fig. 4. This reaction and several others were removed from our reaction rule data base in order to prevent their nonselective use.

As one can see, the nature and selection of reaction rules has placed limitations on SYNLMA. The reaction rule data base not only contains the rule itself, but also "must have-must not have" information/constraints concerning functional group incompatibility (10). These mandated constraints, often invoked in lieu of selectivity knowledge, protected us from incorrect use of some reactions, but, in numerous cases, also caused SYNLMA to eliminate potentially useful reaction rules - rules that a chemist might have considered in spite of the constraints. For example a chemist might be happy to sacrifice one equivalent of a cheap Grignard reagent to a compound containing both a ketone and an alcohol in order to have the second equivalent add to the ketone. More insidious to us was the inability, with constraints on, to use many double addition reactions required to make bifunctional cocaine starting materials. Some examples are shown in Fig. 5. Tetra bromide 17 was not considered as a potential starting material since the first bromine addition to give 16 was not allowed. The reaction rule says you cannot add bromine to a non conjugated alkene if there is another alkene present. In the second example, the selective constraints prevent SYNLMA from adding hydride or Grignard reagents arbitrarily to the carbonyl of its choice to give alcohols 20 or 21 when reacting with dione 18.

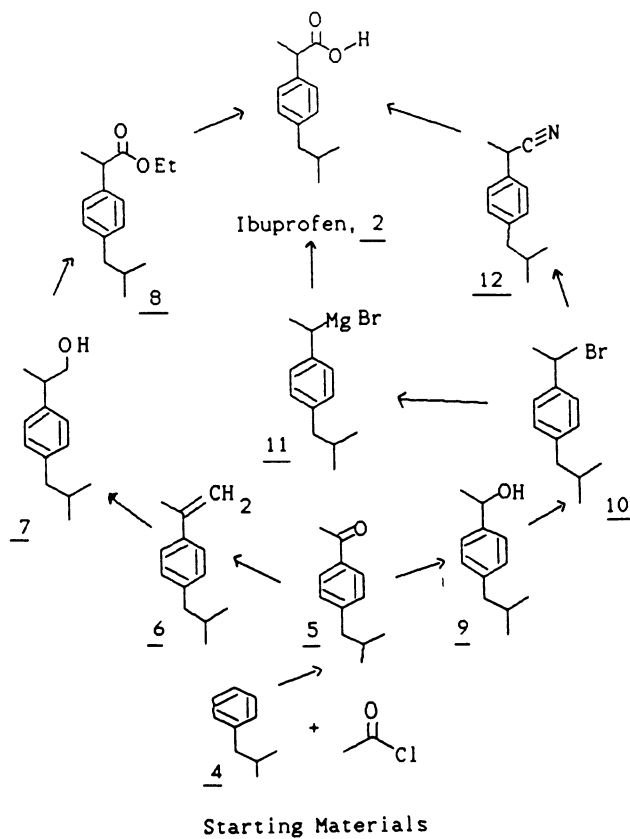


Figure 3: Pruned SYNLMMA Synthesis of Ibuprofen

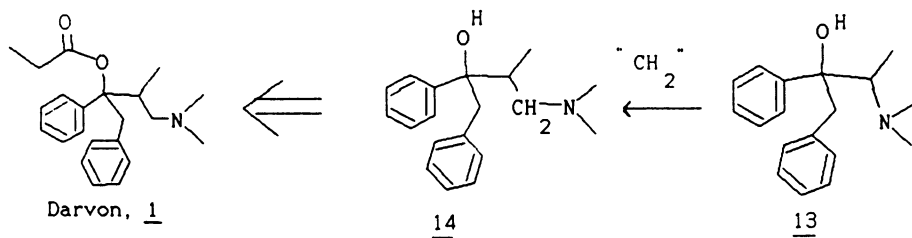


Figure 4: Darvon Synthesis, Nonselective Carbene Insertion

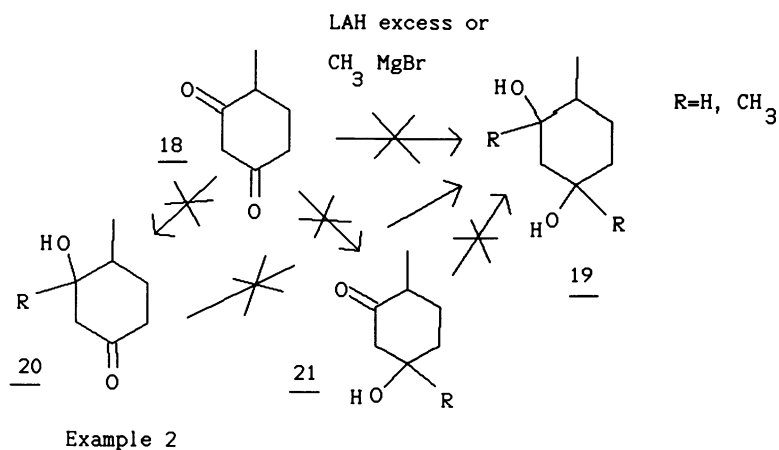
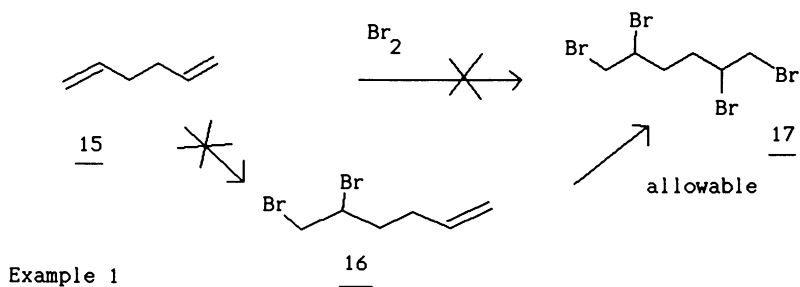


Figure 5: Reaction Rule Constraints

This protection against nonselectivity also stops addition of two equivalents of reagent to 18 to give diol 21, a chemically reasonable route to this compound.

It is clear now that we could rewrite the reaction rules to invoke subrules or layers of qualifiers as a means of effecting reaction selectivity but checking every qualifier of every rule called would slow SYNLMA significantly. This level of consideration would more reasonably be done after several strategies had been chosen for further investigation. Our experiments with reaction taxonomies are discussed later in this paper. Without the constraints SYNLMA finds more paths but is less efficient in its generation of viable synthetic pathways.

While having too many "chemical restrictions," the reaction rules have no "structural restrictions." In Fig. 6, which shows the first retro-synthetic steps SYNLMA considered for cocaine synthesis, we see that four Bredt's rule violations, enamines 27a,b and 28a,b were accepted as subgoals. While discussing Fig. 6, it should be noted that structures 22, 23, 24, and 26 are not allowed when constraints are on. Structures 23, 24, 27, and 28 are typical of current SYNLMA output. When it finds a reaction rule, it applies the rule exhaustively. Structures 29 and 30 are not synthetically demodulated and represent wasted CPU time. Finally, one second generation structure, 31, is shown because it represents an interesting variation of an N-oxide ene cycloaddition reaction that has been used to synthesize tropanol (16), the basic cocaine ring system.

From the above examples, it is clear we need to build effective planning strategies into SYNLMA and restructure our data base of chemical information. This will improve the efficiency of our system and make it a viable assistant to the synthetic organic chemist.

Modeling Strategic Planning For The Synthesis Process

Our new system design involves planning and organizing the synthesis process so that it closely models the human expert's approach. How do chemists deal with a complicated organic synthesis problem? They seem to organize their work into three successive stages which we call the tree-definition, tree-building, and tree-verification stages (17-18). We are now in the process of redesigning and upgrading SYNLMA to reflect this new approach.

In the first stage, the tree-definition stage, the main thrust is to identify potential starting materials and/or methodologies by noting resemblances between the target compound and (1) classes of available starting materials or (2) substructures or superstructures produced by name reactions such as the Fischer indole synthesis. Examples of these two approaches are shown in Fig. 7 (19) and Fig. 8 (20) respectively for the synthesis of the alkaloid ibogamine, 34. In the substructure driven approach to ibogamine, the indole substructure 34 (Fig. 7) is recognized as an abstracted starting material. As outlined below in the discussion of the Tree Definition Stage, the abstracted starting materials are linked to increasingly specific, less abstract potential starting materials (see Fig. 9). The identification of a potential starting material drives the retrosynthetic analysis in a manner which preserves that component, generating in Fig. 7 the chemical island 35. In the methodology

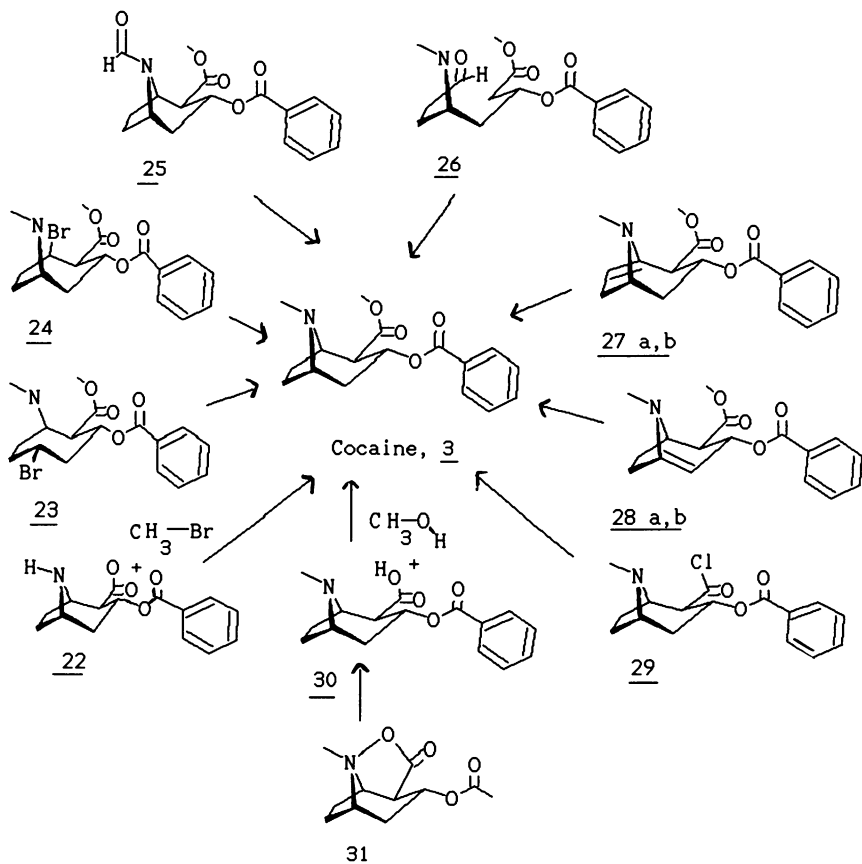
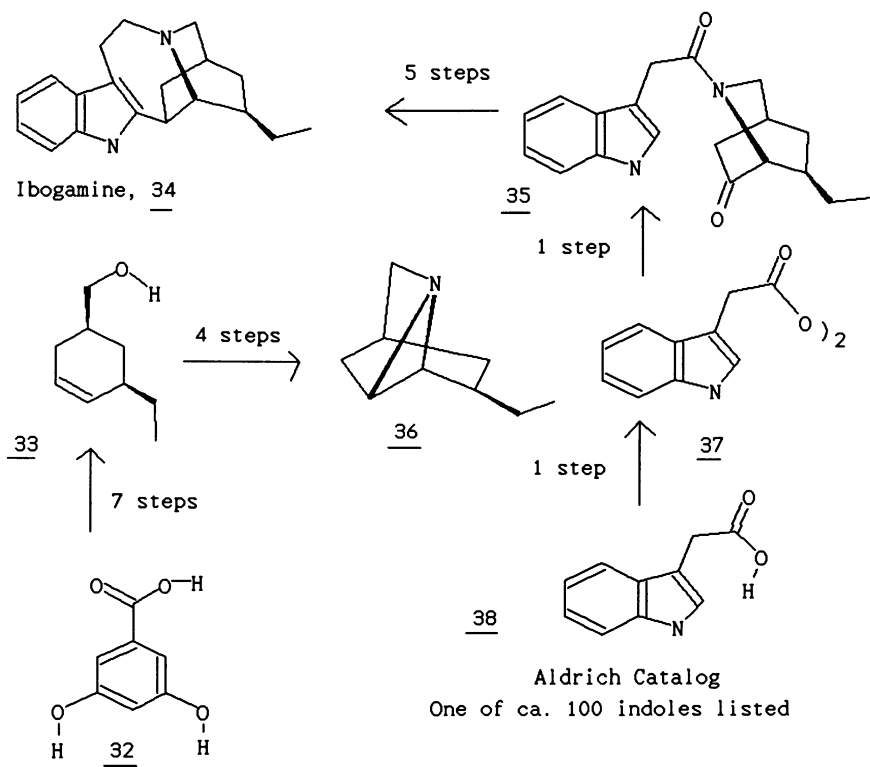


Figure 6: SYNLMAs Syntheses of Cocaine
Initial Retro-synthetic Paths



Aldrich Catalog

1 of thousands of 6 membered rings

Figure 7: Ibogamine synthesis: Substructure driven.

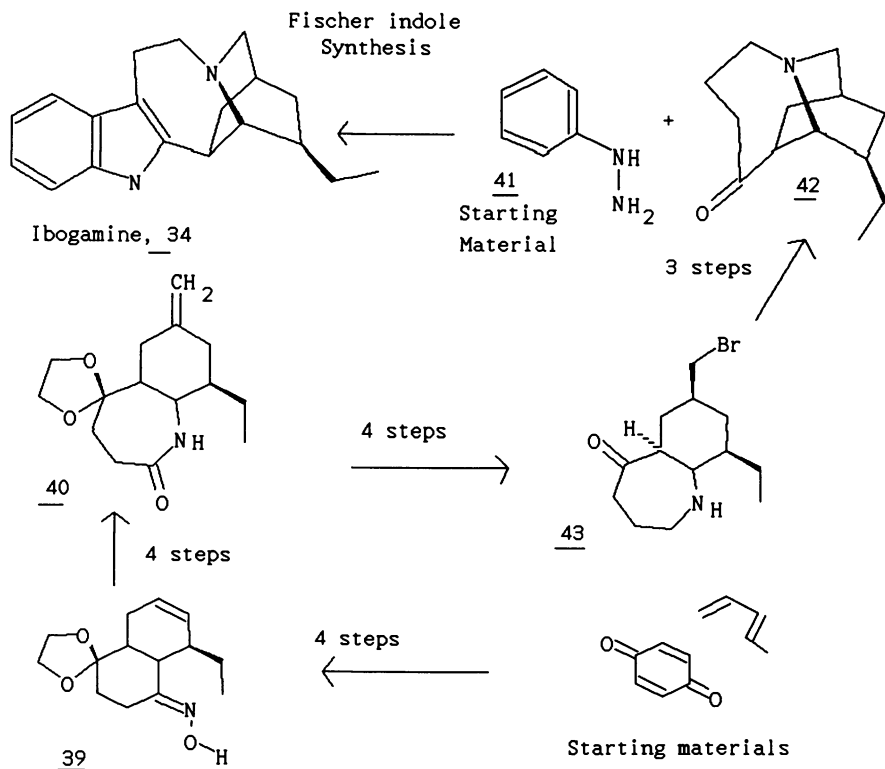


Figure 8: Ibogamine Synthesis. Methodology Driven

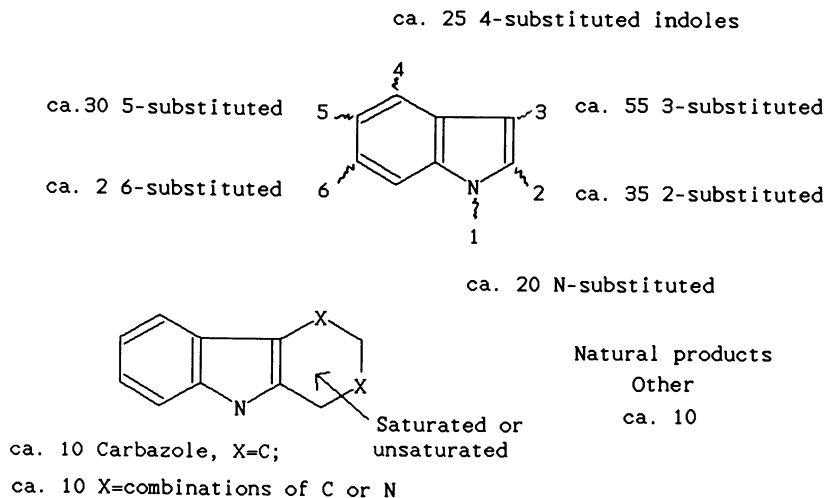


Figure 9: Indole Substitution Patterns Found in Aldrich Catalog

driven approach to ibogamine, the indole substructure **34** (Fig. 8) is recognized as the abstracted end result of a named organic reaction. In this case the retrosynthetic analysis using the named organic reaction will indicate that **41** and **42** are potential synthetic subgoals for further synthesis.

The discovery of these resemblances gives a tentative shape to the problem solving tree. This is the highest level planning stage where chemists use induction to limit the search for starting materials and to determine where to focus their deductive processes, in many cases making what has been called "the intuitive leap" (21). Resemblances between the target and starting materials, or methodology-produced intermediates can be described in terms of chemical substructures or superstructures and translated into graph overlay techniques which can be implemented by SYNLMMA.

During this stage the system can fill in the root node and many of the leaf nodes in the problem solving tree. The leaf nodes are the classes of starting materials or synthons which have been identified by the system as having significant or strategic structure in common with the target compound.

During the second stage, the tree-building stage, a collection of crude, imprecisely defined problem solving trees will be generated. The chemist goes through an analogous stage. Once an analysis of the target has been completed, rough synthesis outlines/routes are constructed usually reflecting the individual's knowledge, creativity, and prejudices. As part of the process, the chemist will often insert, potentially at any node along the path, intermediate structures which are expected to be convertible to the target or higher level intermediates in the pathway. These intermediate structures also have a reasonable chance of being synthesized from some starting materials available, in the abstract (21). In summary, these intermediate compounds, which can be considered "chemical islands" have a structure which is related to the abstracted starting materials or methodology derived synthons and the target molecule. They can be represented as leaf nodes along a crudely defined synthetic pathway. To reach from starting materials or synthons to these "chemical islands" and then to the target may involve several multi-step synthesis processes which can be filled in through successive tree-building stages, each providing a skeleton plan for the next stage; each using more detailed or selective reaction rules. What is required to implement this approach is a new organization, or taxonomy of reaction rules for SYNLMMA, ranging from the very general to the more specific. The more general rules, which represent multi-step reactions or processes, are applied in the earlier planning stages. The more specific single-step reactions are applied later.

During this second stage we will model the expert's approach to initial path generation and "chemical island" derivation by having SYNLMMA call a version of Ugi and Gasteiger's "Principle of Minimum Chemical Distance" (PMCD) program which offers a computer-assisted combinatorial solution for connecting two graphs (22). Incorporation of this strategy will help SYNLMMA choose efficient connective paths between target and generalized (abstract) starting materials or synthons. The nodes along PMCD paths connecting target graph and synthon or starting material graphs represent basic chemical structures. Addition of bond-making functional groups to last

disconnection points on these basic structures will constitute our first attempts at generating "chemical islands". The PMCD program, which is based on minimum structure change, has been used to demonstrate that many classic syntheses closely follow the most efficient graph representation disconnections between target and starting material.

For the chemist, the third stage in the synthesis process is usually a detailed analysis. During this stage all the steps are filled in, bridging the chemical islands to the target and to the starting materials. Factors such as yield, cost, and safety are considered at this point. For SYNLMA this phase will result in the completion of the problem solving tree using single step reaction rules chosen on the basis of functional group information. The system will have to examine adjacent nodes of the tree, find appropriate single step reactions rules and check cost and yield factors. Structural information will be incorporated into the reaction rules as constraints.

System Implementation

The new system resembles SYNLMA in overall structure; we have continued to use the three layered approach. The bottom layer is much like the bottom layer of the old system; that is, a custom built theorem prover calling LMA routines to do much of its work.

Argonne Laboratory is in the process of updating LMA, with particular emphasis on speeding it up. Any improvements made by the Argonne group will be incorporated into the new system. The middle layer continues to be a network of environment pairs, but with the additional pairs needed for graph overlay and PMCD implementation. The top layer is being entirely reorganized into three stages, the tree-definition, tree-building and tree-verification stages described above.

The Tree-Definition Stage. Often a chemist will choose a set of appropriate starting materials by noticing resemblances between the target molecule and classes of available compounds, be they starting materials or name reaction synthons. Resemblances between the target and starting materials can be described in terms of chemical substructures or superstructures. If we visualize a chemical structure as a graph, resemblances in the form of substructures or superstructures can be revealed by the overlaying of one graph on top of another.

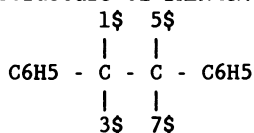
To implement a substructure identification process in SYNLMA we are adding to our knowledge base a group of chemically meaningful substructures. The substructures data base will be culled from the Aldrich Chemical Co. catalog and the 500 Organic Name Reactions listed in the Merck Index. These will be stored in clause form and arranged in a hierarchical format according to chemical complexity. The middle layer of SYNLMA will now have several additional environment pairs to handle the new planning strategies. One such environment pair will consist of a call to the theorem prover to find candidate substructures in the target molecule using its pattern matching algorithms. As in the case of Wipke's abstracted structures, exact functional group bonding details will be ignored at this time. In

order to use this substructure information to find potential starting materials, we are organizing the starting materials by generalized classes. Each class is represented by a pattern clause containing the substructure which defines that class, but with variables representing side chains and non-structure bonds. The generalized structures will have pointers to more specific patterns which will in turn have pointers to the unique structures listed in our reference sources. We expect to identify about two hundred generic classes. In the case of potential ibogamine starting materials (see Fig. 7), nearly 120 of the 14,000 compounds listed in the Aldrich catalog have the indole moiety as a substructure. The indole rings listed in the Aldrich catalog can be grouped according to their five substitution sites containing significant members. Fig. 9 shows the breakdown of the indole problem from most general to individual structures. (Some compounds contain multiple substitutions. These are multiply counted, once for each substitution position.)

Once the theorem prover has recognized the substructures in the goal compound, the system will search the abstracted starting material data base for classes of starting materials containing those substructures. If/when an abstracted starting material is recognized, it will point to a more specific possibility. The same type of pattern matching will be implemented for the methodology driven synthons data base. As shown in Fig. 8 for ibogamine, upon recognition of the indole substructure as the product of a name reaction, in this case the Fischer indole synthesis (or one of the other 13 name reactions leading to indole synthesis listed in the Merck Index), the program will construct the structures need to perform the name reaction. Name reaction precursors will become "chemical islands" or new targets. The utility portion of the environmental pair will then store the substructure information and the potential synthons or starting materials chosen.

As outlined above, the starting materials and synthons data base will be organized for efficient search by general structural types (graphs) at the tree definition stage. These general structure types can be organized as linked lists of related structures headed by a general pattern clause representing that particular class of starting materials or synthons. Once a match has been made with the pattern at the head of the list using the graph overlay techniques, the other structurally related potential starting materials on the list could be retrieved through the use of pointers.

The current system recognizes functional groups and ring structures. The new system will recognize larger substructures. For example using this new approach the system would be able to identify the following as a substructure of DARVON:



where 1\$, 3\$, 5\$ and 7\$ are variables representing various side chains. From the general pattern class the system could choose diphenyl ethane, stilbene or diphenyl acetylene as potential starting materials to be examined during the later tree-verification stage. The tree-definition stage is complete when: a) all significant

substructures in the present goal compound have been identified and b) classes of potential starting materials and/or synthons have been found. Significance is defined by a graph complexity algorithm, which counts the incidence of nodes (the numbers of arcs entering a node) and gives preference to those with high incidence values.

Other research groups have used substructure search as a method for selecting suitable starting materials (21). Having the theorem prover as our reasoning component makes this task easier for us to implement because of the theorem prover's ability to use pattern matching to identify the substructures and then match them with pattern clauses representing classes of starting materials. In addition, it is easier for us to represent abstractions of substructures by the use of clauses containing variables which substitute for atoms and side chains.

The Tree-Building Stage. In this stage we begin to sketch out the shape of the problem solving tree and construct pathways from the target molecule to starting materials. Our plan includes the modeling of the "Principle of Minimum Chemical Distance" (PMCD), developed by J. Gasteiger and coworkers (22). The use of the PMCD will help the system devise "chemical islands"; these are compounds which are structurally related to both the target and starting materials. Our implementation of the PMCD will differ from that of the Gasteiger group in that we represent chemical knowledge in terms of clauses rather than matrices.

To determine the minimum chemical distance, the largest ensembles (sets) of largest substructures in the target molecule must be identified. The next step is to find the largest substructures common to both the potential starting materials (subgoals) and target compound using the unification routines imbedded in the theorem prover. Some of this work has been done in the tree-building stage. From the trace of the proof we can discover where the target and starting material (subgoal) structures differ. A utility program paired with a call to the theorem prover can calculate the necessary "chemical distances" between target and subgoal from this information. These tell us, in graph form, which bonds need to be made and which need to be broken to produce, in a retrosynthetic sense, subgoals which can lead to targets. At this point we can construct the "island" molecules between the target and starting materials that will satisfy the PMCD. Our strategy is to have SYNLMA choose, based on PMCD information, the bonds to break to generate the "chemical islands." The bonds to be broken in the target molecule will lead to subgoals marked with reactive centers at the positions where the bond was previously attached. Ultimately SYNLMA will select functional groups to be placed at the reactive centers that would allow simple functional group interconversion and/or bond making reaction rules to "chemically" reconstruct the bonds suggested for breaking at the retrosynthetic planning stage. As can be seen then, the "chemical island" formed as a result of the bond breaking will contain the major substructures found in the target and starting materials. They will be represented as abstractions having marked reactive sites of intermediate compounds; that is substructures common to the target and starting materials will be present and other groups such as side chains will be represented as variables. The next task is to

construct bridges between the floating islands and the target. This may require many passes if the compound to be synthesized is very large and complex. Essentially each pass will make a general plan for the next, more specific pass. During each pass the PMCD will be used as a screening criteria to "turn off" random or inefficient synthetic paths thus keeping the tree from combinatorial explosion.

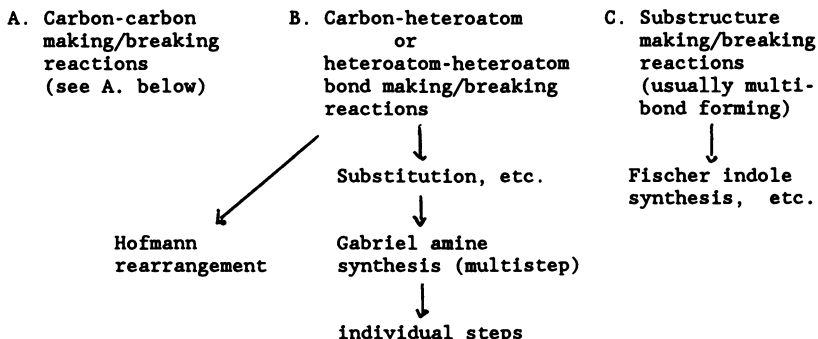
To implement such a scheme we are organizing our reaction rule data base into a taxonomy of reaction rules headed by (1) bond making/breaking reactions and (2) functional group interconversion reactions as shown in Fig. 10. This organization is modeled after the systems developed by Sacerdoti (23) and Stefik (24). At the top level of the taxonomy reactions are entered in as abstract a form as possible. These will often be multi-step reactions. Lower in the taxonomy the reaction rules are still in somewhat general form, but they have more chemical and structural detail. Gross functional group incompatibility and structural requirements (i.e. Bredt's rule information), will be included at this level. Finally, specific single-step reactions and their limitations are entered at the lowest levels. These would be similar to the reaction rules in the present version of SYNLMA which consist of a goal pattern (product) and a subgoal pattern (reactants). The theorem prover will choose a particular reaction rule whether multi-step or single-step, when it finds a match between the compound to be synthesized and the goal portion of the reaction rule (10). The top of Fig. 10 shows a fragment of a reaction type taxonomy. Each reaction type can be further subdivided according to structure reactivity differences. The bottom of Fig. 10 shows a fragment of a structure type taxonomy for substitution reactions.

By use of the taxonomy of reaction rules the system only searches a small relevant portion of the data base at each stage. In the early planning stages only the most general of the bond making/breaking reactions and functional group interconversions will be required; those at the head of the reaction taxonomy. As the synthesis progresses, the more specific rules are applied. For example, the system might choose the general category of substitution reactions for functional group interconversion at the beginning of a search. Further passes will involve the choice of the type of substitution reaction, for example SN1 or SN2. Finally, a specific reaction rule will be needed for the type chosen. Presently, the reaction rules in SYNLMA are stored in files indexed by the functional groups involved in the reaction. In the new system, the reaction rule data base will be larger and more complex. The partitioning of this data base is crucial to our ability to solve complex problems. Our plans are to organize the partitioning in terms of significant substructures. The design must also allow easy progress from the more general reaction types to the more specific. We believe the best approach would be to store the taxonomy in tree form with a secondary index of substructure pointers.

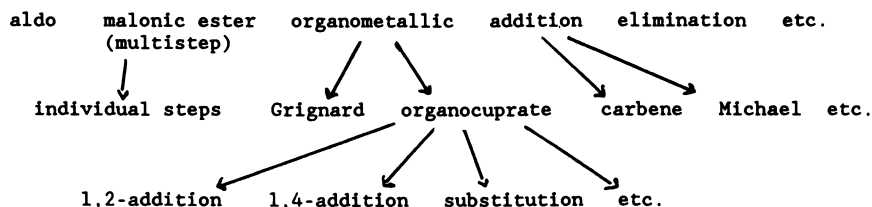
The Tree-Verification Stage. In this final stage the system attempts to refine the tree produced by the previous stage using the detailed analysis approach that SYNLMA uses in its unsophisticated version. In fact the latter form of SYNLMA can be used by making only slight changes to the logic. In the tree verification stage we examine each

Taxonomy of Reaction Rules

1. Bond making/breaking Reactions (see below 1.)
2. Functional group interconversion Reactions (see below 2.)

1. Bond making/breaking reactions

- A. (from above) Carbon-carbon making/breaking reactions (selected name reactions, some generic reactions)



2. Functional group interconversion reactions (many name reactions, all generic reactions)

Fischer Wolff-Kishner oxidation substitution reduction etc.

Typical taxonomy shown for: Substitution

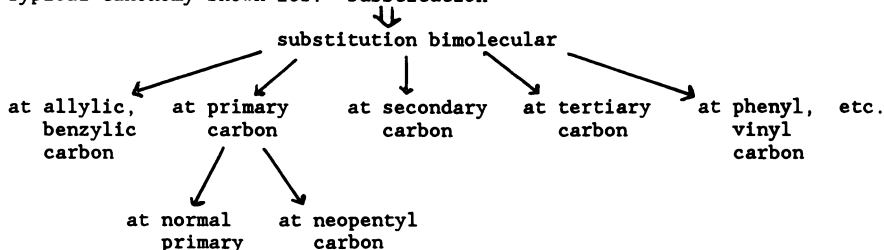


Figure 10: Reaction Taxonomy

pair of adjacent nodes in the tree, treating the upper node as a goal and the lower node as starting material. Because the distance between the two nodes is very small at this point, it is feasible to make a thorough examination of their best connection paths. The final listing of all node connecting paths results in a complete synthesis. During this stage starting materials need to be identified as specific compounds rather than general classes of compounds. The starting material data base will have to be organized at this stage, for easy access to this information. The organization will be based on functional group and substructure information.

The system can evaluate the proposed reaction pathways by use of the PMCD in this stage. A rough evaluation of the cost of a particular path can also be made in terms of the ranking of the general applicability of the reaction and the number of steps required. An industrial chemist who needed to make a more accurate determination of cost and efficacy would have to proceed with a literature search at this point. It is interesting to note that our approach would allow a chemist, in SYNLMMA's interactive mode, to insert a "chemical island" or structure in order to force or guide its use in a synthesis pathway.

Interfacing with Commercial Data Bases

For SYNLMMA to be of practical use to an organic chemist, it must be able to interface with large commercial data bases. Our plans are to continue work on building interfaces to the ISI and CAS (Chemical Abstract) data bases. We are also interested in the machine readable form of the Aldrich Chemical catalog, the Merck Index, and Beilstein Collection among others. ISI has added several useful features which make it especially attractive for us to use. For example, one can search the data base using what is called Generic substructures which may represent many actual compounds. A user can retrieve a specific compound by identifying the groups desired as side chains. Darc-Chemlink allows off-line phrasing of substructure queries on a PC making chemical structure searching easier for both the experienced and infrequent online user. Both systems should be accessible through the Chemical Abstracts Connection Table Format. We are in the process of moving to this format ourselves for internal use. This change will require a change in only one program, the program that converts connection table information to clause form.

Summary

Our initial research efforts were directed toward the development of an expert system that could solve chemical synthesis problems using a theorem prover as its inference engine. We have been successful in implementing such a system that can carry out syntheses of simple molecules such as Darvon, Ibuprofen, and the bicyclic cocaine. To develop a system capable of handling more complex molecules with access to large commercial data bases we are augmenting our initial design to include the planning strategies used by human experts. The new plan involves a three stage approach; the three stages defined as tree-definition, tree-building and tree-verification. In the tree-definition stage graph overlay techniques will be used to do

substructure searches which will allow for wiser choices of starting materials. The tree-building stage involves use of a taxonomy of reaction rules which will help guide the system through successive layers of detail appropriate to each pass. Gasteiger's PMCD will be used to evaluate the feasibility of proposed intermediate compounds. At the tree-verification phase actual details will be filled in. In each stage the theorem prover plays a significant role in the derivation of new chemical information for the system. Incorporation of our new strategies, induction to guide deduction to determine how to attack the problem, and planning at successive levels of generalization to manage complexity, will add sophistication to our system. The new system will be much smarter than SYNLEA. Instead of trying to build bigger and bigger trees, it will build better trees.

Literature Cited

1. Bersohn, M. ACS Symposium, 1977, 61, 128.
2. Gelernter, H.; Sanders, A.; Larsen, D.; Agarwal, K.; Boivie, R.; Spritzer, G.; Searleman, J. Science, 1977, 197, 1041.
3. Corey, E.; Pensak, P.A. J. Am. Chem. Soc., 1974, 96, 7724-37.
4. Wipke, W.; Grund, P.; Grabowski, Z.; Huff, P.; Smith, G.; Andose, J.; Rhodes, J. J. Chem. Inf. Comp. Sci. 1980, 20, 88.
5. Salatin, T.; Jorgenson, W.; J. Org. Chem. 1980, 45, 2043.
6. Vernin, G.; Chanon, M. Computer Aids to Chemistry, Ellis Horwood Limited, West Sussex, England, 1986.
7. Funatsu, K.; Sasaki, S.; Tetrahedron Comput. Method. 1988, 1, (1), 39-51.
8. Wang, T.; Ehrlich, S.; Evens, M.; Gough, A.; Johnson, P. Proc. Conference on Intelligent Systems and Machines, 1984, 176-181.
9. Wang, T.; Burnstein, I.; Ehrlich, S.; Evens, M.; Gough, A.; Johnson, P. Proc. 1985 Conference on Intelligent Systems and Machines, 1985.
10. Wang, T., Burnstein, I. Corbett, M., Evens, M., Gough, A., Johnson, P. ACS Symposium, Artificial Intelligence Applications in Chemistry, T. Pierce and B. Hohne, Eds.; 1986, 244-257.
11. Crary, J. M.S. Thesis, Illinois Institute of Technology, 1988.
12. Zehnacker, M.; Brennan, R.; Milne, G. W.; Miller, J.; Hammell M. J. Chem. Inf. and Comput. Sci., 1986, 26, 193-197, and refs. cited therein.
13. Lusk, E.; McCune, W.; Overbeek, R. Proc. Sixth International Conference on Automated Reasoning, D. Loveland, Ed.; Computer Science Lecture Notes, #138, Springer-Verlag: New York, 1982, 85-108.
14. Lusk, E.; McCune, W.; Overbeek, R. Proc. Sixth International Conference on Automated Reasoning, D. Loveland, Ed., Computer Science Lecture Notes, #138, Springer-Verlag: New York, 1982, 70-84.
15. For examples of Ibuprofen syntheses see Pinhey, J. and Rowe, B., Tet. Let., 1980, 21, 965, and refs. cited therein.
16. Tufariello, J.; Mullen, G. J. Amer. Chem. Soc., 1978, 100, 3638.
17. Bindra, J.; Bindra, R. Creativity in Organic Synthesis, vol. 1, Academic Press, New York, 1975.

18. Warren, S. Organic Synthesis: The Disconnection Approach, John Wiley and Sons, New York, 1982.
19. Sallay, S. J. Amer. Chem. Soc., 1967, **89**, 6762.
20. Nagata, W.; Hirai, S.; Kawata, K.; Okumura, T. J. Amer. Chem. Soc., 1967, **89**, 5046.
21. Wipke, W.; Rogers, D. J. Chem. Inf. Comput. Sci., 1984, **24**, 71-81.
22. Jochum, C.; Gasteiger, J.; Ugi, I. Angew. Chem. Int. Ed. Engl., 1980, **19**, 495-505.
23. Sacerdoti, E. A Structure of Plans and Behavior, Elsevier North Holland, New York, 1977.
24. Stefik, M. Artificial Intelligence, 1981, **16**, 111-140.

RECEIVED June 26, 1989

Chapter 10

Verification and Validation of Decision Support Expert Systems

Chemical Process Risk Management in International Operations

Myron J. Miller

Factory Mutual Engineering, Norwood MA 02062

Expert Systems are being designed or considered for both on-line process control and off-line decision support in the handling of hazardous materials. There are possibilities for both positive and negative impacts on chemical industry fire/explosion loss potentials. We will review some guidelines and tools available for the verification and validation (V&V) of expert systems. Finally, we will illustrate some of the points discussed by describing the development of a prototype expert system for hazardous materials classification and loss prevention engineering decision support.

OBJECTIVE - EARLY RECOGNITION OF V & V NEEDS

We hope to stimulate an interest in a very interesting and challenging subject with much work yet to be done. Consideration of V & V questions at an early stage of expert systems development should improve the chances of producing systems that are useful, safe and acceptable in the marketplace.

BACKGROUND

Catastrophic accidents have been a lesson in geography. Flixborough, Seveso, Feysin, Mexico City, Bhopal, Wesseling, Chernobyl, Schweizerhalle, Pampa, Norco and Piper Alpha are but a few of the recent incidents. Each issue of the Financial Times' World Insurance Report adds a few more to the list. Together they have had a major impact on both government and industry. At the same time the litany of less visible incidents continues. For instance, the Factory Mutual (FM) System Loss Prevention Publication "Programmable Logic Controllers" (1) indicates that in the period 1980-85 there were 729 losses involving programmable controllers with a net FM liability of about \$49 Million. The potential exists for both positive and negative impacts of expert systems technology on industry loss experience.

0097-6156/89/0408-0126\$06.25/0

© 1989 American Chemical Society

Since many failures of process control and protective systems are due to human element failures, reliable automation and decision support systems can help to control losses. On the other hand, inadequate design, deployment or configuration control can cause or contribute to significant losses. Artificial intelligence concepts and expert systems technology hold forth the promise of significant improvements in the safety of chemical processes, but only if these tools themselves are safe and reliable. Verification and Validation (V&V), Testing and Evaluation (T&E) will be necessary throughout the entire lifecycle of the system if maximum benefits are to be obtained.

DEFINITIONS - VERIFICATION, VALIDATION, LIFECYCLE

Verification is the process of testing and evaluation of both the software and the knowledge base to determine if they work properly together and according to specifications.

