
RESEARCH IN PROGRESS

AI Research at The Ohio State University

B. Chandrasekaran and Thomas Bylander

AI Group, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio 43210

The AI Group at The Ohio State University conducts a broad range of research projects in knowledge-based reasoning. The primary focus of this work is on analyzing problem solving, especially within knowledge-rich domains, in information processing or knowledge-level terms. B. Chandrasekaran has been the director of the group since its inception in the late 1970s.

Our approach to the design of knowledge-based systems is based upon treating these systems as interacting communities of different kinds of problem solvers. This approach has led us to design systems that are highly organized symbolic structures made up of active specialized knowledge-using agents. This design represents a move away from systems built up mostly of numerical algorithms, or those that separate the knowledge base from the inference engine.

Our research on the fundamental types of knowledge-based problem solving has led us to think in terms of generic information processing tasks (Chandrasekaran, 1983). The idea is that each fundamental type of problem-solving activity accomplishes a certain generic task, and has its own characteristic way of using knowledge. A grasp of the "atoms" of intelligent information processing should provide the building blocks out of which more complicated forms of intelligent problem-solving can be built.

We propose that a complex task be broken down into a number of subtasks, each of which is an example of a generic task. Each subtask is then solved by an appropriately organized community of active agents who specialize in the concepts of the subtask's domain. That is, the knowledge structure corresponding to a problem-solving type can be decomposed into a number of specialists who cooperate in solving that class of problems. We have developed approaches for a number of problems based on this overall approach.

Recently we have been concerned with developing deep models of expert reasoning. Most of the expert systems that have been developed in medicine and other domains have been called "compiled" or "shallow," pointing out that the knowledge base encodes in a fairly direct way the relationships between data and hypotheses. Yet often a human expert's knowledge of how a device functions is used to generate new relationships during the reasoning process. So far we have developed a primitive language for representing the functioning of devices, and a compiler capable of building a diagnostic problem solver from a device representation made in this language.

Types of Problem-solving

We have identified several generic tasks from our work on medical reasoning and reasoning about mechanical devices. Some examples are:

- *Classification* This is the identification of a description with a specific node in a predetermined classification hierarchy. Each node represents a potential hypothesis about the description. Higher nodes represent more general hypotheses, while lower nodes are more specific. Each node is associated with a specialist, which contains the knowledge to evaluate the plausibility of its hypothesis and to cooperate with the other specialists to solve the problem. Classification is used in a number of real-world tasks; it is particularly useful in diagnostic reasoning. In the following section, we discuss some of our approaches to classification problem solving.
- *Prediction* This determines the consequences that will occur when a system is in a particular state. Again, the problem solver will be organized hierarchically with specialists corresponding to subsystems of the system. Each specialist infers state

changes in the immediately larger system based on the state changes within its associated subsystem. These latter state changes include those that led to the current state and those that are inferred by lower-level specialists. We expect that this problem type will be important for expert systems that need to determine the consequences of proposed actions in complex systems.

- *Knowledge-Directed Data Retrieval* In this task, descriptions of data are inferred from raw data and from data in related data concepts. In medicine, for example, drugs, organs, and procedures are data concepts. The specialist associated with a data concept contains or inherits rules about how to infer values for various attributes of the concept, -e.g., the ANESTHETIC specialist can infer that its GIVEN? attribute is true if a particular type of anesthetic has been given or if major surgery has been performed. One of the projects discussed below is to build a general-purpose tool for implementing problem solvers of this type.
- *Design.* A certain kind of design problem-solving can be performed by a hierarchy of specialists who are associated with the conceptual parts of the design. The problem-solving is top-down, in which each specialist chooses from a set of design plans, using the plan to partially fill in its portion of the design and to refine it by activating lower levels of the hierarchy. A section below discusses a project investigating this problem-solving type.

Diagnostic Reasoning

Participants: B. Chandrasekaran, Tom Bylander, Jim Davis, Sia Hashemi, John R. Josephson, Don W. Miller, V. Sembugamoorthy, D. D. Sharma, Jack W. Smith, Jr., M.D., Jon Stucklen, John Svirebely, M.D., Michael C. Tanner.

