AI Magazine Volume 1 Number 1 (1980) (© AAAI)

Over the past two years we have started a program of research into the development of VLSI systems. Professor Gerald Sussman, John Holloway, Tom Knight, Howie Shrobe, and others have developed a number of systems, including a single ship SCHEME processor. We have TTL implementations of a number of our vision algorithms, and we are in the process of switching some of them to VLSI.

Expert Problem Solving and VLSI Design

Traditional automated synthesis techniques for circuit design are restricted to small classes of circuit functions for which mathematical methods exist. Professor Gerald J. Sussman and his group have developed computer-aided design tools that can be of much broader assistance. The work has developed the idea of analysis by propagation of constraints. Guy L. Steele developed a language to support such programming, Johan de Kleer studied causal and teleological reasoning in the recognition of circuit function from schematics, and Howie Shrobe has worked on constraint satisfaction and the development of an interactive knowledge-based system for substantially supporting VLSI design. Jon Doyle has studied belief revision via truth maintenance and non-monotonic logics, as well as self-conscious adaptive deliberate reasoning programs. Richard Waters, Charles Rich, and Howie Shrobe have developed the idea of a programmer's apprentice.

Learning and Language

Professor Patrick H. Winston has studied systems that reason by analogy. During the past year, he has worked out a theory and implemented a system. Professor William Martin has continued his work on computer understanding of natural language. Robert Berwick has developed a theory and system that can learn a substantial part of the syntax of English, based on a theory of syntactic recognition developed in the laboratory by Mitchell Marcus. Other work on Natural Language understanding has centered on determination of focus during discourse and response generation.

Personal Machines and Computing Concepts

Intelligent information processing places enormous demands on computing resources, and the computational facilities required to effectively support research in Artificial Intelligence are equally large. About five years ago it became clear to Richard Greenblatt and his associates that our computing needs would be best served by powerful personal computers which execute LISP extremely efficiently and which maintain the level of system support. Currently we have eleven LISP machines and plan a further nine over the next couple of years. All our machines are interconnected through an eight-megabit packet switched network called the CHAOSNET.

On the theoretical side, Professor Randall Davis has developed a theory of distributed problem solving which he calls the Contract Net, in which goals are broadcast to sub-contractors, who then bid for them based on their current work load, the description of the contract, and their abilities. Professor Carl Hewitt and his colleagues have developed an architecture for large scale distributed systems which they call the Apiary, and which is based on the migration of processes to free processors to optimize throughput and achieve load balancing. They have introduced a descriptive formalism called OMEGA, which contributes to many of the issues of current concern in knowlege representation, and they have applied it to describe the various structured entities such as messages in a study of cooperating sequential programming.

Basic Theory

Professor Marvin Minsky has worked on a theory of human thinking, which likens the mind to a society of agents and attempts to combine a number of insights from psychoanalytic, developmental, and cognitive theories of mind. Further work by Richard Greenblatt and Dr. Lucia Vaina develops the idea of thread memory.

Information Sciences Institute
University of Southern California

ISI is an off-campus research center in the University of Southern California's School of Engineering. The Institute engages in a broad set of research and application oriented projects in the computer sciences. These projects range from advanced research efforts aimed at producing new concepts, through development of prototype systems, to operation of a major Arpanet computer facility. Except for the Computer Services group, these projects range from one up to ten people. ISI is composed of about 130 full time people plus a number of part-time graduate research assistants and, except for a small operation in San Diego, is entirely housed in facilities at Marina del Rey, California. ISI currently operates about a half-dozen Tenex and TOPS-20 systems as network facilities. Part of these facilities support in-house research; the remainder supports research at other ARPA-supported sites.