Validation is the process of determining whether the system actually helps the user as it was originally intended - and will continue to help him in the future. Validation should also make sure that the limits of applicability are defined and tested, the level of confidence specified and the consequences of failure due to misuse emphasized.

Dolores Wallace of the National Computer and Telecommunications Laboratory suggests that software V&V be treated as a single entity encompassing verification, validation and testing. Formal definitions are found in FIPSPUB 101.

The Lifecycle of a system should include the whole spectrum of activities from the critical "design for testing" phase to the final decision on when the system should be abandoned.

CURRENT AI ACTIVITIES IN CHEMICAL ENGINEERING

Much is happening already. More interaction between those studying chemicals and their reactions, and those who ultimately design and run - (and insure) - the process plants would be beneficial. A listing of titles from some of the recent Artificial Intelligence/Expert Systems sessions of AIChE meetings is appended to give you some idea of the interests of your counterparts in the Chemical Engineering Community. Most of these papers are available from the Engineering Societies Library in New York City, but have not been published in book form by AIChE as the ACS has done.

Other chemical engineering applications oriented activities are also in evidence. Ohio State University conducted a summer "Workshop on Applications of Artificial Intelligence Tools in the Process Control Industry". Another session, on "Expert Systems in Process Engineering & Process Development, Design, Control and Operations", was organized by Professor George Stephanopoulos at MIT. A session on "Artificial Intelligence in Process Engineering" was also

scheduled for the August 88 meeting of the American Association for Artificial Intelligence (AAAI).

There is much activity, both published and unpublished, on expert systems applications in chemical process analysis, design and control. A paper in *Computers & Chemical Engineering* titled "An Expert System Approach To Malfunction Diagnosis in Chemical Plants" (2a) is one of the Ohio State efforts that has been published. Related articles are also appearing in the Journals *Computers in Chemistry and Computational Chemistry*. The *Journal of Computational Chemistry* recently contained an article by Klopman and Raychaudhury at CASE Western Reserve describing a program called *Computer-Automated Structure Evaluation (CASE)"* (2b). Wu and Lidsky at MIT have published a report (3) on their Prolog based system designed to aid nuclear power plant operators to function within NRC guidelines.

Real-Time process control and diagnostic expert systems are only one area where improved loss control can result. The decisions relating to the design and protection of safe facilities are also areas equally amenable to the application of improved risk management decision aids. A paper by Volker Pilz (4) outlines many of the chemical reactivity, compatibility and regulatory considerations that must be addressed in the design and protection of facilities for storing and handling hazardous chemicals. Expert Systems would be useful for database access, hazard estimation and decision making for both normal and emergency operations.

The focus of this paper is on applications of artificial intelligence to improve industry loss experience. Considering safety and system verification, validation and testing requirements in the earliest stages of materials evaluation and process design will help attain this goal.

PURPOSE - AN OVERVIEW OF V & V

The purpose here is not to present a tutorial on software reliability or quality assurance, but to pose some of the questions relating to the verification and validation of knowledge-based systems, to outline some of the current research in this area, and review a few pieces of recent literature.

In the past three years AI Technology has spawned a whole new publishing industry. It is now almost impossible to read everything being published about AI and Expert Systems. *AI Magazine*, *IEEE Expert*, *Expert Systems*, *AI Journal*, *AI/Expert*, *International Journal of Intelligent Systems*, *International Journal of Logic Programming* and the *International Journal of Expert Systems Research and Applications* are only a sampling of journals in English. Conference proceedings, both U.S. and foreign, are proliferating rapidly (5,6,7). At the same time, much relevant material is being published under the headings of cognitive science, computational linguistics, and a wide variety of applications-oriented titles.

Universities world-wide are publishing a steady stream of papers and dissertations. One of particular note (8), describing an expert system called "LEILA" for "Estimating The Rates of Chemical Reactions", was spotted in the listings published by the ACM SIGART. There are also real live expert systems being sold commercially. An example is one called "SOPHIE" (9), which provides a systematic method for choosing procedures for analyzing risk. It is based upon the publication Battelle produced for the AIChE's Center for Chemical Process Safety (10).

INTERNATIONAL FACTORS

The use of expert systems for both on-line and off-line risk analysis and decision support becomes much more complicated in an international environment. Human input and output interpretation becomes a much more challenging exercise. The user model upon which the system depends most certainly will have to account for both linguistic and cultural factors if the system is to operate reliably and effectively. Multicultural human factors must certainly be addressed if we are to succeed in international trade and multinational enterprise.

About a quarter of the world's population speaks some English - just different versions. Questions of terminology and interpretation abound. Three-fourths of the world's population speaks no English at all, and they constitute the largest potential market for future development. Of course, physical chemists and chemical physicists often have trouble communicating due to the technical jargon and differing perspectives. The same is true of other disciplines.

Machine translation of "natural" languages is in my estimation the real test of AI. Research publications are increasing in both quantity and quality and the Center for Machine Translation at Carnegie Mellon University is giving current efforts some focus.

Recent developments such as the Monroe keyboard for entering Chinese characters (Boston Globe 7/25/88) is only one example of significant recent progress toward multilingual computing and machine translation.

SOFTWARE SYSTEMS SAFETY - TRIPLE REDUNDANCY FOR SOFTWARE?

Mil STD-882B (11) outlines the general concepts of Systems Safety. Notice 1 (12) appended to that document specifically covers requirements for software systems safety. The Air Force also has a Software Systems Safety Handbook (13). A number of recent articles in Hazard Prevention (14), the Journal of the Systems Safety Society, expanded on the technical content of this handbook. The concepts were covered in a more explicit and useful manner by Erwin Schoitsch of the Austrian Research Center Seibersdorf in a recent forty-two page, two-part, article (15).

Heinz Luck and Udo Schlossarek (16) at Duisberg University discussed the relative merits of a "perfectionist" versus a "fault tolerant" approach in the design of "Software Controlled Fire Detection Systems". This document also discusses the concept of triple redundancy of software, employing different algorithms, different computer languages, different analysts and different programmers for parallel channels with the same input/output objectives. This is akin to using the devices of different manufacturers, employing different principles in a redundant hardware system to decrease the chances of common mode failure.

A document called "The Engineers Responsibility for Computer Based Decisions" (17a), published by the Institution of Chemical Engineers, recommends triple redundancy - with voting - for software in "critical" applications.

Several of FM's Loss Prevention Data Publications (1, 17B, 17C) discuss the concept of triply-redundant, fault-tolerant, high-reliability hardware/software systems for manufacturing operations. Risk analysis and systems reliability research is currently underway to develop better guidelines for the design and application of reliable process control systems.

None of the documents noted above covers the questions specific to V&V for knowledge-based expert systems, although many of the concepts discussed are applicable. And triple redundancy is useless unless the top level problems of safety have been addressed in system specifications.

THE FUTURE OF SOFTWARE SYSTEMS SAFETY

R&M 2000, the Air Force Reliability and Maintainability Action Plan does not discuss the unique problems (and potential) of expert systems, particularly those which rely on the massive "knowledge bases" which can now be delivered on optical disc. Nor have they addressed the problems (and potential) associated with parallel processing, neural networks and "systems that learn".

Discussions with the Navy's representative to the Triservice Software Systems Safety Group and the NBS (Now NIST) representative to the Compass 88 Conference indicate that the impact (both positive and negative) of AI on software safety is recognized but has not as yet been documented or incorporated into an action plan. The state of the art can best be summarized by a recent RFP from Rome Air Force Base (19) calling for "The development of a self-diagnosing, self-repairing, active, intelligent data base capability whose architecture will utilize knowledge engineering techniques, system constraints and intelligent integrity rules in order to provide a data management capability resilient to logical faults, erroneous and spurious inputs/updates and integrity problems caused by malicious updates or incomplete data." Quite a wish list.

An overview of procedural software verification is covered by Mili (20a). Quirk (20b) covers V&V of real time software and DeMille, et al (20c) covers the general subject of software testing and evaluation.

There is also an increasing body of literature (21, 22, 23) concerning the reliability of programmable controllers (or programmable logic controllers (PLC's) as they are now called). In the UK these fall under the Health and Safety Executive's (HSE) guidelines for programmable electronic systems (PES). See Chemical Engineer (UK), for a discussion (24).

The book (25) "Programmable Controllers: Theory and Implementation" is a good overview which includes a chapter on "Artificial Intelligence and PLC Systems". More and more devices, such as Foxboro's EXACT system, include "artificial intelligence based" adaptive, self-tuning features.

V&V BY AND FOR EXPERT SYSTEMS

Peter Politakis' PhD thesis entitled "Empirical Analysis for Expert Systems" (26) is an excellent introduction to the subject of expert systems V&V. Politakis and Weiss (27) cover the use of case experience to refine the rules of a model in a medical diagnostics system.

An excellent survey article appeared in *AI/Expert* (28). Papers by Stachowitz, et al (29a, b, c, d) provide a good overview of most of the V&V component questions. The effort uses Prolog for its inherent logic checking potential. The elements of Lockheed's V & V expert system EVA (Expert Systems Validation Associate) are shown in Figure 1. Three other papers from the SOAR Conference also cover V&V concerns (30, 31, 32).

The book "Algorithmic Program Debugging" by Ehud Shapiro (33) is interesting for its in-depth description of debugging algorithms written in Prolog.

A 1986 ICOT report (35) from Japan describes "ARGUS/V: A System for Verification of Prolog Programs".

For those of you interested in Prolog, the book "Prolog Programming in Depth" (34), has a good discussion of program debugging and verification, as well as an example representation in Prolog of a connectivity table for molecular structure representation.

A paper by Richardson and Wong (36) indicates that the KBS lifecycle V&V process is now being formalized and will be the first to incorporate new quality standards for software. They state that the purpose of V&V is "To ensure that software is 1) designed to be testable, 2) completed according to defined specifications which lead to testability, and 3) tested according to requirements."

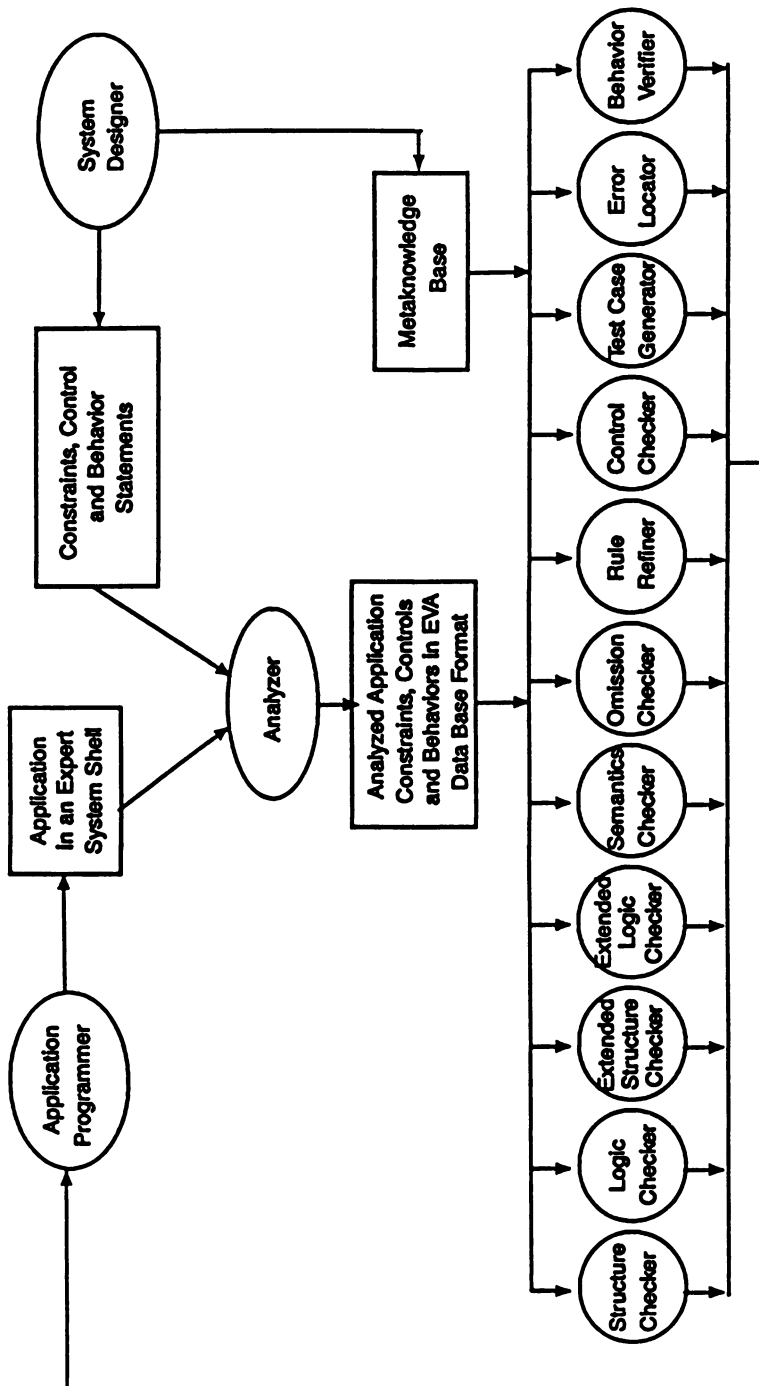


Figure 1. Validation functionality. (Adapted from ref. 29b.)

Application of V&V to expert systems for nuclear power plants is described in an Electric Power Research Institute report (37). Table 5.9-1 in that report indicates the "V&V Lifecycle Phases" for a number of expert system types. Again, the focus is on Lifecycle V&V.

We will not discuss the problems of bugs, viruses, Trojan horses, logic time bombs and hackers. You should all recognize by now the questions of software security that must be addressed in both the design and configuration control processes. Factors that might affect the performance of software/hardware/humanware systems are being addressed in depth at annual conferences of the Computer Security Institute, as well as by various government and industrial bodies.

V&V TOOLS AND COMPUTER AIDED SOFTWARE ENGINEERING (CASE)

Now, what about tools to aid the V & V process. Good, systematic software development, maintenance and documentation techniques are of course assumed (but don't let them get in the way of progress). Martin, Orr, Yourdon and others make a good living teaching and writing about these. In addition to the environments delivered with ART, KEE, M1/S1 and expert system development shells sold by TI, Gold Hill, Arity, KDS and others, a couple of recent demonstrations have been very impressive.

AI Corp's Knowledge Base Management System (38a) for the design of systems containing knowledge base and expert system components contains modules to aid the design process with forward and backward chaining, hypothetical reasoning and object oriented programming that incorporates the INTELLECT™ concepts of natural language user interface to generate working code.

Knowledgeware Inc's Information Engineering Workbench (38b) package is notable for its reported origin in 2000 rules and 60,000 lines of (Arity) Prolog code to manage several system development methodologies with alternate representations and logic and consistency checking of input via a single shared encyclopedia concept. The involvement of James Martin will give you some idea of the systems analysis, design and diagramming methodologies used. The output is C code.

Perceptronics' Knowledge Shaper (38c) advertises that it contains a "knowledge base validation and verification tool".

A combination of these three approaches would seem to give the best of all possible (expert system development) worlds.

Texas Instruments' Personal Consultant and Procedures Consultant (39) series of expert system development shells could also be considered as specialized CASE tools.

There are of course many other CASE products and a multitude of expert system development "shells", languages and "tools". Case

tools incorporating expert systems V & V technology should find a ready market.

CASE STUDY - V&V DURING PROTOTYPE DEVELOPMENT

Factory Mutual engineers and consultants are involved in two main activities, risk analysis and loss prevention engineering. Our parent insurance companies (Allendale, Arkwright, Protection Mutual, Factory Mutual International) provide property damage and business interruption insurance coverage and related risk management services for "highly protected risks (HPR)" worldwide. The generic problem faced by our field personnel is the definition (or classification) of hazards followed by decisions on appropriate recommendations for cost-effective forms of both active and passive protection.

Three years ago we conducted a survey of AI Technology and decided that developing a prototype expert system would give us a better feel for both the potentials and the problems associated with this "new" technology.

We chose to focus on the classification and protection of organic peroxides mainly because of the existence of FM Loss Prevention Data Publications (48a, 48b) on the subject and the existence of some in-house expertise.

Our exercise in defining the hazard classification logic and appropriate loss prevention measures has raised a number of questions, disclosed some potential problems and in general given us a better understanding of the subject matter. This in itself has made the exercise worthwhile.

After reviewing a number of AI languages and expert system development shells we chose one called KDS (40) for Knowledge Delivery Systems from the KDS Corp. in Wilmette, Illinois to serve as a development tool. The choice was based on consideration of price/performance, ability to run on a PC, ease of use and lastly - the developer, Barbara Wallace, consistently answered the phone and fielded our questions patiently and competently. In addition, the KDS product has seen a steady stream of improvements and enhancements which make it a very powerful tool for a very competitive price. It is written in Assembly language, which makes it fast and compact.

Our prototype hazard classification system was intended to be based on three approaches, 1) look up the answer in a database (either ours or others), 2) infer an answer by comparison of unclassified materials with materials whose class is already "known" with some degree of certainty, or 3) ask for sufficient input data (either basic properties or "test" data) on the material to allow calculation of the potential hazard from something akin to "first principles: - an algorithmic or logical solution.

This approach is still considered quite valid; however, our expectation of the availability of unambiguous baseline data and well-defined, universally-accepted classification logic turned out to be overly optimistic.

A summary of some current efforts in this area of material hazard classification follows:

FM/NFPA: The classification processes of both FM and NFPA (The National Fire Protection Association) turned out to be essentially a committee decision based on an undocumented review of material properties and unvalidated test data submitted by manufacturers, either individually or through the auspices of the Organic Peroxides Producers Safety Division (OPPSD) of the Society of the Plastics Industries (SPI). There was and is to date no published protocol defining the logic of the hazard classification, testing and decision making process, although as a result of this effort a logic diagram has been proposed for discussion by the NFPA Technical Committee on the Storage, Handling and Transportation of Hazardous Chemicals. It is still under discussion. In the meantime there continue to be some significant differences between FM and NFPA conclusions (45, 48a, 48b) regarding both the classification and protection of packaged peroxide formulations.

UN/TNO: The UN and TNO (the Netherlands) are grouped together because of TNO's major role in coordinating the efforts of various U.N. organizations dealing with the transport of hazardous materials. The U.N. Working Group of Experts on the Classification of Organic Peroxides and TNO are also involved with related questions of the hazard classification of oxidizers. At present there seems to be no basis for the construction of an anion-cation matrix defining the relative hazard of all inorganic oxidizer possibilities, much less one for organic materials. The current TNO guidelines for storage of organic peroxides (49) includes classification procedures.

For those who wish to keep abreast of the activities of national and international groups dealing with the transport of hazardous materials, the "Hazardous Cargo Bulletin", published in the UK, has timely technical updates.

The paper by Pilz (4) previously mentioned indicates that these transportation hazard classes are also being used to determine warehousing requirements, although transport hazard test procedures as presently conducted do not adequately address the sprinkler system design and density requirements for large quantities of hazardous materials in a variety of packaging and storage configurations. It is, however, a significant first step toward minimizing the potential for a Schweizerhalle-type incident. The present U.N. classification logic diagram for organic peroxides as published in the latest U.N. "Orange Book" is shown in Figures 2 and 3. The relationship between the flow chart "exit" (A-G) and the U.N. Packing (Hazard) Groups (I-III) is still somewhat elusive.

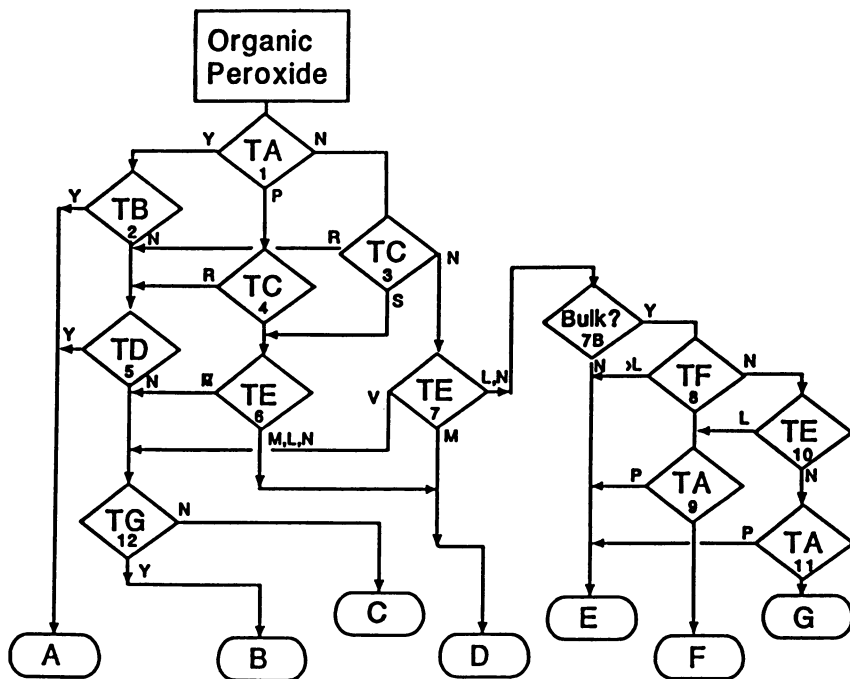


Figure 2. U.N. organic peroxide classification system. (Adapted from ref. 46.)

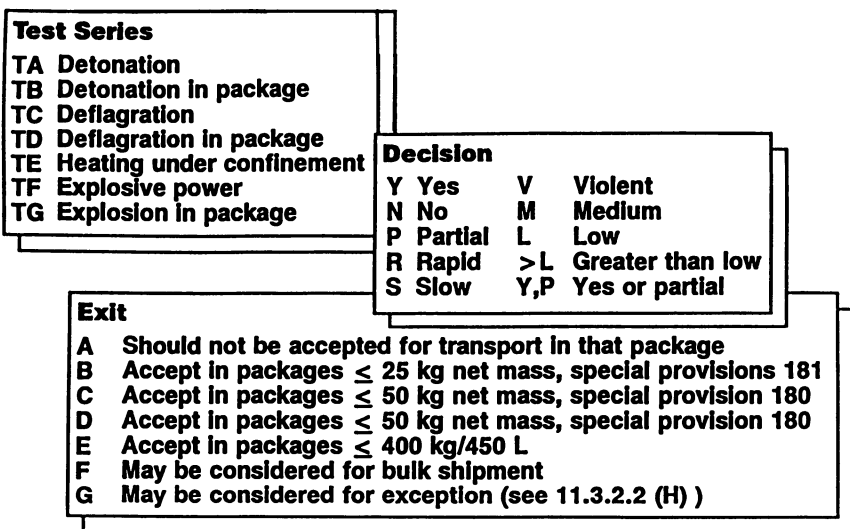


Figure 3. Key to U.N. organic peroxides test protocol. (Adapted from ref. 46.)

Work on peroxides is currently being coordinated by TNO; a database of test information from laboratories worldwide has been developed and the classification protocol is being refined. The next step is to reduce the number of "Orange Book" entries for organic peroxides from over a hundred to twenty generic (class) entries. Hopefully the working groups will also give some consideration to warehousing and warehouse protection requirements to account for problems at both ends of the transport process. The hazards during manufacture and in-process use of peroxides are another matter entirely. It is anticipated that more consideration will also be given to the arrangement and protection of hazardous materials during transport as well as quantity and compatibility limitations.

ASTM: Saul Patai's book "The Chemistry of Peroxides" (41) was reviewed with some hope of finding either a systematic evaluation of reactivity as it relates to the potential hazard of peroxides or some basic concepts that would allow formulation of a unifying theory that would permit an algorithmic solution to the classification problem. Unfortunately, when Saul got to the chapter on safety he acknowledges that he gave up. Bretherick's "Handbook of Reactive Chemical Hazards" (42) presents some useful information and incident histories, but provides little in the way of coordinated insight based on molecular composition and structure to allow for systematic extrapolation to new materials. Discussions have also been held with ASTM E27 Committee (On The Hazard Potential of Chemicals) members regarding the use of the CHETAH (Chemical Thermodynamics and Hazards Estimation Program) software to aid in the classification process. Work is proceeding (by E27.07 Estimation Methods) on both a PC version of the existing CHETAH program (with considerable updating) and a new mainframe (and possible PC) version with expanded capabilities. CHETAH is described in detail in references 43a, b, c.

Copies of the TNO peroxide test databases have been provided to E27.07 and the new versions of CHETAH are expected to contain an extensive database as well as pattern-recognition techniques for estimating the hazard of new materials. The CHETAH software will continue to rely on bond energy data and group contribution calculations to estimate energy release potential. Hopefully, the new versions will also incorporate natural language expert system-type front ends so that the CHETAH program(s) will see expanded use in both analytical and tutorial modes. Copies of the LEILA (8) dissertation have also been provided to E27.07 as an example of an expert system approach to selection and use of appropriate theories and computational methods for the solution of problems in chemical kinetics.

The ASTM Committee E27.02 (Thermal Stability/Condensed Phases) has also recently published a document (44) entitled "Standard Practice for Calculation of Hazard Potential Figures of Merit for Thermally Unstable Materials". This includes calculations of 1) Time to thermal runaway, 2) critical half thickness, 3) critical temperature, and 4) adiabatic decomposition temperature rise.

Unfortunately, or fortunately depending on your perspective, the kinetic parameters needed in this calculation are obtained by Differential Scanning Calorimetry (DSC) and to my knowledge no systematically-acquired data has been obtained on a large number of pure peroxides and commercial formulations and their variations. The exercise could have both practical and theoretical benefits.

SUMMARY-STATUS OF RESEARCH BY THE UN, TNO, ASTM, FMRC

In summary, work relating to various approaches to the classification of hazardous materials is currently underway within committees of the NFPA (45), ASTM (43a, b, c, 44) and the UN (46). The TNO in Holland is spearheading much of the work on peroxides and oxidizers. It appears that all of these groups will produce significant new results by the end of 1988. In the meantime, the Factory Mutual Research Corporation (FMRC) has been making significant progress in developing test methods for classifying the hazard of both materials and commodities. One of these programs is summarized in a paper entitled "Generation of Heat and Chemical Compounds in Fires" by Dr. A. Tewarson (47) which appears as a chapter in the new SFPE (Society of Fire Protection Engineers) Handbook published this fall.

CONCLUSIONS

Because of cost and time constraints, we have not had the opportunity to actually work with and evaluate many of the tools mentioned. Consequently our conclusions are based in large measure on the comments received from those working in the fields of AI/Expert Systems and Software Verification and Validation. Also, my conclusions are those of a potential user rather than of a specialist in the field.

The principle conclusion is that the development of "expert" systems and the necessary audit tools for verification and validation of those systems is still in infancy. There are a number of interesting concepts under development which bear promise but which will probably see their first application within critical space and defense applications. Tools useful to those of us interested in more generalized risk and reliability analyses are probably at least two to three years away.

Mike Brown, chairman of the DOD Tri-Service Software Systems Safety Working Group (SSSWG), emphasizes that "Safety must be designed into a system and particularly the software at the earliest possible stage of development." "We can't V&V a requirement that isn't there or doesn't exist." "The most common problem that we encounter in software systems safety is incomplete specifications and inadequate design from the safety perspective." The Tri-Service Working Group is currently working to produce a "how to" handbook to augment or supersede current guidelines. Again, good software engineering practice should include safety considerations at the earliest stage and throughout the lifecycle.

And software, expert systems included, should be designed to be tested.

Expert systems technology can be useful. We have demonstrated the ease of use and logic-documentation capabilities of a prototype using the UN flow chart and the KDS expert system development software. KDS Corp personnel have also demonstrated the relative ease with which our Loss Prevention Data Sheets can be represented in expert systems format. They will be happy to explain their demo "PEROX" based on our D.S. 7-81 (48). At present, however, the "PEROX" demo has been neither funded nor endorsed by the FM System.

Peroxide hazard classification expert systems development at FM/Norwood is presently on hold until some of the resources noted become available. We have a way to go before a fully validated classification model is complete. At the same time, we are continuing to explore the possibility of using chemical database programs such as those available from Molecular Design Inc. and the University of Santiago, Chile (ARIUSA) as components of our chemical hazard expert systems. We are also looking at chemical databases on optical disc such as those available from DuPont, Aldrich, Micromedix and the Canadian Center for Occupational Safety and Health as components of future systems.

THE FUTURE

Combinations of portable microcomputers, optical disc/card knowledge bases, expert systems technology, natural language processing and machine translation will have a major impact on the way we do business in the future.

The acceptance of expert systems in process control applications and in risk assessment, risk management and loss prevention engineering will depend to a large extent on the degree to which verification and validation questions are addressed early on in the design of the systems.

References

- 1 Anon "Programmable Controllers" Factory Mutual Loss Prevention Data 5-27/14-27 (Feb 1988).
- 2a Shum, S. K., Davis J.F., Punch, W.F. and Chaudrasekaran, 13 "An Expert Systems Approach to Malfunction Diagnosis in Chemical Plants". Computers in Chemical Engineering, Vol. 12, No. 1, 1988, pp 27-36.
- 2b Klopman, Gilles and Raychaudhury, Chandan, "A Novel Approach To The Use of Graph Theory in Structure-Activity Relationship Studies. Application to the Qualitative Evaluation of Mutagenicity in a Series of Nonfused Ring Aromatic Compounds." Journal of Computational Chemistry, April/May 1988

- 3 Wu, Yun-Fu and Lidsky, L.M. "Nuclear Power Plant Technical Specifications Computerization: Application of Prolog" MIT-NPI-TR-008, May 86.
- 4 Pilz, Volker (Bayer), "Sichere Lagerung von Stoffen in der chemischen Industrie". Chemie Ingenieur Technik, 60 (1988) No. 6, S.456-463.
- 5 Anon. Various "German Workshops on Artificial Intelligence (GWA) published annually by Springer Verlag as part of the Fachberichte Informatik Series
- 6 Anon. "Proceedings of The Second Annual Artificial Intelligence Research Forum" NASA/Ames, April 1987.
- 7 Anon. "Proceedings of the First Annual Workshop on Space Operations Automation and Robotics (SOAR-87) NASA/Johnson Space Center, Houston, TX, August 1987.
- 8 Sauk, Brian Edward, "LEILA: An Expert System for Estimating Chemical Reaction Rates" PhD Dissertation, Carnegie Mellon Univ., April, 1987.
- 9 Nicholosi, Stephen L. (Battelle-Columbus) "SOPHIE: An Expert System for Selection of Hazard Evaluation Procedures" 99 pp + disc, Oct. 28, 1987
- 10 Anon "Guidelines for Hazard Evaluation Procedures" prepared by Battelle-Columbus for the AIChE Center for Chemical Process Safety (CCPS), 1985
- 11 Anon "System Safety program Requirements: MIL-STD-88213 (30 March 1984)
- 12 Anon "Software Systems Safety" MIL-STD-882B-Notice 1 (3/84)
- 13 Anon-Software Systems Safety handbook: AFISC SSH-1-1 (5 September 1985) (U.S. Air Force)
- 14 Anon "Software Systems Safety" Hazard Prevention-Journal of the Systems Safety Society, Vol. 22, Nos. 1, 2, 3, 6, 1986
- 15 Schoitsch, Erwin "Software Safety and Software Quality Assurance in Real Time Applications". Computer Physics Communications, 50 (1988) 169-211.
- 16 Luck, Heinz O. and Schlossarek, Udo "Software Controlled Fire Detection Systems". Fire Science and Technology, Vo. 7, Nov. 2, 1987, pp 53.60.
- 17a Anon "The Engineers Responsibility for Computer Based Decisions". Published by the Institution of Chemical Engineers, UK, 1987.

- 17b Anon "Chemical Process Control and Control Rooms" Factory Mutual Loss Prevention Data Publication 7-45 (7/74)
- 17c Anon "Chemical Recovery Boilers" Factory Mutual Loss Prevention Data Publication 6-21/12-21 (9/85).
- 18 Lipow, Myron and Book, Erwin "Implication of R&M 2000 on Software" IEEE Transactions on Reliability, Vol. R-36, No. 3, August 1987, pp. 355-361
- 19 Anon "Intelligent Data/Knowledge Base System" Sol F30602-88-R-0071, February 1988, Rome Air Development Center, USAF
- 20a Mili, Ali "An Introduction to Formal Program Verification" VNR 1985
- 20b Quirk, W.J. "Verification and Validation of Real Time Software", Springer Verlag, 1985
- 20c DeMillo, Richard A., McCracken, W. Michael, Martin, R.J., Passafiume, John F. "Software Testing and Evaluation" Software Engineering Research Center. Georgia institute of Technology, Benjamin/Cummings, 1987
- 21 Wilson, Dean K. "Failure Mode Management: A Loss Prevention Philosophy for Programmable Logic Controllers" AIChE Loss Prevention Symposium, Mar 1988.
- 22 McMillan, Gregory K. "Can You Say 'Process Interlocks'?" Intech Journal, May 1987. Presented at the AIChE Loss Prevention Symposium, Mar 1988.
- 23 Thomas, Harold W. "A Quantitative Approach To The Use of Programmable Controllers in Safety Circuits". AIChE Loss Prevention Symposium Mar 1988.
- 24 Anon "PES's and the HSE Guidelines "The Chemical Engineer, Loss Prevention Supplement, No. 439, August 1987, pp. 16-17.
- 25 Bryan, L.A. and Bryan, E.A. "Programmable Controllers - Theory and Implementation" Industrial Text, 1988.
- 26 Politakis, Peter "Empirical Analysis for Expert Systems" PhD Thesis, Rutgers, 1985.
- 27 Politakis, P. and Weiss, S. "Using Empirical Analysis to Refine Expert System Knowledge Bases" Artificial Intelligence Journal 22 (1984) 23-48.
- 28 Geissman, James R. and Schultz, Roger D. "Verification and Validation of Expert Systems" AI Expert, February 1988, pp. 26-33.

- 29a Stachowitz, R.A., Chang, C.L. and Combs J.B. "Performance Evaluation of Knowledge Based System" Proceedings of the Third Aerospace Applications of Artificial Intelligence" Conference (AAAIC), Dayton, Ohio, Oct. 5-9, 1987 (Keynote Address).
- 29b Stachowitz, R.A., Chang, C.L., Stock, T.S. and Combs, J.B. "Building Validation Tools for Knowledge-Based Systems" Proceedings of the First Annual Workshop on Space Operations Automation and Robotics (SOAR), Houston, TX, Aug. 1987, NASA Conf. Pub. 2492 pp. 209-216.
- 29c Stachowitz, R. A., Combs, J.B. and Chang, C.L. "Validation of Knowledge-Based Systems:", Proceedings AIAA/NASA/USAF Symposium on Automation, Robotics and Advanced Computing, Arlington, VA, Mar. 1987
- 29d Stachowitz, A.H. Combs, J.B. "Validation of Expert Systems" Proceedings Twentieth Hawaii International Conference on System Sciences, Kona, Hawaii, Jan 1987
- 30 Goodwin, Mary Ann and Robertson, Charles C. (Rockwell) "Expert System Verification Concerns in an Operating Environment NASA SOAR Conference, August 1987, pp. 203-207
- 31 Castore, Glen (Honeywell) "A Formal Approach to Validation and Verification for Knowledge Based Control Systems". NASA/JSC SOAR Conference, August 1987, pp. 197-202.
- 32 Culbert, Chris, Riley, Gary and Savely, Robert T. (NASA) "Approaches to The Verification and Validation of Rule Based Expert Systems" NASA/JSC SOAR Conference, August 1987, pp. 191-196a.
- 33 Shapiro, E. "Algorithmic Program Debugging - Using Prolog" MIT Press.
- 34 Covington, Michael A., Nute, Donald, Vellino, Andre, "Prolog Programming in Depth" Scott, Foresman, 1988
- 35 Kanamori, Tadashi, et al (Mitsubishi) "Argus/V; A System for Verification of Prolog Programs" ICOT report TR-176, May 1986
- 36 Richardson, Keith and Wong, Carla, "Knowledge Based Systems Verification and Validation" Proceedings of the Second Annual Artificial Intelligence Research Forum, April 1987 NASA/Ames.
- 37 Groundwater, E.H. et al (Science Applications) "Approaches to the Verification and Validation of Expert Systems for Nuclear Power Plants" Electric Power Research Institute, EPRI NP 5236, July 1987
- 38a AI Corp - "Knowledge Base Management System" 617-890-8400
- 38b Knowledgware, Inc. - "Information Engineering Workbench" 404-231-8575

- 38c Perceptics (Westinghouse) AI Applications Division -
"Knowledgeshaper" - 615-966-9200
- 39 Texas Instruments "Personal Consultant Series" 1-800-527-3500
or in Europe (44) + (234) + 22 41 87
- 40 KDS Corp "Knowledge Delivery System - KDS 3.0" 312-251-2621
- 41 Patai, Saul "The Chemistry of Peroxides". Wiley, 1983
- 42 Bretherick, L. "Handbook of Reactive Chemical Hazards", Third
Edition, 1985. Butterworths, Stoneham, MA
- 43a Davies, C.A. (Dow), Kipnis, I.M. (FMC), Chase, M.W. (Dow) and
Treweek, D. N (consultant/Battelle), "Estimation of
Thermochemical and Hazard Data of Chemicals Using the ASTM
CHETAH (Chemical Thermodynamics and (Energy Release) Hazard
Evaluation). Proceedings of the spring 1984 meeting of ACS in
St. Louis. ACS Symposium Series 274 (Chemical Process Hazard
Review) PP 81-90 (1985)
- 43b Yoshida, Tadao "Safety of Reactive Chemicals" Industrial Safety
Series Vol. 1, Elsevier, 1987
- 43c Ducros, M and Sannier H. "Application du Programme Chetah a
L'Etude de la Sensibilisation de Composés Oxygénés et a
l'Estimation des Limites Inférieures d'Inflammabilité", Journal
of Hazardous Materials, 19 (1988) 33-49
- 44 Anon "Standard Practice for Calculation of Hazard Potential
Figures of Merit for Thermally Unstable Materials", ASTM, 1987
Standard E 1231
- 45 Anon "Standard for Storage of Organic Peroxides" NFPA, 1986
- 46 Anon "Recommendations on the Transport of Dangerous Goods -
Tests and Criteria". First edition of Addendum I, Part III
"Tests and Criteria for the Classification of Organic
Peroxides." United Nations Publication
- 47 Tewarson, A. "Generation of Heat and Chemical Compounds in
Fires". Factory Mutual Research Corporation, J.I. ON1E1.RC
(NS), March 1987
- 48a Anon "Organic Peroxides". Factory Mutual System Loss
Prevention Data 7-80, March 1972.
- 48b Anon, "Organic Peroxides Hazard Classification (tentative)"
Factory Mutual System, Loss Prevention Data 7-81, August 1986.
- 49 deKoning, J. et al "Storage of Organic Peroxides" CPR 3E,
Committee for The Prevention of Disasters Caused by Dangerous
Substances (CPR), First Edition 1982, from the Labour
Inspectorate, The Netherlands

APPENDIXAIChE MEETING AI PAPERSApril 1986 - New Orleans

An Expert System Approach to On-Line Optimization
An Expert System for Process Control Troubleshooting (Prolog)
Object-Oriented Integration in Process Engineering Computation
Dynamic Simulation in the Construction of Expert Systems for
Process Fault Diagnosis
PROVIAL - A Tool for Batch Process Design and Evaluation

November 1986 - Miami Beach

A Computer-Aided Laboratory For Artificial Intelligence Applications
in Chemical and Biochemical Engineering
Chemical Network Problems Solved on NASA/Goddard's Massively
Parallel Processor (MPP) Computer
Symbolic Computation and Chemical Engineering Model Development
An Overall Strategy for the Systematic Synthesis and Optimization
of Large-Scale Chemical Processing Systems
Synthesis of Solids Processing (Heuristics & Selection)
The Application of Automata and Formal Language Theory to
Chemical Process Control
Systematic Analysis of Chromatographic System for Protein
Purification (Knowledge-Based)
A Fuzzy Expert System for Separation Sequencing of Bioproducts
Computer Aided Modeling of Bacteria Cells: The Use of Expert
Systems
Prediction of the Dynamic Behavior of Temperature Sensitive
Recombinant Cells Using a Hybrid Heuristics and Mathematical
Model Based Expert System
An Expert Approach Towards State Estimation of Bioreactors
Stop Reinventing Pilot Plant Control Systems
Flexible Batch Processing Control System (Logic Block)
Artificial Intelligence - Emerging Science for Controls Systems
Applications
Self Learning Advisory Technology On the Design of a Block
Diagram Based Signal Processing and Simulation Package

Mar/Apr 1987 - Houston

An Intelligent System for the Design of Plant-Wide Control
Configuration
Design Kit: An Intelligent Interface and Database for Process
Engineering
The Development of (PIP Expert System) an Interactive User
Interface for Chemical Flowsheet Synthesis
The Falcon Project: An Application of an Expert System to Fault
Diagnosis
Narrowing Diagnostic Focus by Control System Decomposition
Model-Based Reasoning for Fault Diagnosis
An Expert System Approach to Diagnosis of Product Quality
Deviations
PRISIM - An Expert System for Process Risk Management
Analyzing Process Alarms When Your Expert System Fails

Disturbance Analysis Using Digraphs
Real-Time Diagnosis of Chemical Processes: A Model Base Algorithm
Optimal Logic for Multi-Channel Protective Systems During On-Line
Maintenance
Integrated Risk Assessment Program for Risk Management

August 87 - Minneapolis

The Use of Artificial Intelligence in Distributed Control
Model Based Reasoning Approach to Chemical Plant Design
Use of Expert Systems in Nuclear Power Plants
Application of AI to Management and Analysis Problems

November 1987 - New York

Heuristic Manipulation of Process Identification and Adaptive
Control Algorithms
POPS: The Prototype Operating Procedure Synthesis Program
Expert Multivariable Control
An Operator Advisor for Controlling Corrosion in a Crude
Fractionator
COMA: A Configurable Operator Monitor and Advisor Integrated
into a Real Time Control System
Knowledge-Based Real Time Sensor Interpretation for Process Plants
Operation Instruction System
Qualitative Modeling of Dynamic Systems
Heat Exchanger Network Synthesis: A Knowledge Engineering
Approach
An Expert System for Designing Distillation Column Plates
STES: A Separation Processes Expert System
RIP - A Prototype Expert System for Retrofitting Chemical Plants
Design of Polymer Composites: A Blackboard Approach
POPS: The Prototype Operating Procedure Synthesis Program
Operator-Assisted Learning in Expert Systems for Fault Diagnosis
A Connectionist Expert System Approach to Fault Diagnosis in the
Presence of Noise and Redundancy
Expert System in a Wastewater Treatment Process Diagnosis
An Operator Aid for Analysis of Disturbances in Distillation
Columns
MOLDOCTOR: An Expert System for Fault Diagnosis and Remedies in
Injection Molding of Plastics
A Method of Fault Diagnosis: Presentation of a Deep-Knowledge
System
Real Time Hazard Aversion and Fault Detection: Multiple Loop
Control Example
Overview of Artificial Intelligence
The Expert System for Chemistry: Planning Multistep Syntheses
and Conversion of Polymeric and Hazardous Chemical Wastes to
Useful Materials
The Use of Expert Systems in Negative Manufacturing
Qualitative Simulation of Continuous Chemical Processes
The Use of Expert Systems in Process Design of Offshore Oil and
Gas Production Systems
Application of Expert Systems to Environmental Problems
Cellular Metabolism: An Approach Through Artificial Intelligence
An Intelligent System for the Design of Biochemical Pathways

Expert Systems, Metabolism and Pharmacokinetics
Strate Gene, An Expert System for Use in Molecular Biology
An Expert System Based Intelligent Control Scheme for Bioreactors
An Intelligent System for the Design of Protein Recovery Processes

March 1988 - New Orleans

Symbolic Computing in Process Flowsheeting
Implementation of an Expert System for On-Line Fault Diagnosis in
a Commercial Scale Chemical Process
Impact and Requirements of Knowledge-Based Software
Expert Systems in the Pulp and Paper Industry
ICI Process Engineering Expert Systems - Present and Future
Development of a Diagnostic Expert System for a Whipped Toppings
Process
Experience Using an AI Language and an Expert System Shell for
Developing "BUS1"
Process Scheduling Using Artificial Intelligence
To AI or not to AI, That is the Question
Expert Systems for Control of Plant Operations in Real Time
Using Expert Systems to Monitor Safety Margins for High Hazard
Processes
A Natural Language Interface to Relational Databases for Waste
Management
Intelligent Computer-Assisted Instruction Programs

August 1988 - Denver

Artificial Intelligence at Wang Laboratories
A Natural Language Interface to Relational Databases for Waste
Management
Expert Systems "To Go": Laptop Personal Computers & KB Software
An Application of Object Oriented Programming To Process
Simulation
Hands-On Demonstration of AI Programs

RECEIVED June 9, 1989

Chapter 11

ERICE

Corrosion Rate Monitoring and Diagnosis in the Heating Circuits of Nuclear Power Plants

A. G. Federico, C. Ronchetti, and A. Tripi

ENEA Thermal Nuclear Department, V. Anguillarese 301, 00060 Rome, Italy

The corrosion of the metallic materials, is one of the most damaging factors that destroy economical and ecological resources, and many problems encountered in thermal plants for electrical power production are related to corrosion. In order to improve results in controlling corrosion, the authors decided to build an expert system, which performs on-line corrosion rate monitoring by means of a number of probes connected to an automatic corosimeter, evaluates expected corrosion rate values and behaviours. If there are discrepancies, the expert system performs a diagnosis, providing suggestions to overcome the difficulty and to execute the right water treatments.