We are engaged in a number of related projects in diagnostic reasoning, both in the medical domain and in the domain of mechanical systems and devices. In the medical area, the original work started by viewing diagnosis as a form of classification, and this process resulted in a system called MDX (Chandrasekaran and Mittal, 1983). Our recent thrust has been to expand the research to cover the following aspects of diagnostic reasoning.

Abduction and Overview Criticism

Going from data describing a situation to an explanatory hypothesis that best accounts for the data is a commonly occurring knowledge-based reasoning problem. Sometimes the need is to assemble interacting hypothesis parts into a unified hypothesis. In a medical diagnosis, for example, several diseases might be present, and they might be related causally. Disease hypotheses sometimes overlap in what they can explain.

We have developed a general mechanism for accomplishing the unification of sub-hypotheses with some possibly overlapping domains of explanation (Josephson *et al.*, 1984). This mechanism makes use of plausibility information concerning the sub-hypotheses, along with information about what a sub-hypothesis can explain in the particular situation, to build toward a complete explanation. The novel capability arises of confirming a sub-hypothesis on the basis of its ability to explain some feature for which there is no other plausible explanation.

The mechanism we have developed accommodates several types of hypothesis interaction: additive hypothesis cooperation in accounting for the features of the situation, substantive hypothesis interactions of mutual compatibility and incompatibility, and interactions of the sort where one hypothesis, if it is accepted, suggests some other hypothesis. Prospects seem good for extending the mechanism to accommodate other forms of interaction too.

We have used this mechanism successfully as the basis for an expert system, called Red, designed to solve real-world problems of red-cell antibody identification. These are problems that arise in the hospital blood bank and are solved by specially trained human experts.

We have proposed an architecture for abduction (*i.e.*, inference from data to an explanatory hypothesis) that consists of two main cooperating modules:

- A module for selecting sub-hypotheses appropriate to the case at hand.
- A module, which we call "Overview," for assembling these sub-hypotheses into the overall best available conclusion for the case, and for critically assessing this conclusion.

Overview and the other module communicate through a shared language of the plausibility of sub-hypotheses, and of the findings that are to be explained.

For many reasons, we decided in Red to treat sub-hypothesis selection as a problem of classification, using the hierarchical establish-refine method used in the MDX expert system to accomplish the task. An Overview module, similar to the one described here, was envisioned by Gomez and Chandrasekaran (1981). Thus Red represents an extension and elaboration of the architecture pioneered by the MDX system.

Deep Models

Human experts often use in their problem solving a deeper understanding of their knowledge domain than has been captured in the first generation of expert systems. We have developed a functional representation for one aspect of this deeper knowledge, corresponding to an expert's understanding of how the functioning of a complex device results from its structural properties (Sembugamoorthy and Chandrasekaran, 1984). We have also built a compiler that

automatically generates a diagnostic expert system from the functional representation of a device.

The idea is that an agent's understanding of how a device works is organized as a representation that shows how an intended function is accomplished as a series of behavioral states of the device, and how each behavior state transition can be understood as either due to a function of a component, or in terms of further details of behavior states. This can be repeated at several levels so that ultimately all the functions of a device can be related to its structure and the functionality of the components in the structure.

The power of this method for representing how a device works is due in large measure to explicitly distinguishing five aspects of an agent's understanding of the device, and treating each aspect appropriately. The distinctions hold at every level of organization on which the device is represented. The five aspects are:

- *Structure.* This specifies the relationships between components
- *Function.* This captures the intended purpose of a device or component, specified as what the response is to a stimulus.
- *Behavior.* This specifies how, given a stimulus, the response is accomplished.
- *Generic Knowledge.* Chunks of deeper causal knowledge that have been compiled from various domains to enable the specification of behavior.
- *Assumptions.* Other specifications of the conditions under which various behaviors or conditions occur.

