prepared in advance of the implementation.

Knowledge engineering tools are rarely exactly suited to a particular engineering problem. More often, they require adaptation and evolve during the knowledge acquisition and implementation process. Engineering problems are inherently diverse and will continue to pose challenges to the builders of such tools.

Higher level program strategies, such as interfaces and interactions among modules, are often represented in the same formalism as the knowledge base. Thus, these aspects of the program also grow gradually and interactively as the program evolves to handle more and more test cases. Exploratory programming environments seem appropriate for these tasks.

Four articles

This issue reports on software engineering aspects of knowledge-based expert systems that involve diagnosis, control, simulation, and design.

Diagnosis. In the first article, "A Qualitative Modeling Shell for Process Diagnosis," Timothy F. Thompson and William J. Clancey describe the use of an existing shell to develop an expert system for another domain. Caster, a system for diagnosing malfunctions in industrial sandcasting, is constructed in the shell environment of Heracles, a heuristic classification shell generalized from the Neomycin medical diagnostic system.

The heuristic classification technique helps develop knowledge-based expert systems for the derivation of the engineering problem spectrum. The authors show that a well-defined diagnostic procedure and a relational language to state a qualitative domain model are general tools that can be applied to engineering as well as medicine. The authors also used manuals and available literature, rather than a human expert, to build the system—a useful knowledge acquisition strategy.

Control. In "An Expert System for Real-Time Control," M. Lattimer Wright and his coauthors touch on a number of practical software engineering issues. Their experimental system for dealing with control problems in military and advanced industrial applications has some unique problems. A hybrid system, Hexcon combines algorithmic real-time control techniques with knowledge-based approaches to higher level reasoning to organize the inference engine at different levels of detail.

Among the Hexcon's design goals are a capacity of 5000 rules in a 512K-byte microcomputer, a response time of 10 to 100 milliseconds, control of about 1000 objects, and continued functioning in an uncertain environment. A prototype has demonstrated these capabilities in several problem domains.

Simulation. In the third article, YV. Ramana Reddy and his coauthors describe KBS, a knowledge-based simulation system developed as part of a comprehensive, intelligent factory management system. Designed to help managers analyze complex systems, KBS takes advantage of several artificial-intelligence programming paradigms to provide a simulation-based decision support tool.

KBS incorporates a number of features not found in general-purpose simulation environments. One essential difference lies in the use of heuristics to evaluate the output from the simulation program. The frame-based Schema Representation Language, along with graphics utilities, provided the environment needed for system development.

KBS offers facilities for automatic generation of model scenarios, goal-directed instrumentation, rule-based diagnosis, cause-and-effect analysis, causal path analysis, and interactive model building. It has been used successfully in simulation applications that include printed circuit board manufacturing, light bulb manufacturing, flexible assembly, and corporate distribution and inventory.

Design. In VLSI circuit design, implications and consequences of a particular design may not become clear until it has been partially—or almost fully—worked out. At that point, the layout may have to be completely reworked.

The complexity of IC design has made it difficult to apply traditional programming methods to the task. Recently, however, artificial intelligence techniques have been used successfully in computer-aided IC design systems. In the final article, Jin Kim and John McDermott describe and evaluate one such system, Talib.

Talib assists the VLSI designer with cell layouts for NMOS technology. Talib's rule base, implemented in the OPS5 general-purpose language, is rich in domain knowledge. Talib also combines planning with