The corrosion of the metallic materials, which are largely used to build industrial plants, is turning out to be one of the most damaging factors that destroy economical and ecological resources.

Indeed, it wastes about 3-5% of Gross National Product in industrialized countries, and also diminishes the availability of raw materials, both directly, through the destruction of strategic materials (such as iron, chromium, nickel, copper etc.) and indirectly, through the amount of energy necessary for obtaining such materials, starting from their minerals.

Other negative aspects to be taken into account are the decrease of efficiency, the standstill for maintenance and the reduction of plant reliability.

There is a worldwide effort to highlight the corrosion and to prevent it. Such effort is characterized by two fundamental objectives:

- to reduce the amount of the corrosion through the improvement of the alloys and of the active and passive protection treatments, and through the use of more effective control systems, which should be able to restrain uncontrolled corrosion;
- to develop new alloys, possibly characterized by low amounts of high quality metals, which can stand up corrosion better.

0097-6156/89/0408-0147\$06.00/0

© 1989 American Chemical Society

American Chemical Society

Library

1155 16th St., N.W.

Washington, D.C. 20036

The Economic European Community is also actively pursuing these goals, through the BRITE project, which encourages the cooperation of the research centers of various countries, in order to obtain pre-competitive projects.

An interesting example of this large and complex project is the one regarding thermal and thermo-electrical plants. This project, through constant improvements of the design, the materials, the water chemistry and the monitoring systems, induces a continuing improvement of the plant performances.

Problems Induced by Corrosion Phenomena

Many problems encountered during the operation of thermal plants for electrical power production are related to corrosion in a more or less direct way.

The water-metal interaction produces a series of processes that cause the metal oxidation and the consequent diffusion into the water in different chemical states. The result can be the reduction of the reliability and of the availability of the plant. These processes are based on electrochemical laws, which are clearly related to the water chemistry and to the material properties.

With regard to the consequences on the plant, it is possible to divide the corrosion processes in two broad classes:

- processes which deteriorate the structural integrity of the components
- processes which reduce the efficiency of the plant.

The former class includes all the processes of localized corrosion (stress corrosion, pitting, fatigue corrosion, etc.) and some cases of generalized corrosion.

The latter class includes all the processes of generalized corrosion which give rise to the formation of insoluble corrosion products (crud). These products can either accumulate in some areas of the plant which are subject to particular thermo-hydraulic conditions (heat transfer, low water circulation etc.), or else they can be included in oxide layers growing on other surfaces.

Evolution of Corrosion Monitoring Systems

Recently, the management of large nuclear and fossil fuel thermal power plants have stimulated the development of diagnostic systems, in order to improve primary coolant quality. Such improvements constitute indirect tools for increasing the plant reliability and safety, both depending on the corrosion of the structural materials.

At the same time, systems for the direct control of corrosion have been developed, as well as systems for measuring parameters directly related to the corrosion itself (redox and corrosion potentials).

With regard to chemical and physical-chemical specifications (contents of additives and impurities, conductivity, acid conductivity and pH) more and more sophisticated, reliable and sensitive systems have been developed [1].

Investigations into the reliability of a system as a whole were also performed. The topics which were investigated regard the localization of sampling points, the probe shapes, the sizes and the characteristics of the sampling lines [2].

More recently, there was a tendency to perform electrochemical measurements (free corrosion and redox potentials) directly on the plant. Such parameters integrate the effects of many chemical and physiochemical characteristics, and it is also possible to correlate them to the behaviour of the materials with respect to corrosion.

Since 1973 [3] and 1975 in Dresden 2 [4], on line measurements of redox potential have been performed in the BWRs' (Boiling Water Reactors) primary coolant in order to verify the IGSCC (Intergranular Stress Corrosion Cracking) susceptibility of austenitic stainless steel components. Such measurements turned out to be fundamental for estimating the effect of hydrogen addition as a remedy for IGSCC [5-6].

Later, it was proposed to use redox potential for monitoring the materials of the primary circuit of PWRs as well [7].

In 1977 it was proposed to use analogous measurements for monitoring the feed-water line of BWRs starting from measurements carried out in fossil power plants [8]. This measurement enables one to control both low concentrations of oxygen (which are indicative of a fault in the degassing system of the steam condenser) and addition of 50 to 100 ppb of oxygen (which is used for reducing the production of iron oxides).

Such measurements, however, constitute only indirect information about material corrosion, while the direct measurements of corrosion rates give straight information about anomalous behaviours of the structural materials.

The direct measurement of corrosion was formerly obtained by inserting material specimens in autoclaves connected with the primary circuit. This method requires periodic weighings performing a visual controls and metallographic investigations.

More recently, sophisticated SSRT (Slow Strain Rate Technique) equipment has been installed on some BWRs, in order to measure the tendency to IGSCC of some sensitized austenitic steels [6]. These methods have lately been also used for directly measuring crack propagation rate in TC samples [5].

Systems able to supply in situ monitoring of the corrosion rate were developed as well in the recent years [9]. Such systems make use of different methodologies, and consist of a probe inserted in the plant component, and of an instrument which converts the probe signal into a corrosion rate measurement.

It turns out that the reliability of the corrosion rate measurements depends on the probe and instrument performance, on a good understanding of the phenomenon (as far as materials and chemistry are concerned) and on a good knowledge of the plant operating conditions.

The compliance with the hydraulic conditions on the electrode surfaces, which is impossible by the usual three-rod probe, becomes possible with tubular electrodes. Flush mounted probes are convenient to perform measurements on large components, such as vessels, tanks, heat exchangers etc.

The problem of high water resistivity is overcome by means of advanced commercial corrosimeters and computer aided design of the probes, thanks to the availability of finite element codes. The electric problems, such as streaming potentials and

noise, related to the mentioned high water resistivity, are solved by the geometry of the probe, and by proper electronic adjustment at the corrosimeter inlet. Temperature and pressure problems are overcome by a suitable insulator and gasket material.

These studies allowed the development of corrosion probes with high sensitivity and reliability. They have been applied on steam condensers [10] and cooling water circuits [11].

In the case of the steam condensers, the probes, installed on the cooling water side, controlled the behaviour of the tube bundle materials. The most interesting applications regarded the procedures for the passivation of copper alloys even in polluted water conditions, and the cleaning of tube bundles.

With regard to heat exchange circuits, probes which operate in experimental circuits at temperatures of 250 C have been developed. Other probes which will operate at even higher temperatures are under construction.

The measuring equipment (corrosimeters) have been improved as well, using microprocessors and programmable memories which can correct the potential drop in the water and verify the reliability of the probe.

ENEA (Comitato Nazionale per la Ricerca e per lo Sviluppo dell'Energia Nucleare e delle Energie Alternative) in cooperation with CISE (Centro Informazione Studi e Esperienze) developed an advanced automatic corrosimeter for monitoring corrosion in power plants, consisting of two modules: the first one acquires data from the probes, and the second one (an IBM personal computer) controls the first module and performs all the required calculations. Several probes can be connected to a single corrosimeter [12].

In order to simplify and improve the use of the corrosimeter, an expert system has been installed on the corrosimeter itself, for controlling the corrosion rates measured by any single probe, and for verifying that the values, the trends and the transients of such corrosion rates are not anomalous. This expert system is called ERICE (Expert Reasoning Instrument for Corrosive Environments).

Anomalous behaviours of the corrosion rates mean that something is going wrong, either with the plant or with the corrosimeter itself. If this is the case, ERICE makes a diagnosis, finding the cause of the anomalous behaviour. Once the diagnosis is performed, it advises the operator about the actions to be performed to overcome the difficulty.

Design of a Model Emulating the Expert Reasoning Behaviour

Writing a computer program which is able to reproduce the reasoning of a human expert in a specific domain offers several remarkable advantages. They are the availability of such experience in case of non availability of the human expert, the formalization of the expertise in the domain, the standardization of the decisions taken under the same conditions, the availability of the expertise even in the case the expert is missing (e.g. because of retirement), etc.

The use of expert systems [13-14] allows the solution of problems which lack reliable mathematical models and are characterized by a great amount of empirical knowledge acquired in heterogeneous ways and by a high level of abstraction. The study of corrosion phenomena has such characteristics and is a very promising field for the realization of expert systems.

Among the knowledge representation methods, the one based on production rules is probably the most widely used. Anyway a production rule is only the smaller indivisible part of a Knowledge Base relative to an application domain.

As the reasoning of a human expert is goal-oriented and driven by the external situations, so the organization and the correct instantiation of the production rules inside a Knowledge Base depend on the system goal and on the external constraints.

A Knowledge Base system emulating the human reasoning must contain rules of at least two different grades: rules of low grade emulate the management of the conceptual objects relative to the application domain and an rules of high grade (metarules) emulate the reasoning in terms of strategies to be used to reach the desired goals and to realize the required functions. Usually metarules control the correct firing of the lower grade rules during the inference execution.

The definition of the higher grade Knowledge Base structure is the primary goal during the Knowledge Elicitation phase.

The knowledge of the human expert is characterized by an high degree of intuition and a low level of formalization and abstraction: starting from some elements acquired from the interviews and from documents analyses, avoiding to examine exhaustively all the application domain content, the knowledge engineer must build a general model of the expert reasoning [15] that can be applied to the real situations.

The definition of this model will be used to design the best higher grade Knowledge Base structure. If this structure matches with a commercial expert system developing tool characteristics, it can be selected and used allowing a minor effort to build the final system.

In this case Personal Consultant Plus shell, installed on an IBM PC and realized by TEXAS INSTRUMENT, was selected as developing tool to build the ERICE expert system. This shell allows the possibility to group the production rules in different knowledge base elements named units: a unit contains a base of rules, a set of relevant parameters and is associated to a goal that must be solved. More units can be connected together realizing a tree composed by parent and descendant units: the descendant unit inherit the parameters of the parent units. When a unit is instantiated it tries to solve its goal using its own rules: if not, it search a descendant unit that has this capability and, when found, it is instantiated.

In other words this type of shell allows the object oriented programming style [16] to be used for building the complete rule based system.

Knowledge Base Structure and System Functionalities

The ERICE system Knowledge Base is composed at top-level by a parent unit (CORROSION) as shown in Figure 1 and by three descendant units (FORECAST, CHECK and DIAGNOSIS). The CORROSION unit has the goal to perform a complete cognitive process relative to forecasting and checking the correct/incorrect corrosion rate.

To execute this job it activates external software routines acquiring the corrosion rate from the corrosimeter and

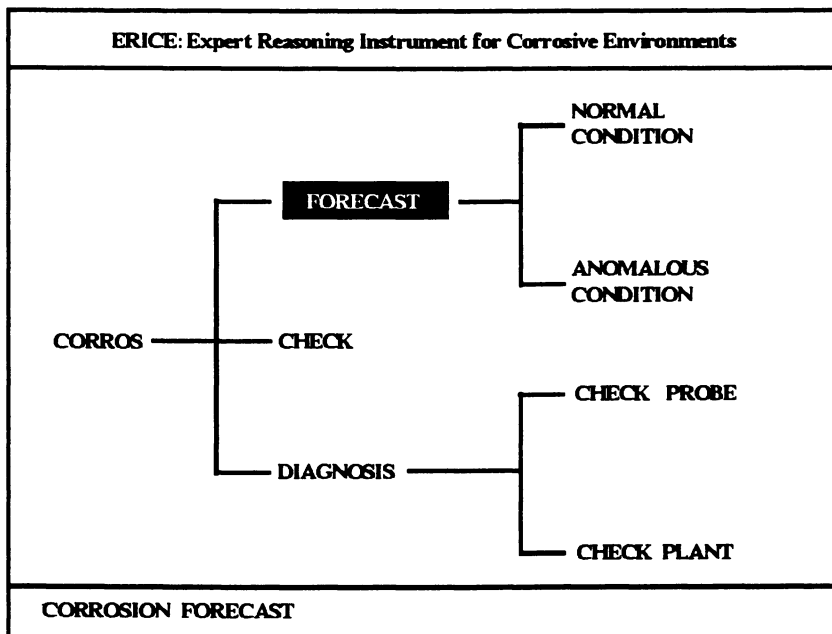


FIGURE 1. ERICE Knowledge Base tree structure.

ERICE: PLANT OPERATING CONDITIONS UPDATING SYSTEM												
<p style="text-align: center;">MAIN MENU</p> <p>Date (D-M-Y): 18-3-1988</p> <p>Hour (hh:mm): 15:00</p> <p style="text-align: center;">CONDITION</p> <p style="text-align: center;">SUBCONDITION</p> <p style="text-align: center;">MALFUNCTIONS</p> <p style="text-align: center;">OTHER PARAMETERS</p>	<p style="text-align: center;">Condition and previous condition</p> <p>BUILD-UP</p> <p>STAND-BY</p> <p style="text-align: center;">PASSIVATION</p> <p>NORMAL OPERATION</p> <p style="text-align: center;">RINSING</p> <p style="text-align: center;">Subcondition</p> <p style="text-align: center;">BEGINNING</p> <p>HEATING</p> <p>REGIME</p> <p>COOLING</p>	<p style="text-align: center;">Malfunctions</p> <table border="0"> <tr> <td>FAULT PROBE</td> <td style="text-align: right;">NO</td> </tr> <tr> <td>CORROS. OUT-OF-SERVICE</td> <td style="text-align: right;">NO</td> </tr> <tr> <td>AIR IMLET</td> <td style="text-align: right;">YES</td> </tr> <tr> <td>IONIC POLLUTION IMLET</td> <td style="text-align: right;">NO</td> </tr> <tr> <td>SOLID POLLUTION IMLET</td> <td style="text-align: right;">NO</td> </tr> </table> <p style="text-align: center;">Other parameters</p> <p>TEMPERATURE (C): 48</p> <p>CONDUCTIVITY:</p> <p>ACID CONDUCTIVITY:</p> <p>pH:</p>	FAULT PROBE	NO	CORROS. OUT-OF-SERVICE	NO	AIR IMLET	YES	IONIC POLLUTION IMLET	NO	SOLID POLLUTION IMLET	NO
FAULT PROBE	NO											
CORROS. OUT-OF-SERVICE	NO											
AIR IMLET	YES											
IONIC POLLUTION IMLET	NO											
SOLID POLLUTION IMLET	NO											

FIGURE 2. Plant Data Base multi-windows updating system.

performing some required elaborations (evaluation of fast/slow trends in the measurements). At the same time informations regarding the actual and the previous status of the plant are acquired from an external Data Base that the operator updates during the plant operations by means of the multi-windows interface visualized in Figure 2. The interface displays the actual time and date, the current plant condition and subcondition, the previous plant status and the detected plant malfunctions.

Then, utilizing these data, the FORECAST unit is instantiated to evaluate the forecasting range of corrosion rate values.

Obviously the forecasting corrosion rate value will be different if the plant was operated in normal or anomalous chemical conditions: the two different evaluations are executed by the units NORMAL CONDITION and ANOMALOUS CONDITION respectively.

After the forecasting process the CHECK unit is instantiated to compare the estimated corrosion rate values with the measured one: if a match of the values occurs the current goal of the system is reached and the expert system waits until a new corrosion rate value is available. Otherwise the DIAGNOSIS units is instantiated.

The DIAGNOSIS unit considers firstly the possibility of a fault in the probe: the check includes both visual controls by the operator and statistical elaborations on the data acquired by the corrosimeter.

If it establishes that the corrosimeter is properly working, a diagnosis is performed about the plant. At this point the instantiated knowledge unit hypothesizes with different credibility values the possible causes of the anomalous corrosion behaviour: in Figure 3 an example of two different hypothesizing rules is showed. The generated credibility values depend only on the malfunction historical probability based on statistical and design considerations; it will be used to determine the best order to follow in verifying

RULE 071 [SLOW-INCREASE-RULES]

**IF CONDITION = STAND-BY AND SLOW-INCREASE-CORROSION
THEN AIR-INLET = GREAT-QUANTITY CF=20**

RULE 087 [SLOW-INCREASE-RULES]

**IF CONDITION = PASSIVATION AND
SLOW-INCREASE-CORROSION
THEN IONIC-POLLUTION-INLET = SMALL-QUANTITY CF=10**

FIGURE 3. Examples of hypothesizing production rules

the hypotheses. These are verified, using a backward-chaining reasoning strategy, trying to find out from the Plant Data Base the facts able to increase or decrease the credibility values (hypotheses verifications). PERSONAL CONSULTANT PLUS shell uses a method to propagate the credibility values during the inference execution that is quite similar to the method used by MYCIN expert system [17-18].

In addition it was established to divide the verifications in three different groups to be performed in the following order:

- 1) the more simple ones obtained checking the plant conditions and by means of on line measurements evaluation;
- 2) the chemical-physical measurements;
- 3) the more complex verifications obtained by means of chemical analyses.

In this way the order of verifications depends also on the time to be spent to perform the verifications itself: the verification process stops when a single malfunction is identified with a sufficient credibility value or when all the verifications are already performed. Naturally if ERICE is not able to identify only one possible malfunction, more malfunctions will be suggested as candidates to the operator with different credibility values.

If, during the verification process, ERICE needs the value of a parameter that is not contained in the Plant Data Base, it asks the operator for such value.

First Applications

The ERICE applications currently under development at ENEA refer to the management of power plant standard chemical treatments (acid cleaning, passivation, additive treatments, etc.), and to the plant operation. Since these applications demand the specific knowledge for each plant to be monitored, they will be developed in close cooperation with firms active in the specific sectors.

The whole system for corrosion rate monitoring has been operating in a plant for testing PWR primary pumps, developed by ENEA and FIAT. The system comprises :

- 16 probes, 8 of which tubular, 6 flush mounted and 2 projecting;
- an automatic corrosimeter, ENEA-CISE model 1200;
- the ERICE prototype.

The first version of ERICE was tailored for this plant. The information peculiar to the plant with which the expert system was supplied consists of material properties, water chemistry specifications and operating conditions for different circuits, etc.

The plant phases that were taken into account are: passivation, stand-by, acid cleaning and steady state. The passivation is a treatment that consists of causing the formation of a very compact layer of oxide on a clean metallic surface. To perform such treatment, the circuit is heated uniformly, then a forced circulation of water with adequate additives is executed. When the passivation is correctly

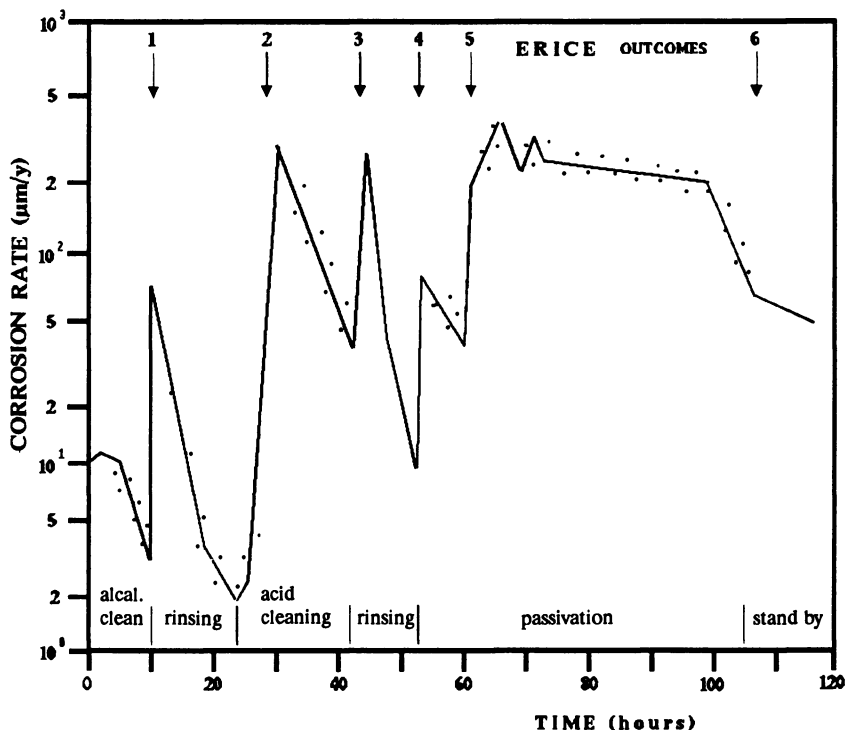


FIGURE 4. Corrosion rate behaviour during different plant conditions.

accomplished, the resulting layer of oxide protects the underlying metallic material, preventing a further formation of oxide.

Stand-by consists of keeping a plant out of service, in such a way that corrosion of plant materials is kept as low as possible. It can be accomplished either by supplying appropriate additives to the water, or by emptying the circuit and filling it with inert gases.

Acid cleaning consists in using very aggressive additives for cleaning metallic surfaces of any trace of oxide. When an acid cleaning is performed, the previous history of the plant becomes irrelevant. It can be useful in order to fix the consequences of an incorrect handling of the plant, but conversely it also destroys a layer of metallic material, weakening the strength of the structure. For this reason acid cleaning should be performed only few times during the life of a plant, and in the case of nuclear plants, it usually should not be performed at all for safety reason.

Beside these principal phases, a few supplementary phases are considered, for example, alkaline cleaning, a variant of acid cleaning, or rinsing, a minor phase following acid cleaning.

An interesting example of ERICE performance during a period of plant operation is shown in Figure 4. It shows the

corrosion rate behaviour during alkaline and acid cleaning, passivation, out of service and transient phases of one of the plant circuits. During that period some interesting situations happened, some of which were not expected by the operator.

As it can be seen, there have been six significant ERICE outcomes:

- 1 - ERICE detected an increase in corrosion rate, during rinsing after an alkaline cleaning. It found the cause, consisting of the absence of a proper inhibitor, and updated the database.
- 2 - ERICE detected an increase in corrosion rate at the beginning of an acid cleaning. Since the database had been previously updated by the operator, it just indicated the beginning of acid cleaning.
- 3 - ERICE detected an increase in corrosion rate during rinsing after an acid cleaning. It found the cause, which consisted of the absence of a proper inhibitor, and updated the database.
- 4 - ERICE detected an increase in corrosion rate at the beginning of a passivation in reducing conditions. The database had been previously updated by the operator, and therefore ERICE just indicated the beginning of passivation.
- 5 - ERICE detected an increase in corrosion rate during a passivation. It inferred the possible causes, asking the operator for the necessary checks and information. At the end ERICE identified the cause as an air inlet, and asked the operator whether the plant could operate in the new conditions. Since this was the case, it updated the database.
- 6 - ERICE detected a high corrosion rate at the end of the passivation, but such rate was considered reasonable, since the system took into account the history of the plant, which had been previously updated (air inlet). It indicated that the plant was operating in bad corrosion conditions, since the passivation treatment was handled improperly.

Conclusions and Future Developments

The use of an on line corrosion monitoring system in power plants is currently possible due to the marked technological improvements in both probes and measuring instruments. The addition of an Expert System provides the operator with easy to use informations.

Such informations can be correlated with those coming from standard chemical monitoring systems (conductivity, acid conductivity, pH, redox potential, chlorine, sodium and oxygen concentration, etc.).

The first version of ERICE applied to the CPPP plant was strictly tailored for this plant.

The future development which we are considering is the development of a general Expert System, accessing very large corrosion Data Bases, that could be applied to different plants with little effort, and able to effectively manage nearly every situation of generalized corrosion.

To perform an effective management of both direct and indirect corrosion effects, this system should be integrated with a water chemistry monitoring Expert System.

Literature Cited

1. R. Rizzi, C. Ronchetti, F. Sigon and G. Quadri. Paper n. 66, Fourth BNES Conference on Water Chemistry of Nuclear Reactor System, Bournemouth, 1986.
2. L. Meini, R. Rizzi, C. Ronchetti and A. Perugini. Campionamento dei liquidi di processo, elaborazione di alcuni concetti di base per la stesura della guida al progetto. CISE-NT 79.092 (1979).
3. R.L. Cowan, A.I. Kaznoff. Electrochemical measurements of corrosion processes in a Boiling Water Nuclear Reactor. Corrosion Vol. 29 (4), 123, (1973).
4. M.E. Indig and R.L. Cowan. Paper n. 15, Second BNES Conference on Water Chemistry of Nuclear Reactor System, Bournemouth, 1980.
5. E. Hallden and T. Rooth. Paper n. 16, Fourth BNES Conference on Water Chemistry of Nuclear Reactor System, Bournemouth, 1986.
6. B.M. Gordon, R.L. Cowan, C.W. Jewett and A.E. Pickett. Mitigation of SCC through Suppression of Radiolytic Oxygen. Environmental Degradation of Materials in NPSWR. Myrtle Beach, 1983.
7. A. Molander, B. Rosborg, J. Svenson and P.O. Andersson. Paper n. 48, Fourth BNES Conference on Water Chemistry of Nuclear Reactor System, Bournemouth, 1986.
8. M.L. Binecchio, F. Carassiti, R. Cigna and G. Sturla. Paper n. 20, First BNES Conference on Water Chemistry of Nuclear Reactor System, Bournemouth, 1977.
9. R. Rizzi, C. Ronchetti. On-line Monitoring of the True Corrosion Rate in Problems Related to Power Plants. ASTM STP 908.
10. G. Buzzanca, E. Diacci and P. Ravani. Cleaning by Sand Immission of Steam Condenser Tubes in the Caorso Nuclear Power Plant. Int. Meet. on Nuclear Power Plant Operation. Chicago, 1987.
11. R. Rizzi and C. Ronchetti. Carbon and Low-Alloy Steel Corrosion in the CIRENE Prototype Reference Conditions, Third BNES Conference on Water Chemistry of Nuclear Reactor System, Bournemouth, 1983.
12. C. Ronchetti, F. Vaghi. Automatic Corrosimeter. 2nd ICOSM, Venice, 1986.
13. F. Hayes-Roth, D.A. Waterman and D.B. Lenat. Building Expert Systems. Addison-Wesley, 1983.
14. A. Hart. Knowledge Acquisition for Expert Systems. Kogan Page, 1986.
15. P.C. Cacciabue, G. Mancini, G. Guida, "A Knowledge Based Approach to Modelling the Mental Processes of a Nuclear Power Plant Operator", Paper n. 94, IAEA Conference on MMI Interface in Nuclear Industry, Tokio, 1988.
16. S. Danforth, C. Tomlinson, Type Theories and Object-Oriented Programming, ACM Computing Surveys, Vol. 20, March 1988
17. E.H. Shortliffe and B.G. Buchanan, "A Model of Inexact Reasoning in Medicine", Mathematical Biosciences, Vol 23, 1975
18. E.H. Shortliffe, Computer-Based Medical Consultation: Mycin, Elsevier, New York, 1976

RECEIVED June 9, 1989

Chapter 12

WOOLY: Wool Dyeing

Henderson Azevedo, Jr., and Gertrude Frei

Sandoz Limited, Methods and Models, 210733, CH-4002 Basel, Switzerland

WOOLY, a marketing expert system for wool dyeing is presented. WOOLY is fully operable. It helps the dyer with his daily work of finding adequate dyestuffs for specific quality requirements. The system assists in all steps of the wool dyeing process providing the user with information concerning several topics in the field of wool dyeing, as well as calling his attention to the points he must consider with special care.

WOOLY is a fully operable expert system in the area of wool dyeing. It was developed at Sandoz Ltd Switzerland for the dyestuff departments of the parent company and its subsidiaries as well as for external technicians and clients of Sandoz Ltd. The system supports 5 languages (German, English, French, Italian and Spanish) and is now in use all over the world.

The process of wool dyeing consists of several steps. To obtain the desired color is just one single step in this process. In each step many parameters have to be considered, and even for specialists it is a task that requires time and a considerable amount of effort. A specific education and years of experience are necessary to become a wool dyeing expert, and this kind of specialist is scarce.

The user of the system should not be a novice in the field of dyeing. He must have some knowledge about dyeing but he does not need to be a wool dyeing specialist. WOOLY is intended to help the dyer with his daily work. It will help him to choose in every

0097-6156/89/0408-0158\$06.00/0

© 1989 American Chemical Society

circumstances the right dyestuff and it will inform him about the critical points.

The Wool Dyeing Procedure

A schematic representation of the wool dyeing process is given in figure 1. On the basic screen output of WOOLY (figure 2) the schematic representation of the wool dyeing process is shown in the upper part of the screen. For each item the possible options are shown in a menu at right.

For the wool dyeing procedure it is necessary to know the article (for instance carpet or clothes) that will be produced as well as its care level (dry cleaning, hand washable or machine washable). For each article and care level, some quality properties (enduse-fastness-requirements) have to be achieved. They can be given by international standards or fixed by the customer of the dye company in order to meet his own quality requirements. Standard requirements for hand knitting yarn with care level hand wash are shown in figure 3 (maximum = 5, for Xeno maximum = 8).

The dyeing process is only one step within a sequence of operations, for instance washing, a specific chemical treatment, etc. The dyer chooses this sequence according to the available raw material, his own equipment and experience. On the one hand this sequence of operations determines the type of substrate (untreated, chlorinated or hercosett wool), which is an important parameter for the fastness level of a certain dyestuff. (Fastnesses are quantified quality attributes like resistance to water under different temperatures or light exposure.) On the other hand the operations executed after the dyeing process should not damage the shade. This requires specific quality properties (processing-fastness-requirements) of the dyestuff.

Dyestuffs with similar fastness-properties are grouped into classes. Knowing the requirements and the substrate, the class of dyestuff can be chosen. The desired color can be achieved by combining dyestuffs within one of these classes. The existing programs to calculate recipes can produce (in a time-consuming process) a lot of combinations that yield the right color, but they do not care about the fastnesses that are achieved by a recipe. To restrict the number of possible recipes as well as to ensure the necessary quality, a preselection of dyestuffs according to the specific requirements must be made. WOOLY can provide the user with all the necessary information about the dyestuffs, and it can help the dyer to make a reasonable preselection of dyestuffs. The diagram in figure 4 shows the range of shade depth (minimum = 0, maximum = 4) where each dyestuff in a class meets all

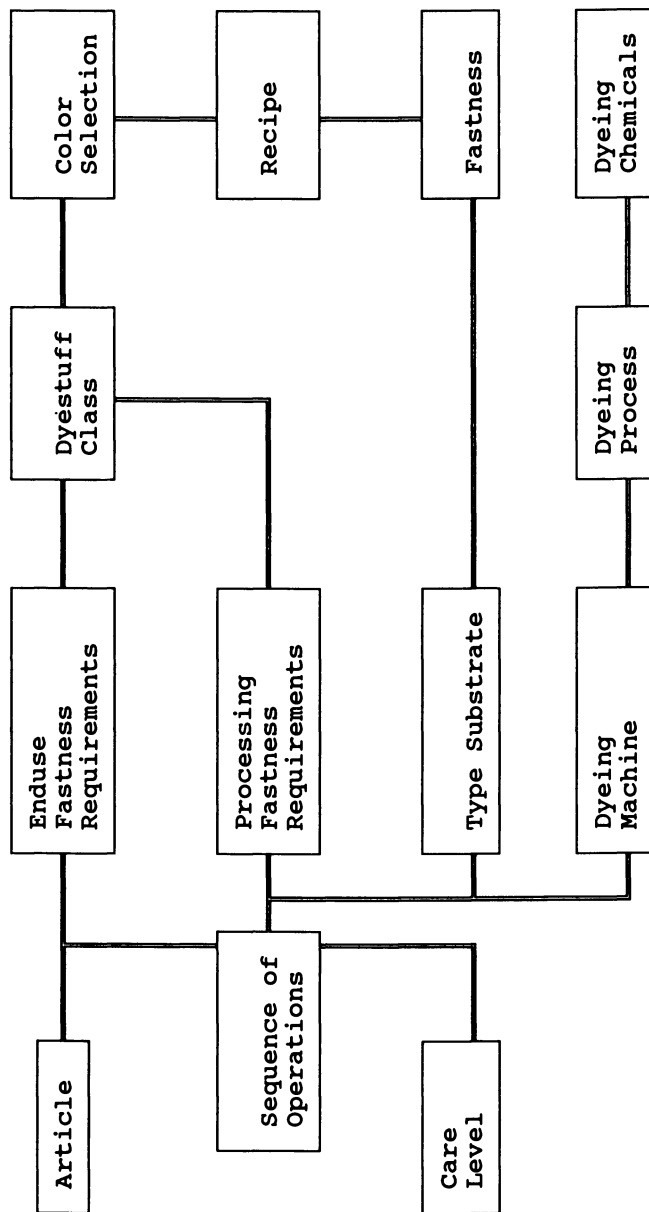


Figure 1. Schematic representation of the wool dyeing process.

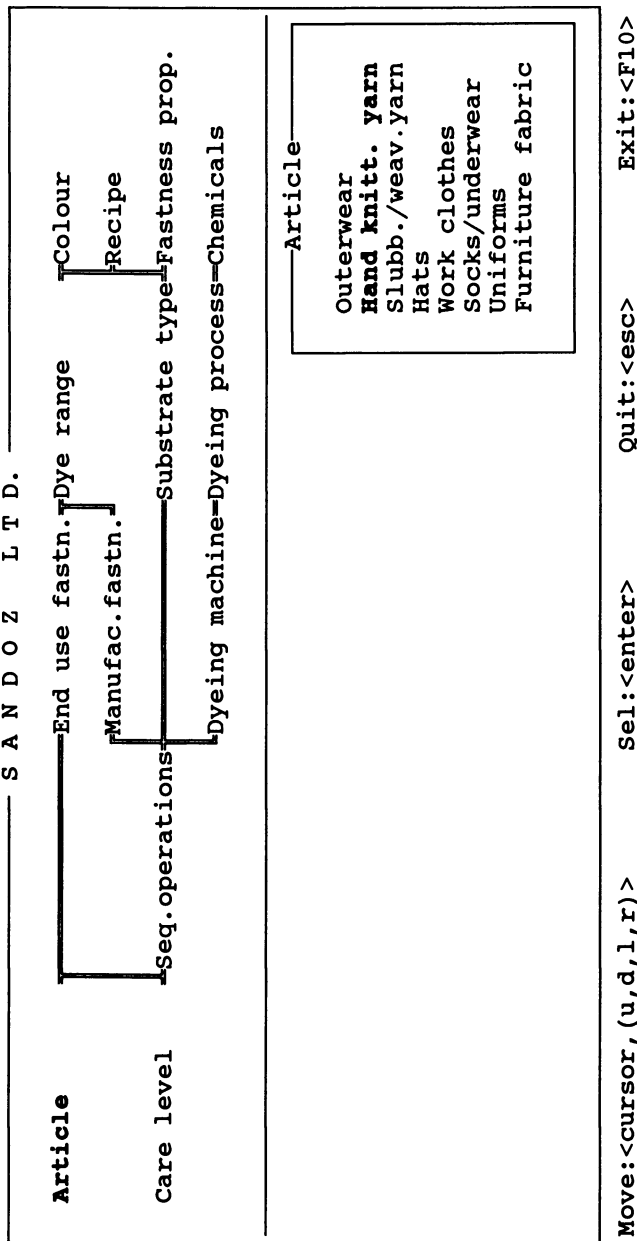


Figure 2. Main screen output of the WOOLY expert system.