AI-Related Projects

Specification Formulation, Testing, and Implementation

Robert Balzer, Lee Erman, Martin Feather, Neil Goldman, Philip London, and David Wile

This is an area of three interrelated projects aimed at facilitating the creation, testing, and optimized implementation of program specifications. The first effort, called Specification Acquistion From Experts (SAFE), is directed at helping people create unambiguous, consistent, and complete formal program specifications through informal description. While end users are quite capable of providing informal process-oriented descriptions of the task being automated, formalisms of any kind provide major impediments. The informal descriptions are characterized by partial, rather than complete, constructs. The system uses a knowledge base of program well-formedness rules to disambiguate and complete the informal natural language descriptions. An early version of the system has successfully converted several small informal specifications into formal specifications (Balzer 78). Attention is now focused on handling large specifications through incremental formalization; to support this, a generalized AI architecture for knowledge-based systems, based on the Hearsay-II blackboard model, has been designed and implemented (Balzer 80a).

The second project (Balzer 80b) seeks to provide a methodology guaranteed to produce only valid implementations of the formal specifications by involving the computer in the implementation process to maintain consistency of the evolving implementation with the original formal specification. This is accomplished by limiting the activity of programmers to the choice of optimizations to be utilized. These optimizations are embodied in a catalog of transformations, applied by the system in response to the programmer's choices, after verifying that they are validly applicable.

The third project seeks to ensure that a formal specification matches the user's intent, by enabling its behavior to be investigated systematically through symbolic execution. The focus of this project will be in explaining the simulated behavior meaningfully.

Designed for use by all of these projects is a new abstract specification language, called GIST (Goldman 79).

Cooperative Interactive Systems

David Wilczynski, Robert Lingard, and William Mark

Current interactive systems must be made more habitable for a wide variety of users, especially those having little experience with computing. We are currently building the Consul (Mark 80) system to explore methods for allowing natural interaction—natural-language requests, explanation of system facilities, user-understandable error handling—between users and a set of online tools (for text manipulation, message handling, etc.) Consul works by mapping user requests into appropriate system actions as dictated by a detailed model of system capabilities and requirements. This model is based on built-in tool-independent knowledge, but is tailored to individual tools as part of the tool building process. Our research therefore includes not only modelling and mapping methods that generalize across tools, but also a programming environment for tool builders that integrates their tools into Consul's knowledge base.

Knowledge Delivery in Multiparagraph Text

William Mann and James Moore

This project is developing new methods for autonomous text composition by machine, with the focus at the larger-than-sentence level. The project is producing a series of knowledge delivery systems. The first of these (Mann 79a), which was called KDS (Knowledge Delivery System), was developed as a testbed for methods of text composition from computer-internal knowledge representations such as semantic nets. Innovations in KDS included a new expressive paradigm that does not rely on dividing the semantic net into sentence-sized pieces; a problem-solving text organizer that delegates part of its work to other processes; explicit principles for ordering and grouping concepts for expression; rules for aggregating small expressions into larger, more natural ones; and a novel text improvement methods based on a rule-competition-oriented variant of the classic hill-climbing algorithm. The ideas in KDS were tested on two principal domains: description of the structure and use of a computer message system and description of a multi-actor, multi-contingency policy on how to respond to indications of fire in a computer center.

A second system, called PENMAN, is currently being developed to explore the problems of creating a portable knowledge delivery facility which is useful in multiple knowledge domains. PENMAN will seek to deliver knowledge (in English) from inside a system which was not designed to have a knowledge delivery component.

Computer Comprehension of Natural Language

William Mann and James Moore

Despite its great social significance, human communication remains a complex, poorly understood process. Particularly obscure is comprehension, the process by which a sequence of words effects the cognitive state of the hearer. Human-computer interaction could be made easier and more effective if computers could comprehend strings of symbols in more human-like ways. Our prior research has created a model of human communication as a kind of goal-pursuit activity (Mann 79b). The model represents four principal levels of knowledge in communication: goals being pursued, speech acts, and linguistic symbols used to convey all of these. This project is building an experimental system called DCS (Dialogue Comprehension System), which will read and follow transcripts of actual human dialogues. In particular, it will contain processes that can recognize the establishment, pursuit, and satisfaction of individual goals, including representation of state changes in the communicating individuals.

InterLisp Developement

Mel Pirtle, David Dyer, and William Rizzi

This effort is developing and will maintain a portable, large address-space InterLisp implementation. The first version is currently being done for the VAX computer.