Our work also proposes an approach for compiling diagnostic problem-solving structures from the functional representation. To illustrate how the functional representation can be used to generate a piece of diagnostic knowledge, suppose that some function of a device is not being achieved. The representation will indicate what behavior accomplishes the function, as well any structural relationships and assumptions that are required for the function. The compiler uses this specification to generate a list of hypotheses about the reason for the malfunction. Each hypothesis in turn corresponds to an example of one of the five aspects above, which is compiled accordingly.

Directions for future research include the following: We need to develop methods to check the correctness/consistency of a given device representation. We need to investigate the design of two other needed dimensions of device representation, namely the temporal dimension, and the dimension of interactions of functional units by way of feedback and communication. Also the causal dimension, which we have discussed, has to be integrated with the other two in a disciplined, practically useful, and cognitively meaningful framework. We need to identify the compilation processes that come into play to generate

other types of expert problem-solving structures, such as those that can predict the functional and behavioral consequences of changes of structure

Integrated Diagnostic Reasoning

It is becoming increasingly clear that diagnostic reasoning involves distinguishing different types of knowledge structures and associated problem-solving types. For example, depending upon the knowledge that is available, an agent might do classificatory, functional, or abductive reasoning as mentioned earlier. In some instances, even more basic "naive physics" reasoning may be required. Reasoning from the structure of a device (or anatomical system) to possible malfunctions is also often involved. In some cases, the diagnostic task needs to come up with a hypothesis that best explains the data, while in others data gathering itself may be a problem, *i.e.*, deciding what tests to order will be an issue.

In order to see how to integrate multiple types of reasoning tasks and provide proper control of reasoning, we are building an integrated diagnostic reasoning system in a subdomain of medicine called coagulation disorders. The system will coordinate classificatory problem-solving with multiple hierarchies, abductive reasoning, functional reasoning, and structural reasoning.

Complex Engineering Systems

We have joint projects under way with researchers in nuclear and chemical engineering departments on reasoning about complex engineering systems. Maintenance of safety and increase of product quality are the aims in such systems, and we are investigating issues of detecting malfunctions, sensor validation, and combining qualitative and quantitative reasoning.

Languages for Building Diagnostic Systems

Many kinds of problem-solving for expert systems have been proposed within the AI community. Whatever the approach, there is a need to acquire the knowledge in a given domain and implement it in the spirit of the problem-solving paradigm. Reducing the time to implement a system usually involves the creation of a high-level language that reflects the intended method of problem-solving. For example, EMYCIN was created for building systems based on MYCIN-like problem-solving. Such languages are also intended to speed up the knowledge acquisition process by allowing domain experts to input knowledge in a form close to their conceptual level.

CSRL, (or Conceptual Structures Representation Language) is a language for implementing expert diagnostic systems that are based on our approach to diagnostic problem-solving (Bylander *et al.*, 1983). It facilitates the development of diagnostic systems by supporting constructs that represent diagnostic knowledge at appropriate

Expert Systems for Design

Participants: David C. Brown and B. Chandrasekaran

levels of abstraction. A specialist represents the diagnostic knowledge about a diagnostic hypothesis. Message procedures describe the specialist's behavior in response to messages from other specialists. Knowledge groups determine how data relate to features of the hypothesis. Rule-like knowledge is contained within knowledge groups.

We have used CSRL in the implementation of two expert systems. Auto-Mech is an expert system that diagnoses fuel problems in automobile engines. It consists of 34 specialists in a hierarchy that varies from four to six levels deep. Red is an expert system whose domain is red blood cell antibody identification. CSRL is used to implement specialists corresponding to each antibody that Red knows about (around 30 of the most common ones) and to each antibody subtype.

Explanation in Planning and Problem-Solving Systems

Participants: B Chandrasekaran, John Josephson, and graduate students.

This is a newly-initiated DARPA-funded project for investigating the role of explanation in problem-solving systems, with emphasis on planning systems. Issues in explanation can be decomposed into two components: one dealing with the structure of the problem solving system, and the other having to do with the user and the issues of presentation. We are developing a framework for the first kind of explanation by explicitly identifying approaches to explain the control strategies and knowledge structures of a system.