S A N D O Z L T D.

Article	Hand knitt. yarn	End use fastn. Standard
Care level	Hand wash	
Fastness	Test	DE Wo Co/PA
Dry cleaning perchlorethylene	SN-ISO D01	4 3.5 3.5
Perspiration alkaline	SN-ISO E04	4 3.5 3.5
Rubbing dry	SN-ISO X12	3 3.5 3.5
Rubbing wet	SN-ISO X12	3 3.5 3.5
Water severe	SN-ISO E01	3.5 3.5 3.5
Xeno <1/12 SD	ISO-B02	3 3.5 3.5
Xeno >1/12 SD	ISO-B02	4 3.5 3.5

Move: <cursor,pgup,pgdn>

Quit: <F10,esc>

Figure 3. Standard enduse fastness requirements for hand knitting yarn.

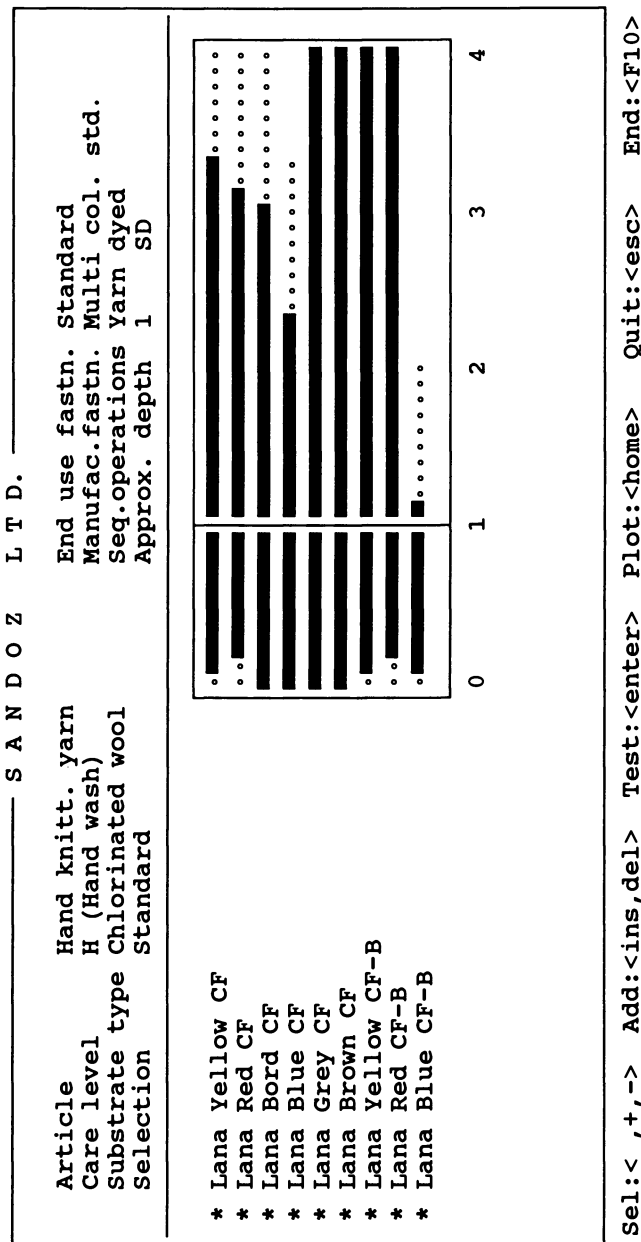


Figure 4. Evaluation of a class of dyestuffs for given fastness requirements.

standard requirements for hand knitting yarn (solid bar), and where minor problems occur (dotted line).

The properties of the resulting recipes must then be compared with the quality requirements so that a suitable one can be chosen. In figure 5, for the fastness perspiration alkaline, the values achieved for a recipe are compared with the requirements.

Some other factors of influence for the wool dyeing process are the used dyeing machine, the applied dyeing process as well as the added dyeing auxiliaries (chemicals). WOOLY informs about suitable dyeing processes, possible dyeing machines and dyeing auxiliaries. Figure 6 for example shows the temperature-time diagram for a dyeing process.

Characteristics of WOOLY

WOOLY includes information about the requirements of 15 groups of articles, depending on the care level and on the sequence of operations that are made during the fabrication process.

There is data about 2 classes of dyestuff (with about 20 elements each) on 3 different types of substrate.

WOOLY takes 34 fastnesses into account. For each dyestuff, each type of substrate and each fastness, there is a model estimated to analyze the behavior of the fastness depending on the concentration of the dyestuff and to describe the fastness for a given recipe.

The system runs on every IBM compatible personal computer (with a minimum of 512 kB RAM). It consists of 15 different programs (most of them written in Turbo Prolog) that call each other mutually and of eight databases containing the data about the considered products and the standard values of the requirements. The user can build up a database containing his own requirements, sequence of operations and recipes. The system is fully flexible. At any time the user can inform himself about all the different topics and he can reset the relevant variables according to his own preferences and needs. The current process can be saved and restarted at any state.

Types of knowledge

In the considered area of wool dyeing an expert has different forms of knowledge. Therefore, an expert system cannot be based on a single form of knowledge representation. The following examples are intended to point out the complexity of the problem domain as well as the different approaches that are implemented in WOOLY to cope with this variety.

S A N D O Z		L T D.	
Article	Hand knitt. yarn	Lana Yellow CF-B	0.123 %
Care level	H (Hand wash)	Lana Yellow CF	0.893 %
Seq. operations	Yarn dyed	Lana Blue CF	0.098 %
End use fastn.	Standard		
Manufac.fastn.	Multi col. std.	L= 49 C= 38.1 H= 90.4	
Substrate type	Chlorinated wool	Approx. depth 1 SD	
Perspiration alkaline SN-ISO E04			
Dyestuff	Weight	DE	Wo
Lanasan Yellow CF-B	0.07	4.4	4.5
Lanasan Yellow CF	0.83	4.7	3.9
Lanasan Blue CF	0.1	4.6	4.2
Recipe		4.7	4
Requirements		4	3.5
Hit any key to continue			

Figure 5. The perspiration alkaline fastness for a given recipe.

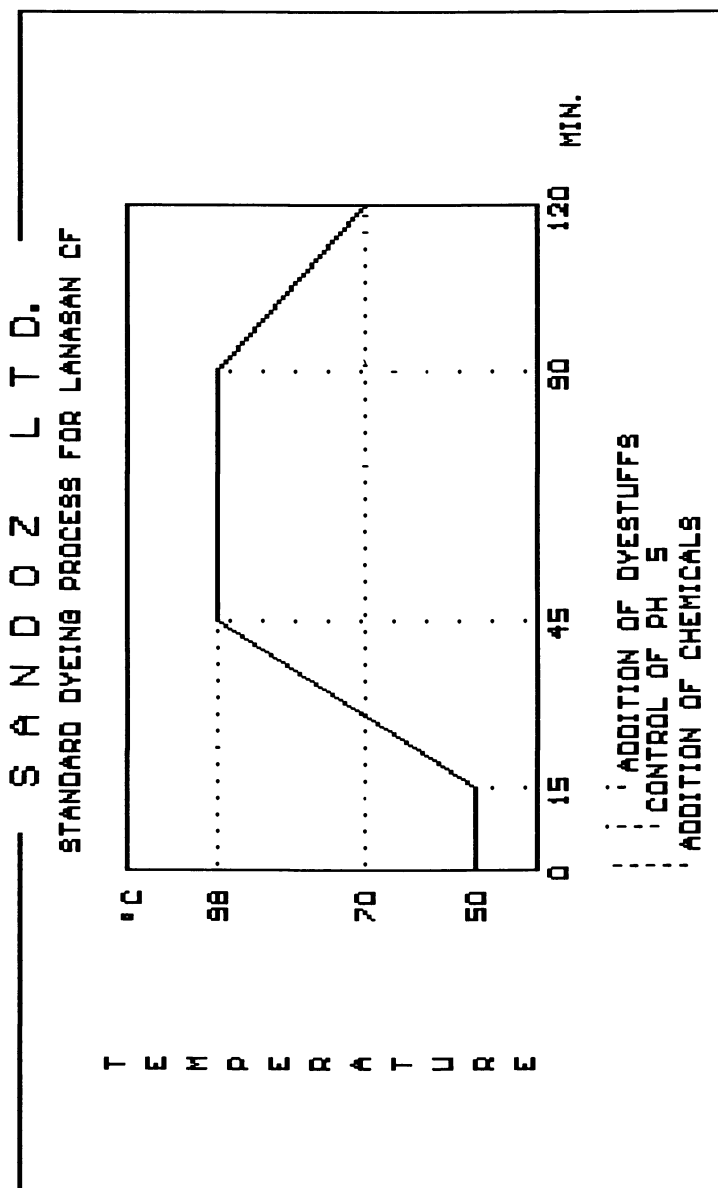


Figure 6. Temperature-time diagram for the standard dyeing process for Lanasan CF dyestuffs.

Firstly, there are many facts or data to be considered. In the field of wool dyeing there is for example the international standard for the enduse-fastness-requirements given by the International Wool Secretary (IWS). For WOOLY this means that it must contain a component of simple data retrieval.

On the other hand, the processing-fastness-requirements depend on the very special sequence of operations that are made around the actual dyeing process. The best representation of the expert's knowledge in this area is given by a set of rules that catches the expert's considerations when he derives these requirements.

Both kinds of fastness-requirements can only be considered as a reasonable set of default values. Depending on the customer, the requirements can deviate considerably from this standard. So the user must have the possibility to change the default values and set his own standard. But not all changes are possible in a given situation. To prevent the user from unreasonable requirements the system must know what can reasonably be required in a given context. For the system this means, that it must not only know the usual requirements in each situation, it must also know its own limits.

Moreover the properties of a dyeing process depend strongly on the actual recipe, i.e. on the very specific combination of dyestuffs as well as on the depth of the resulting color. An expert has quite a good feeling about the influence of the recipe on the different fastnesses. To seize this influence the building of a quantitative model is a suitable approach.

Facts, rules and quantitative models are the chosen forms of representation for the different areas of knowledge in the field of wool dyeing. But these parts are not independent, they must be tied together, i.e. an inference mechanism is needed. The theoretical concepts of forward and backward chaining in their pure forms turned out to be insufficient to catch fully the expert's knowledge about the connections and interdependencies between the various aspects. Only a careful detailed analysis of the expert's behavior allowed implementation of an adequate inference mechanism, which turned out to be a mixture of several theoretical approaches.

Usually an explanation facility is considered to be an essential part of an expert system. As the basic problem of WOOLY is very complex, this aspect had to be an integral part of the system. Therefore, special care was given to combine every result with the relevant information about the reasons why it was achieved. In most cases this simply means that all the necessary background information appears on the

screen. If this is not possible because of the amount of important information, there are always several approaches to look at a specific result to stress different aspects.

Final Considerations

WOOLY is a system that provides non experts in the complex field of wool dyeing with the knowledge of some outstanding specialists. Several important aspects of this field are covered and tied together by this expert system. WOOLY combines qualitative and quantitative aspects. Most of the collected knowledge is for the first time brought together and considered in a systematic way. The system can even replace some of the dyer's laboratory work. With WOOLY the dyer can find in a short period of time better solutions for his daily problems than he could get up to now.

RECEIVED June 9, 1989

Chapter 13

G2: Chemical Process Control

Robert L. Moore

Gensym Corporation, 125 CambridgePark Drive, Cambridge, MA 02140

The requirements for expert systems for process control have inspired new designs based on real-time knowledge base inferencing. Object oriented representation of plant equipment, knowledge representation of the interactions of processes and models of process behavior -- heuristic as well as analytical -- are incorporated into a real-time expert system for process control. The application of inference in real-time requires using metaknowledge to focus the inferencing resources of the expert system. Finally truth maintenance requires a temporal model of the time dependence of the truth of data and inferred results. A structure which includes these considerations is presented. Over 100 installations have been implemented as of this writing.

This paper describes the G2 expert system technology developed for real-time applications. Current installations are primarily in large chemical process plants, where the need for this technology is to advise operators for safety and economic reasons. Other applications include manufacturing in the aerospace and microelectronics industries, network monitoring, telemetry data monitoring, robotics and financial transaction monitoring. In this paper we discuss the basis of the technology of the G2 real-time expert system.

G2 allows the representation of deep knowledge, analytic and heuristic knowledge, and other aspects required for the implementation of real-time expert systems. Graphics and structured natural language interfaces allow the user to construct knowledge bases of dynamic applications, to test expert system behavior and to validate knowledge bases under various dynamic scenarios. The interactive developer interface allows short development-

0097-6156/89/0408-0190\$06.00/0

© 1989 American Chemical Society

and-test cycles. Built-in data interface facilities allow the engineer to implement an application interactive with live data sources.

Knowledge Representation

Several considerations of dynamic domains impose requirements on the knowledge representation:

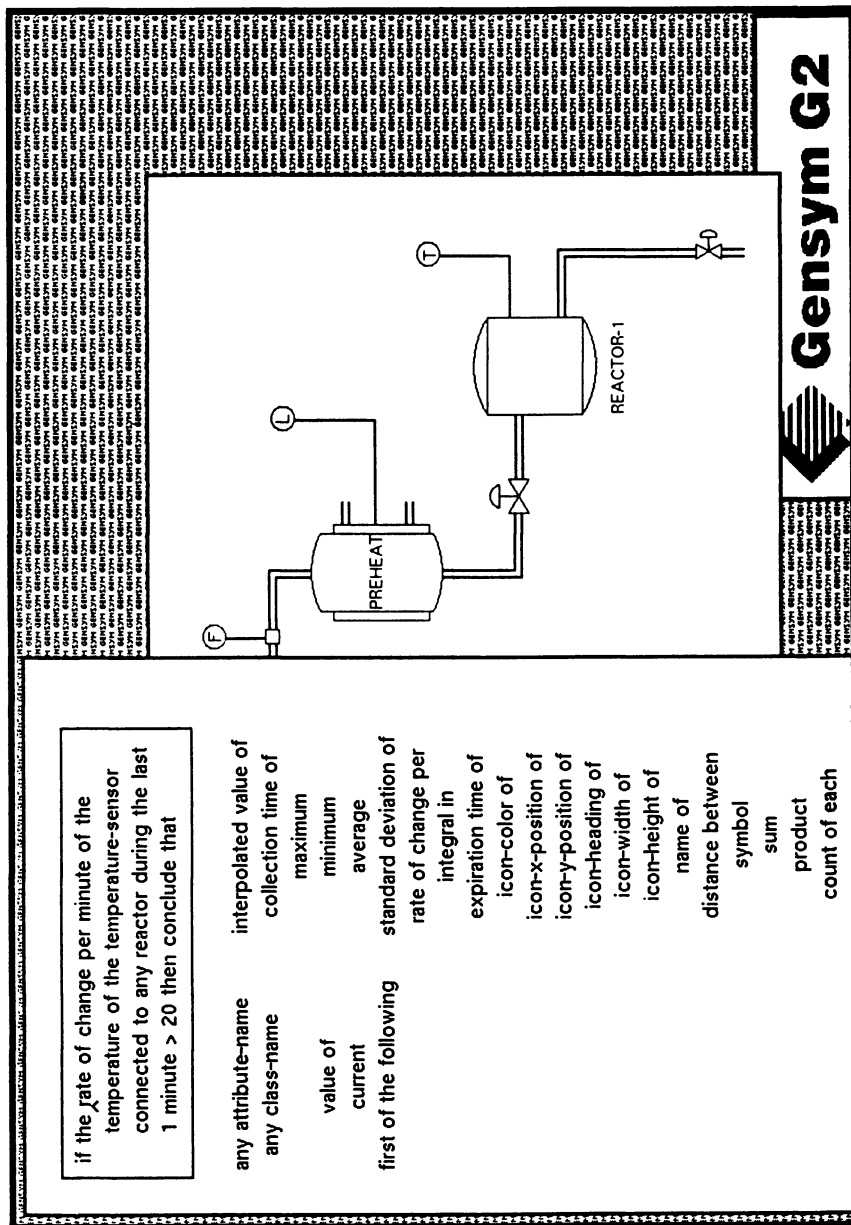
1. *The concurrent use of analytic and heuristic models.* Conventional simulation methods allow analytic models. Conventional expert systems allow heuristics, but leave the analytic part for the user to program. The combination of analytic and heuristic knowledge in an object oriented framework allows the applications to be addressed in a unified way.

2. *Interaction between objects.* The structure of an application is frequently important in predicting behavior, performing diagnosis or in scenario simulation. Structure is generally expressed as connectedness of objects, or proximity of objects. Structure may also be expressed in an object's attributes, especially where connections may vary in time. A framework which has the built-in capability to reason in terms of object connectedness or proximity, and to integrate analytic as well as heuristic knowledge in these terms, allows construction of the knowledge for the application.

3. *Dynamic behavior and live data.* Many problems have a real-time aspect, including dynamic knowledge in differential equation form, such as equations of motion. Live data may be needed for the eventual deployment, and data access and real-time processing may be important. A framework which includes these real-time considerations in the expert system design is required. The framework allows simulation to provide real-time values for prototyping and development, to be supplanted by sensor-based data at installation. Data servers provide interfaces to other systems with a minimum of user work, so the prototype can become the actual application.

In addition to the general characteristics of the applications, which call for a unified framework, the general desirability of rapid implementation calls for the use of high level interfaces. In G2 these include graphical construction of the application domain and structured natural language for expression of knowledge, models, and other information. Modern parsing techniques allow the user to express the knowledge in reasonably natural form, and G2 checks the user input as it occurs. Look-ahead menus help the user, and errors are immediately flagged. This eliminates a whole level of debugging which conventional programming requires.

Fig. 1 shows connected objects. The domain was defined on the workstation screen by connecting the separate objects. The G2 expert system then "knows about" the connection and interaction of these objects. In Fig. 1, the engineer is defining a rule. A "look ahead" menu helps the engineer to define knowledge in a form that G2 can understand, while the rule itself is in a natural form. The engineer can refer to current or past



if the rate of change per minute of the temperature of the temperature-sensor connected to any reactor during the last 1 minute > 20 then conclude that

- any attribute-name
- any class-name
- value of
- current
- first of the following
- interpolated value of
- collection time of
 - maximum
 - minimum
 - average
- standard deviation of
- rate of change per
- integral in
- expiration time of
- icon-color of
- icon-x-position of
- icon-y-position of
- icon-heading of
- icon-width of
- icon-height of
- name of
- distance between
- symbol
- sum
- product
- count of each

Fig. 1. Defining heuristic knowledge.

values, or to behavior over time. The rule can invoke conclusions or initiate actions.

In many domains, there are multiple objects which have related knowledge. For example, there may be many pumps, all of which need inference analysis of a similar sort. Fig. 2 shows generic knowledge, which can be applied across a class of objects. The G2 inference engine can interpret all specific applications of generic knowledge by actively interpreting the schematic representation of domain structure.

In the case of connected objects, the behavior can be defined in terms of the connection, which is understood by G2 "live" from the screen. For example:

"if the rate of change per second of the temperature of the object X connected at an input of any heat-exchanger during the last 3 seconds > 5 then invoke diagnostic rules for X"

This heuristic considers dynamic behavior of a connected object, and searches for causes of problems by tracing back through connections. The "X" in the above rule is a local name, within the rule. If a new object in the heat-exchanger class is created, through cloning for example, and it is connected to other objects, then the above heuristic will apply also to the new object. This allows the rapid implementation of diagnostics.

Behavior of objects can be defined by classes or specific instances. An example of a class behavior might be:

"state variable: $d/dt(\text{the velocity of any intercept-vehicle } V) = (\text{the thrust of } V) / (\text{the mass of } V) \dots \text{ etc.}"$

Here the behavior is generic, across the class, but individual instances can behave according to their own attribute values. A new object in the class can be created, by cloning for example, and it will inherit this behavior. The prototyping of hundreds of such objects becomes relatively trivial.

Fig. 3 shows a robot navigation application. This was inspired by an installation of G2 at the Robotics Technology Group at the Savannah River Laboratory. The robots in the example seek the shortest path toward one of the "goals." The user can connect and disconnect the nodes of the example, and G2 will understand if the path exists between the nodes. The animation feature of G2 is used to move the robots during the scenario. The use of robots in a nuclear plant is primarily intended to avoid sending humans into hazardous environments. There are many examples in conventional chemical plants where a similar use of robots would be well justified.

Fig. 4 shows critical path planning using G2. The user defines the possible paths by graphical connection of objects. Alternate structures can be considered, even by changes while running. This allows interactive design of planning strategies. In the particular example, a complex batch chemical processing plant is being scheduled by G2, with critical path activities explained by G2, to help the user with rescheduling the plant.

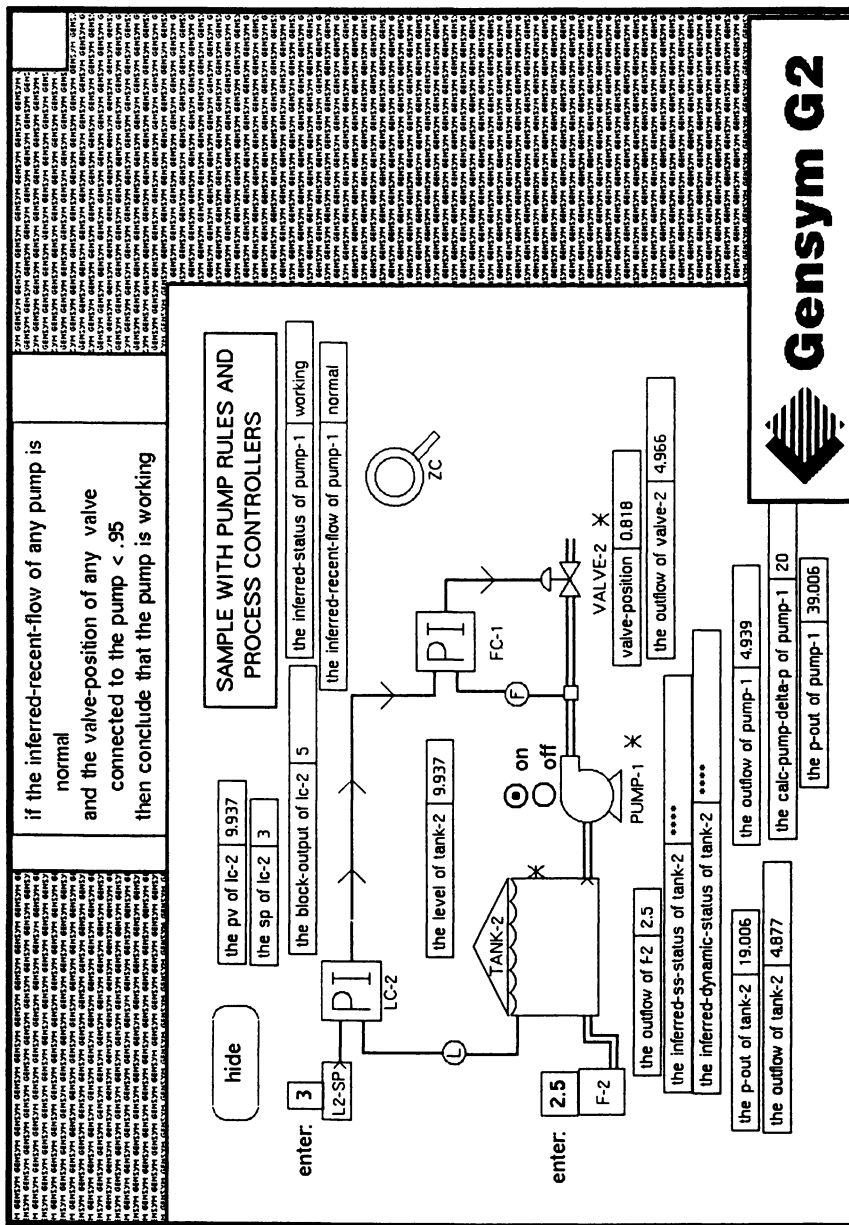


Fig. 2. Generic knowledge for classes of objects.

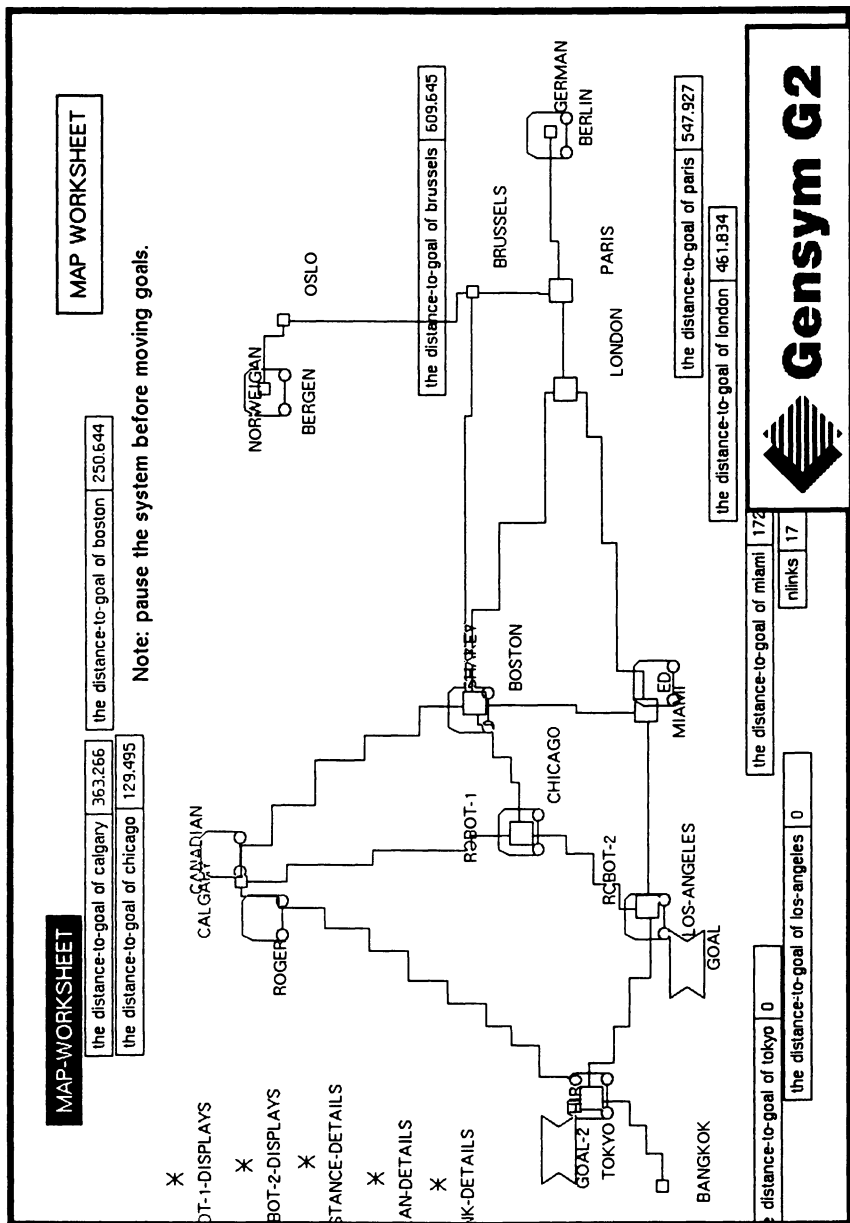


Fig. 3. Robot navigation example.

Publication Date: September 1, 1989 | doi: 10.1021/bk-1989-0408.ch013

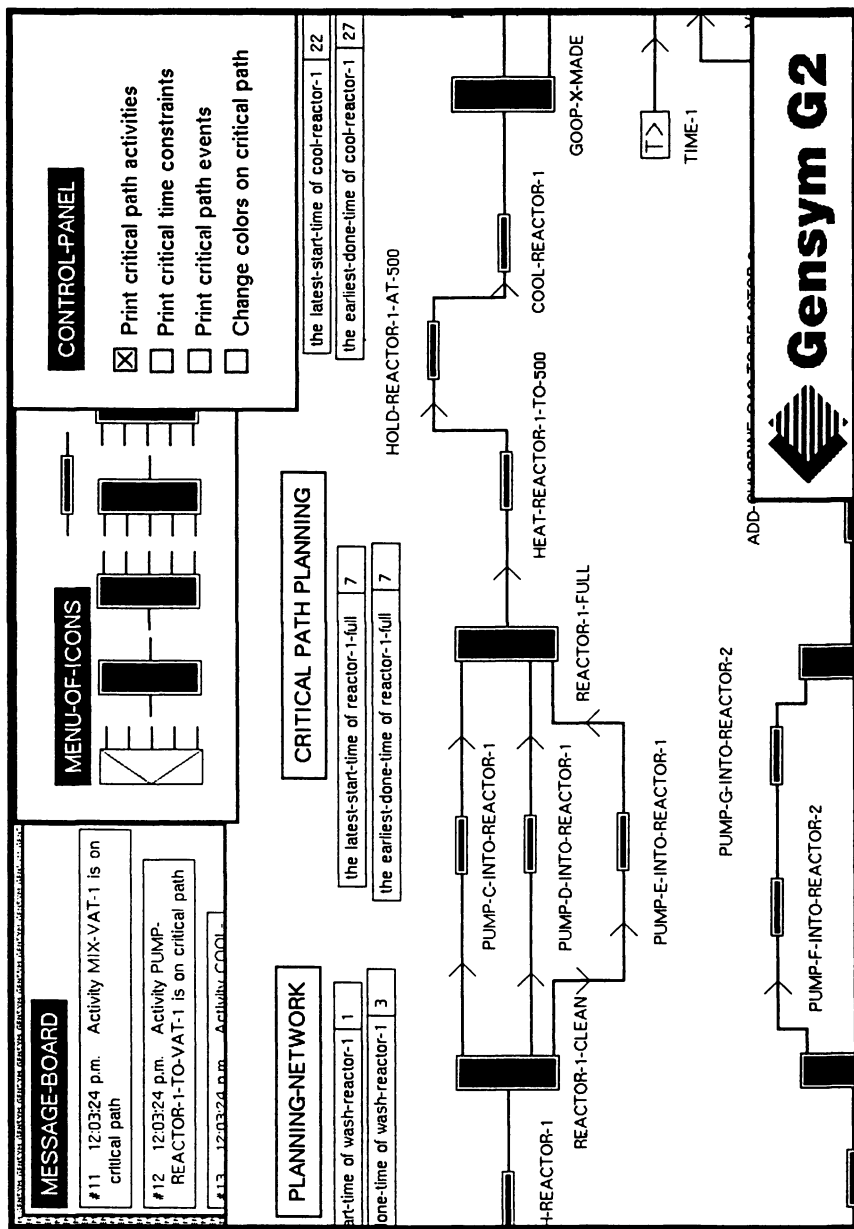


Fig. 4. Critical path planning.

Knowledge Base Management

Knowledge can be arranged on workspaces, on which the user can organize the knowledge for ease of management. Fig. 5 shows a variety of such workspaces, which can be stacked, expanded or contracted, and otherwise manipulated. Each workspace can be treated as a separate execution process, or they can be executed concurrently.

In the case of large applications, there may be many workspaces, perhaps hundreds of them, each with hundreds of objects, rules, dynamic models, displays and other items. G2 provides a relational-knowledge-base retrieval facility to enable the user to manage such a large application. For example, the user may request:

"show on a workspace every statement containing the word schedule"
in which case a temporary workspace is created with copies of every model, rule or other statement with the chosen word or phrase.

or "show on a workspace every rule where categories includes safety"
in which case the temporary workspace contains all rules about safety This is an example of one use of metaknowledge (knowledge about knowledge).
or "show on a workspace every object where the outlet-temperature of the object > 780"

in which case the temporary workspace shows the objects which have the named attribute, and where the value of the attribute satisfies the request.

In the temporary workspace, the user may interactively edit the knowledge, or may say "go to original" in which case the workspace containing the original of the item is brought up, with the cursor centered on the requested item. In this way, a user can navigate through workspaces containing thousands of items.

Reasoning About Knowledge

Two, apparently conflicting, requirements dominate the inference paradigm considerations in the real-time domains. One is the need for truth maintenance. With thousands of data changing rapidly, the validity of conclusions at all levels of inference are in question. The other requirement is for real-time performance, where real-time means fast enough to advise the human operator and/or control the robot or other process.

First attempts at using expert systems for real-time applications involved taking a "snap-shot" of data and using a static expert system paradigm to perform inference. This process is then repeated after inferences are completed. Conventional pattern-matching paradigms which examine all possible conclusions for the current data values are too slow for most real applications. The static expert system approaches lead to slow performance on even small prototypes of a few hundred rules and a few hundred data values. This has been widely recognized.

Code improvements and computer improvements can help. However, a fundamentally different inference approach is appropriate for real-time problems. The approach that a human expert uses in a real-time situation is to maintain a peripheral awareness across the domain, watching for

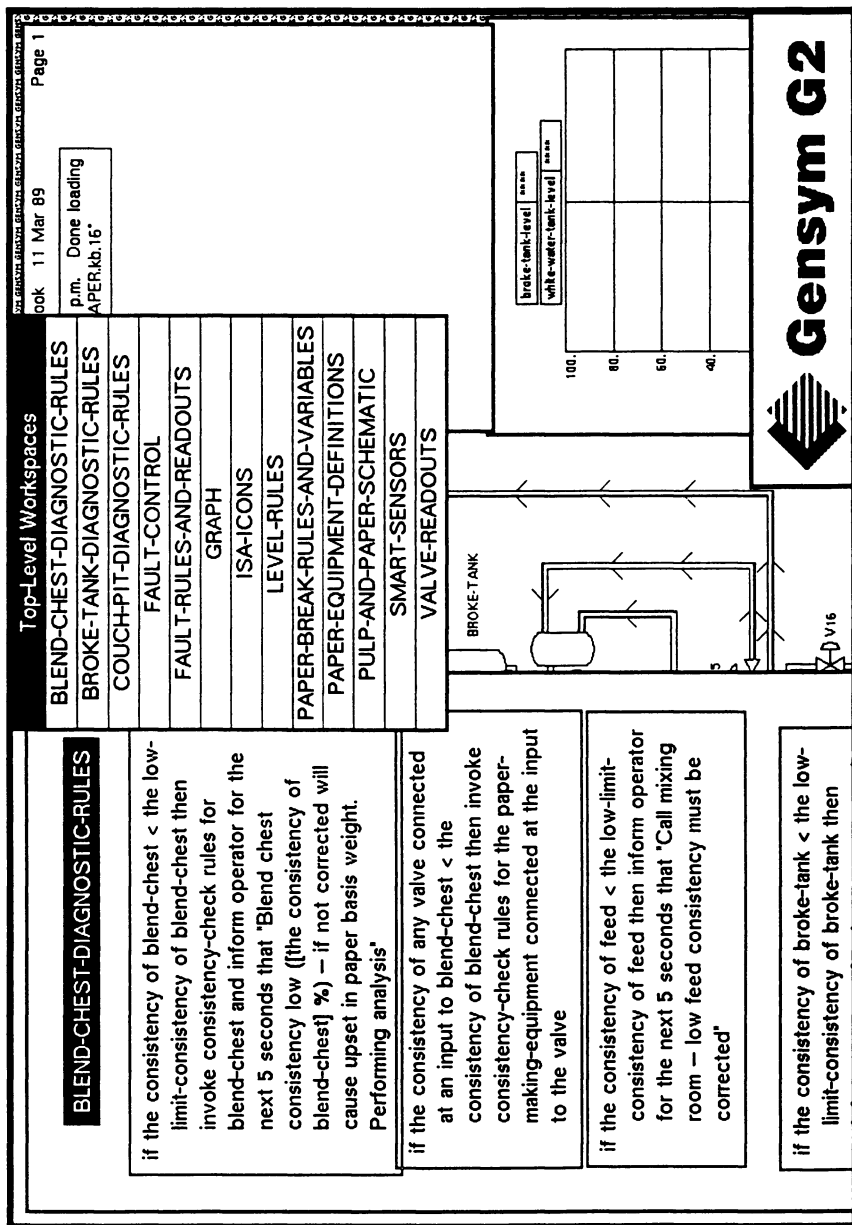


Fig. 5. Managing knowledge on workspaces.

performance exceptions, and then focusing on areas of interest. The G2 inference engine operates similarly. The inference engine continually scans knowledge which the expert has specified for peripheral awareness. If a safety-threatening condition occurs in a reactor, for example, the G2 inference engine uses metaknowledge to determine which knowledge to invoke, thus focusing on the area of interest.

One benefit of the metaknowledge approach is that very large knowledge bases can be run in real time. Since many types of problems and behaviors are represented in the knowledge base, it can get quite large, with thousands of rules. However G2 does not consume computer time looking for patterns in all of this knowledge all the time. Rather it focuses attention on the knowledge needed. The concept is like the human thought process, in that a human does not use knowledge of swimming or driving when walking in the park. The human mind focuses, using the knowledge relevant to the task.

In static expert systems, truth maintenance involves changing inferences when data changes. In real-time problems there is an additional requirement to change inferences even if no new data is available, since time is a factor in validity or certainty of inference. One way to express this temporal validity information is to attach an expiration time to each value maintained by the inference engine, and propagate this when inference is carried forward. Generally, when a conclusion is based on several time sensitive variables, the earliest of their respective expiration times will be carried forward. Expiration times can be propagated forward through multiple levels of inference, but there are also ways to limit this propagation.

Application Example

Plant safety is an important application area for G2. Many of the early installations are for chemical and nuclear safety purposes. Of course, plants are already provided with extensive alarming and other safety related equipment. G2 does not replace any of this -- it is used to add another intelligent observer of the plant operation -- an observer that watches thousands of variables with tireless attention, applies knowledge of expected behavior and interactions, and indicates to the operator an intelligent description of unexpected or unsafe behavior. It is questionable whether a plant should be run without such an observer.

As an example of the methodology involved with such an application, consider the following:

"if abs(the valve position of any valve V - the valve position of V 2 minutes ago) > .05 and abs(the flow of V - the flow of V 2 minutes ago)/(the flow-maximum of V) <.01 then conclude that the valve-is-stuck of V and invoke diagnostic rules for the process equipment connected to V"

Such a rule will look for stuck valves across the plant, detect the stuck condition for those where flow can be inferred or calculated to have not

responded to valve movement, and will start to trace the upstream and downstream consequences. Note that the generic form of the rule allows a concise expression of a form of knowledge which will be applied in perhaps hundreds of instances, actively using the plant schematic to trace connectivity. This is the use of object-oriented knowledge representation.

The above is a simple example. A more typical case involves the use of dynamic models of expected behavior, compared to multiple plant measurements. When a significant difference occurs, then some form of problem exists, and the operator can be notified, in many cases before alarm conditions are reached. Furthermore, G2 can try to determine the cause of the problem. For example, a measurement may be faulty, or a process reaction, leak or other process problem may be occurring, or the model may simply be in error. These causes have different effects on downstream units, so G2 can reason about connected process equipment to try to isolate the true cause.

Summary

The real-time expert system technology described in this paper represents a departure from static expert system design, as the issues of time relationships and dynamic behavior have been addressed. The resulting expert system is capable of applying thousands of rule-frames of knowledge, and of performance in real time for reasonably complex operations. Installations of the technology include robotics (1), chemical process plants (2,3), manufacturing, and network management (4).