Distributed Sensor Nets-DSN

Danny Cohen, Jeff Barnett, Iris Kameny, and Yechiam Yemini

The objective of this project is to develop a methodology for defining DSN components and their organization that is both sensor and scenario independent. The three main DSN problem areas that are currently being addressed are: how to combine different kinds of data into higher level knowledge and assign appropriate levels of confidence to the knowledge; the development of sophisticated, goal-oriented communication protocols; and the design of a closely coupled user-DSN interface to carry out DSN command, control, communication, and intelligence functions.

A software testbed has been designed and is currently being implemented to test and evaluate different DSN configurations. A major goal is to discover how to incrementally automate some of the command, control, communication, and intelligence functions of the user as soon as they can be understood and specified. This should allow the user to spend increasingly larger portions of his time on more complex tasks and decision making, and should indicate the appropriate distributed process structure and communication needs of the system.

Selected References

(1) (Balzer 78) Balzer, R., N. Goldman, and D. Wile, "Informality in Program Specifications," IEEE Trans. Software Eng. SE-4, (2), March 1978.

(2) (Balzer 80a) Balzer, R., L. D. Erman, P. London, and C. Williams, "Hearsay-III: A Domain-Independent Framework for Expert Systems," in 1st National Artificial Intelligence Conf., Palo Alto, CA, August 1980. (To appear)

(3) (Balzer 80b) Balzer, R., N., "Transformational Implementation: An Example," IEEE Trans. Software Eng., November 1980. (Also appeared as Technical Report, USC/Information Sciences Institute, RR-79-79).

(4) (Goldman 79) Goldman, N. M. and D. S. Wile, "A Relational Data Base Foundation for Process Specifications, " in Chen, P. P. (ed.), Int. Conf. on Entity-Relationship Approach to Systems Analysis and Design, pp. 413-432, December 1979.

(5) (Mann 79a) Mann, W. C., and J. A. Moore, Computer as Author—Results and Prospects, USC/Information Sciences Institute, Technical Report RR-79-82, 1979.

(6) (Mark 80) Mark, W., "Rule-Based Inference in Large Knowledge Bases," in 1st National Artificial Intelligence Conf., Palo Alto, CA, August 1980. (To appear)

Other Projects

Program Verification

Susan Gerhar, Ralph London, Ray Bates, Rod Erickson, David Thompson, and David Wile

Study of methods to prove logical correctness of programs.

Formal Specification of Systems

Steve Crocker, Richard Gillmann, Vittal Kini, Leo Marcus, David Martin, Wiliam Overman, and Dono van-Mierop

Three interrelated projects aimed at understanding how to specify and verify computer systems and languages.

VLSI Implementation System

Danny Cohen, Vance Tyree, Ron Ayres, Leroy Richardson, and George Lewicki.

The process of decoupling the design process from the idiosyncracies of the various fabrication facilities, the information management of Multi-Project Chip runs, and high-level design methodology.

Wide Band Communication

Steve Casner, William Brackenridge, Danny Cohen, Randy Cole, and Eric Mader

Exploration of satellite-based wide-band packet computer communication, both the technology needed to support it and of opportunities for its use.

Command and Control Graphics

Richard Bisbey, Dennis Hollingworth, and Ben Britt

Development of a device-independent graphics system and graphics-oriented command and control applications programs.

Internet Concepts Research

Jon Postel, Danny Cohen, Carl Sunshine, and Greg Finn

Exploration of aspects of protocols for the interconnection of computer communication networks, specifically the design and prototype implementation of an internetwork multimedia computer message system and the design and analysis of internetwork host and gateway protocols.

QPRIM

Lou Gallenson and Joel Goldberg

Building a facility for producing computer emulators, with emphasis on debugging target programs.

Application Downloading

Robert Balzer, Al Cooperband, Martin Feather, Philip London, and Dave Wile

Work to improve utility of processing terminals by permitting the division of responsibility between host and terminal to be made after application is designed, with the partitioning and synchronization to be achieved via designerdirected machine transformations of a checked-out and debugged single-machine implementation.