We propose that explanation of problem-solving activity can be categorized into three types:

- How the data match the local goals This should describe how, at run time, problem data matched pieces of the knowledge base, and certain conclusions were drawn
- How the knowledge itself is justified We are concentrating on explanations that justify knowledge by showing how it can be derived from a deeper understanding of the domain.
- How the control strategy can be justified. A particular control strategy that was used in a certain situation can be justified by reference to the type of problem-solving that is being used by the system

Thus our general approach to generating explanations will be based on:

- Analyzing the domain and cataloging the types of explanations needed in terms of the taxonomy of explanations that we are developing;
- Synthesizing a complete complete explanation from the above elements.

This research is concerned with the design of mechanical components, and views design as a problem-solving activity (Brown and Chandrasekaran, 1984). The theory explains the activity of a human designer when solving a problem that falls into a particular subclass of mechanical design. An expert system called AIR-CYL has been implemented that embodies the theory. The system designs a particular type of air cylinder according to a set of user given requirements. The behavior of the system closely follows that of the human designer

We have established three classes of design activity which are distinguished by their problem-solving components. Our work refers only to the third class, which requires that at every stage of the design the designer knows both which sequences of design steps are appropriate and also what knowledge is required. The theory hypothesizes that the activity is organized around a hierarchy of concepts, where each concept is active in the design, and may be considered to be a specialist about some portion of the design. The hierarchy reflects the way that the designer thinks of the object during design, and it shapes the design process.

Each specialist has its own set of plans from which to select depending on the current stage of the design. The plans may request portions of the design from other specialists lower in the hierarchy, or may use tasks to make small additions to the design itself. Constraints may be planted at any point in order to test the validity of the design. The design data-base contains the current state of the design and a record of its progress, plus the collected requirements from the user.

The complete design process proceeds by first obtaining and checking the requirements for consistency. It then does rough design to establish whether full design is worth pursuing. If the rough design succeeds, then the full design is attempted by requesting a design from the top-most specialist. If a constraint fails, a redesign phase is entered until the problem can be fixed and design can continue.

To facilitate the building of the AIR CYL system, and class 3 design problem-solvers in general, a language has been provided in which to declare design specialists and describe plans. The Design Specialists and Plans Language (DSPL) has been used to capture the air cylinder design knowledge.

Much work remains to be done in this area before we fully understand what design is and how best to build systems to do it. However, we feel that we have captured the essential qualities of routine design, while discovering many interesting and difficult issues.

A Shell for Intelligent Databases

Participant: Jon Sticklen

The ASIM (A Shell for Intelligent Medical Databases) project is aimed at developing high-level support for the construction and use of intelligent databases, especially in the medical domain.

The ground work for understanding database reasoning was the PATREC database assistant in the original MDX implementation (Mittal *et al.*, 1984). PATREC provided both a data abstraction function and a coarse grained inference function for the diagnostic system, and was capable of temporal reasoning and reasoning about medical units. For example, PATREC was able to determine whether the patient's white blood count was normal, elevated, very low, etc., on the basis of the actual white blood count, and able to infer that anesthetics were recently administered to a patient if the patient recently had major surgery.

This kind of reasoning is essential for diagnosis (as well as other kinds of problem-solving), but it is not appropriate to embed this knowledge within the diagnostic knowledge structure. One reason is that the inferences are not diagnostic in nature, *i.e.*, they do not relate data to diagnostic hypotheses. Also, it would be redundant to associate these inferences directly with each diagnostic hypothesis that requires them. Rather, there is a need for a separate, knowledge-based data inference system.

PATREC was coded in a local implementation of FRL, with most of the "demons" written directly in UCI LISP. Thus most of the inferences of PATREC could not be altered except by expert LISP programmers. In an attempt to allow access to a broader user community, the ASIM project was initiated.

ASIM is being implemented on XEROX 1108 workstations. By making full use of the graphic display tools provided by the 1108s and the LOOPS object oriented language, ASIM will provide a database language as well as an environment for the construction and updating of intelligent medical databases. Our plans call for ASIM to be ready for alpha testing by the end of 1984.