Literature Cited

1. Byrd, J.S.; Fisher, J.J.; Mallett, W.R. "Expert Robots for Process Environments," Robotics and Autonomous Systems, Elsevier Science Publishers, New York, N.Y. (1989).
2. Moore, R.L.; Hawkinson, L.B.; Levin, M.; Hofmann, A.G.; Matthews, L.B.; David, M.H. "Expert System Methodology for Real-Time Process Control," Proc. 10th World Congress on Automatic Control, IFAC, Munich, July 1987, Vol. 6, pp. 274-281.
3. Moore, R.; Lindenfelzer, P.; Hawkinson, L.; Matthews, B. "Process Control with the G2 Real-Time Expert System," Proc. the First International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Univ. of Tennessee Space Institute, June 1988.
4. Noren, C.S. "Rapid Prototyping Network Management Systems," Proc. Milcom '88 (IEEE), San Diego, CA, (1988).

RECEIVED July 11, 1989

Chapter 14

Optimizing Combustion in Multiple-Burner Installations

Terence C. Fogarty, D. Paul Jenkins, and Jon Mortimer

Bristol Transputer Centre, CSM Department, Bristol Polytechnic,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Rules for adjusting the air inlet valves to each of a number of burners with a common flue, depending on the concentrations of oxygen and carbon monoxide monitored in that flue, were elicited from experts. The rules have been embodied in an expert system, written in PROLOG, which is used to optimise combustion in multiple burner installations. The system was tested on PROLOG simulations of multiple burner installations, each burner being modelled by a number of perfect burners. It was then run on a twelve burner zone of a 108 burner furnace on a continuous annealing line for rolled steel. A system which learns its own rules by modifying the expert rules in the light of experience is being developed and will be tested against the expert system on a multiple burner boiler plant.

The Problem. The problem is to automatically and continually optimise combustion in a multiple burner furnace or boiler plant by altering the air inlet valve to each of the burners depending upon the carbon monoxide and oxygen readings taken from their common flue. The optimum air/fuel mixture will be different for each burner, because of their varying type, age and condition, and it can be achieved for a given supply of fuel by individually adjusting the air inlet valve to that burner.

A typical multiple burner furnace has over one hundred small burners arranged in zones of one or two dozen burners with two exhaust stacks per zone each taking the waste gases of half the burners of that zone. A typical multiple burner boiler plant has six double burner boilers exhausting via three shared stacks.

At present, the air/fuel ratio of each burner in a multiple burner installation in the steel industry will be manually adjusted about every 3 to 9 months by a fuel technician using portable analysers to measure the carbon monoxide and oxygen concentrations in the waste gases of each burner.

0097-6156/89/0408-0180\$06.00/0

© 1989 American Chemical Society

Related Work. As the cost of analysers has come down, many single burner boilers have had air/fuel ratio set point controllers added with a feedback loop from either an oxygen (1) or a carbon monoxide (2) analyser. Presser and Semerjian (3) give a good review of this work up to the mid 1980's. The optimum oxygen setpoint varies depending upon the kind of burner, firing level, load, type of fuel, temperature, humidity, amount of tramp air and the deterioration in the burner. The carbon monoxide reading is very volatile at near optimum levels. A combination of the two readings is thought by most experts to be the best approach (4-7).

A rule based approach to process control has for many years provided an alternative to traditional methods in the form of fuzzy logic control (8,9). Since the advent of expert systems, rulebases have been used for fault diagnosis [10], to advise operators (11), to aid control engineers when installing PID controllers (12), to provide expert on-line tuning for PID controllers (13), and to control processes without the use of fuzzy logic (14,15).

The Rule-Base

The Origin of the Rules. Rules for adjusting the air inlet valves to each of a number of burners with a common flue, in which the concentrations of oxygen and carbon monoxide are monitored, were elicited from energy engineers (16) and coded into PROLOG. The resulting rulebase embodies the practice of the fuel technician who periodically tunes multiple burner installations together with the expertise of the energy engineer who has devised a system for the continuous optimisation of such installations.

The Type of Rule. The rules are basically of the situation-action type, where the situation is governed by carbon monoxide and oxygen readings, classified according to constant or variable system parameters, and the action is a combination of adjustments to air inlet valve settings and system parameters.

All the rules are of the form: "IF situation THEN action" but chains of up to three of these are accommodated to encapsulate the procedures outlined by the experts. The longest chains are therefore of the form:

```
"IF situation 1 THEN action 1
  THEN IF situation 2 THEN action 2
    THEN IF situation 3 THEN action 3"
```

which means that the rulebase is guaranteed to react to a completely changed situation within three interactions with the installation. This goes some way towards satisfying the real-time constraints.

Associated with each of the rules is a report clause which updates the operators console and/or logs a message depending upon the situation which caused the rule to fire.

Classification of Situations. Three kinds of situation are covered by the rules. First, there are problem situations which require fault diagnosis and possibly safety measures. Second, there are

situations in which all the equipment is assumed to be working correctly but the air/fuel ratio of some or all of the burners is far from optimum. Fairly decisive control measures are required here. Third, there is the most desirable situation, in which all of the burners are giving near optimum performance, requiring only checking and further optimisation measures.

Each of the readings, oxygen and carbon monoxide, is classified as one of: very low, low, O.K., high or very high, according to parameters which remain fixed except for those delineating the optimum oxygen reading. The change in carbon monoxide since the last reading is classified as one of: neg(ative), nil or pos(itive), nil being positive but less than a calculated amount. This amount is the difference between the last reading and the upper optimum limit for carbon monoxide divided by the number of burners.

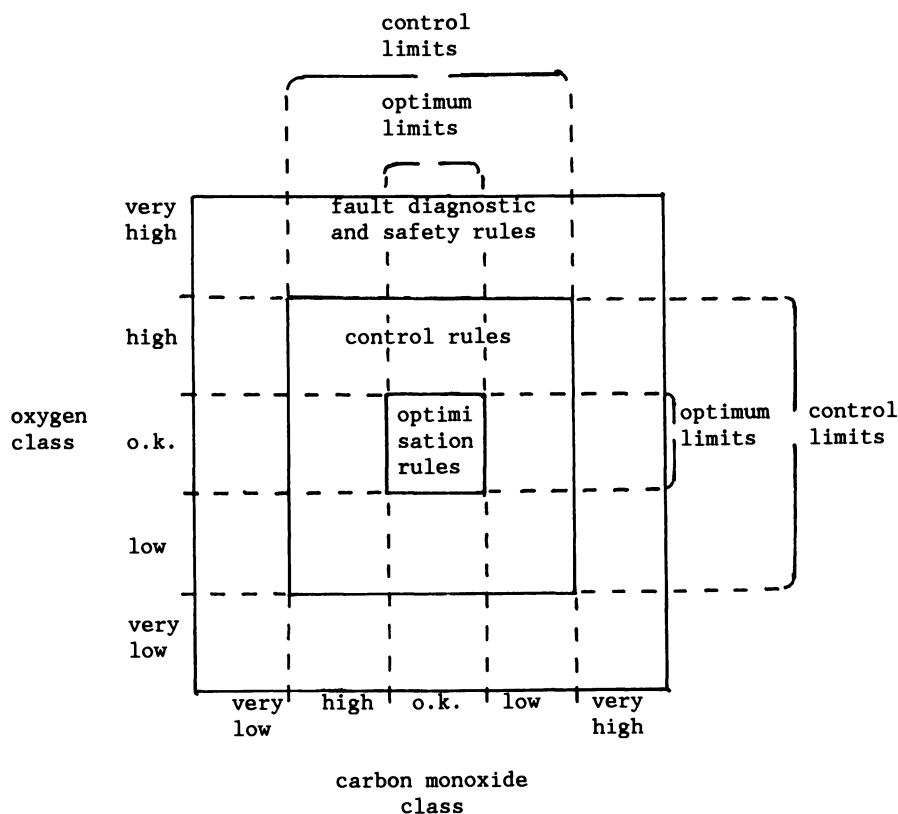


Figure 1. Classification of situations and classes of action

The classification of situations and classes of action required is illustrated in Figure 1. Every eventuality is covered by a rule and there are no conflicting rules for any situation. In PROLOG a rule takes the form:

```
action(O2,CO,CO_change,Rule) :- {list of actions}
```


where "O2" is instantiated to the oxygen, "CO" the carbon monoxide and "CO_change" the change in carbon monoxide classifications, respectively, and "Rule" to the number of the rule in the chain. The maximum number of rules possible in the system is thus $5*5*3*3 = 225$ which has a corresponding maximum search time, thus also helping to satisfy real-time constraints.

Composition of Actions and Examples of Rules. All the procedures described by the experts can be broken down into four types of action. There are actions which alter one or all of the valve settings, actions which alter one or all of the increments by which the valves are adjusted, an action which focuses attention on the next burner and one which lowers optimum oxygen limits. An example of a fault diagnostic rules is:

```
action(very_high,ok,_,1).
```

which comes into effect when the oxygen reading is classified as very high and the carbon monoxide reading as O.K. No action is taken in this case but a message will be generated by the associated report clause to the effect that the oxygen sensor should be checked. One of the chains of control rules is:

```
action(high,high,_,1) :- next_burner,
                        reduce_air.
```

```
action(high,high,pos,2) :- increase_increment,
                           increase_air,
                           reduce_increment.
```

```
action(high,high,nil,3) :- reduce_air.
```

which fire successively if, initially, both oxygen and carbon monoxide readings are high and then, after the air inlet valve to the burner being focused on is closed somewhat, there is a positive change in the carbon monoxide reading and then, after the air inlet valve has been opened somewhat more, there is a negligible change in the carbon monoxide reading returning the air inlet valve to its original position.

Rule-Based Optimisation of a Simulation

The Simulator. A computer simulator of multiple burner installations (17), represents each burner in an installation by a number of "perfect" burners, each coded as a PROLOG fact, whose attributes are distributed within a given range to model a particular kind of burner. Alterations to the air inlet valves are represented by changes to these attributes, as are changes in firing level and small changes in the burners over time. Readings are averaged functions of the attributes based on experiments with a near perfect burner (18) and the given characteristic of air inlet valves.

The Simulation. A simulation of a twelve burner furnace with a common flue was created. Each burner was set up with randomly

generated average excess air levels of between -5% and 95% with its air inlet valve fully open and between -25% and 2% with its air inlet valve closed as far as it would go. The particular model generated is shown in Figure 2. Each valve can be closed from 0% to 100% of possible closure and the simulation reduces the level of excess air

burner number	excess air level		burner number	excess air level	
	closed	open		closed	open
1.	-5%	58%	7.	-16%	44%
2.	-19%	58%	8.	-20%	66%
3.	-10%	11%	9.	-24%	29%
4.	-19%	73%	10.	-8%	16%
5.	-13%	50%	11.	-3%	3%
6.	-24%	77%	12.	-2%	32%

Figure 2. The randomly generated model

to that particular burner by an exponential function of that percentage of the difference between the maximum and minimum levels of excess air to that burner. Oxygen and carbon monoxide readings for each burner are then produced according to the simple chemical equations, (plus or minus randomly generated interference factors), and these are averaged over the twelve burners to give single, final, oxygen and carbon monoxide readings.

Run	Initial Valve Positions	Final Valve Positions											
		1	2	3	4	5	6	7	8	9	10	11	12
A.	0,.....	93	76	64	80	80	77	74	77	61	69	75	96
B.	90,....	93	76	64	80	80	77	74	77	61	69	75	96
C.	75,....	94	76	64	80	80	77	74	77	61	69	75	96
D.	60,....	93	76	64	80	80	77	74	77	61	69	75	96
E.	0,90,..	93	76	60	80	80	77	74	78	56	73	58	96
F.	90,0,..	93	76	64	80	80	77	74	77	61	69	92	94

Figure 3. Rulebased optimisation of the simulation

Optimisation of the Simulation. The final valve positions arrived at after running the rulebased optimisation program on the simulation from a number of initial valve positions are shown in Figure 3.

Initial valve position "N,..." indicates that each valve is set to position N, while initial valve position "N,M,.." indicates that valves were set, alternately, to positions N and M, where N and M are percentages of possible closure.

There is a high correlation between the final valve positions starting from the different initial conditions. In fact only one valve shows a marked difference, this being the one for burner 11. It can be seen from Figure 2 that altering the air inlet valve for burner 11 can make very little difference to the amount of excess air reaching that burner anyway.

Rule-Based Optimisation of a Twelve Burner Zone of a Furnace

The Hardware. Having been developed, tried and tested on simulations of multiple burner installations, the rulebase was then run on half of a twenty-four burner zone of a 108 burner furnace on a continuous annealing line for rolled steel. The program ran in compiled Turbo Prolog on an IBM PC XT connected to a Thinklab 32 channel A/D and 24 channel D/A converter. This operated electrically actuated air inlet valves on eight of the burners via a unity voltage gain current amplifier. The monitored signals were from a zirconia oxygen probe and thermocouple in the common flue and an infra-red absorption carbon monoxide analyser with a sample point in the common flue.

Optimisation of the Furnace. Initial performance of the rulebase was highly erratic due to the irregularity of the carbon monoxide readings. Instead of a constant reading corresponding to each set of valve positions there was a noisy band of readings with random peaks above it. The noise was due mainly to the volatility of the combustion process at near optimal air/fuel ratios while the peaks were induced by changes in firing levels of other zones in the furnace. This problem was circumvented by writing into the interface module, statistical procedures to filter the signal from the carbon monoxide analyser with the consequent lengthening of response time and loss of some information.

The constant system parameters used by the rulebase had to be learned before it could be run satisfactorily. Upper and lower carbon monoxide optimum limits and maximum and minimum increments for the air inlet valves all needed adjusting.

The rulebase contains no rules regarding the detection of and action to be taken in the case of the changing of firing levels. The furnace has only one firing level for each zone and a control system maintains the required temperature by turning zones on and off as necessary. All the rulebase had to achieve was to optimise one zone when it was firing. If the firing level changed during a chain of rules this did not register.

The program was run 14 times in all using various parameters and different initial configurations of air inlet valve settings. On each occasion stackloss, a measure of energy wasted in the exhaust gases as a percentage of energy supplied in the fuel, was reduced to about 15%. This compares well with the stackloss of 20% registered when the system was not being used. On each run valves were closed by certain amounts and an optimum was arrived at, but there was no correlation between the final configurations of valve settings even when the same parameters were used.

A System which Learns its own Rules

The Need to Learn. No conclusions were reached as to how responsive the rulebase will be as regards long term change in an installation but it was unable to cope with short term change in the form of the rapidly varying carbon monoxide readings. It was obviously unable to initially adapt to and learn about the particular plant on which it was installed and it will always need an expert to initialise and possibly maintain it. When the rulebase is used on a boiler plant it may not be equipped to deal with the constantly changing firing levels that the boilers undergo. In common with most Expert Systems it performs like a competent novice instead of like a real expert who exchanges his/her current knowledge for the opportunity to work on problems from which they can learn more (20).

Ideally, a system that learns is required: one that learns the individual characteristics of the plant on which it is installed; one that learns its own rules when it meets situations that it either has not met before or was not prepared for by the experts that set it up; one that learns simple rules of the same format that are easily modifiable according to changing circumstances.

A Simple Learning System. A learning system, based on a genetic algorithm (21), which generates, tests and improves a rule for optimising combustion in multiple burner installations using simulated evolution has been developed. The system randomly generates 50 rules of a given format and evaluates them on a simulation of a multiple burner installation. It then chooses 50 times from those rules, based on their performance (some will be duplicates and some will not be chosen), and, using the genetic operations of crossover and mutation, generates 50 new rules for evaluation. This process is continually repeated and the system converges on the best rule it can find.

Experiments have been conducted to compare the performance of the rulebase and the genetic learning system in dealing with noise. Ten different simulations of multiple burner furnaces were set up and the signals from them distorted using four different noise levels from none to high. The rulebase and the genetic learning system were used to optimise each of the simulations at each noise level. It was found that although the performance of the rulebase is superior when there is no noise, it deteriorates considerably and proportionally as the level of noise is increased. The performance of the genetic learning system, on the other hand, was the same no matter what the level of noise. The performance of the two systems was approximately comparable at the low noise level.

These results form the basis for the development of a learning classifier system (22,23) for combustion control in multiple burner installations.

Conclusion

Provided it is used under usual conditions with the sort of data expected the Expert System performs as well as a human expert. It has the added advantage of being able to do this 24 hours a day, 7 days a week and not just every 6 months. On the other hand two weaknesses can be identified. Firstly, a human expert has to learn some specific parameters corresponding to characteristics of the

particular installation in order to equip the rulebase with them. Secondly, the rulebase is brittle in the face of noisy data. In general, it is not known how well the Expert System will perform in unusual or unexpected circumstances.

At present the Expert System can only learn by being told, i.e. by having new or revised rules entered by an expert. In order to make it more robust other ways of learning need to be introduced. The genetic algorithm provides one approach to learning control rules for the system and this is being vigorously pursued.

Acknowledgement

This work was supported by a co-operative award from the Science and Engineering Research Council and British Steel. (Award No.86518692).

Literature Cited

1. Slevin, M.J.: Boiler oxygen trim control. Heating/Piping/Air Conditioning, pp.95-98, November 1984.
2. Coe, D.: Improving small boiler combustion control. Control and Instrumentation, pp.33-35, March 1987.
3. Presser, C. and Semerjian, H.G.: Evaluation of Industrial Combustion Control Systems, U.S. Dept. of Energy, 1984.
4. McFadden, R.W.: Multiparameter trim: it pays in combustion control, InTech, pp.57-61, May 1984.
5. Sunderland, R.H.: Improving combustion efficiency by multivariable analysis and control, Measurement + Control, Volume 18, July/August 1985.
6. Langridge, S.: Combustion control in process heaters, Control and Instrumentation, pp.37-39, March 1987.
7. Cremer, C. and Hertz, J.: A fast control algorithm for near optimal control of industrial combustion, Proceedings of Control-88 Oxford, IEE, 1988.
8. Zadah, L.A.: Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans on Sys, Man and Cybernetics, vol 3, no 1, pp.28-44, Jan 1973.
9. Mamdani, E.H., Efstathiou, H.J. and Sugiyama, K.: Developments in fuzzy logic control, 23rd. Conference on Decision and Control, pp.888-893, Dec 1983.
10. Efstathiou, J.: Knowledge based systems for industrial control, Comp. Aided Eng. Journal, pp.7-20, Feb 1987.
11. Sargeant, R.A.E.: Expert systems for process control rooms, Measurement + Control, pp.239-244, Nov 1986.
12. Jeffreys, S.: Software simplifies batch control design, Control Engineering, pp.107-109, Sep 1987.
13. Highham, E.H.: A different approach to self tuning in process controllers - the case for introducing an expert system, Meas. and Control, vol.19, pp.253-257, Nov 1986.
14. Francis, J.C. and Leith, R.R.: Artifact: A real-time shell for intelligent feedback control, in Research and Development in Expert Systems, M. Braemer (Ed), 1985.
15. Darius, I.H.: Rule based control, a new tool, Proc.2nd Int.Conf.Machine Control Systems, pp.121-128, May 1987.

16. Edmundson, J.T., Jenkins, D.P. and Mortimer, J.: Multiple Burner Stoichiometry Control in Innovation in Process Energy Utilisation, IChemE, pp 47-60, 1987.
17. Fogarty, T.C.: A Prolog Simulator for Combustion Control, Bristol Polytechnic Technical Report, 1987.
18. Wakefield, A.C.: An Expert System for Methane Flame Stoichiometric Control, MSc Diss., Bristol Polytechnic, 1986.
19. Gaines, B.R.: How Do Experts Acquire Expertise?, Proc. of AAAI Workshop on Knowledge Acquisition for Know.-Based Systems, 1987.
21. Holland, J.H.: Adaption in Natural and Artificial Systems, 1975.
22. Holland, J.H.: Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-Based Systems, Machine Learning: An Artificial Intelligence Approach, Volume 2, eds. Michalski, Carbonell, Mitchell, pp 593-623, 1987.
23. Wilson, S.W.: Classifier Systems and the Animat Problem, in Machine Learning, vol. 2, no. 3, pp 199-228, 1987.

RECEIVED June 9, 1989

Chapter 15

CACSS

Computer-Aided Characterization of Solids and Surfaces

D. P. de Bruijn and H. P. C. E. Kuipers

Koninklijke/Shell-Laboratorium, Amsterdam (Shell Research B.V.),
P.O. Box 3003, 1003 AA Amsterdam, Netherlands

The Computer-Aided Characterization of Solids and Surfaces (CACSS) project is aimed at streamlining and improving the multi-technique analytical approach. Together with conventional computer programs (for example, those used to simulate an experiment), expert systems and 3d graphics will be combined to assist in analyzing the problem under study, choosing the most suitable set of analytical techniques and, finally, simultaneously interpreting the various experimental results. A discussion of the analytical strategy, the general outline of the resulting project and the implementation of the first parts are presented.

The characterization of complex solids, such as heterogeneous catalysts, composites or ceramics, involves the determination of many parameters. Several techniques are always needed to obtain all the necessary data to yield these parameters. For example, to characterize a catalyst surface, one might apply a combination of XPS (X-ray Photoelectron Spectroscopy), various forms of electron microscopy and LEIS (Low Energy Ion-scattering Spectroscopy) to produce a correct model. These techniques involve separate areas of expertise, the coordination of which, essential to the success of the project, requires a dedicated effort.

The Matrix Discussion Group. To this end, a discussion group was established in the analytical department of our laboratory about four years ago. This 'Structural Analysis of Solids' (SAS) group brings together the experts in the relevant techniques in order to study, improve, streamline and stimulate the multidisciplinary approach. Case studies are the principle means of doing this. Our experience has taught us that only the simultaneous presence of all the required skills can guarantee both a balanced strategy and optimal performance of the characterization process.

Besides the definite strengths of a 'matrix discussion' team, several of its shortcomings have become obvious. First, because of

0097-6156/89/0408-0190\$06.00/0
© 1989 American Chemical Society

its time-consuming nature, such an approach can only handle a moderate number of selected cases. Second, during the discussions, the relevant data (whether they be the results of simulations or basic data) cannot all be produced on the spot, at least not quantitatively. Third, the relevance of available extramural facilities (e.g. a spallation source for neutron diffraction) is sometimes difficult to judge due to the lack of sufficient in-depth expertise. A fourth problem encountered by the panel members, namely the lack of a basic understanding of each other's techniques, was gradually overcome by the performance of these exercises. The problem of communicating structural aspects of solids and surfaces, however, remained.

The CACSS Project. The SAS approach has proved to be most useful for studying and improving the multi-technique approach, but much less so in day-to-day practice. Clearly, a vehicle which makes a larger number of problems amenable to the same synergistic benefits as produced by the discussions has to be constructed. For this purpose, the SAS group continues to make the multidisciplinary strategy behind characterizing complex materials explicit. This requires an in-depth study of the complementary nature of all the techniques involved. Further, a new research project was started two years ago, the 'Computer-Aided Characterization of Solids and Surfaces' (CACSS) project. The final goal of this project is to produce a computer-based system which makes the multidisciplinary approach practical for day-to-day use. For individual analytical exercises CACSS must aid in:

1. developing a working hypothesis (a model of the material),
2. planning the multi-technique analytical strategy, and
3. interpreting jointly the analytical results.

The Role of Expert Systems. Several developments in computer science have made us believe that the CACSS project is feasible, although it will be a major effort. These developments are the fast growth in CPU power, 3d computer graphics (to model a structure), database technology, and last but not least: artificial intelligence. To achieve the required functionality, the CACSS system has to provide, among other things, user-friendly access to:

- basic expertise in fields alien to the user
- state-of-the-art physical simulation software
- interactive crystallographic and molecular modelling
- mathematical refinement of models using analytical results
- chemical, physical and spectroscopic databases.

The first generation of expert systems, for which reasonably developed commercial 'shells' are available, is capable of capturing to some degree the behavior of one expert in a well-understood area of expertise. In this respect our application is atypical because it will derive its strength from combining the expertise of a group of specialists. Moreover, part of the knowledge will actually be gained through the project. Therefore, we have chosen to first create 'toolboxes', mainly filled with conventional programs but with easy access to all the tools needed, to test and optimize possible analytical strategies manually. At a later stage, the experience gained will be translated into an expert system to guide the analyst through the system, enabling him to apply the best possible strate-

gies. Subsequently, expert systems will be used to actually solve appropriate parts of the problem.

Artificial intelligence technology is regularly used, especially in research environments, to solve 'puzzles' involving many pieces of data. Programs of this type are usually written directly in a computer language: Prolog, Lisp or in a conventional language like Fortran. Good examples in the field of chemistry can be found in references (1) and (2). The latter concerns a program by Bremser for the combined interpretation of NMR (Nuclear Magnetic Resonance), IR (Infrared Spectroscopy) and MS (Mass Spectrometry) results to find the structure of an organic molecule by searching large spectroscopic databases. In the CACSS project, this type of program could be used, for example, to derive a crystallographic structure from powder experiments using several techniques. Another advantage in this case is that diffraction experiments (with X-rays, neutrons or electrons) can be fully simulated.

A Case Study

To illustrate the strength of a multi-technique analytical approach and the kind of strategy which can be chosen, we will briefly describe the characterization of a titanium/vanadium denoxing catalyst. This work was carried out in our laboratory and was used as a case study by the SAS group (3).

The TiVO_x Catalyst. Industrial waste gases frequently contain nitrous oxide emissions (NO_x). This pollutant can be removed by co-proportionation with ammonia in the presence of a suitable catalyst. Vanadia (vanadium oxide) is an active material to this end and titania (titanium oxide) is often used as a support because it also has a promoting effect. To improve this catalyst, and to make it cheaper, titania and vanadia were deposited on a high-specific-surface-area silica. The most successful version was obtained by co-impregnation of the active compounds.

To gain insight into the working of the catalyst in order to further improve it, an analytical characterization was carried out. The experiments focused on the surface of the catalyst where the catalytic reaction takes place. Some surface-sensitive techniques to consider, depending on the material and the kind of results desired, are AES (Auger Electron Spectroscopy), EELS (Electron Energy Loss Spectroscopy), LEIS, SAXRD (Small Angle X-Ray Diffraction), SEXAFS (Surface Extended X-ray Absorption Fine Structure), SIMS (Secondary Ion Mass Spectrometry), STM (Scanning Tunneling Microscopy), UPS (Ultraviolet Photoelectron Spectroscopy) and XPS. The models derived from the surface technique(s) subsequently have to be reconciled with the results obtained with the bulk solid-state techniques.

Surface Characterization. In the present case we started with XPS which, due to the low energy of the ejected electrons, yields results on the top 5-15 atomic layers. Since the energy of the electrons is characteristic of the element found, and even of its valence state, a semi-quantitative picture of the surface coverage of elements can be deduced. If accurate results of an overall elemental analysis are available and a set of models is applied: for example, one of the models states that vanadia is only found mixed with titania, then the analytical results can be quantified and

particle sizes estimated (4). All these models predicted low dispersions for the active materials. The XPS signal ratios relative to silica were rather low, and consequently rather large particles (20–30 nm for the mixed phase model) were calculated. Furthermore, no indications were found of Ti and V being in a valence state different from 4+ and 5+ respectively.

Using a complementary technique, Transmission Electron Microscopy (TEM), the particle sizes were measured in order to choose the best model(s). In this way, the lateral resolution of one technique (TEM) is combined with the surface sensitivity of the other (XPS) to yield some parameters. In this case, however, the results appeared contradictory. Measuring with a resolution of 1 nm, TEM could not find titania or vanadia particles and with EDX (Energy Dispersive X-ray analysis) it was corroborated that, even on a smaller scale, the metals were evenly spread through the catalyst.

Hence at least one of the basic assumptions used in the set of models for quantitative XPS had to be incorrect in the present case. Most likely the XPS signals for Ti and V are attenuated by a third compound. The only other material present is silica from the support. No model covering this combination of compounds is included in the standard set. The new Ti/V/Si model had to be proven with another analytical technique. Sputtering of the possible silica overlayer in a LEIS experiment was chosen. This technique has a surface sensitivity superior to XPS, is sensitive to essentially only the topmost atomic layer, but is not capable of separating the Ti and V signals. LEIS proved that the main fraction of Ti/V was buried under a thin layer of silica.

Similarly, a combination of powder XRD (X-Ray Diffraction) and EXAFS (Extended X-ray Absorption Fine Structure) was used for the structural characterization of the phases of titania and vanadia. A relatively rare, mixed phase was found, which may explain the high catalytic activity.

Conclusions from the Case Study. Exercises such as these are quite common in the characterization of complex solids and do indeed require the combined expertise of a group of specialists. The set of techniques required varies from case to case, but the more or less standard combination of two or more complementary techniques as part of the arsenal is very useful. In retrospect, we were able to identify the techniques which were crucial to solving this problem: XPS/TEM, LEIS, XRD and EXAFS. A number of others (Magic-Angle-Spinning NMR (MAS-NMR), Raman Spectroscopy and FTIR (Fourier Transform Infrared Spectroscopy) were applied, but did not add significantly to the final result. The study of various samples which were synthesized in different ways and which showed different catalytic activities did prove relevant, but is not described in detail here.

From case studies like this we learn how to use complementary techniques and how to choose the most promising starting set. It is clear that in the present case the use of only one technique (XPS, TEM or LEIS) would have led to erroneous, or at least very incomplete, results on the surface topology of the catalyst. Only the use of a complementary set of techniques and the combined interpretation of their outcomes yielded a trustworthy model.

Outline of the CACSS Project

The process of characterizing a complex solid via a truly multi-technique analytical approach has been examined, case studies like the one described above providing valuable information. The result is a general outline for the CACSS project which we will describe in some detail here. It reflects our basic ideas about how to proceed through the analytical process in the optimum way, guided by an expert system when appropriate.

In Figure 1 the different stages of an analytical exercise and the necessary utilities are illustrated schematically. The left-hand sequence of actions resembles the procedure described above. The large box on the right symbolizes the most important computational facilities the analyst(s) may invoke during all stages of an analytical process. In the following pages we shall describe the symbolic framework of CACSS, discussing the different steps and the decisions to be made.

The Choice of the Optimal Starting Set of Analytical Techniques.

Seldom is a solids and surfaces problem presented in pure analytical terms. Therefore, the first step is to derive a working hypothesis which can relate the observed behavior of a material to its possible composition and structure. This should result in a 'microscopic model', which can range from a complete, quantitative proposition to a mere classification of the problem area. The detail of the model, in general, reflects the detail of the desired analytical result. This step, an a priori definition of a working hypothesis, is an important facet of the CACSS approach. It is an efficient way to make explicit the existing knowledge of the material and the exact nature of the analytical problem. At the same time this model is conveyed to the computer, enabling an automated search for an optimal set of analytical techniques.

Even in this first step the CACSS system can be used beneficially. The modelling of structural proposals (in the case of a newly synthesized crystalline material, for example) can be performed with the aid of powerful 3d graphics. To follow the example further, one might choose a known structure from a crystallographic database as a first model, to be refined during the process. However, anticipated structural changes can be made at the modelling stage and can be combined with theory (5).

In the next step, the most efficient starting set of analytical techniques has to be constructed. This set should be able to produce information on the most prominent characteristics of the first model. This can be done by formalizing the heuristic reasoning process of an (ideal) analyst, making the problem a clear-cut case for a first generation (rule-based) expert system. The reasoning must include the set of parameters which has to be determined and the required precision, given the general model as 'facts' (e.g. the elemental composition). The system should contain basic information on most existing techniques and how they complement each other. This enables the user to select the most appropriate set of candidates (including ones unfamiliar to him/her). Other practical considerations like the availability of the technique, can be easily incorporated.

After having found one or more promising set(s) of techniques at the end of this heuristic process, the user can decide whether to

check the proposed strategy. To this end, CACSS will predict the outcomes of the candidate experiments. The degree of physical understanding underlying the various techniques will determine whether the full spectra or only some properties, which can be derived from the spectra in a data reduction step, are simulated. For some techniques a spectroscopic database can be utilized to predict a spectrum. The result may be that the selected set of techniques provides insufficient information, necessitating the return to the heuristic search step.

The Interpretation of Experimental Results. Once the user has deduced the set of techniques which can serve as a crucial test for the first model, the actual experiments have to be performed. The spectra of those techniques which can be fully simulated have to be transferred to CACSS via the laboratory computer network. This enables a combined interpretation, in its simplest form by using the results of one technique as constraints for the interpretation of the other. For the remaining techniques a standard format has to be defined in which the reduced data are presented to CACSS. These data are produced by specialized software on the dedicated computers of the analytical instruments. This approach is preferred to incorporating all the data reduction steps into the CACSS system because interpretation of the raw data requires the experience of the expert in the individual technique. If, however, a stage of interpretation has been reached where presumptions about the model of the material have to be made, the results have to be transferred to CACSS.

The experimental results can now be used for confirming or rejecting the first model. This involves checking the match between predicted and measured data. This step is critical because rejection implies that an entirely new model and strategy be developed. If the rejection of the original model was not justified, one could spend a long time following the wrong track before returning to the original model. However, if one has the evidence and the mutual implications of several techniques at hand, the risk of making such a mistake is greatly reduced.

The criterion for rejection is that the free parameters of a model cannot be adjusted such that all the experimental data are reproduced. If the model cannot be rejected, or fully confirmed, other decisive experiments have to be identified. Since, however, some of the originally free parameters are now fixed by the quantitative requirements of the experimental results, this has become easier. Once a model is regarded as being both sufficiently trustworthy and detailed, the problem has been solved.

The chance that a different model also fits the experimental data can never be fully excluded. The multi-technique approach, however, reduces this chance considerably.

For some cases, it is possible to use refinement techniques (6) to obtain very accurate structural data. This is especially true for the well understood diffraction techniques and, therefore, for characterizations of polycrystalline material. With an increasing level of physical understanding the number of techniques which are suitable for use in refinement is bound to grow. Because of the 'local minimum problem', the starting model should already be close to reality for single-technique refinements. We expect that the

implementation of a simultaneous, multi-technique refinement (7) will partially solve this problem because of a stronger convergence in the iteration process.

Implementation of the Project

The implementation of the full CACSS project is a major effort, requiring quite a number of man-years. Therefore, it was decided to first develop a useful subset of the system. The structural elucidation of polycrystalline material was chosen, because this task can be well separated and because this subset would be very useful to us now. Further, this smaller project will prove the feasibility of the overall project. In this respect, it can be considered as a case study for the full CACSS system. The project has been christened NUT, after the Egyptian goddess of the sky.

NUT will have a modular construction like CACSS itself (Figure 1). Structural modelling with the aid of 3d graphics will play a more central role here. The creation of a dedicated computer program for truly crystallographic modelling was the first sub-project we embarked upon two years ago. Work is also under way to design and implement a framework to integrate this program, and existing conventional programs used for the interpretation of experimental results, into a 'toolbox' for the analyst. By using this system we hope to acquire sufficient experience on the most efficient way to use the multi-technique approach in this field. This experience will enable us to build an advisory expert system to guide the analyst through the toolbox. An expert system to find the optimal starting set of techniques is less important here because diffraction techniques always play a central role.

To enable us to gain experience in the use of expert systems in analytical chemistry, better understood areas have been chosen. An expert system shell is being used to develop a prototype of an expert system for selecting analytical techniques in the field of elemental analysis. In addition, a Prolog program is being developed to assist in the planning and interpretation of combined XPS and TEM experiments of heterogeneous catalysts, as described in the case study. This program tests the topological models, and combinations thereof, using the experimental results obtained by XPS, TEM and EDX. There are also rules implemented to check for violations of the basic assumptions made in the set of models. Finally, if required, advice is given on additional experiments.

Structure Elucidation from Crystal Powders. For many practical materials, such as polymers and zeolite catalysts, it is impossible to synthesize large crystals. Therefore the structure has to be found from powders. Powder XRD (preferably using synchrotron radiation) and neutron diffraction are the most important techniques, but experiments using other analysis methods like High Resolution Electron Microscopy (HREM) and Electron Diffraction (ED), MAS-NMR and EXAFS can add valuable information (8).

The first step in the characterization is indexing of powder diffractogram(s), if possible also using ED data. This yields, if successful, the parameters of the unit cell and, through extinguished diffraction peaks, indications of the possible symmetry groups. In the next step, the positions of the atoms in the

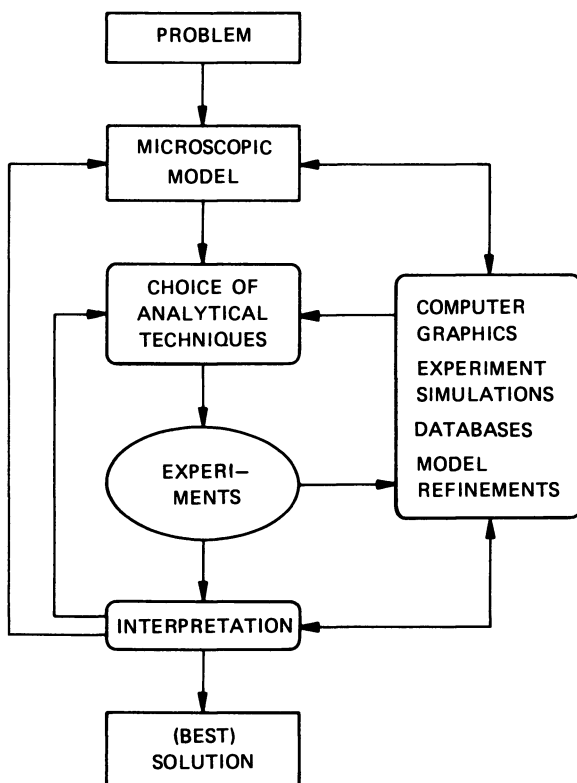


Figure 1. Schematic outline of the Computer-Aided Characterization of Solids and Surfaces project.

asymmetric unit have to be found. Several approaches exist for this. In favorable cases (9), with a large number of well-resolved peaks, 'direct' methods can be applied (10). Such methods, which iteratively seek the phases of the diffracted beams, have been developed for single crystal XRD. The result can be improved with Rietveld refinement (6), a least squares refinement of the structural model against the full diffractogram. This step can be best conducted on a neutron diffractogram, because neutrons mainly 'see' the atomic nuclei.

In many cases, however, one has to rely on trial and error methods to find the atom positions from the intensities of the dif-

fraction lines. Through simulations of the diffractograms one has to try to find the atom coordinates. In this process, data obtained by other techniques can be used as important constraints. Distance geometry (11) or other calculations (5) can be used to optimize structural proposals and databases can be used to find comparable structures (e.g. only differing in atom types and unit cell lengths).

To facilitate the whole exercise, bridging the gap between indexing and automated refinement, we apply a 3d crystallographic modelling program which is capable of conserving the parameters (unit cell dimensions and symmetry) which are fixed by the indexing.

Crystallographic Modelling. In collaboration with the crystallography group at the University of Amsterdam (Prof. H. Schenk), the program PLUVA for the modelling of crystal structures has been written (12). PLUVA runs on an Evans & Sutherland PS390 3d graphics computer. In contrast to molecular modelling programs, which consider a (molecular) structure as an entity, PLUVA uses the symmetry of the structure in the display itself. If an atom, or a group of atoms, in one asymmetric unit is moved interactively, all symmetry equivalent atoms move at the same time. In that way, the symmetry of the structure is preserved. The facilities to view the structure in three dimensions, manipulate it in real time, and display it in a convenient style (stick, stick and ball, etc.) provide a very powerful aid for structure elucidation from powders. In fact, the available search space has been made visible and directly accessible.

A powder XRD simulation program is coupled interactively to PLUVA, allowing almost real time simulation of a diffractogram from the structure on the screen and comparison with the experimental diffractogram. In this way a fast 'manual' refinement can be performed, with data introduced from other techniques serving as constraints. One possible constraint would be keeping two atoms together whose internuclear distance has been measured. In addition the analyst can apply his experience and intuition. For example, two atoms cannot occupy the same space (the task is in fact to fill a fixed volume with a fixed set of spheres), a fact which is very hard to take advantage of in direct methods.

