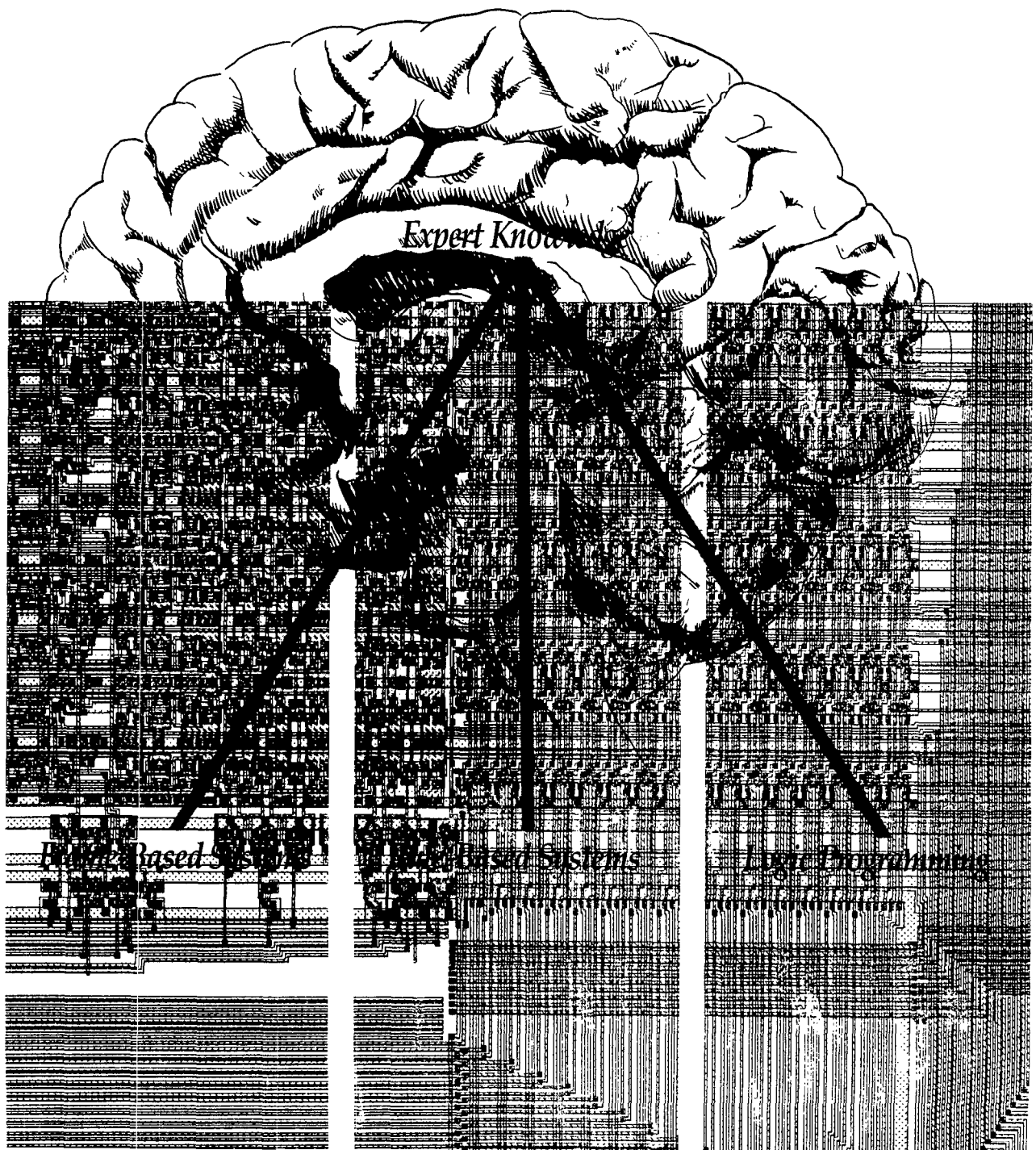


# ***Special Section on ARCHITECTURES FOR KNOWLEDGE-BASED SYSTEMS***



A fundamental shift in the preferred approach to building applied artificial intelligence (AI) systems has taken place since the late 1960s. Previous work focused on the construction of general-purpose intelligent systems; the emphasis was on powerful inference methods that could function efficiently even when the available domain-specific knowledge was relatively meager. Today the emphasis is on the role of specific and detailed knowledge, rather than on reasoning methods.