Reasoning about Physical Systems

Participant: Tom Bylander

A recent AI approach for reasoning about the behavior of physical stems is qualitative simulation. The structure of the physical system and knowledge about the behavior of its components are used to derive a collection of constraints. Using these constraints, the simulation is performed and its results are interpreted. This research investigates a new method of reasoning for this problem, which we call consolidation.

The major processing sequence of consolidation is to hypothesize a composite component consisting of a selected subset of components, and then to infer the behavior of the composite from the behaviors of the components. Successful application of this sequence on increasingly larger composite components results in inferring the

behavior of the whole system. As a byproduct, a hierarchical behavior structure is produced that explains how the overall behavior is caused by the components' behavior. Also note that each reasoning step is localized over a small number of components and subsystems, avoiding the global problem-solving required for qualitative simulation.

This research also proposes a novel representation for behavior. Current theories describe behavior as arithmetic constraints on variables and their derivatives, which would imply that consolidation is purely a matter of mathematical manipulation. Instead, we describe the behavior of a component by the actions that the component performs upon "substances," *e.g.*, fluids, electric currents, control activations, or other things that can potentially move. We claim that there is a small set of behavior schemas that can directly represent these actions and that allow inferences about the behavior of composite components. Examples of schemas include permitting a substance to move from one place to another and influencing a substance to move.

We are implementing a version of consolidation, which will depend upon a few simplifying assumptions. The structural description will be limited to connection of components and containment of substances, thus reducing the amount of spatial reasoning required. Numerical attributes of behaviors (such as amount of influence or rate of movement) will be specified qualitatively. We hope to discover the limits of consolidation under these assumptions, and to learn how more complex spatial and temporal reasoning can be integrated into this process

Computing Facilities

Computing facilities used by the AI Group include four XEROX 1108 LISP workstations, a DECsystem 2060, and a VAX-11/780 running 4.2 BSD Unix (tm).

Research Support

Research by the AI Group has been supported by many sources over the years. In 1984-85, these sources include the Defense Advanced Research Projects Agency, the Air Force Office of Scientific Research, the Computer Science as well as the Chemical and Process Engineering sections of the National Science Foundation, and the University Distribution Program of the Battelle Corporation.

References

- Brown, D C , & B Chandrasekaran. (1984) *Expert systems for a class of mechanical design activity* In Proceedings of the IFIPS WG5 2 Working Conference on Computer-Aided Design, Budapest
- Bylander, T, S Mittal., & B Chandrasekaran (1983) *CSRL A language for expert systems for diagnosis*. In IJCAI-8, 218-221 An expanded version will appear in a special issue of *International Journal of Computers and Mathematics* on practical artificial intelligence systems
- Chandrasekaran, B (1983) Towards a Taxonomy of Problem-Solving Types *AI Magazine* Vol. 4, No 1: 9-17

Chandrasekaran, B (1984) Expert Systems: Matching Techniques to Tasks In W Reitman (Ed.), *Artificial Intelligence Applications for Business* Norwood, New Jersey: Ablex

Chandrasekaran, B , & S Mittal (1982) *On deep versus compiled knowledge approaches to medical diagnosis* AAAI-82, 349-354 An extended version appears in *International Journal of Man-Machine Studies* 19(5): 425-436, 1983

Chandrasekaran, B , & S Mittal (1984) *Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems* In M Yovits (Ed) *Advances in Computers*, Vol. 22 New York: Academic Press

Gomez, F , & B. Chandrasekaran (1981) *Knowledge organization and distribution for medical diagnoses*, *IEEE Transactions on Systems, Man and Cybernetics* SMC-11(1): 34-42.

Josephson, J R , B Chandrasekaran., & J W Smith (1984) *Assembling the best explanation* In *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, Denver , 185-190

Mittal, S., B Chandrasekaran , & J Sticklen (1984) PATREC: A Knowledge-Directed Database for a Diagnostic Expert System *Computer* Vol 17, No 9: 51-58

Sembugamoorthy, V , & B. Chandrasekaran (1984) *Functional Representation of Devices and Compilation of Diagnostic Problem Solving Systems* Tech Rep , AI Group, Dept. of Computer and Information Science, The Ohio State University To appear in *Cognitive Science*.