A framework for NUT. To make the aforementioned 'toolbox' operational, a user-friendly environment is needed. For this purpose a framework is now under construction through which the user can model a crystal structure and invoke the various programs and utilities. This can be done with minimal knowledge about the way these programs are to be operated and without the distracting and time-consuming tasks of reformatting data and maintaining extensive file management. The data on a partially solved structure are stored in a database in a standard way, the history of the case being included. All kinds of facilities, from indexing programs to refinement, are invoked through pre- and post-processing routines which gather and transform all necessary data, only querying back to the user for unique data. The programs are called from menus and at every stage of the elucidation process the structure can be viewed and further modelled with PLUVA, including semi-real-time simulations of several diffraction techniques. The possibility of implementing an advisory expert system has been explicitly taken into account in the design of the framework.

Conclusions

Analysis of the multi-technique approach to the characterization of complex solids and surfaces has demonstrated the need for far-reaching computer assistance and, at the same time, that this need can largely be met by appropriate integration of systems available today. Expert systems and conventional programs will be combined in the Computer-Aided Characterization of Solids and Surfaces (CACSS) project. The strategies required to perform the analytical task in an optimal way are gradually being discovered during the project.

Implementation has started with NUT, which is the part of CACSS concerning the structural elucidation of polycrystalline powders. A framework to connect a large number of existing programs needed during the elucidation process is under construction. A truly crystallographic modelling program has been created which allows fast navigation through the remaining search space. In addition, a Prolog program is under development which is able to interpret, in combination, XPS, TEM and EDX results for surface characterization.

Literature Cited

1. Bremser, W.; Angewandte Chemie 1988, **27**, 247.
2. Liebman, S. A. In Artificial Intelligence Applications in Chemistry; Hohne and Pierce, Eds.; ACS Symposium Series No. 306; American Chemical Society: Washington, DC, 1986.
3. Kuipers, H. P. C. E., Boxhoorn, G., De Bruijn, D. P., Brongersma, H. H. et al.; submitted for publication in Applied Catalysis.
4. Kuipers, H. P. C. E., Van Leuven, H. C. E., Visser, W. M.; Surf. and Interf. Analysis 1986, **8**, 235.
5. Catlow, C. R. A., Mackrodt, W. C. Computer Simulations of Solids; Lecture Notes in Physics No. 166; Springer-Verlag: Berlin, 1982.
6. Rietveld, H. M.; J. Appl. Cryst. 1969, **2**, 65.
7. Maichle, J. K., Ihringer, J., Prandl, W.; J. Appl. Crystallogr. 1988, **21**, 22.
8. Cheetham, A. K.; Proc. Indian Natn. Sci. Acad. 1986, **52**, A, No. 1, 25.
9. Christensen, A. N., Lehmann, M. S., Nielsen, M.; Aust. J. Phys. 1985, **38**, 497.
10. Woolfson, M. M.; Acta Cryst. 1987, **A43**, 593.
11. Barrer, R. M., Villiger, H.; Z. Kristallogr. 1969, **128**, 352.
12. Driessen, R. A. J., Loopstra, B. O., De Bruijn, D. P., Kuipers, H. P. C. E., Schenk, H.; J. Comput.-Aided Molec. Design 1988, **2**, No. 3, 225.

RECEIVED June 9, 1989

Chapter 16

ES-EPA

Environmental Pollutant Analysis

Hikaru Hirayama¹, Robert Wohlsen², and Constance Brede²

¹Osaka Gas Research Center, 6-19-9 Torishima, Konohana-ku,
Osaka 554, Japan

²SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025

ES-EPA (Expert System for Environmental Pollutant Analysis) is an expert system for producing laboratory test plans, including the appropriate sampling methods, pretreatments, test methods and their order. ES-EPA generates test plans in a stepwise manner from abstract plans called "templates" to detailed plans, using a hierarchical planning mechanism. The knowledge base contains information on analysis items, test methods, test equipment, pretreatments and other necessary information. The prototype system has been successfully tested for various cases in the domain. The development of a delivery version has been completed and it will soon be used in the field on a daily basis to further verify its feasibility.

The Analysis Center of Osaka Gas Co., Ltd., performs laboratory tests to analyze environmental pollutants on about 1,000 samples per year. The goal of these tests is to measure the concentration and other characteristics of regulated substances in the sample. Prior to executing the tests, correct test plans must be made in order to get correct results. A test plan consists of sampling methods, pretreatment methods, test methods and their order. Test planning procedures for pollutant analysis are so complex that only experts who have been doing this job for more than ten years perform adequately. First, an expert must collect a broad range of necessary information on the sample: type, source, amount, sample conditions, purpose of the analysis and analysis items. The information needed varies with each case. Then, the expert makes a test plan using the information, his knowledge of the chemical characteristics of the analysis items, and the conditions and constraints for application of the laboratory test procedures.

A great deal of effort was expended in an attempt to write a manual for this job. It has not been completed because of the wide

0097-6156/89/0408-0200\$06.00/0
© 1989 American Chemical Society

variety of cases which must be described in it. For example, procedures for a very typical example of waste water analysis might be given in a 20 page manual. But if the sample contains a substantial amount of suspended solids, special pretreatment is required, while if you want know the amount and contents of the suspended solids, then you need two more bottles of the sample and additional test procedures for analysis of the solids. If the sample contains oil or sea water, yet other kinds of pretreatments and test procedures are required. There are many more conditions and requirements which may affect the test procedures. If you consider the various possible combinations, hundreds of different test procedures are possible for waste water analysis alone.

The development of the ES-EPA (Expert System for Environmental Pollutant Analysis) started in September 1986. At the beginning of the project, we spent two months on a feasibility study on the application of expert system technologies to this domain, and concluded that this problem perfectly matched the methodologies both of knowledge representation and inference of expert systems. The development of the prototype system took thirteen months and was completed in October 1987. The prototype system was implemented on a dedicated lisp machine and had limited knowledge, only enough for verification of the system concepts. It was tested by human experts for various cases in the limited domain and the generated plans were proved to be practical. The development of a delivery system with larger knowledge bases and better user interface but implemented on a conventional engineering work station, started in January 1988 and was completed in March 1989. It will soon be tested in the field on a daily basis. This paper describes the methods of planning and the implementation of the ES-EPA system.

Planning Methods used by Human Experts

After extensively interviewing human experts and doing experiments using an early version of the prototype system, we found two important characteristics of the test planning methods used by human experts in this domain. Firstly, experts do not always make a detailed plan. They make a plan at an appropriate level of abstraction, a rough plan for one purpose and a detailed plan for another purpose. Secondly, they do not make any plan from scratch. Experts have various kinds of "templates" which they use as the starting point of their planning or as part of a plan. These two characteristics are explained in more detail below.

Different Abstraction Level Plans. The experts in this domain can change the abstraction level of the test plan according to the amount of available information and the purpose of the plan. Below are three typical situations that require plans with different levels of abstraction:

(1) An analysis case starts with inquiries from a client about cost and duration. Because information about the sample at this stage is often incomplete and occasionally inaccurate, the expert usually makes a rough plan which is only enough to estimate cost, time and sampling instructions.

(2) When the sample arrives, the expert can get most of the information necessary to produce a detailed plan. At this stage the

detailed plan is necessary for instructions to the laboratory equipment operators. This detailed plan includes pretreatments, preliminary tests, final tests and their order. However, some parts of the plan may still be in abstract form if the necessary information could not be obtained prior to a pretreatment or a preliminary test.

(3) As the laboratory tests are carried out, the test plan becomes more and more detailed. It may be modified and added to according to the results of previous tests and pretreatments. There are three reasons to change the test plan: first, if new information is revealed after executing these tasks; second, if a measured value is outside of the range of the test equipment; and third, if a measured value is so close to the regulated concentration that a more accurate test method must be applied to further verify it.

Plan Templates. When a new class of test plan made by an expert works well, he seems to store abstract forms of the plan along with the conditions for using them. The next time he meets similar conditions, he starts planning from the stored abstract plan. He uses these as "templates" for test planning. This kind of stored abstract plan was discussed first by Friedland (1).

Templates range from general and abstract ones to specific and detailed ones. General, abstract templates are used to represent a large part of a plan or even a whole plan. For example, if a sample is some kind of industrial waste, a human expert starts planning with a very rough plan which is typical for industrial wastes, instead of making a plan from scratch. Later, he will modify this plan considering any atypical requirements and constraints of the analysis.

Specific, detailed templates are used for smaller parts of the plan. For example, if the test plan should include a series of procedures related to an IC (ion chromatography) test, a typical template for IC test procedures will be applied to the plan. These typical procedures for the IC test will be modified later, too.

Hardware and Software Environment of the ES-EPA System

The prototype system of ES-EPA was implemented using IntelliCorp's KEE and Common Lisp on a Symbolics 3600 series computer. Before we chose the development environment, we spent two months on a needs analysis. During this stage, we first made a conceptual design of the system, then we clarified the hardware and software requirements. Major requirements are: first, the software tool should have a frame representation, forward chaining production rules and an object oriented programming language; second, both hardware and software should be able to handle Japanese characters, especially kanji characters; third, the combination of software and hardware should be fast enough to handle large knowledge bases. We made very small toy prototypes using several different combinations of software and hardware, and did experiments on these to choose the prototype development platform.

For the delivery environment, we continued to use KEE and Common Lisp as the software platform, but chose a conventional engineering workstation, Sun 3/60, as hardware. Sun 3/60 is powerful

enough to deliver ES-EPA. We considered the possibility of using a 386 machine, but we gave up on this idea because Japanese language was not supported in KEE on 386 machines when we had to decide. If we have to choose less powerful machines, probably we have to consider the possibility of porting ES-EPA to a lighter software environment. Although we used KEE, we restricted ourselves to using only the fundamental functions of KEE, namely, the frame representation, the production rules and the object oriented programming. We did not use the KEE provided user interface and other facilities because we thought that it would reduce the amount of work when porting to a different environment became necessary.

Architecture and Knowledge Bases of ES-EPA

The system consists of three major modules: the Control Module, the Knowledge Base Module, and the User Interface Module. Figure 1 shows a schematic of this architecture. Arrows between modules represent the flow of information. The flow of the session and dialogue with the user are controlled by the Control Module. The Knowledge Base Module provides all the necessary information for making test plans: analysis items, sample types and purposes of analysis, analysis tasks and customers. All input and output, in various forms including menus, tables and graphics, are processed by the User Interface Module.

ES-EPA has four major knowledge bases in its Knowledge Base Module: Analysis Item KB, Sample Type and Purpose KB, Analysis Task KB, and Customer KB. The frame representation and its slot value are used for expressing declarative knowledge. Procedural knowledge is expressed as methods of the object programming language or forward chaining production rules. ES-EPA currently has about 1,000 frames and 300 production rules.

In the Analysis Item KB, analysis items of environmental pollutant analysis, such as concentration of elements or ions are represented by frames. Each analysis item frame has information about applicable analysis methods or equipment, legal regulations, appropriate material for sample bottles and other applicable information.

The Sample Type and Purpose KB contains knowledge about sample types and possible analysis purposes for each sample type. For example, there are many kinds of industrial wastes: polluted mud, slag, embers and so on. Possible analysis purposes differ according to the type of sample, and typical analysis items differ according to the analysis purpose.

The Analysis Task KB contains all kinds of analysis plan elements from abstract task components to specific tasks. Abstract components have information for plan refinement in the form of templates. Specific tasks include sampling methods, pretreatments, equipment tests, analysis methods defined in the Japan Industrial Standards and others. A specific task has information on the conditions and constraints of the task application, instructions for the task execution and so on.

In the Customer KB, an index of all past customers and the results of past analysis are stored. Past records are retrieved easily and used as references and sometimes as a large template of a new plan.

The ES-EPA Planning Mechanism

In order to have multiple representations of a plan with different abstraction levels, a hierarchical planning mechanism is necessary. The method is first to sketch a plan that is complete but vague and then gradually refine the vague parts into more detailed sub-plans until the plan consists of a complete sequence of detailed components (2).

A simplified schematic of the hierarchical planning mechanism of ES-EPA is shown in Figure 2. There are four representations of a plan with different abstraction levels: Level 1, 2, 3 and 4. In this figure, the rectangular boxes represent specific tasks that can not be expanded any further, and boxes with rounded corners represent abstract components.

In the ES-EPA system, all of the abstract test plan components have templates. Some abstract components have only one template, while others have multiple templates. If a component has only one template, the same template must be used for all situations. In this case, the template is stored as a default value in the Template Slot of the frame. An example of such a template is shown in Figure 3.

If a component has multiple templates, then the most suitable template for the situation must be chosen. In this case, multiple templates are represented as a set of production rules. The antecedents (the left hand side or the "if part") of this kind of rule represent the conditions for using that template, and the consequences (the right hand side or the "then part" of the rule) put the template in the Template Slot of the frame. Figure 4 shows an example of such rules.

On Level 1 in Figure 2, the most abstract test plan is represented by a single component which is always the Test-Plan-of-Sample. The Test-Plan-of-Sample has tens of different templates in production rule form. In this case, as the sample is some kind of water sample and the purpose is the ordinary waste water analysis, the Waste-Water-Task-Set-Rule shown in Figure 4 is applied.

On Level 2, as a result of the execution of the Waste-Water-Task-Set-Rule, the Test-Plan-of-Sample of Level 1 is expanded into a set of less abstract components: Sampling, Waste-Water-Task-Set and Report. The Report component is specific and cannot be expanded any more, but, Sampling and Waste-Water-Task-Set are still abstract components on this level. Both components have multiple templates as sets of production rules, and the most appropriate ones are applied.

On Level 3, Sampling is changed to a specific task which is Sampling-in-Plastic-Bottle. Waste-Water-Task-Set is expanded into four task components. The IC-Test-Set has multiple templates. Among the template rules, the IC-Filter-Rule is selected because there are suspended solids in the sample. The ICP-Test-Set has only one template as a slot value of the frame which is shown in Figure 3.

On Level 4, all the abstract components are expanded into specific task components. The hierarchical planning mechanism stops on this level. Then rules for detailed modification are applied to the plan. Certain components might be added or deleted after the application of these rules.

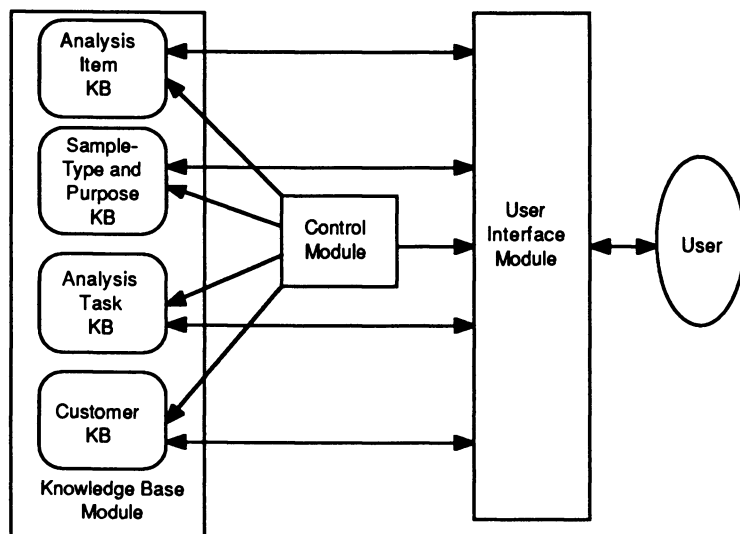


Figure 1. Architecture of ES-EPA

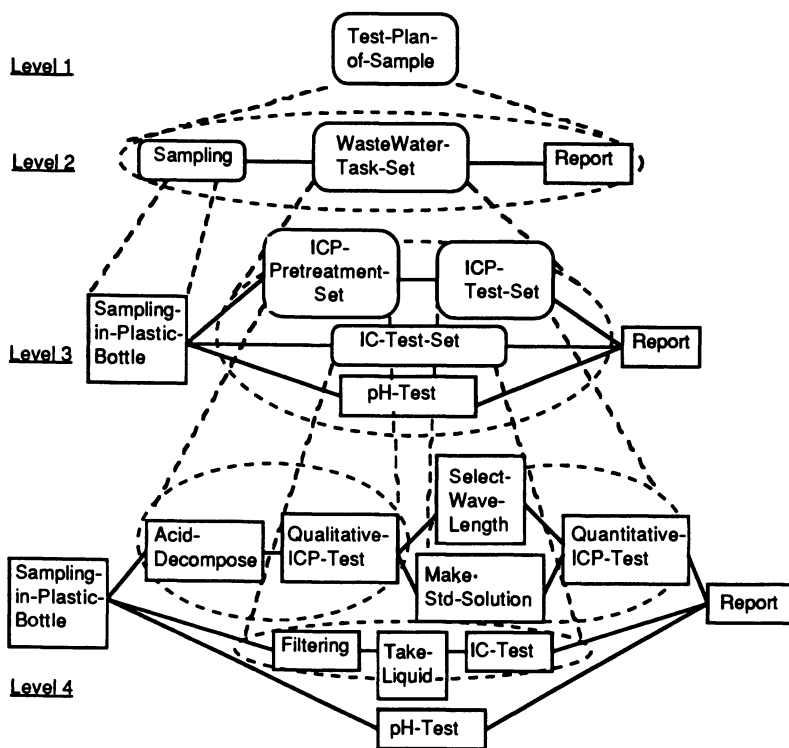


Figure 2. Hierarchical Planning in ES-EPA

```

(Unit: ICP-Test-Set
-----
(Slot: Instruction (Value -----))
-----
(Slot: Template
(Value (:Actions Make-Std-Solution
      Select-WaveLength Quantitative-ICP-Test)
      (:Order Make-Std-Solution Quantitative-ICP-Test)
      (:Order Select-WaveLength Quantitative-ICp-Test)
      (:Entry Make-Std-Solution Select-WaveLength)
      (:Exit Quantitative-ICP-Test))))
-----)

```

Figure 3. Single Template

User Interface of ES-EPA

Generally speaking, if an expert system is expected to be used in the real world on a daily basis, users demand better user interfaces and knowledge engineers tend to spend substantial amounts of time adding and changing user interface facilities. The User Interface Module of ES-EPA was developed not only to enable efficient interaction between the user and the system, but also to reduce the amount of effort needed for knowledge engineers to add and modify the user interface. The User Interface Module has many functions for generating input and output windows and these can easily be modified and combined. The module also has knowledge base editing facilities.

Figure 5 shows a typical layout of some user interface windows. Window 1 is the "ES-EPA Logo Window". By clicking this window using the mouse, the main menu, which is Window 2, will appear. Window 3 is the "Test Plan Graphic Window". It shows the order of tasks and represents the duration of each task by the width of the box. Window 4 is the "Table Window". In this example, the Table Window shows the test methods, analysis items and costs in table form. Window 5 is the "Menu Window". In this example, analysis items can be selected from this window. Window 6 is the "Help Window". The help window guides users through ES-EPA. For example, if a user cannot understand the meaning of a question from ES-EPA, he can get a detailed explanation of the question in this window by clicking the question.

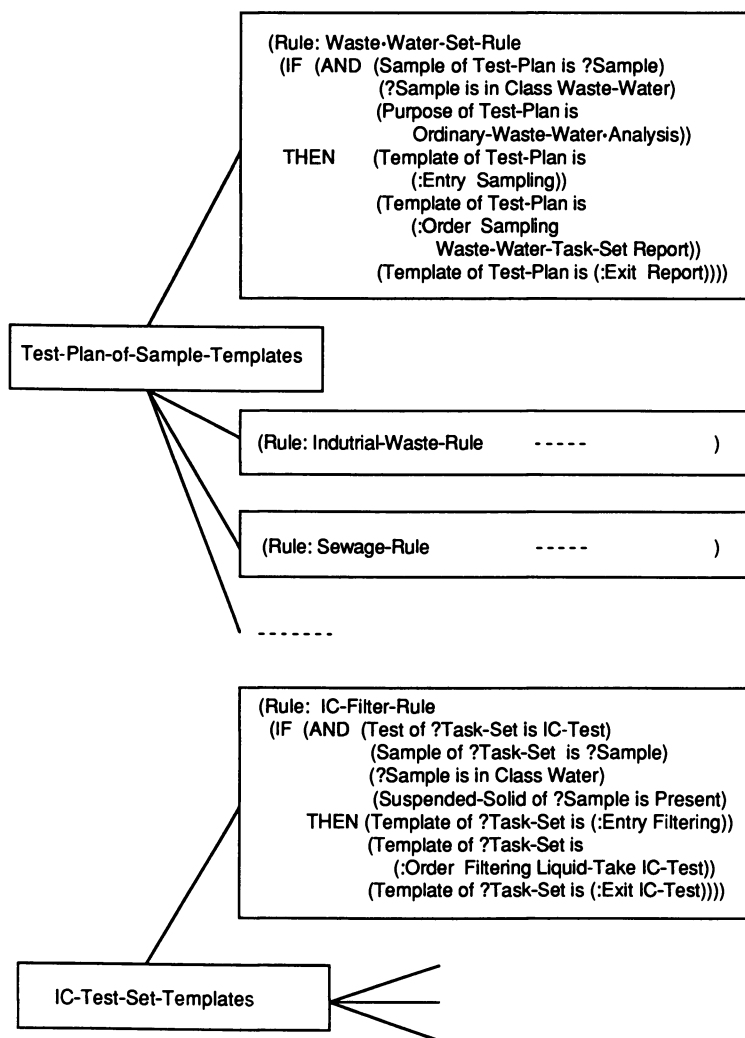


Figure 4. Multiple Template

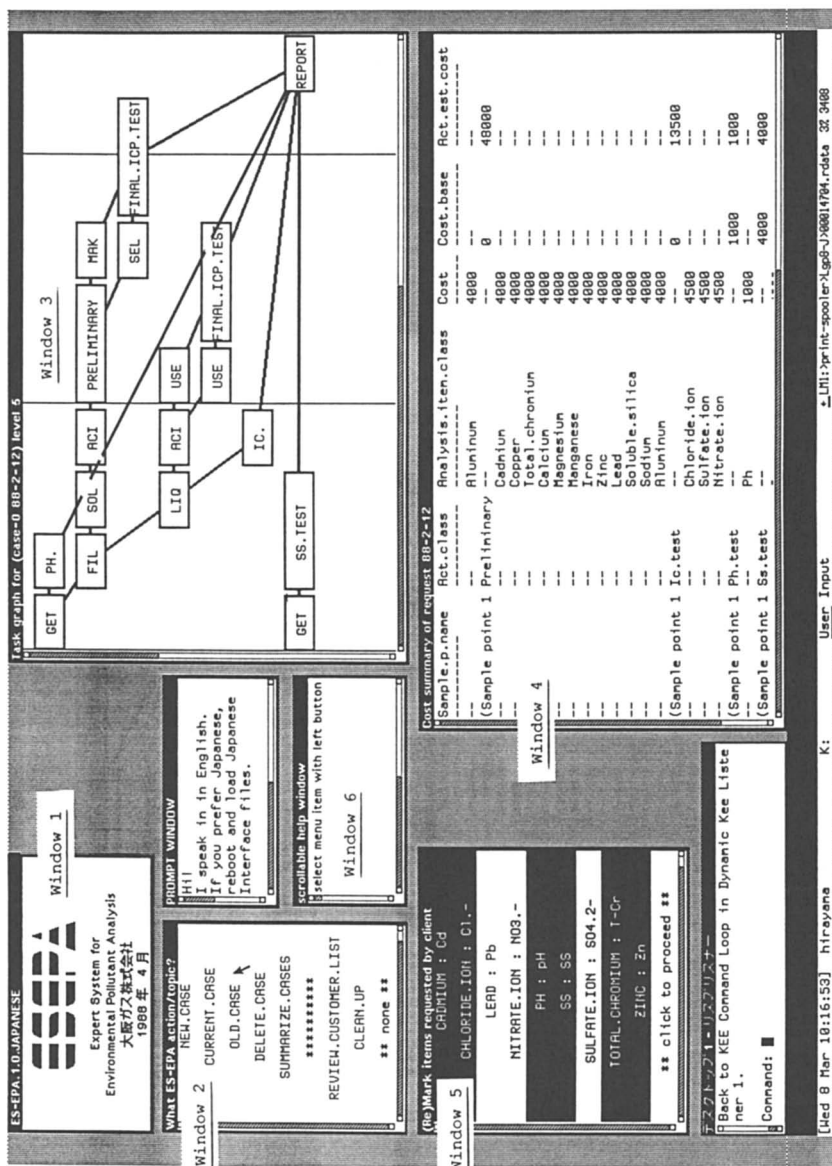


Figure 5. User Interface of ES-EPA

Conclusions and Future Directions

The prototype of ES-EPA was tested by human experts for various cases in a limited domain, and most of generated test plans proved to be practical. The hierarchical template planning mechanism was proved to be suitable for representing the knowledge of the human experts and also extremely efficient. The system can generate a complex plan consisting of more than 100 components in one minute on a Symbolics 3650 machine, which is substantially faster than the human experts.

The development of the delivery system has been completed and will soon be tested in the field on a daily basis. The tests will be done for all the domain and by all possible users. These tests will show us the real feasibility of the ES-EPA approach to this problem.

Among several problems of ES-EPA, maintenance of the system is the most serious. Although it has knowledge base editing facilities specially developed for ES-EPA users, adding or changing knowledge to the large expert system without causing contradictions is a very tough job. The maintenance and expansion of ES-EPA are still done by trained knowledge engineers. The development of a better knowledge acquisition facility is one of our future research goals.

Literature Cited

1. Friedland, P.E., Proc. 6th International Joint Conference of Artificial Intelligence, 1979, p 285.
2. Cohen P. R.; Feigenbaum, E. A., The Handbook of Artificial Intelligence; William Kaufmann Inc.: California, 1982; Vol. 3, p 517.
3. Sacerdoti, E., Proc. Third International Joint Conference of Artificial Intelligence, 1973, p 412.
4. Schank, R.; Abelson, R., Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates, 1977.
5. Iwasaki, Y.; Friedland, P., Proc. National Conference on Artificial Intelligence, 1982, p 341.

RECEIVED June 9, 1989

Chapter 17

ACexpert

Automated Metal Analysis by Atomic Absorption

William R. Browett¹, Timothy A. Cox, and Martin J. Stillman²

Department of Chemistry, University of Western Ontario, London, Ontario
N6A 5B7, Canada

A set of modular expert systems, which together form the program **ACexpert**, have been designed to provide real-time assistance to the analyst in completing metal analyses using automated sample introduction and atomic absorption spectrometry. Each of the modular expert systems of **ACexpert** performs a separate task that is modelled on existing laboratory procedures. The preliminary specifications of the overall controlling expert system, **ACexpert**, and four of its constituent modules, **ACmethods**, **ACdiagnosis**, **ACteach** and **ACcontrol**, are described. Prototypes of the first three of these modules have been implemented using the expert system shell, **KDS3**. **ACcontrol** was written in "C", within a Windows (Microsoft) environment, to provide extensive real-time control of the solution dispensing pump, autosampler, and the atomic absorption spectrometer.

Automation of analytical instrumentation, sample preparation, laboratory data management techniques, and process analytical chemistry (1-6), has been an important component in the improvement in productivity in the analytical laboratory. The integration of expert systems into the automation process will become increasingly commonplace for laboratories in which large amounts of data must be managed, and complex quality assurance programs maintained (7,8). When instrument control and data

¹Current address: Upjohn Company, 865 York Mills Road, Don Mills, Ontario M3B Y16, Canada

²Address correspondence to this author.

0097-6156/89/0408-0210\$07.50/0

© 1989 American Chemical Society

acquisition takes place in real-time, automation of the quality control steps becomes even more important. The development of automation in instrumentation has proceeded hand-in-hand with the move towards considering complicated, computer-controlled instruments as "black boxes". While automation of instrumental analyses can lead to dramatic improvements in productivity, the technicians who carry out the analyses often have little time to study the underlying principles being applied. As a consequence, the quality control functions which were previously performed by the analysts who were skilled scientists, must now be carried out automatically.

Expert systems can be used to replace the functions of the skilled analyst and at the same time maintain the productivity gains of instrumental automation. Expert systems will become more and more essential as increasingly complicated "black box" instruments are used in the analytical laboratory, and as audit control by regulatory agencies becomes more stringent. In our view, expert systems will both perform the analyses of the skilled technician and carry out the quality control duties of the analyst.

The development of **ACexpert** began with an investigation of the need for sophisticated instrument control software that would take advantage of newly-developed, graphical, user interfaces for low priced microcomputers. During the course of this work, it became apparent that although computerized control of individual instruments was being provided by instrument manufacturers, a much more critical problem was the integration of chemical expertise into the computerization of the instrumentation. Completion of this step would allow the completely automated collection of high quality analytical data to be carried out without constant monitoring by a skilled analyst.

At present, expert systems in the chemical laboratory are commonly used for separate, interactive applications involving consultations. These expert systems may be described as applying diagnosis or prescription, prediction, monitoring and control, planning, instruction, or interpretation (17). With the development of more powerful software for the microcomputer, there is considerable interest in integrating individual expert system applications into systems which will incorporate the rules of the expert system with numeric computation, database access, and simulation of the chemical reaction or instrumental procedure (9).

Chemists who are used to the development path of procedural programs for instrumental control or data analysis, will find the development path of an expert system to be a far more convoluted process. Expert systems should be viewed as evolving rather than being constructed as a one time program. The specification and prototyping stages encompass a much larger portion of the system development time. Furthermore, the

validation of the rules used in the expert system will pose some unique problems. Like the human expert, but unlike the conventional procedural program, the expert system must be expected to make mistakes, because judgmental information, and even imprecise information, is incorporated into the knowledge base of the expert system.

Recent advancements in the development of expert system shells suggested that these tools could be used effectively to integrate the analytical chemist's knowledge of chemical methods and quality control procedures, with the instrument control program. The expert system **ACexpert** has been designed to model the laboratory procedures used in the completion of metal ion concentration measurements. In this paper, we describe how the manual model of quality control and quality assurance, which is applied in the analysis of metals using atomic absorption spectrometry, can be used to outline the specifications of the set of modular expert systems that together make up **ACexpert**. Our experience in prototyping four of the system modules, **ACmethods**, **ACdiagnosis**, **ACteach** and **ACcontrol**, is also described.

Metal Analysis: The Manual Model

Metal concentration determination using the AAS technique requires the completion of several tasks, each of which utilizes extensive knowledge and expertise. The organization, operation and control of these complex tasks are ideally suited to the present generation of expert system shells. Although the decisions made by the analytical chemist during the course of a series of analyses are complex, the procedures and the methodologies used for the successful completion of the decisions are well understood. Before beginning work on an expert system to aid in instrumental analysis, we need to consider the following three criteria:

- (i) Is this a significant problem for which there is a demand for a solution?
- (ii) Are acceptable expert system development tools available?
- (iii) Is the expertise, which will be used to create the expert system, readily available?

We believe that each of these important criteria are fulfilled with respect to metal analysis and an analytical assistant that can dispense advice on the chemical methods to be used, on how to modify and improve analyses, and can also control and monitor data acquisition in real-time, can be developed at the present time.

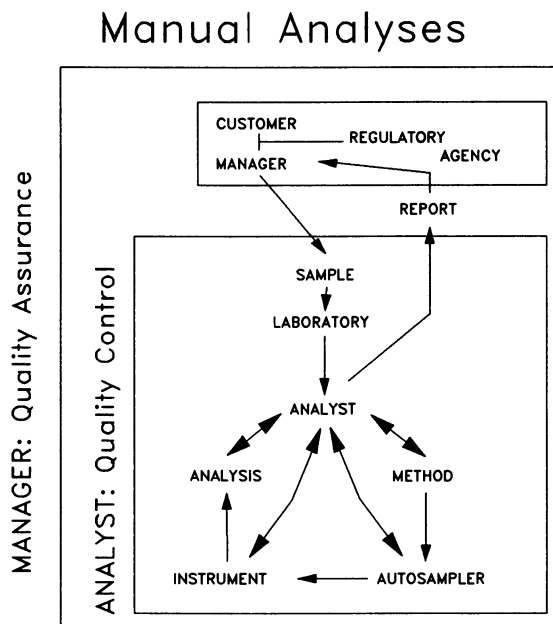


Figure 1. Manual Analysis for Metals. (Reprinted with permission from ref. 17. Copyright 1989 Pergamon.)

Our model of the laboratory procedures which are required for the completion of metal analyses, is presented in Figure 1. The normal flow of information and the sample through the analytical laboratory is illustrated diagrammatically with arrows.

The **MANAGER** and **ANALYST** make all the decisions and perform all the tasks that are involved in completing an analysis. Typically, the **MANAGER** consults with both the **CUSTOMER** and the **REGULATORY AGENCY** to determine the method that is to be used. In this model of the nonautomated system, the quality control and quality assurance programs, as well as the report writing, although assisted extensively by computers, are carried out manually as separate tasks by the **MANAGER** and the **ANALYST**. Once the analytical method has been defined, the sample preparation is completed by the **ANALYST**.

In analysis by atomic absorption techniques, a portion of the sample containing the analyte, will usually be placed in an autosampler for automated introduction into the instrument that measures the light absorbance and determines the concentration of metal by empirical comparison with a calibration curve for that analyte. The ANALYST checks the precision of the replicate readings for each analysis, as well as the precision and accuracy of the instrument over both the short and long term, in considering the quality control. Techniques to recognize and compensate for matrix effects, physical interferences, and spectral interferences, are among the ANALYST's expert knowledge. This specialist knowledge is used in combination with the methods of the quality assurance program to certify that the results and procedures are reliable.

The tasks of the analyst and the manager of the laboratory, as outlined in Figure 1, are redefined with the incorporation of expert systems into the laboratory procedures. Initially, the tasks which are to be performed by the expert system will be rudimentary and repetitive. Nonetheless, the manual performance of these tasks still requires the judgement and expertise of an analytical chemist or a skilled technician. An array of expert system modules can be incorporated into the normal functions of a fully automated laboratory information management system (a LIMS). The LIMS software would then be able to perform the following operations.

- (i) Reduce data transcription errors by having a single-point of entry for all data, and, through real-time data acquisition, eliminate the manual entry of analytical results.
- (ii) Reduce information processing turnaround by automating tedious tasks that are more precisely carried out by computers, such as reviewing calculations and performing data transcriptions.
- (iii) Optimize personnel resources by shifting the most talented staff from easily-automated tasks, such as data review, to solving problems that require the expertise of the expert.
- (iv) Supervise and improve the quality of the sample analysis achieved, by providing expert assistance to the technicians and laboratory managers through the quality control and quality assurance programs.

Many of the more rudimentary quality control and quality assurance tasks will then be shifted from the analyst to the expert system.

Metal Analysis: ACexpert

The model of laboratory procedures shown in Figure 1, was translated into the hierarchy of expert systems shown in Figure 2. **ACexpert** encompasses a complete system that will be used to carry out automated metal concentration determination. Each expert system will perform a separate task and each will be capable of acting independently or as a module of the full system.

ACexpert is subdivided into two managing expert systems, **ACassurance** and **ACanalyst**, which in turn are subdivided into the individual expert system modules that perform specific tasks. As in Figure 1, the flow of information, and the movement of the sample through the analytical laboratory, begins with the receipt of the sample from the customer by the manager, Figure 2. Each expert system is listed in bold typeface and is named beginning with the letters "AC".

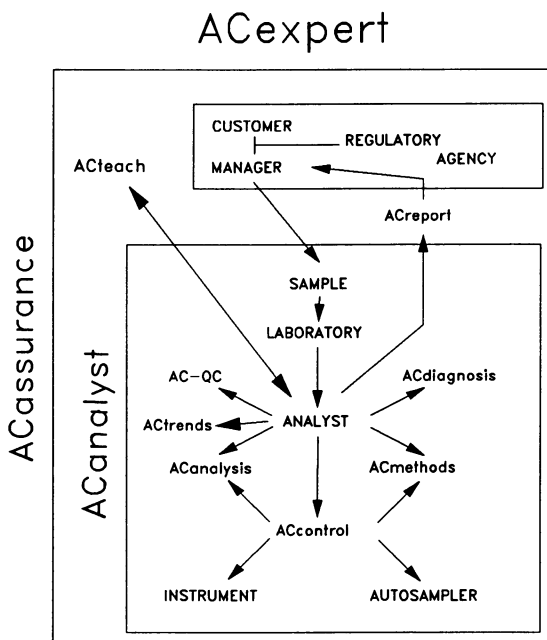


Figure 2. ACexpert: A metal analysis expert system. (Reprinted with permission from ref. 17. Copyright 1989 Pergamon.)

In this model, the ANALYST's role fundamentally changes from one of a technician to that of a supervisor of the expert system, **ACanalyst**. However, the MANAGER's role remains the same, although he or she now has the assistance of both the ANALYST and **ACexpert** in monitoring the

completion of analyses. **ACexpert** can be seen as an expert advisor which can also carry out the mechanical and electronic aspects of the advice and present these results to the ANALYST.

ACassurance is the quality assurance expert system that will be used by the **MANAGER** and the **ANALYST** to assist in the execution of a laboratory quality assurance program. Incorporated into this expert system are modules that will be used to provide instruction and advice to the laboratory personnel in carrying out their tasks and completing the analysis procedures. **ACanalyst**, the quality control and process control expert system, will comprise five modules that will be used in methods selection, process control, analysis, fault diagnosis, and quality control.

In the way that we have specified the **ACexpert** system, the **CUSTOMER** and **MANAGER** will begin using the **ACassurance** expert system to assist them in selecting the appropriate methods and analysis criteria that are consistent with the requirements of the **REGULATORY AGENCY** and the capabilities of the laboratory. The analysis criteria, together with the recommended methods, will then be available to **ACanalyst** for use in the selection of specific chemical and instrumental procedures. **ACcontrol** will carry out the metal determination of each sample. In normal operation, the **ANALYST** would manually confirm the method selected by **ACmethods** and monitor the performance of the analyses. All other aspects of the analysis would be carried out automatically, although still under the direct supervision of the **ANALYST**.

Design Specifications for ACexpert

Our initial objective in the development of **ACexpert**, was to develop a system that could offer 'expert' assistance during the analysis of metals. We also intended that the modules developed here for atomic absorption spectrometry would become more generally usable for analysis by other instruments.

In detail, the **ACexpert** system will provide (i) advice on methods selection, (ii) diagnostic information for improving analytical quality, (iii) a simulation role for teaching users how to optimize their analyses, (iv) real-time control of solution preparation, sample dispensing, and sequence control, (v) control of the instrumental settings and data acquisition, and (vi) an estimation of the quality of the data obtained by the instrument. Bearing in mind that the program must be used by experts and novices alike, these requirements impose quite severe design constraints on the system. The system must be able to solve problems based on a knowledge base of facts and solution strategies. At the same time, the human-machine interface, must also be able to describe in detail the current status of the analysis in an easy to read format.

Each of the modular expert systems will be limited to a relatively narrow domain of expertise. However, together, these modules will encompass the many varied aspects of decision-making that would be normally carried out by the human expert: from method selection, to real-time process control and then to reporting and documentation.

Implementation of the **ACexpert** concept as outlined above requires a system design process that includes thorough prototyping and validation before a production system can be completed. The development of **ACexpert** proceeded through several stages of prototyping. As starting points, we selected for development the control of the instrument and solution handling, and the setting up of the expert system knowledge bases for methods development and diagnosis. The instrumental control program and the user interface development proceeded along a traditional route using the Microsoft Windows (MS Windows) multitasking and graphical interface environment, with the procedures written in the C computer language. We chose to use expert system shells to aid in the development of the knowledge base (10,11), because shells contain specific strategies for knowledge representation, inference engines, and a database access interface.

Design of an Expert System using Expert System Shells

The development will proceed through four stages (10):

- (i) Problem determination and specification.
- (ii) Development of the initial demonstration prototype.
- (iii) Development of the expanded prototype.
- (iv) Completion of the deliverable system.