The first successful application of this method, which goes by the name of knowledge-based or expert-system research, was the DENDRAL program at Stanford, a long-term collaboration between chemists and computer scientists for automating the determination of molecular structure from empirical formulas and mass spectral data. The key idea is that *knowledge is power*, for experts, be they human or machine, are often those who know more facts and heuristics about a domain than lesser problem solvers. The task of building an expert system, therefore, is predominantly one of "teaching" a system enough of these facts and heuristics to enable it to perform competently in a particular problem-solving context. Such a collection of facts and heuristics is commonly called a *knowledge base*. Knowledge-based systems are still dependent on inference methods that perform reasoning on the knowledge base, but experience has shown that simple inference methods like *generate and test*, *backward-chaining*, and *forward-chaining* are very effective in a wide variety of problem domains when they are coupled with powerful knowledge bases.

If this methodology remains preeminent, then the task of constructing knowledge bases becomes the rate-limiting factor in expert-system development. Indeed, a major portion of the applied AI research in the last decade has been directed at developing techniques and tools for knowledge representation. We are now in the third generation of such efforts. The first generation was marked by the development of enhanced AI languages like Interlisp and PROLOG. The second generation saw the development of knowledge representation tools at AI research institutions; Stanford, for instance, produced EMYCIN, The Unit System, and MRS. The third generation is now producing fully supported commercial tools like KEE™ and S.I.™ Each generation has seen a substantial decrease in the amount of time needed to build significant expert systems. Ten years ago prototype systems commonly took on the order of two years to show proof of concept; today such systems are routinely built in a few months.

Three basic methodologies—frames, rules, and logic—have emerged to support the complex task of storing human knowledge in an expert system. Each of the articles in this Special Section describes and illustrates one of these methodologies. "The Role of Frame-Based Representation in Reasoning," by Richard Fikes and Tom Kehler, describes an object-centered view of knowledge representation, whereby all knowledge is partitioned into discrete structures (*frames*) having individual properties (*slots*). Frames can be used to represent broad concepts, classes of objects, or individual *instances* or

components of objects. They are joined together in an *inheritance hierarchy* that provides for the transmission of common properties among the frames without multiple specification of those properties. The authors use the KEE knowledge representation and manipulation tool to illustrate the characteristics of frame-based representation for a variety of domain examples. They also show how frame-based systems can be used to incorporate a range of inference methods common to both logic and rule-based systems.

"Rule-Based Systems," by Frederick Hayes-Roth, chronicles the history and describes the implementation of production rules as a framework for knowledge representation. In essence, production rules use **IF conditions THEN conclusions** and **IF conditions THEN actions** structures to construct a knowledge base. The author catalogs a wide range of applications for which this methodology has proved natural and (at least partially) successful for replicating intelligent behavior. The article also surveys some already-available computational tools for facilitating the construction of rule-based knowledge bases and discusses the inference methods (particularly backward- and forward-chaining) that are provided as part of these tools. The article concludes with a consideration of the future improvement and expansion of such tools.

The third article, "Logic Programming," by Michael Genesereth and Matthew Ginsberg, provides a tutorial introduction to the formal method of programming by description in the predicate calculus. Unlike traditional programming, which emphasizes *how* computations are to be performed, logic programming focuses on the *what* of objects and their behavior. The article illustrates the ease with which incremental additions can be made to a logic-oriented knowledge base, as well as the automatic facilities for inference (through theorem proving) and explanation that result from such formal descriptions. A practical example of diagnosis of digital device malfunctions is used to show how significant and complex problems can be represented in the formalism.

A note to the reader who may infer that the AI community is being split into competing camps by these three methodologies: Although each provides advantages in certain specific domains (*logic* where the domain can be readily axiomatized and where complete causal models are available, *rules* where most of the knowledge can be conveniently expressed as experiential heuristics, and *frames* where complex structural descriptions are necessary to adequately describe the domain), the current view is one of synthesis rather than exclusivity. Both logic and rule-based systems commonly incorporate frame-like structures to facilitate the representation of large amounts of factual information, and frame-based systems like KEE allow both production rules and predicate calculus statements to be stored within and activated from frames to do inference. The next generation of knowledge representation tools may even help users to select appropriate methodologies for each particular class of knowledge, and then automatically integrate the various methodologies so selected into a consistent framework for knowledge.

—Peter Friedland, Guest Editor

Knowledge Systems Laboratory, Stanford University