Packet Radio Terminal System Evaluation
Tom Ellis and Steve Saunders
Work intended to result in a demonstration-level portable terminal to test and evaluate various solutions to the issues raised by extreme portability in the packet-radio environment.

The Stanford Heuristic Programming Project: Goals and Activities

by the Staff of the Heuristic Programming Project

The Heuristic Programming Project (HPP) of the Stanford University Computer Science Department is a laboratory of about fifty people—faculty, staff, and graduate students—whose main goals are these:

> ...to model, and thereby to gain a deep understanding of, the nature of scientific reasoning processes in various types of scientific problems, and various areas of science and medicine;
> ...as part of the methodology, and as a coordinate activity, to construct "Expert Systems"—programs that achieve high levels of performance on tasks that normally require significant human expertise for their solution; the HPP therefore has a natural applications orientation.

The HPP was started by Professor Edward A. Feigenbaum and Professor Joshua Lederberg (now President, Rockefeller University) as the DENDRAL project in 1965. Professor Bruce Buchanan joined shortly thereafter, and is Co-Principal Investigator of the HPP.

For its computing facilities, the HPP uses the Stanford-based SUMEX-AIM National Resource for Applications of AI to Medicine and Biology (a pair of DEC KI-10s and a DEC 2020); and the SU-SCORE machine (a DEC 2060).

HPP research includes basic AI research, applications-oriented research, and development of community tools useful for building expert system. The programs developed can be used as problem solving assistants or tutors, but also serve as excellent vehicles for research on representation and control of diverse forms of knowledge.

The main issues of building expert systems are coincident with general issues in AI. We build working programs that demonstrate the feasibility of our ideas within well defined limits. By investigating the nature of expert reasoning within computer programs, such processes are "demystified." Ultimately, the construction of such programs becomes itself a well-understood technical craft.

The foundation of each of the projects described in this article is expert knowledge: its acquisition from practitioners, its accommodation into the existing knowledge bases, its explanation, and its use to solve problems. Continued work on these topics provides new techniques and mechanisms for the design and construction of knowledge bases, its explanation, and its use to solve problems. Continued work on these topics provides new techniques and mechanisms for the design and construction of knowledge-based programs; experience gained from the actual construction of these systems then feeds back both (a) evaluative information of the ideas' utility and (b) reports of quite specific problems and the ways in which they have been overcome, which may suggest some more general method to be tried in other programs.

One of our long-range goals is to isolate AI techniques that are general, to determine the conditions for their use and to build up a knowledge base about AI techniques themselves. Under support from ARPA, NIH/NLM, ONR, NSF, and industry, the HPP conducts research on five key scientific problems of AI, as well as a host of subsidiary issues:

1. *Knowledge Representation*—How shall the knowledge necessary for expert-level performance be represented for computer use? How can one achieve flexibility in adding and changing knowledge in the continuous development of a knowledge base? Are there uniform representations for the diverse kinds of specialized knowledge needed in all domains?
2. *Knowledge Utilization*—What designs are available for the inference procedure to be used by an expert system? How can the control structure be simple enough to be understandable and yet sophisticated enough for high performance? How can strategy knowledge be used effectively?
3. *Knowledge Acquisition*—How can the model of expertise in a field of work be systematically acquired for computer use? If it is true that the power of an expert system is primarily a funciton of the quality and completeness of the knowledge base, then this is the critical "bottleneck" problem of expert systems research.
4. *Explanation*—How can the knowledge base and the line of reasoning used in solving a particular problem be explained to users? What constitutes an acceptable explanation for each class of users?
5. *Tool Construction*—What kinds of software packages can be constructed that will facilitate the implementation of expert systems, not only by the research community but also by various user communities?

Artificial Intelligence is largely an empirical science. We explore questions such as these by designing and building programs that incorporate plausible answers. Then we try to determine the strengths and weaknesses of the answers by experimenting with perturbations of the systems and extrapolations of them into new problem areas. The test of success in this endeavor is whether the next generation of system builders finds the questions relevant and the answers applicable to reduce the effort of building complex reasoning programs.

Thus, the work of the HPP can be described in three ways: by the Expert Systems it writes, by the software tools it creates, or by the basic research issues that motivate the