Sticklen, J , B Chandrasekaran., J W Smith, & J Svirbely (1984) *A comparison of the diagnostic subsystems of MDX and MYCIN* In *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, Denver, 205-212 An expanded version, entitled "MDX-MYCIN: The MDX Paradigm Applied to the MYCIN Domain," will appear in a special issue of *International Journal of Computers and Mathematics* on practical artificial intelligence systems.

INTERESTED IN AI FOR HOME COMPUTERS?

TOPSI - A higher-order AI language for MS-DOS or CP/M systems

TOPSI is:

- Available now for most 5¼ in CP/M systems with 65k bytes and for MS-DOS machines with at least 128k bytes
- Available soon for the Macintosh
- An implementation of OPS-5 with extensions to provide LHS computation and list processing
- An efficient, practical way to develop AI software on home computers

TOPSI is not:

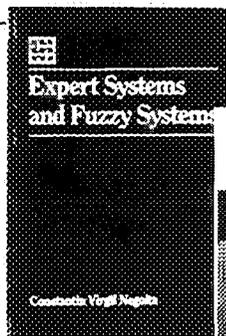
- A Toy - because of its highly efficient storage utilization, even the 65k CP/M version can support an expert system with several hundred rules
- A LISP Add-on - TOPSI is coded in Pascal to provide the most efficient use of memory and CPU time
- Expensive - it costs \$75.00 plus \$5 shipping, plus 3% tax if you live in Georgia Order by Check, C O D . Visa or Mastercard



Dynamic Master Systems, Inc.
 PO Box 566456
 Atlanta, Ga 30356
 (404) 565-0771

New Books and Contemporary Classics...

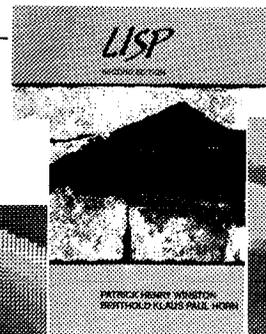
The Best in AI from Addison-Wesley.



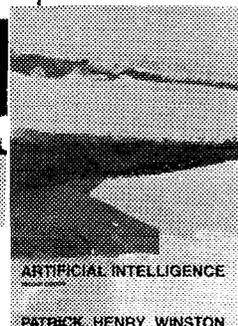
Expert Systems and Fuzzy Systems, Constantin Virgil Negoita, \$47.95



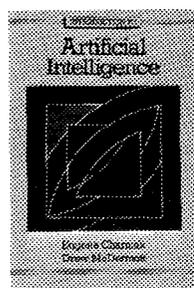
Building Expert Systems, Frederick Hayes, Roth Donald Waterman and Douglas Tenat, Editors, \$34.95



LISP, 2nd ed., Patrick Henry Winston and Berthold Horn, \$23.95



Artificial Intelligence 2nd ed., Patrick Henry Winston, \$31.95



Just released!

Introduction to Artificial Intelligence, Eugene Charniak and Drew McDermott, \$33.95
 A comprehensive "programmer's introduction" to AI using LISP. In-depth examinations of internal representation, language processing, planning, reasoning under uncertainty, and other key topics. Emphasizes underlying mathematical and linguistic theories

Silicon Valley's Bookshop at Your Service

Need a computer book now? Want several books from different publishers? Computer Literacy, the world's largest computer bookstore, has more than 8000 titles ready for **next day shipping**; new books, standard texts, books you won't find elsewhere!

Complete selection of professional books on AI, robotics, LISP, PROLOG, machine architecture, VLSI, human factors-any computer topic!

Order any AI book and receive our "Silicon Valley Favorites" annotated catalog at no charge (regularly \$3.00).

COMPUTER LITERACY BOOKSHOP
 520 Lawrence Expy Dept 310B
 Sunnyvale CA 94086; (408) 730-9955/ext 15