The first stage requires the completion of a detailed description of the concepts. This stage is also required for the identification of criteria and case studies that will be used to produce a small version of the expert system in order to demonstrate its overall feasibility. The specifications and concepts, which represent the knowledge that is to be captured, are established by describing the key concepts, interrelationships, and the flow of information that is needed to describe the problem-solving process of the expert system (12).

Three of the more important types of expert systems, which are used in the **ACexpert** system, are the consultation, interpretation, and monitoring and control expert systems. **ACmethods** and **ACdiagnosis** can be identified as consultative and interpretive types of expert system (17). The monitoring and control, and interpretation, types of expert

system are combined into the analytical process control expert, **ACcontrol**. The quality control and quality assurance expert, **ACanalysis**, will be based upon an interpretation type expert system.

Each of the modules is made up from four essential components: (i) a knowledge base of facts and rules, (ii) a problem solving inference engine, (iii) a knowledge acquisition module, and (iv) an explanatory interface (13). The knowledge acquisition module and interface module will be very different for the monitoring and control experts, compared with the methods selection or data interpretation experts. The knowledge base for each expert will contain symbolic knowledge, such as a description of the objects, their attributes, and corresponding data values. As an example, the **ACcontrol** knowledge base contains a description of the instrument components, ranges of valid settings, and a list of error codes. In addition, the knowledge base contains rules that consist of judgmental knowledge, such as the relationship between the stoichiometric ratios of the fuel and oxidant gas flows and the stability of the flame, and the dependence of the sensitivity of the spectrometer on the stoichiometric ratios of the gas. The inference engine is the component of the expert system that allows rules and logic to be applied to facts in the knowledge base.

The rules may be analyzed by two different strategies, forward-chaining and backward-chaining that may be constrained by selecting a search technique. Forward-chaining solves problems by asserting new facts or conditions, and examining the consequences or conclusions. The backward-chaining inference strategy attempts to solve problems by hypothesizing a conclusion and then examining the truth of the conditions that are associated with the conclusion.

The KDS3 expert system shell

Prototype versions of the **ACmethods**, **ACdiagnosis** and **ACteach** modules were developed using the expert system development shell named **KDS3** (from the KDS Corporation, 934 Hunter Road, Wilmette, Illinois, USA 60091).

KDS3 is an inductive expert system shell that requires that the developer specify expert knowledge in terms of examples of conditions and conclusions. The relationship between each condition and conclusion is used by **KDS3** to construct a set of rules. This tool, like many other inductive tools, generates rules from examples. The development system then uses an algorithm to convert the examples into rules and to determine the order that the system will follow when questioning the user and making a recommendation. Unlike other production rule-based systems, **KDS3** does not deal with attributes or values, but only with conditions, which are input into **KDS3** as text strings that are either true, false,

or known with some degree of certainty. To KDS3, every entry is simply a string of text that has some factors associated with it and a certain place in the logical hierarchy, as outlined in Figure 2.

Expert systems developed with KDS3 are set up by the developer in much the same way that they are used. The developer is prompted with questions about analysis conditions, until a conclusion, in the form of an analytical method, is reached. During this stage, if the KDS3 system selects an analytical method that is not the analytical method that the developer had in mind when the questioning began, the developer is prompted for a new analytical method by the KDS3 system together with the condition that would distinguish the analytical method initially selected by the inference engine in the KDS3 shell, from this separate analytical method. The developer continues this process until all the analytical methods are considered and distinguished from each other. In this part of the development of the expert system, the KDS3 shell is used to construct and test the rules and facts that will become part of the system knowledge base. Once completed, each KDS3 module can be used as a stand-alone expert system, as well as part of a larger system.

The KDS3 environment provides the developer with substantially the same interface and performance characteristics that the user will encounter. Therefore, testing is an on-going process. In general, the system developer or user simply answers YES, NO, or ? (don't know or don't care) to conditions that are presented, which are already assigned in the program, according to a specific case history that is being considered. If no conclusion can be found with the answers provided to the conditions, or an ambiguity arises because of an answer during the system development process, then the developer is requested to supply the answer.

An example of the use of an analytical methods selection expert which was developed with the KDS3 system is shown in Table 1. In this example the user is attempting to select a method for the determination of total sediment aluminum concentration according to the requirements of the Water Quality Branch of Environment Canada in Ottawa. From the main menu, aluminum, Al, is selected as the analyte of interest. The presentation of a set of values in a menu is a useful technique in the KDS3 shell when mutually exclusive options are to be selected. The KDS3 shell then prompts the user with a series of questions that can be answered with Y(yes), N(no), or ?(don't know or don't care), based on the conditions set up by the developer.

The standard screen interface provided by the KDS3 shell, is shown in Figure 3. A series of these screens can be used to present information to the user that the developer considers to be important for the user to see in order to make the selection of the analytical method. In the

right hand panel, a further description of the condition, or additional information can be presented to the user. Further details concerning the use of the KDS3 system are described below under the ACdiagnosis heading.

ACmethods

ACmethods will assist the user in the selection of the appropriate method for the analysis of metals based on the procedures included in the Environment Canada NAQUADAT dictionary (14). **ACmethods** also offers standard "Cook Book" methods for multiple element analyses in which the module checks for interferences between the elements.

ACmethods uses a forward-chaining strategy and the following information as conditions: type of analyte, instrumental technique, sample type, regulatory agency, and if necessary, details of the method if this is necessary in order to discriminate between similar methods.

Table 1. Conditions and responses that are required for the ACmethods module implemented in the KDS3 shell, to select the Environment Canada NAQUADAT method number 13053 for the analysis of Al

#	ACmethods Statements	Answer
1.	Analysis of dissolved metal in an aqueous sample.	No
2.	Analysis of mineral acid-extractable metals.	No
3.	Use the flame AAS technique.	Yes
4.	Analysis of metal from the sample as a suspension.	No
5.	Analysis of metal in sediments.	Yes
6.	Metal extracted with solvent to eliminate interferences.	No
7.	Required by the Water Quality Branch of Environment Canada in Ottawa.	Yes

The final screen presented to the user of the ACmethods prototype implemented with KDS3, is presented below.

If question 7 in Table 1 was answered ambiguously with a ? (don't know or don't care), then two answers, which in this case represent two methods, would be left unresolved. The KDS3 system would warn the user that insufficient information was provided to resolve the differences, as seen in Figure 4. Since only one method, either method 13050 or 13053, can be used at any one time this ambiguity would have to be resolved before the other modules in the ACexpert system, or the human technician, could complete the aluminum determination.

<p>Title: Actinium to Carbon</p>	<p>Sample type: total sediments Sample prep.: Open digestion with HNO₃, and HF.</p>
<p>THIS IS MY BEST ANSWER. 13053 Al, Flame AAS, total sed., required by A-WQB-O.</p>	<p>Analysis: Flame atomic absorption by direct aspiration. measured spectrophotometrically at 309.3 nm and compared with identically prepared standards. An acetylene-nitrous oxide reducing flame is used. Detection limit: 1 mg/kg Required by: WQB Ottawa. **** Approved WQB Method ****</p>

Figure 3. ACmethods: Conclusion. (Reprinted with permission from ref. 17. Copyright 1989 Pergamon.)

<p>Title: Actinium to Carbon</p>	
<p>Possible conclusion(s):</p> <p>1 13050 Al, Flame AAS, total sed.</p> <p>2 13053 Al, Flame AAS, total sed., required by A-WQB-0.</p> <p>I have been unable to resolve 1 condition(s) either directly from the answers or through inference. As a result please regard the above as conjecture, not advice.</p>	<p>You may press the BACKSPACE key if you wish to go back and see unresolved conditions which you previously answered with a "?".</p> <p>Conditions to which you already gave a Y or N answer will not be asked again except the last one.</p>

Figure 4. ACmethods: An ambiguous response. (Reprinted with permission from ref. 17. Copyright 1989 Pergamon.)

In this example the expert system presents the user with both menus and individual questions during the course of the session to select the appropriate method to be used. In a more complex expert system, the KDS3 standard interface could have been replaced with a customized user interface or an instrument interface, if real-time data input and analysis, were desired.

When ACmethods is integrated with the other modules of ACAnalyst, such as ACcontrol and ACanalysis, the information presented in the right hand panel of Figure 3, which describes in more detail the procedures to be carried out, will need to be incorporated into a database that each of the expert system modules can access.

ACdiagnosis

The first version of this routine was incorporated into AAexpert, an expert system developed by Jim Stanton and Mohamed Moussa as part of their 4th year B.Sc. Honors research projects at the UWO, using KDS3+ linked to Quick Basic (Microsoft) 4.0 for graphical displays. ACdiagnosis will solve both instrumental and chemical problems encountered with flame atomic absorption spectrometry analysis. The instrumentation section solves problems with solution transport caused by factors such

as high viscosity, a burner blocked with salt, or a flame that is too oxidizing or too cool, by interrogating the user about the quality of the signal. From the conditions and cases describing the operation of the AAS instrument itself, the KDS3 inference engine generated 92 rules. Addition of conditions and case conclusions that describe chemical interferences increased the rule base to 240. The module can now diagnose both instrumental and chemical problems.

Conditions used in ACdiagnosis

25 conditions are used in the first complete implementation of ACdiagnosis. The statements are not set up as questions, simply as conditions of the system, as the user is requested to state, "True", "False", or "Don't Know", to each statement.

1. The calibration curve is sigmoidal.
2. You are analyzing for a readily-ionizable element.
3. Low sensitivity.
4. You are using a nitrous oxide-acetylene flame.
5. You are using an oxidizing flame.
6. Signals for the same sample are inconsistent.
7. The calibration curve levels off at high concentrations.
8. The slit width is correctly set.
9. The absorption signal is abnormally noisy.
10. The monochromator is correctly set.
11. The blank value drifts.
12. The signal has a slow rise time.
13. The signal value pulsates.
14. The fuel pressure is low.
15. Abnormal noise at 0.00 absorption with no flame.
16. Abnormal noise at 0.00 absorption with the flame lit.
17. Occasional pulses in the absorption signal.
18. There are bubbles in the aspiration capillary.
19. The sample solutions are viscous.
20. The signal integration time is less than 1 second.
21. A refractory element is present.
22. The spectrometer is in the single beam mode.
23. The hollow cathode lamp will not strike.
24. The hollow cathode lamp will strike, but no signal is obtained.
25. The burner is above 8 on the vernier scale.

As described below, these conditions, and their associated case-conclusions, are processed into a much larger body of rules by the inference engine in the KDS3 shell. The rules can be set up in a matrix so that the least number of answers supplied by the user, will lead to

a conclusion. The statements offered to the user follow a watershed condition. In the example shown here, the "Low Sensitivity" question is critical in separating out the various instrumental problems that may be solved by the module. To lead to a conclusion that the burner is blocked, the module asks the following questions. We should note that because of the structure of the rule base, answers to other questions were implied and so answer were not requested.

1.Statement: Low Sensitivity.

Response: True

Explanation: This is when you are unable to detect a weak signal normally observed with a sample.

2.Statement: Signal Pulsing Regularly.

Response: False

Explanation: Regular and consistent fluctuations during the integration period.

3.Statement: Pulse at the beginning of the aspiration.

Response: True

Explanation: An abnormally large absorption signal for a short period at the as the solution is aspirated into the flame.

Conclusion: The burner slot is contaminated, probably by salt build up. The module also suggests a remedy, both mechanical (clean the slot) and chemical (dilute the sample; choose a more volatile matrix).

Consider the generation of rules concerning the calibration curve. The first condition that we enter is "The calibration curve is sigmoidal", and the respective case conclusion for this being True could be "Ionization is occurring". If False, one conclusion could be: "There is no fault". KDS3 then generates the following rules: "If the calibration curve is sigmoidal is TRUE, THEN Ionization is occurring" and "If the calibration curve is sigmoidal is FALSE, THEN Ionization occurring is FALSE". While the generation of these rules is straightforward, because KDS3 is an inductive shell, it attempts to relate all subsequent conditions to this calibration condition. The next condition to be entered may be "The analyte is a refractory element". Because analysis of refractory elements by flame AAS does not lead to a sigmoidal calibration curve, after the appropriate case-conclusions for True and False have been entered, the development module will check the relationship of this condition to others already in the knowledge base.

In this simple example, a third rule will be generated "If the calibration curve is sigmoidal is TRUE, THEN The analyte is refractory is FALSE". The logic of this rule is implied and therefore, during operation of this module, only if the answer to the sigmoidal condition is FALSE will the refractory element condition be asked.

Clearly, generation of this type of rule can lead to problems. During development it is acceptable to respond to the questions linking conditions with "?", to imply, "There is no relationship between these conditions". Analysis of the rules generated from the condition and case-conclusions is an important step in checking the chemical logic used in this type of expert system. The best method of analyzing the logic is to analyze the truth table that links conditions and case-conclusions. In this table, "T", "F" and "?" show the relationship between the conditions. In many cases the KDS3 shell implied the logic based on answers that relate a specific condition to all cases already entered into the knowledge base. We show in Figure 5, the truth table for 6 of the 25 conditions and cases used in the prototype stage of ACdiagnosis.

Conclusion	Condition						
	sigmoidal curve	low sensitivity	inconsistent curve levels	signals off	noisy signal	bubbles in capillary	refractory element
1 ionization	T	T	T	F	F	F	F
2 self-reversal	F	F	F	F	T	F	F
3 contamination	F	T	T	F	F	F	F
4 viscous samples	F	T	T	T	F	F	?
5 slit too large	F	T	F	T	T	F	F
6 check lamp	F	F	F	F	T	?	T
7 integration time	F	T	F	F	T	F	F
8 no problem	F	F	F	F	F	F	F

Figure 5. Part of the conditions (top) and case-conclusions (1-8) used in the AAdiagnostic module.

ACteach

As the system has been developed from the specification stage, it has become clear that the addition of a "learning" module is valuable. Simulation of the effect of changing important instrumental parameters and chemical procedures on the absorption signal, greatly aids in the selection of these parameters at run time. Examples in the analysis

of metals, includes the choice of oxidizing or reducing flames, the choice of air or nitrous oxide as oxidant, and the choice of standardization method, either the use of a standard curve or through standard additions.

We have found that simulation of the effect of altering the flame chemistry on the signal obtained during analysis of metals, greatly helps in the understanding of the role of the chemistry in the production of atoms in the light path. **ACteach** can demonstrate via graphical displays how addition of an ionization suppressant will increase the signal observed for metals like K, Na and Ca. Similarly, addition of releasing agents will increase the signal observed in the analysis of Ca in the presence of phosphate.

ACcontrol: The Monitoring and Control Expert

ACcontrol is an expert system that must function in real-time, using a series of procedures to carry out instrumental control and data acquisition. This expert system must be structured so that it has access to the appropriate information at the right time from the sensors on the analytical instrument, and at the same time can make decisions about what to do next rapidly, usually within the measurement time of the instrument. Combining the monitoring, control, and interpretation tasks of the expert systems, requires that individual tasks be performed at the same time, either on separate computers, which can communicate with each other, or within a multitasking environment on a single computer. In either case, computational and reasoning efficiency is extremely important. Generally, the system must have complete information to allow the interpretation expert to return a definite answer to the controlling expert. The occurrence of incomplete information, long reasoning chains, unreliable data, and contradictory input information, will result in this fault-finding expert sending an alarm to the operator.

Instrument Control

Before the monitoring and control expert could be implemented, a traditional procedural user interface, a database manager, and an instrument controlling program, had to be written. The hardware of the **ACexpert** system consists of a Gilson 222 autosampler and a 401 dilutor, and a Varian 875 AAS, each controlled by an IBM PS/2 model 50 microcomputer. The autosampler and dilutor act in concert to provide solution handling capability, from which both preparation and sample injection into the AA spectrometer can be carried out automatically.

The user interface makes extensive use of graphical displays to control the instrument setup, sample setup, and metal determination procedures in an interactive environment based on Windows (Microsoft Corporation). This graphical user interface is similar to the type depicted as a "Human Processor Interface for ICP-AES" by Karanassios and Horlick (15) or as presented for the automation of a chromatography work station (16). Each of the MS Windows applications has access to all the features of MS Windows, such as pull-down menus, and mouse driven window and icon selection. Because menus are used, any number of these functions may be active at any time. All of these features are necessary to provide a simple to use, robust system for instrumental control. It is worth mentioning here that Windows is in fact not a true multitasking system because there is no scheduling program that ensures a particular application won't use all of the CPU time. Rather, it is up to the application to yield control to the other applications periodically. Figure 6 shows a schematic representation of the instrumentation under control. **ACcontrol** is taught where the beakers and test tubes are located so that it can set up the "Tray Layout". Using information provided directly by the user, or passed to it from other members of the **ACExpert** system, **ACcontrol** establishes the "Sequence Control", prepares the necessary "Standards" and "Modifiers", then "Dispenses" each solution. "Analytical Data" is acquired in real-time, and saved in the "Database". While these functions are interrelated, at run time, many of the tasks must operate independently and concurrently.

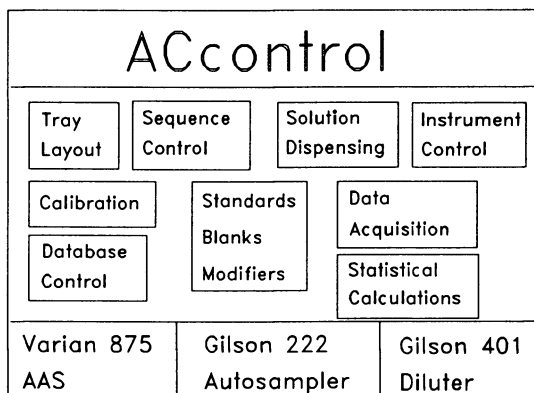


Figure 6. Instrument control by ACcontrol.

User and Instrument Interface Development

Instrument control, sample preparation, and sample injection into the AAS are performed by two MS Windows applications. The first, called AAS, provides the user with a "Control Panel" (Figure 7) that will allow complete control of the AAS operating parameters. Using a mouse, the user simply clicks the appropriate control button on the screen and the application (through a RS-232 serial line) sends the AAS the command. Instrument settings are saved on disk for future use.

The second application, called Sampler, is the heart of the instrumental control application. This application sets up three window procedures to handle (i) control of the Model 222 Autosampler and the Model 401 dilutor as slave devices, (ii) the sequencing of the samples to be analyzed within a batch, and (iii) the data values received from the AAS following every measurement. Control of the Gilson devices is carried out as follows. Each device is assigned a unit number on the GSIOC (a bus with specifications similar to the IEEE 488 specifications), where the computer acts as the master device. Sample placement is performed by providing the user with a picture of the rack they will use from template designs stored for a number of racks and beakers. The user identifies the appropriate rack, which is then drawn on the screen simulation of the Gilson tray. By using the mouse, the user designates which tubes contains the standards, samples, blanks, or controls. Containers for larger amounts of blank or reslope solution may be put on the sampler tray, these too are displayed on the computer screen in the way they appear on the sampler tray. Templates for specific tray layouts may be designed, stored and then printed out for future use. The data-feedback-window procedure reads the data that appears on the RS-232 line from the AAS and dispatches the data to the data output and quality control procedures. The quality control procedure will later calculate the relative standard deviation of the replicate data for each sample and decide if the sample should be re-analyzed.

Once tubes have been allocated to samples, standards or blanks, the database entries are completed with textual information for each sample, batch ID numbers can be added, and incremental numbers can be assigned automatically. The elements to be analyzed for in each tube are assigned from a default list of available lamps. For standards, the concentration is also given at this step, Figure 8. Advice, as "Help", is available from a "Cook Book" of standard procedures, arranged according to the lamps in use. At a later stage, the "Help" facility will be linked directly with the methods selection experts so that the user can readily confirm why a particular method has been selected. The Gilson pump can be set up with both flow rate and aliquot size, allowing full control over the aspiration of the sample into the flame.

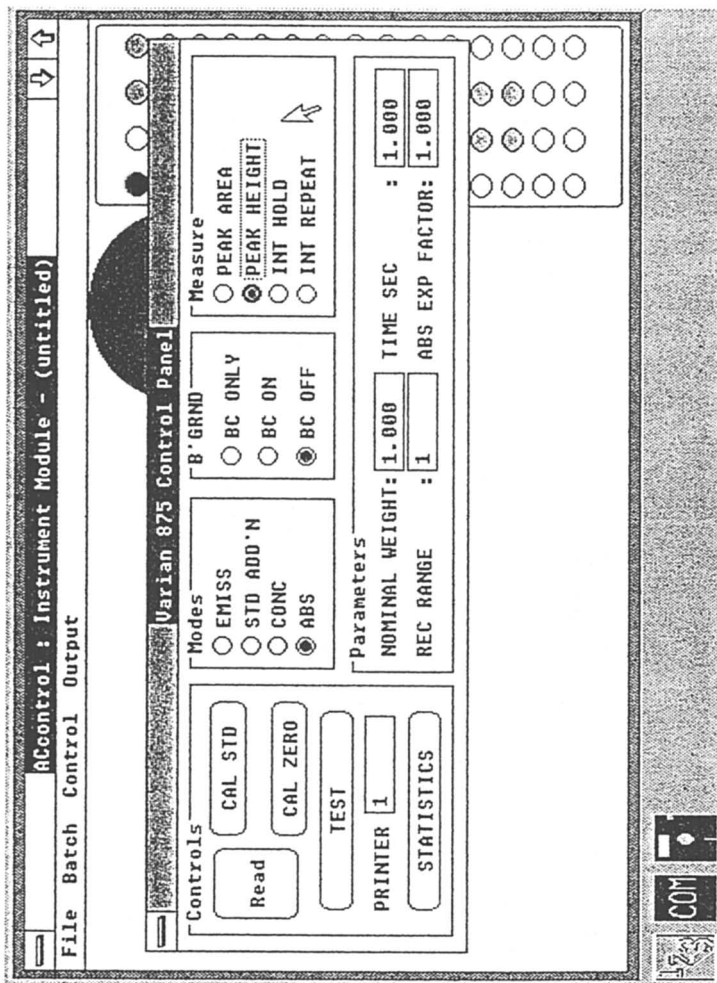


Figure 7. Software simulation of the front panel of the Varian 875 Atomic Absorption Spectrometer. All functions operate when activated by the mouse as if activated on the instrument itself.

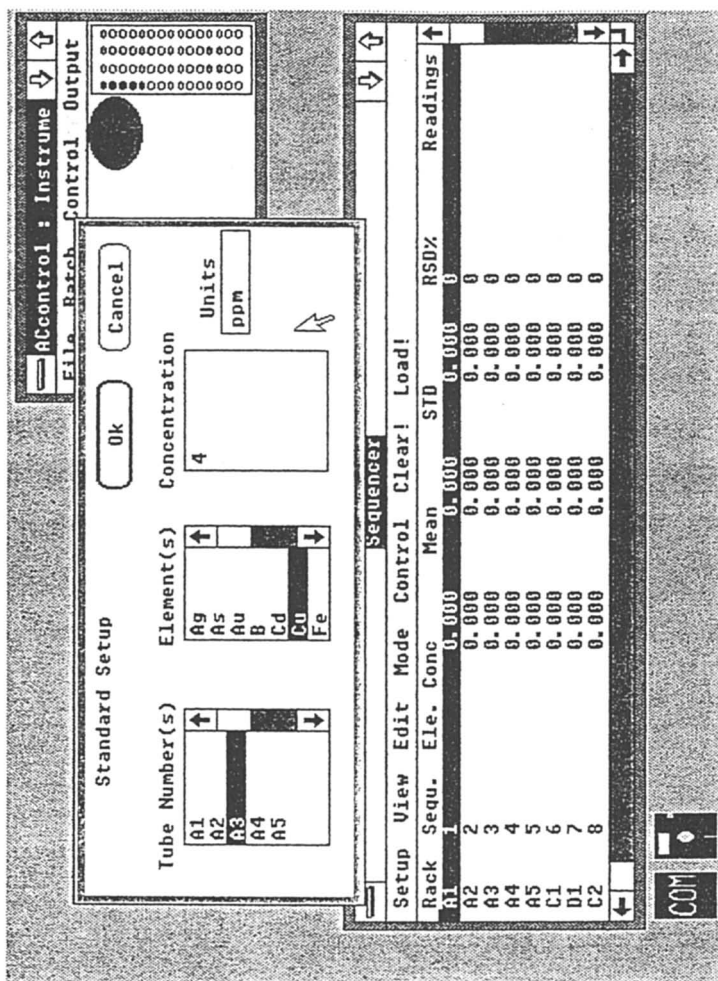


Figure 8. Assignment of the elements in each tube to be used to obtain the standard curve. The tubes can contain multiple elements, each with a different concentration. The concentration of each standard can be entered in one of several user-selected units.

The sequence control procedure, uses the entries in the database to arrange the operating sequence so that all tubes with the same element are analyzed in one batch. The user, or **ACcontrol**, can modify the sequence. For each analysis, the signal graphics obtained directly from the phototube response as the sample enters the flame, is recorded, Figure 9. This trace can be recorded from any point in time after the start of the sample selection by the autosampler, so that effects due to the pump operation on the flame stability can be observed. The graphical information can be stored and used in assessing the quality of the AAS itself. Control conditions, such as when air, water, or a standard is injected into the flame, can be used to check the operation of the instrument automatically during a run. Finally, the calibration curve is displayed and can be compared with stored averages of previous calibration curves for this element in this matrix, Figure 10.

The goal of **ACcontrol** is to provide a completely accessible control module such that information obtained from the other modules can be used to interrupt the assigned schedule allowing for repetitive sample analysis and unscheduled quality control programs to be started.

Discussion

We have described in this paper the design and first stages in prototyping of an extensive expert system that will operate in real time to control the measurement of metal concentrations. While our project is centered on metal analysis, the concepts described here, may applied to any instrumental analysis.

The future research involves encoding expert knowledge about the methods to be used for the analysis. This information includes the role of chemical modifiers, and interferences from other metals and the matrix. **ACmethods** will be the key module in the next stage of our work. We have to set up the chemical information in the knowledge base and develop a user-friendly interface that will allow the analyst to use the module as an advisor. **KDS3** provides a useful environment for coding the chemical information. We find that it is organizing the chemical knowledge that becomes the most difficult task, not the writing of the linking code.

Ultimately, one of the goals in the development of the expert system **ACexpert** is to incorporate the system into the automated control program of any instrument. This would require that the modules be sufficiently generic that only the control statements, and the specific chemistry encoded as conditions and cases would have to be changed. For this goal to be realized, the knowledge base would have to be created by some form of automated knowledge acquisition module.

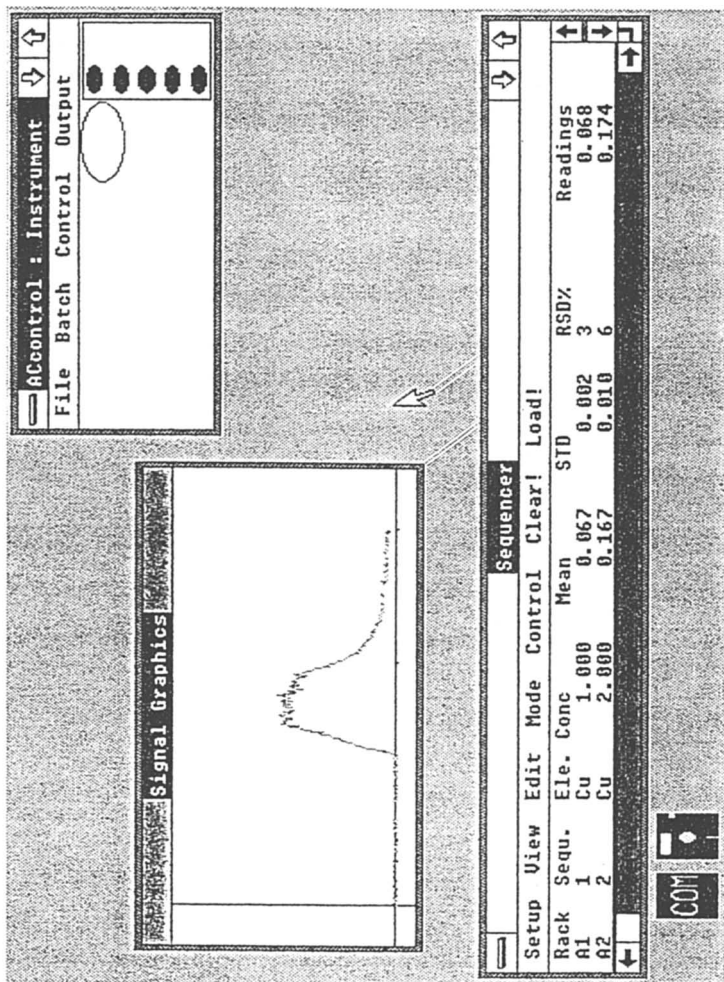


Figure 9. The phototube response showing the change in absorption as a solution is aspirated into the flame.

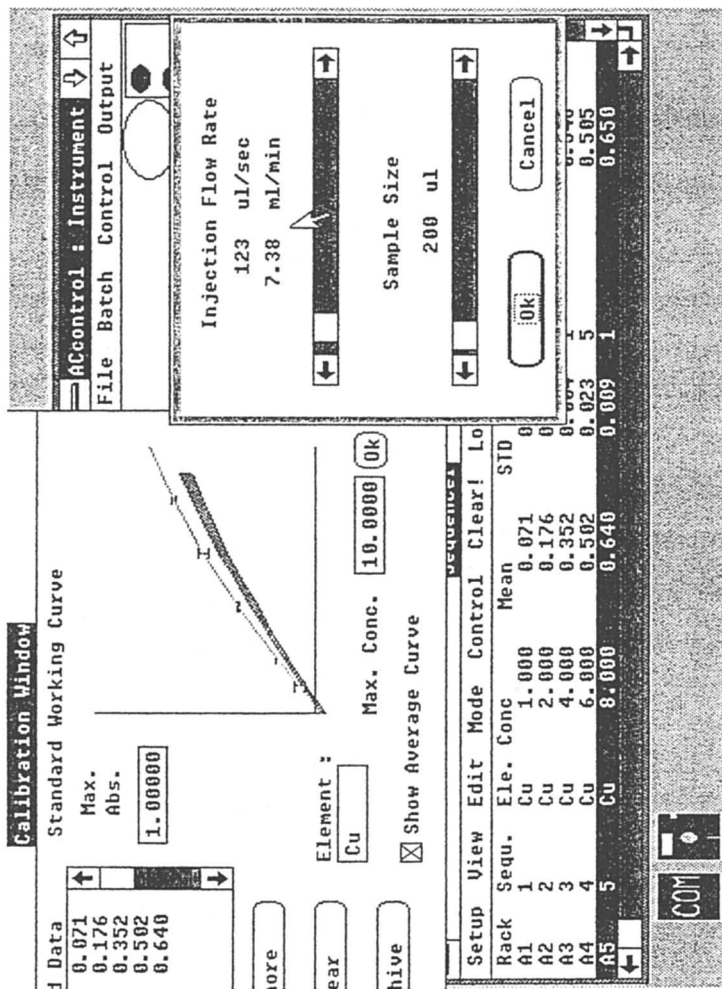


Figure 10. The calibration curve plotted as each standard is measured. The shaded curve represents the range of all previous calibration curves for this element that has been retrieved from the database. In this experiment, multiple aliquots of 200 μ l were injected into the flame.

In summary, we have found that the KDS3 shell provides an excellent prototyping environment. For the chemist, the inductive logic of the knowledge acquisition engine is an important feature as it results in early validation of the chemical rules generated. The rigorous logic that the developer must follow in order to create a usable expert system with the KDS3 system, results in a well-implemented system that is a learning experience for both the developer and the user. Experts use their expertise in a complicated manner and are often unable to define how a particular solution was arrived at. However, incorporation of their expertise into an inductive system like KDS3, in which the knowledge base is built up through step by step example-driven programming, provides a method for resolving the ambiguity of unclear rationalizations. An interfacing capability which allows several modules to be linked together, as well as the capability of linking to external programs, is essential in an expert system in which information must be acquired from and dispensed to a variety of modules.

Acknowledgements

We gratefully acknowledge the continuing support of this work by the Ontario Ministry of the Environment, and by the Academic Development Fund at the UWO. We wish to thank Jim Stanton and Mohamed Moussa for their contribution to the project. MJS is a member of the Centre for Chemical Physics at the UWO.

Literature Cited

1. Fucini, E.; Spadaro, G.; Pennisi, M. Anal. Chim. Acta, 1986, 191, 425.
2. Martin, J. M.; Ihrig, P. J. Applied Spectroscopy, 1987, 41, 986.
3. Granchi, M. P.; Biggerstaff, J. A.; Hillard, L. J.; Gray, P. Spectrochimica Acta, 1987, 42B, 169.
4. Berridge, J. C. Anal. Chim. Acta, 1986, 191, 243.
5. Bach, R.; Karnicky, J.; Abbott, S. in "Artificial Intelligence Applications in Chemistry", T. H. Pierce and B. A. Hohne, eds. American Chemical Society, Washington, 1986, pp 278-296.
6. Callis, J. B.; Illman, D. L.; Kowalski, B. R. Anal. Chem., 1987, 59, 624A.
7. Ennis, R. L.; Griesmer, J. H.; Hong, S. J.; Karnaugh, M.; Kastner, J. K.; Klein, D. A.; Milliken, K. R.; Schor, M. I.; Van Woerkom, H. M. IBM J. Res. Develop., 1986, 30, 14.

8. Smith, D. H. in "Artificial Intelligence Applications in Chemistry", Pierce, T. H.; Hohne, B. A., Eds., American Chemical Society, Washington, 1986, pp 1-16.
9. Gaines, B. R. in "Intelligent Simulation Environments", Luker, P. A.; Adelsgerger H. H., Eds., Society for Computer Simulation, San Diego, California, pp 144-149, 1986.
10. Citerbaum, R.; Geissaman, J. R.; Schultz, R. AI Expert, 1987, 2, 30.
11. Harmon, P.; Maus, R.; Morrissey, W. "Expert Systems Tools and Applications", John Wiley & Sons Inc., Toronto, 1988.
12. Keen, M. J. R.; Williams, G. in "Research and Development in Expert Systems", Bramer M. A., Ed., Cambridge University Press, Cambridge, pp 13-21, 1985.
13. Forsyth, R. in "Artificial Intelligence Principles and Practice", Yazdani, M., Ed., Chapman and Hall, New York, pp 186-199, 1986.
14. NAQUADAT Dictionary of Parameter Codes, Data Systems Section, Water Quality Branch, Environment Canada, Ottawa, 1986.
15. Karanasslos, V.; Horlick, G. Applied Spectroscopy, 1987, 41, 360.
16. Henshall, A.; Schibler, J. A. Amer. Lab., 1987, Sept., 42.
17. Browett, W. R.; Stillman, M. J. Progress in Analytical Spectroscopy, in press, 1989.

RECEIVED June 9, 1989

Chapter 18

Developing Knowledge-Based Systems

A Learning Process

Sterling A. Tomellini¹, Barry J. Wythoff¹, and Hugh B. Woodruff²

¹**Department of Chemistry, University of New Hampshire,
Durham, NH 03824**

²**Merck Sharp and Dohme Research Laboratories, Rahway, NJ 07065**

A great deal of effort and resources are being dedicated to developing knowledge-based systems in Analytical Chemistry. Creating such complex systems is not always a straightforward process. For example, during the approximately eight years which it has taken to develop PAIRS (the Program for the Analysis of Infrared Spectra), many complications have arisen and modifications made which were not originally anticipated. The knowledge gained while developing PAIRS indicates a number of difficult questions must be addressed before, during, and after creation of such a knowledge-based system. For example, questions concerning the source(s) of knowledge, the scientific merit of the project, the anticipated users of the system and their underlying knowledge of the area, the long-term durability and viability of the system, the methods to be used to evaluate the "final" product, and user accessibility of the knowledge base, must at some point, be addressed.

The Program for the Analysis of Infrared Spectra, PAIRS, is a knowledge-based system designed to assist chemists in determining which functional groups are likely to be present in an unknown compound based on its infrared (IR) spectrum. PAIRS can be further classified as a rule-based expert system. While PAIRS was certainly not the first such system to be developed, it was one of the pioneering applications of knowledge-based techniques in analytical chemistry. Many advances have occurred in computer resources during the decade since development work first began on PAIRS. Knowledge-based and expert systems have become the objects of much attention, largely due to this progress, which has placed far more powerful, less expensive, and easier to use computational tools in the hands of researchers. Watson and Mann (1) reported that 150 of the Fortune 500 companies collectively have spent an estimated \$1 billion on expert system development. They also cited predictions of \$5-10 billion in spending by 1990, \$30-40 billion by 1995, and \$50-110 billion by the year 2000.

The actual impact of this activity on the way scientists do research and the way people live their daily lives is only

0097-6156/89/0408-0236\$06.00/0
© 1989 American Chemical Society

beginning to be felt. For most people, the question, "Excluding the systems which you have developed, how many knowledge-based systems do you use routinely?", results in an answer of "zero". During the 1990s, it is expected the answer to the above question will likely change for many people.

With all of the development efforts currently underway, it is useful to reflect on some of the problems encountered and the solutions implemented while creating the PAIRS system. Starting from the precursors of PAIRS, Munk's CASE program (2) and Wipke's SECS program (3), and leading up to the current version of PAIRS, it has been a continuous development process and learning experience. Figure 1 summarizes PAIRS progressions and cites relevant references. Some of the issues that have had to be addressed during this period will be described in this chapter. Where appropriate, these issues will be generalized in an attempt to provide some insight for other researchers in this field.

What Are the Expectations for the System?

Obviously, before undertaking any project, knowledge-based systems included, the objectives should be clearly established. The intended audience needs to be determined. Decisions must be reached about whether the system will be narrowly focused to solve a specific problem, or will be larger and more ambitious. When developing knowledge-based systems, the source(s) to be used to obtain the required expertise is a complex issue. Once this expertise has been transformed into the knowledge base (or rules) to be used by the system, a decision needs to be made about whether or not the rules will be available and modifiable by the user. Finally, decisions must be made as to what will constitute an answer and what information will ultimately be provided by the system to the user.

Intended Audience and System Magnitude. The expectations of PAIRS from its inception were to assist the practicing chemist in interpreting an IR spectrum. This goal was to be accomplished by creating a system which mimicked the thought process used by an experienced spectroscopist. Thus, much of the effort in creating PAIRS involved incorporating the knowledge and logic used by a spectroscopist into an extensive rule base. Simply making predictions about a limited number of chemical functionalities or otherwise artificially limiting the domain of the system to a few chemical functionalities would not suffice in meeting the expectations established at the outset. Thus, interpretation rules were initially written for 175 chemical functionalities and subfunctionalities. A schematic diagram of the PAIRS system is given in Figure 2.

To use PAIRS, the scientist was required to supply the system with peak intensity, width, and position information derived from the unknown compound's spectrum, as well as information concerning the sample state (e.g., solvent, neat, mull). In addition, the scientist was allowed to indicate the presence or absence of given atom types in the empirical formula of the unknown compound. Since the goal in performing a PAIRS interpretation was to provide the user with a functional group analysis, the results of an early PAIRS interpretation were limited to a numerical indication of the likelihood of presence or absence for a given functionality. An example of the interpretation results provided by an early version

HISTORY OF PAIRS

1980	IBM mainframe version of PAIRS	(4)
1981	Minicomputer version of PAIRS	(5)
1984	First attempts at automated rule generation	(6)
1985	Rule trace incorporated (VAX version)	(7)
1986	A version of PAIRS which allows interactive optimization developed	(8)
1987	PAIRS modified for personal computers	(9)
	- explains rationale for decisions	
	- allows interactive interpretations	
	- CAI of the spectral interpretation process possible	

Note: References in parentheses.

Figure 1. Significant advances during the development of PAIRS.

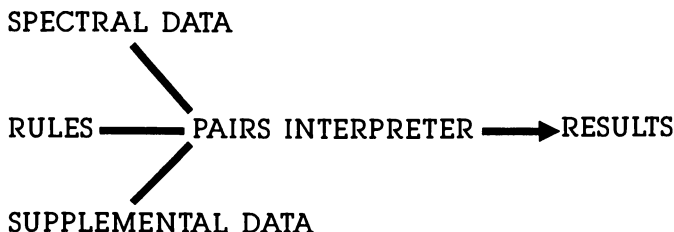


Figure 2. A schematic diagram of the PAIRS system.

of PAIRS for the spectral data for the compound nitrotoluene is presented in Figure 3.

Source of Expertise. Does one always need an expert to create a knowledge-based system? This question stems from the often stated assumption that to make a knowledge-based system, one needs to find an expert in the chosen field from whom to extract the required knowledge. While this may often be the case, it is not always necessary to have an expert available to develop a knowledge-based system. Clearly, the needed commodity is knowledge, not necessarily in the form of a person. If the required knowledge is available from other sources, then an expert may not need to be present for system development.

An example of a knowledge-based system created without the direction of an expert is PAIRS. The knowledge and logic used to interpret IR spectra are well documented in the scientific literature. Thus, the basis for the system is readily available. While a number of gifted infrared spectroscopists participated in the development of the PAIRS knowledge base, the knowledge base was prepared primarily using literature resources by analytical chemists with some background in spectral interpretation but who were (and are) by no means infrared spectroscopists. The importance of the ability to consult with experts is, however, very useful and can add greatly to the overall performance of the knowledge-based system. Our experience with PAIRS demonstrated that experts are especially helpful in diagnosing where the preliminary system failed. The availability of such information aided greatly in the refinement of the PAIRS system.

If one chooses to utilize an expert during the development of a knowledge-based system, then one must be conscious of the biases which that expert has. For example, if one were to make a knowledge-based system to assist with chromatographic separations using an expert chromatographer to prepare the knowledge base, then it is reasonable to expect that the system produced will contain the biases of the chromatographer. Chromatographers, for obvious reasons, are generally forced to limit their thinking to chromatographic experiments which can be performed in the laboratory. The knowledge-based system, however, will necessitate that a somewhat powerful computer will be available to the person performing the chromatographic experiments. It is possible that the way one approaches the chromatographic problem having a computer available may be different than the way one would approach the problem without a computer, just as the way one approaches a pH titration depends on whether one must use visual indicators or a pH meter. Clearly, the availability of the computer changes the resources available to the chromatographer and possibly the approaches taken to solve the chromatographic problem. The knowledge and logic used by the "expert" in chromatography may, therefore, not be appropriate as the basis for a knowledge-based system. Further, if the knowledge and logic used by the chromatographer is incorporated into the knowledge-based system, then the system will have the same limitations and biases as the chromatographer.

Availability and Modifiability of the Knowledge Base. A decision was made early in the development of PAIRS that the interpretation rules should be made available to the user in a relatively easy-to-understand format. The reasons for making this decision included:

- 1). To allow the user to determine the decisions which were made to arrive at the results of a given interpretation;
- 2). To allow for the expansion of the rule base by those persons with expertise in interpreting spectra for functionalities not included in the PAIRS rule base;
- 3). To allow users who are dissatisfied with the results provided by the interpretation rules to modify, adjust, or replace the existing PAIRS rules for the functionalities in question.

The importance of allowing the user to determine quickly the rationale used to arrive at the results of a given spectral interpretation cannot be overestimated. Simply making the interpretation rules available does not adequately fulfill the needs of the user. In fact, information concerning the rationale for the determinations being made by the system is of much greater benefit to the user faced with questionable results than the rule base itself. It is for this reason that versions of PAIRS were developed which provide the user with the ability to trace the decision making process. The system was later expanded to include explanatory statements describing the rationale for the questions asked and decisions being made. Clearly, these later versions of PAIRS go well beyond simply making the rule base available and add a new dimension to the interpretation process.

The expertise required, both in spectral interpretation and development of interpretation rules, has severely limited the number of users willing to extend the PAIRS rule base. Further, those capable of developing PAIRS rules are often interested in interpreting spectra of compounds which are proprietary or of great specificity and, therefore, of limited general interest. Developments in the availability and use of expert system shells and the limitations of the approach used by PAIRS have also reduced the interest in the expansion of the PAIRS rule base by scientists not in the original PAIRS development group. Thus, it appears further expansion of the generally available PAIRS rule base to include the expertise of others will proceed at a slow pace, if at all.

It was originally envisioned that the availability of the rules would provide users with a base for modification and improvement to suit their needs. However, the complexity of the rule base and lack of imbedded comments made rule modification difficult. Just as the meaning of a key sentence in a paper can be changed by simple rewording by a well meaning editor, the results of a spectral interpretation can be completely changed by seemingly modest modifications to the interpretation rules. Our experiences in this area have indicated it is probably better to allow the informed user to develop his or her own interpretation rules to augment the current rule base and limit the modification of existing rules to those intimately familiar with the original development of the specific rules in question.

While it may not be in the best interest of system developers to provide user access to the knowledge base for reasons of system security and integrity, the ability to understand the reasons for decisions which are made and conclusions which are reached is clearly of importance to the user. It will be interesting to see how commercially available knowledge-based systems provide such information to the users while protecting the investment of the

system developers. This issue is especially important to those scientists using such systems in the practice of their research.

What Information Will Constitute an Answer? The question of what information will be provided by the knowledge-based system to the user as the results of a consultation session is very important. In the case of PAIRS, it was initially decided that the question which the system should answer was "Is this functionality likely to be present in the unknown compound?". The answer to such a question clearly falls into a "yes, no, maybe" categorization. For this reason, the original results of the PAIRS system were a numerical indication of the likelihood of presence for a given functionality based on a 0 to 1 scale. An example of the results of an interpretation are presented in Figure 3.

While using such a numerical scale answers the question as stated above, it should be realized that such an answer is insufficient for a researcher faced with a typical functional group analysis problem. Further, it is clear that an answer in this form does not transfer all of the information which is contained in the knowledge base. Realizing that more information could be transferred from the knowledge base to the user, a natural development was to present the user with a trace of the decision making process in addition to a simple numerical indication of presence for a particular functionality. Thus, the system could now answer the question, "How did you determine the likelihood of presence or absence for a given functionality?".

Further enhancement of the information transfer from knowledge base to user was possible once the user was capable of determining which decisions were made to arrive at the given conclusions for a particular functional group. Clearly, the user could benefit from an answer to the question of why a given query was made and what the desired answer might be. To provide the user with this information, it was necessary to code into the knowledge base the reason for the queries, decisions, and actions. A partial interpreter trace, including descriptive comments, for the functionality "nitro" from the interpretation of the spectral data for nitrotoluene is given in Figure 4. With this advance, PAIRS is now capable of providing the answers to the questions, "What is likely to be present?", "What information was used to make these decisions?", and "What was the basis for the information used and the decisions which were made during the interpretation process?".

Clearly, the information which is presently available and provided by PAIRS is quite different from the information which was thought to be necessary and important when PAIRS was in its initial planning stage. Major program and knowledge base changes were required to upgrade the initial system to provide such information. This experience demonstrates clearly that one needs to build into the initial knowledge-based system the flexibility not only to modify the knowledge base (to be covered more fully in a section on system maintenance) but also to modify the basic structure and goals of the system as required in the future.

How Will the System Meet these Expectations?

Having established the expectations for the system, the issues of implementation and maintenance become crucial.

Implementation. Today, there are many options available to the prospective architect of a knowledge-based system. Developments in

1) NITRO	0.99
2) NITRO-AROMATIC	0.99
3) AROMATIC	0.99
4) AROM-1-2-3-SUBST	0.76
5) METHYL	0.72
6) AMINE	0.60
7) AMINE-TERTIARY	0.60
8) AROM-1.2-SUBST	0.57
9) AROM-1.3-SUBST	0.57

Figure 3. The results of an early PAIRS interpretation for the compound "nitrotoluene".

software design tools since the initial development of PAIRS have brought about a host of new computer languages, along with the introduction of "shell" programs (10-12). A brief discussion of the basic components making up an expert system is in order before considering the merits of traditional languages versus shells.

The heart of any expert system is the knowledge base. The knowledge base is a compilation of facts, or data, and rules of inference, to be applied to the problem to be solved. These rules and facts are processed by an inference engine. The inference engine is a central unit which controls the selection and application of "appropriate" rules and facts, and in some cases, may revise the knowledge base as new facts are "deduced". An important characteristic of expert systems is that these two components are separate and distinct entities. The interface is a software window to the inference engine. All user/program interactions, including data input, inference tracing, and the reporting of results, occur through this interface. While not a factor in program performance, the quality of the interface largely determines how useful the system will be in practice.

Which tool is best for developing expert systems? The answer is that it depends on the job to be done and the resources available. How large is the problem to be solved? Are professional programmers and/or knowledge engineers available to create the system? Is broad hardware compatibility an issue? Which is more important, development time or system performance time? Is the system to be distributed? How important is cost? These questions are some of the considerations in the choice of development and deployment tools.

An experienced programmer may elect to implement the system "from scratch" in one of many currently available languages. A thorough discussion of the merits of language choices is beyond the scope of this text, however some general points may be made.

Twenty years ago, artificial intelligence (AI) was considered a subject for academic research, not practical application. A number of languages, most notably LISP (13) and Prolog (14), were considered the only suitable vehicles for AI. Traditionally, these have been offered as interpreted languages, facilitating exploratory AI research and providing the ability for the program to modify itself while running. For programmers skilled in these languages, they offered simpler expression of logic-driven programming than existing procedural languages such as FORTRAN, BASIC, and Pascal. Prospective users should be advised however, that both Prolog and LISP, as well as the object-oriented languages described later, have a reputation for creating CPU-intensive applications requiring large amounts of memory.

As the transition from laboratory to market has occurred, practical considerations have brought a growing trend toward AI programming in conventional languages (15), particularly C and Pascal. These languages offer a much larger base of programming talent to design and maintain the programs, as well as a broader range of hardware for implementation. The resulting systems are generally more compact and offer much faster execution speeds than LISP and Prolog. Size and speed are particularly important for microcomputer-based systems, for very large programs, or for systems requiring rapid response. Additionally, the compilers for these languages (as well as the programming skills) are normally available at reasonable cost.

One area receiving a great deal of recent attention is the application of object-oriented languages (16-18), as exemplified by

Smalltalk. These languages can offer greatly accelerated development due to their high level of abstraction and their inherently reusable code. They feature a large body of predefined "object classes" and associated "methods", which may be expanded with each application. Designed as an exploratory programming environment, these are largely interpreted languages.

Although many people agonize over the choice of a programming language, a general rule is that unless some design criterion dictates a particular choice, the best language to use is the one with which the developer is most familiar. Virtually any mechanism can be implemented with any language. The difficulties associated with becoming proficient in an unfamiliar language usually outweigh any possible benefits.

Those individuals who lack extensive programming experience and/or the aid of a "knowledge engineer" may elect to use an expert system shell to implement their system. These programs provide a ready-made framework for developing an expert system, hence the term "shell". Available products differ greatly in both features and cost. The cost spans a range from less than \$100 to well over \$10,000. Serious developers should expect to spend at least \$500 for base microcomputer systems, more for products offering extended features and/or implemented on mini and superminicomputers. Additionally, distribution of the system may require potential users to purchase the shell interpreter, or may involve licensing fees.

While these programs differ greatly, they share a common set of basic elements. All include one or more knowledge representation schemes which define the structure of the knowledge base. All supply an inference engine which can apply one or more methods of inference, such as backward and forward chaining. Finally, all provide some means of tracing and often inquiring into the problem solving process.

Shell programs may allow a competent scientist with little computer experience to develop a system without the aid of computer professionals. In the hands of a computer professional, they allow rapid development of complex, user-friendly systems. They generally utilize advanced (for most chemist/programmers) database and inference techniques, and provide sophisticated user interfaces. Indeed, it is doubtful that most scientists would be capable of developing a system using a conventional language environment that is comparable to the more advanced shell offerings, without investing a tremendous amount of time and effort. While the limited capabilities of earlier shell programs made them useful largely as a learning environment or as a tool for system development by a domain expert, current shells offer sufficient features to provide a knowledge engineer with a framework for an advanced system.

One final possibility is to use a combined approach, using an object-oriented language or a shell program for system development or prototyping, and implementing the delivery system in a conventional language, such as C or Pascal. Particularly useful for large projects, this approach lets each tool do what it does best.

Maintenance. The long-term maintenance of a knowledge-based system developed to assist researchers has both scientific and practical implications. The scientific implications stem from the fact that the knowledge base in most if not all areas of science continues to grow with time. Even those areas perceived to be stagnant often

benefit from advanced technology and therefore change with time. Just as scientists who do not keep up with the developments in their particular field are quickly obsolete, knowledge-based systems which are not updated regularly in fields where advances are taking place will also quickly become obsolete. If the knowledge-based system is primarily used by researchers in a given field, the consequences of using a non-maintained system can be especially serious. Just as a researcher would question an expert who has not kept up with a field for many years, so too should the researcher be cautious when using a knowledge-based system that has not been updated to include the latest advances, knowledge, and teachings in a given area. The faster the area of expertise is expanding, the greater the chance that the knowledge-based system will not be reliable.

The maintenance of a knowledge-based system is also of practical importance. In this case, we are referring not to updating the knowledge base but instead to updating the software so that it can be run using hardware and software advances as they become available. If the software of the knowledge-based system is not software and hardware compatible with the computers currently used by researchers in a given field, then the system is essentially not available to the end-users and interest among these users will eventually vanish.

Systems which are easily maintained are the result of efforts made in the design stages of development. Proper programming techniques dictate the following (18):

- 1). All source code should be liberally commented.
- 2). Meaningful names should be chosen for variables, procedures, etc.
- 3). Modular (structured) programming using top-down design should be employed.
- 4). Subprograms should be made as general as possible to allow easy adaptation of the code and should "hide" the details of implementation from other program modules.
- 5). Programs should be written for maximum clarity, i.e., the code should be the most literal representation of the ideas possible. For ease of maintenance, this point cannot be overemphasized.

Additional points specific to expert system design include those shown below:

- 6). The use of metarules (20), where applicable, will provide an outline to the knowledge base, along with a modularity to the knowledge base structure.
- 7). The rule base must be structured in some way. This may involve grouping like rules together and/or organizing the rule base in a hierarchical fashion. A hierarchical rule base provides some of the advantages of metarules, namely, generalized upper level rules may be easily "reused".
- 8). The rule base itself should be commented where necessary.

- 9). A separate inference engine should be used to process the rule base, allowing for multiple inference strategies using the same rule base and providing easy rule modification. Additionally, the inference engine itself may be reused for other projects.
- 10). Some available tools provide translation of a rule base into several formats usable by different existing inference engines.

Questions concerning how best to maintain a proposed (or existing) knowledge-based system, who will be responsible for maintaining the system, and how long-term maintenance will be financed should all be rectified prior to beginning development on such a system. It is only by designing for proper maintenance that a knowledge-based system can age gracefully, continue to perform at reasonable levels of expertise, and continue to protect the original investment in its development.

Publishability in Scientific Journals. After determining the expectations and the implementation procedures, at least one other vital issue needs to be addressed. The ability to publish the results of ones efforts is of great importance to analytical chemists.

The publishability of the development of knowledge-based systems raises a number of interesting issues.

- 1). How should one judge whether or not to publish the results of development of a knowledge-based system?
- 2). Should publications be aimed at chemists who are likely to use the system, computer specialists, or both?
- 3). How should improvements to existing knowledge-based systems be treated, as publishable research developments or as non-publishable program modifications?
- 4). Should the publishability of efforts in the area of knowledge-based systems be treated using the same criteria as other areas of chemical research?
- 5). What is likely to happen if efforts are made to limit the publishability of the results of efforts in this area?

A critical concern is that it takes a great deal of time to develop a substantial and useful knowledge-based system. It can easily take years from beginning development of such a system until the first publication of the results of one's efforts. Clearly this rate of publishing will not be sufficient to maintain a successful academic or industrial research career. Further complicating the issue, a single paper on any given subject matter tends to go virtually unnoticed in the wealth of information that is published in scientific journals. Thus the researcher is faced with a rather serious problem, especially if editors and reviewers believe that system improvements are not publishable research. Editors, reviewers, and those involved in creating these systems will need to address the questions raised above if progress in this area is to continue.

American Chemical Society
Library
1155 16th St., N.W.
Washington, D.C. 20036

Conclusions

Improving computer-assisted spectroscopic interpretations continues to be one of the goals of our research activities. As the efforts have evolved, many issues have had to be addressed and reconsidered. It is critical to consider at the outset what the system is to do. What are the expectations? For whom is the system designed? How large (useful) will it be? (i.e., is it a "basic research" effort or is it an effort to develop a usable system?). Next, consideration must be given to how the system will meet these expectations. Fundamental to these issues is whether or not the knowledge base will be made available and modifiable. How one will maintain the system and report alterations and developments are other considerations. By addressing all of these issues at the appropriate times, developers and users of knowledge-based systems will be able to maximize their benefits.

Literature Cited

1. Watson, H.J.; Mann, R.I. Journal of Information Systems Management 1988, 5(4), 39.
2. Woodruff, H.B.; Munk, M.E. J. Org. Chem. 1977, 42, 1761.
3. Wipke, W.T.; Braun, H.; Smith, G.; Choplin, F.; Sieber, W. In Computer-Assisted Organic Synthesis; Wipke, W.T.; Howe, W. J, Eds.; American Chemical Society: Washington, D.C., 1977; pp 97-127.
4. Woodruff, H.B.; Smith, G.M. Anal. Chem. 1980, 52, 2321.
5. Tomellini, S.A.; Saperstein, D.D.; Stevenson, J.M.; Smith, G.M.; Woodruff, H.B.; Seelig, P.F. Anal. Chem. 1981, 53, 2367.
6. Tomellini, S.A.; Stevenson, J.M.; Woodruff, H.B.; Hartwick, R.A. Anal. Chim. Acta 1984, 162, 227.
7. Tomellini, S.A.; Hartwick, R.A.; Woodruff, H.B. Appl. Spectrosc. 1985, 39, 331.
8. Saperstein, D.D. Appl. Spectrosc. 1986, 40, 344.
9. Wythoff, B.J.; Buck, C.F.; Tomellini, S.A. Anal. Chim. Acta, 1989, 217, 203.
10. Citrenbaum, R.; Geissman, J.R.; Schultz, R. AI Expert 1987, 2(9), 30.
11. Arcidiacono, T. PC Tech Journal 1988, 6(5), 44.
12. Fontana, M.; Zeimetz, J. PC Tech Journal 1988, 6(5), 63.
13. Coffee, P. C. AI Expert 1988, 3(3), 38.
14. Clocksin, W. BYTE 1987, 12(9), 147.
15. Coffee, P.C.; Strauss, D.J. AI Expert 1988, 3(5), 38.
16. Stein, J. Dr. Dobb's Journal of Software Tools for the Professional Programmer 1988, 13(3), 18.
17. Thompson, B.; Thompson, B. AI Expert 1987, 2(9), 15.
18. Ramamoorthy, C.V.; Sheu, P.C. IEEE Expert 1988, 3(3), 9.
19. Martin, J.; McClure, C. Structured Techniques for Computing. Prentice-Hall Inc., Englewood Cliffs, N.J., 1985.
20. Nilsson, N.J. Principles of Artificial Intelligence, Tioga Publishing Co., Palo Alto, Ca., 1980.

RECEIVED July 21, 1989

Author Index

- Azevedo, Henderson, Jr., 158
 Bernstein, Zmira, 62
 Birk, James P., 20
 Brede, Constance, 200
 Browett, William R., 210
 Burnstein, Ilene, 102
 Cox, Timothy A., 210
 Crary, John, 102
 de Bruijn, D. P., 190
 Dessy, Raymond E., 34
 Edelson, David, 10,49
 Evans, Martha, 102
 Federico, A. G., 147
 Fogarty, Terence C., 180
 Frei, Gertrude, 158
 Hendrickson, James B., 62
 Hirayama, Hikaru, 200
 Hohne, Bruce A., 2,10
 Jenkins, D. Paul, 180
 Johnson, Peter Y., 102
 Karnicky, Joe, 10
 Klimkowski, V. J., 82
 Kuipers, H. P. C. E., 190
 Lacher, R. C., 49
 Martin, C. E., 49
 Miller, Myron J., 126
 Miller, Todd M., 62
 Momany, F. A., 82
 Moore, Robert L., 169
 Mortimer, Jon, 180
 Parks, Camden, 62
 Pierce, Thomas H., 2,10
 Pitha, R., 82
 Roach, John W., 34
 Ronchetti, C., 147
 Stauffer, Scott T., 34
 Stillman, Martin J., 210
 Toczek, A. Glenn, 62
 Tomellini, Sterling A., 236
 Trindle, Carl, 92
 Tripi, A., 147
 Varma, S., 49
 Venkatachalam, C. M., 82
 Wang, H., 49
 Wang, Tunghwa, 102
 Wohlsen, Robert, 200
 Woodruff, Hugh B., 10,236
 Wythoff, Barry J., 236

Affiliation Index

- Arizona State University, 20
 Brandeis University, 62
 Bristol Polytechnic, 180
 ENEA Thermal Nuclear Department, 147
 Factory Mutual Engineering, 126
 Florida State University, 10
 G. D. Searle & Company, 102
 Gensym Corporation, 169
 Illinois Institute of Technology, 102
 Koninlijke/Shell—Laboratorium, 190
 Merck Sharp and Dohme, 10,18
 Osaka Gas Research Center, 200
 Polygen Corporation, 82
 Rohm and Haas Company, 2,10
 SRI International, 200
 Sandoz Limited, 158
 Supercomputer Computations Research Institute, 10,49
 University of New Hampshire, 236
 University of Virginia, 34,92
 University of Western Ontario, 210
 Varian Associates, 10
 Virginia Polytechnic Institute and State University, 34

Subject Index

A

Academia, goals of expert systems, 11
 ACanalysis, function, 218
 ACanalyst, function, 216
 ACassurance, function, 216
 ACcontrol
 description, 226
 function, 216–218
 instrument control, 226,227f
 user and instrument interface
 development, 228–233
 ACdiagnosis
 conditions, 223
 description, 222–223
 function, 217
 generation of rule base, 223–225
 ACexpert
 AAS application, 228,229f
 ACcontrol, 226
 ACdiagnosis, 222–224,225f
 ACmethods, 220,221–222f
 ACteach, 225–226
 analyst's role, 215–216
 assignment of tube elements, 228,230f
 calibration curve, 231,233f
 components of modules, 218
 design specifications, 216–217
 design using shells, 217–218
 development, 211
 future research, 231
 incorporation into automated control
 program of any instrument, 231
 instrument control, 226,227f
 manager's role, 215–216
 modules, 215–216
 phototube response, 231,232f
 rule analysis strategies, 218
 Sampler application, 228,223–233
 schematic representation, 215f
 software simulation, front panel of atomic
 absorption spectrometer, 228,229f
 stages of prototyping, 217
 Acid, definition, 24,26–27
 Acid cleaning, procedure, 155
 ACmethods
 example of ambiguous response, 221,222f
 example of conditions and case
 conclusions, 225f
 final screen output, 221f
 function, 216,217,220
 strategy and conditions, 220
 ACteach, description, 225–226
 AIChE meeting AI papers, list, 144–146
 Automated metal analysis by atomic
 absorption spectrometry, design of expert
 system, 210–234

Automation in analytical laboratory
 applications of expert systems, 211
 development, 210–211
 evolution of expert systems, 211–212
 use of expert systems, 211

B

Backward-chaining rule analysis,
 description, 218
 Bases, definition, 27
 Brooks data structure, description, 94

C

Chemical language editor, KNODE, 54–55,56f
 Chemical language interpreter,
 KNODE, 54–55,56f
 Chemical process control, expert
 system, 169–179
 Chemistry-based expert system shell (CHESS)
 architecture, 45,46f
 components, 34
 general chemical knowledge, 44–45
 GUESS, 45
 implementation, 45,46f,47
 knowledge extraction utility, 35–43
 molecular structure representation and
 recognition, 44
 parameter estimation and calculation, 44
 purpose, 34–35
 reaction libraries, 44
 CHESS, 34–48
 Complex solids characterization, use of
 CACSS, 190–199
 Computational language interface, KNODE, 53
 Computer-aided characterization of solids
 and surfaces (CACSS)
 case study, 192–193
 choice of optimal starting set of analytical
 techniques, 194–196
 criterion for rejection, 195–196
 crystallographic modeling, 198
 description of project, 191
 implementation of
 project, 196,197f,198–199
 interpretation of experimental
 results, 195–196
 matrix discussion group, 190–191
 outline of project, 194–196
 role of expert systems, 191–192
 schematic outline, 196,197f
 structure elucidation from crystal
 powders, 196,198
 use of artificial intelligence technology, 192

- Computer-assisted instruction
 development, 20
 intelligent tutoring systems, 21
- Corrosimeter, evolution, 150
- Corrosion
 direct measurement, 149
 indirect measurement, 149
- Corrosion of metallic materials
 classes, 148
 damage of resources, 147
 evolution of corrosion-monitoring systems, 148–150
 objectives of prevention efforts, 147–148
 problems induced, 148
- Corrosion-monitoring systems
 design of model emulating expert reasoning behavior, 150–156
 development of probes, 150
 evolution, 148–150
 improvement in corrosimeters, 150
- Corrosion rate measurements,
 reliability, 149
- D**
- Data-directed expert system
 description, 4
 example, 4–5
- Descriptive inorganic chemistry
 database, 24,25*t*
 expert system design, 24
 expert system development, 22–24
 groups of reactant classes, 24,26*t*
 PIRExS, 22
 reactant classes, 24,26*t*,27
 reaction classes, 27–28
 reactions with one reactant, 28
 reactions with two reactants, 28–29,31*t*
 rules for reactivity prediction, 29–30
- DOCENT
 caricature of macromolecule by model matching, 94,95*f*
 deducing atomic-scale geometry from generalized cylinders, 98–99
 features, 93
 function, 93
 manipulation of generalized cylinder representation, 99
 ribbon as set of generalized cylinders, 99–100
 use of acronym-style modeler in molecular displays, 97–98
 use of generalized cylinders in interpretation of scene, 94–95,96*f*,97
- Drug design
 disadvantages of linear methods, 82–83
 method of receptor fit, 82
 use of protein pseudoreceptor, 83–90
- E**
- Environmental test plan
 conditions and requirements, 200–201
 procedures, 200
 use of ES–EPA, 201–209
- Expert reasoning instrument for corrosive environments (ERICE)
 advantages, 150
 applications, 154,155*f*,156
 corrosion rate behavior during different plant conditions, 155*f*,156
 examples of hypothesizing production rules, 153*f*
 future developments, 156
 knowledge base system, 151–154
 knowledge base tree structure, 152*f*
 outcomes, 156
 plant data base multiwindow updating system, 152,153*f*
 system functionalities, 153–154
 use of Personal Consultant Plus shell, 152
- Expert system(s)
 advantages, 7
 application selection, 6
 applications in chemistry, 2–9
 computer platform selection, 6–7
 description, 2,21
 design using shells, 217–218
 development, 6–7,21–22
 disadvantages, 7
 future impact in chemistry, 7
 glossary, 7–9
 inference engine, 3
 knowledge base, 2–3,244
 pieces, 2
 prediction of inorganic reactions, 22–33
 sharing of knowledge bases, 10
 software selection, 6–7
 user interface, 3
- Expert system for environmental pollutant analysis (ES–EPA)
 abstraction level plans, 201
 architecture, 203,205*f*
 characteristics of planning methods used by human experts, 201
 control module, 203
 development, 201
 future directions, 209
 hardware environment, 202–203
 knowledge base module, 203
 multiple template, 204,207*f*
 plan templates, 202
 planning mechanism, 204,205–207*f*
 schematic representation of hierarchical planning, 204,205*f*
 single template, 204,206*f*
 software environment, 202–203
 user interface, 208*f*,209

Expert system for environmental pollutant analysis (ES-EPA)—*Continued*
user interface module, 203
Expert system for instrumental analysis,
criteria for development, 212

F

Forward-chaining rule analysis,
description, 218
FORWARD program
generalized plan, 76,78,80f
procedure, 78–79
refunctionalization reactions, 76,78–79,80f
sample output, 79,80f

G

G2 real-time expert system
application example, 178–179
applications, 169
critical path planning, 172,175f
description, 169–170
knowledge base management, 176,177f
knowledge representation, 170–175
managing knowledge on
workspaces, 176,177f
reasoning about knowledge, 176,178
robot navigation example, 176,174f
use of high-level interfaces, 169
Generalized cylinders
applications, 100
components of ribbon, 99–100
deduction of atomic-scale geometry, 98
examples, 94,95f
manipulation of representation, 99
use in interpretation of scene, 94–95,96f,97
General-purpose expert system shell (GUESS)
description, 45
use in CHESS, 45
Glossary, expert systems, 7–9
Goal-directed expert system
description, 4
example, 4
GUESS, 45

I

Indole, substitution patterns, 111,114f,115
Industry, goals of expert systems, 11
Inference engine
data-directed approach, 4–5
description, 3
goal-directed approach, 4
structure, 3–4
use of meta-knowledge, 5

Inorganic reactions, system for
predicting, 20–33
Intelligent scientific computing systems
description, 49
design of KNODE, 50–60
Intelligent tutoring systems
description, 21
expert system, 21–22
Implementation of knowledge-based system
difficulty of rule modification, 240
ease of understanding rationale for
determinations, 240
expertise required, 240
programming language selection, 244–245
tool selection, 244
use of shell programs, 245

J

Job control, KNODE, 53–54
Job packer–controller, KNODE, 55,57–60

K

KDS3 expert system shell
description, 218–210
development of expert systems, 219
example of analytical methods selection
expert, 219,220t
standard screen interface, 219–220,221f
KNODE
chemical language editor, 54–55,56f
chemical language interpreter, 54–55,56f
comparison to procedure by human
experts, 50
computational language interface, 53
control flow of job-control module, 57,58f
file hierarchy as viewed from job control
module, 57,59f
job control, 53–54
job packer–controller, 55,57–60
mathematical language interface, 52–53
overview, 50–54
phase implementation, 54–60
principal modules and data pathways, 50,51f
scientific language interface, 50,52,54–56
scientific visualization module, 54
syntactic and semantic analyzer function
of scientific language interface, 55,56f
system supervisor, 50
Knowledge base
description, 2–3
encoding of knowledge, 3
frames for encoding, 3
production rules for encoding, 3
scope, 3
sharing, 10–16

- Knowledge base management for G2 expert system, examples for workspaces, 176,177f
- Knowledge base module of expert system for environmental pollutant analysis
 analysis item KB, 203
 analysis task KB, 203
 customer KB, 203
 sample type and purpose KB, 203
- Knowledge-based systems, development, 236–247
- Knowledge extraction utility for chemistry-based expert system shell
 cluster analysis results, 38,39f
 complete grid, 40,41f
 example, 36–43
 example grid, 36,37f,38
 procedure, 35–36
 questions asked by inference engine, 40,43f
 reported results, 40,43f
 typical rules generated, 40,42f
- Knowledge for G2 expert system, truth maintenance, 176,178
- Knowledge representation for G2 expert system
 concurrent use of analytic and heuristic models, 170
 critical path planning, 172,175f
 defining heuristic knowledge, 170,171f,172
 dynamic behavior and live data, 170
 generic knowledge for classes of objects, 172,173f
 interaction between objects, 170
 robot navigation example, 172,174f
- Kuhn statistical segment picture of chain dynamics, use in molecular displays, 97–98
- L**
- Laboratory information management system, development, 34
- Life cycle, definition, 127
- M**
- Maintenance of knowledge-based system
 design specifications, 246–247
 practical importance, 246
 proper programming techniques, 246
 scientific importance, 245–246
- Material hazard classification
 ASTM, 137–138
 FM–NFPA, 135
 UN–TNO, 135,136f,137
- Mathematical language interface,
 KNODE, 52–53
- Matrix discussion group
 establishment, 190
 strengths and weaknesses, 190–191
- Meta-knowledge, definition, 5
- Metal analysis
 manual
 incorporation of expert system, 214
 model, 212,213f,214
 procedure, 212–214
 role of laboratory information management system, 214
 use of ACExpert, 215f,216
- Metallic materials, corrosion, 147–156
- Molecular modeling
 Brooks data structure, 93
 caricature of macromolecule by model matching, 94,95f
 deducing atomic-scale geometry from generalized cylinders, 98–99
 examples of generalized cylinders, 94,95f
 manipulation of generalized cylinder representation, 98
 ribbon as set of generalized cylinders, 99–100
 use of acronym-style modeler in molecular displays, 97–98
 use of generalized cylinders in interpretation of scene, 94–95,96f,97
- Molecular modeling software
 caricaturization of macromolecules, 93
 disadvantages, 92–93
- Multilayered design approach,
 SYNLMA, 104,105f,106,107f
- N**
- NUT
 crystallographic modeling, 198
 framework, 198–199
 schematic outline, 196,197f
 structure elucidation from crystal powders, 196,198
- O**
- Optimization of combustion in multiple-burner installations
 development of simple learning system, 186
 related work, 181
 rule base, 181,182f,183
 rule-based optimization of 12-burner zone of furnace, 185
 rule-based optimization of simulation, 183,184f,185
 statement of problem, 180
 system that learns its own rules, 185–186
- Organic synthesis, designing an expert system, 102–121

P

- Personal construct theory
 applications, 40
 description, 35–43
- PIRExS
 description, 22
 explanation facility, 32
 pedagogical uses, 32–33
 types of predicted reactions, 22,23r
- Plant safety, example of G2 expert system, 178–179
- Polyalanine polypeptide, ribbon structure, 84,85f
- Principle of minimum chemical distance, description, 115–116
- Program for analysis of IR spectra (PAIRS)
 availability and modifiability of knowledge base, 239–241
 description, 236
 example of results, 241,242f
 expectations for system, 236
 implementation, 241,244–245
 information required to answer query, 241,242–243f
 intended audience, 237
 maintenance, 245–247
 partial interpreter trace, 241,243f
 publishability, 247
 schematic representation, 237,238f
 significant advances during development, 237,238f
 source of expertise, 239
 system magnitude, 237,239
- Pseudoreceptor
 calculation of energy and conformational features, 83
 creation, 83
 definition, 83
 methodology, 83
 structure and reaction determination, 83
- Pseudoreceptor method of drug design
 configuration of side chains of muted pseudoreceptor, 88,89f
 enthalpic and entropy contributions to binding and stability, 88,90
 learning cycle, 90
 low-energy structures of substrates, 84,86f
 materials, 84
 packing around substrate, 88
 peptide structure after substrate insertion into pocket, 84,87f,88

R

- Receptor fit method of drug design, description, 82

- Rule-based optimization of simulation of multiple-burner installations
 optimization of simulation, 184f
 randomly generated model, 184,185f
 simulation, 183,184f
 simulator, 183
- Rule-based optimization of 12-burner zone of furnace
 hardware, 185
 optimization of furnace, 185
- Rule base for optimization of combustion in multiple-burner installations
 classification of situations, 181,182f,183
 composition of actions and examples of rules, 181
 origin of rules, 181
 type of rule, 181

S

- Scientific and engineering computation, 49–61
- Scientific language interface, KNODE, 50,52,54–56
- Scientific visualization module, KNODE, 54
- Shape, representation, 92
- Sharing of knowledge bases
 ability for customization and improvement, 13
 accessibility, 13
 advancement of science, 13
 advantages, 10–11
 applications, 16
 dangers, 15
 development costs, 11,12f,13
 different goals for different users, 11
 economic obstacles, 11,12f,13
 modular expert systems, 16
 need for numerical data-sharing mechanisms, 15
 problems, 11
 rule examples, 13,14f
 technical obstacles, 11,13
 trade secret vs. public domain information, 15–16
- SYNGEN program
 bond-set selection from screen, 72,73f
 characterization of structures and reactions, 64,66,67f
 construction—refunctionalization reactions, 69
 description, 66–72
 example of reaction generation, 66,68f
 examples of operator selection choices, 74,75f
 generalized form of construction reactions, 66,68f

SYNGEN program—*Continued*
 generalized plan, 78,80f
 generation of synthetic sequences from
 bond sets, 66,69
 match of SYNGEN to SYNLIB, 76
 mechanism tests for chemical
 solidity, 69,71–72
 operator selections from
 output, 72,73f,74,75f
 output summary from screen, 72,73f
 procedure, 62,64
 protocol, 62
 refunctionalization reactions, 76,78–79,80f
 removal of equivalent reactions, 74
 sample match of SYNGEN to
 REACCS, 76,77f
 sample page of reactions from
 screen, 72,74f,75
 samples of half-reactions, 69,70f
 searching for literature precedents, 74,76,77f
 selection criteria for synthesis routes, 64
 skeletal bondsets and assembly plans, 64,65f
SYNLMA
 advantages of separate reasoning
 component, 102–103
 applications, 103
 chemical islands, 111
 components, 102
 description of present system, 103–104
 implementation of tree-building
 stage, 118–119,120f
 implementation of tree-definition
 stage, 116–118
 implementation of tree-verification
 stage, 119,121
 indole substitution patterns, 111,114f,115
 initial retrosynthetic paths of
 cocaine, 111,112f
 interfacing with commercial data base, 121
 layer structure, 104,105f
 methodology-driven ibogamine
 synthesis, 111,114f,115
 modeling strategic planning for synthesis
 process, 111–116
 multilayered design approach,
 104,105f,106,107f
 network of environment pairs, 104,106,107f
 nonselective carbene insertion, 108,109f
 pruned synthesis of ibuprofen, 108,109f
 reaction rule constraints, 108,110f,111
 reaction taxonomy, 119,120f
 scope of system, 106,108–112
 sophistication, 108
 substructure-driven ibogamine
 synthesis, 111,113f
 system implementation, 116–121
 tree-building stage, 115
 tree-definition stage, 111,113–118
 tree-verification stage, 116

Synthesis tree, schematic diagram, 62,63f
 System supervisor, KNODE, 50

T

Target functionality, definition, 69
 Titanium–vanadium denoxing catalyst
 conclusions from case study, 193
 function, 192
 surface characterization by CACSS, 192–193
 Tree-building stage,
 SYNLMA, 115,116,118–120
 Tree-definition stage,
 SYNLMA, 111,113–114f,115
 Tree-verification stage,
 SYNLMA, 116,119,121

U

Unit reaction, definition, 66
 User interface
 description, 3
 expert interaction, 5–6,
 novice interaction, 6
 parts, 5–6

V

Validation, definition, 127
 Vendors, goals of expert systems, 11
 Verification, definition, 127
 Verification and validation of expert systems
 for chemical process risk management
 background, 126–127
 by and for expert systems, 131,132f,133
 case study for prototype
 development, 134–138
 computer-aided software
 engineering, 133–134
 current AI activities in chemical
 engineering, 127–128
 definitions, 127
 early recognition of needs, 126–139
 example for expert system, 131,132f
 future of software systems safety, 130–131
 future research, 139
 international factors, 129
 prototype hazard classification
 system, 134–135
 purpose, 128–129
 software systems safety, 129–130
 status of research, 138
 tools, 126

W

Wool dyeing

- experience required, 159
- procedure, 159–165

WOOLY

- characteristics, 164
- data retrieval, 167
- default values, 167
- development, 158
- evaluation of class of dyestuffs for given fastness requirements, 159,163f,164
- expertise required of user, 158–159
- explanation facility, 167–168
- inference mechanism, 167
- knowledge types, 164,167–168
- main screen output, 159,161f

WOOLY—*Continued*

- perspiration alkaline fastness for recipe, 164,165f
- quantitative model, 167
- rules, 167
- schematic representation of wool dyeing process, 159,160f
- standard end-use fastness requirements for hand knitting yarn, 159,162f
- temperature–time diagram for dyeing process, 164,166f

X

- XTSYN, description, 103