Aerospace conferees survey AI technology

Galen Gruman, Assistant Editor

The difficulties of applying artificial intelligence systems to the increasingly complex aerospace industry programs was a predominant theme at the American Institute for Aeronautics and Astronautics' Computers in Aerospace Conference held in Long Beach, California, in late October.

The "Future Directions in AI" panel focused primarily on the emerging technology's applications. The reports presented showed that much more needs to be done before AI can be used to dramatically improve software creation, flexibility, automation, and reusability.

Most expert systems follow essentially the same approach: Data is evaluated against a knowledge base and a rule base. An inference engine decides how to correct contradictions between what is expected to happen and what is happening.

A user interface connects this looping, self-evaluating system to people. With the interface, users command the system, which then decides how to carry out commands based on what is occurring in the system. A knowledge acquisition tool in the user interface can send data to the knowledge base and inference engine to teach them about new situations and solutions.

This expert-system model parallels the traditional learning combination of rule-based and experience-based education. Most expert systems use rules—basically sets of if-then procedures—to make their decisions.

Flight monitor. Victoria Regenie of the National Aeronautics and Space Administration's Ames Research and Dryden Flight Research centers in California described work on an expert-system-based flight-systems monitor. The monitor's goal is to downlink satellite telemetry to an on-board expert system so a computer can help pilot an aircraft.

In the foreground, the system would process information at flight control rates for real-time analysis. In the background, it would answer pilot queries using available time. So far, those in the project have designed indicator rules for knowledge acquisition, completed half the expert system monitor, and constructed about three quarters of the knowledge base.

The planned system will have a flight base and simulation base to compare reality with what the system expects to happen. The team is developing a knowledge-acquisition tool that will maintain clause consistency by making new rules from old ones.

The flight system's decision making would be essentially carried out through multiple systems that access a common knowledge base. Each system evaluates the data according to its area of expertise. Voting between the systems' activators determines the solutions followed (this is the system a space shuttle's on-board computers use).

Blackboard approach. William Erickson of the Ames Research Center's space station automation research group described an increasingly popular approach to multiple-expert-system decision making: the blackboard. The blackboard essentially controls the flow of discussion between the systems.

Each expert contributes its knowledge and analysis to a blackboard accessible by all experts. Each can change, delete, or add to the others' contributions. However, only one expert may write on the blackboard at any time. A moderator decides who may contribute when. Usually, a common decision based on the systems' combined expertise is reached. If not, voting takes place.

As had many at the session in their own research, the Ames/Dryden group greatly underestimated the number of rules their system's rule base needed. Writing in Common Lisp, they had expected to need about 200 rules. Instead, they ended up with more than 1000. "It's an escalating process," Regenie said. It can also lead to unwieldy and slow computations.

Fault evaluation. Michael Georgeff of SRI International in Menlo Park, California, discussed fault evaluation on the space shuttle, an area where "there's a strong incentive to automate." However, it is also an area difficult to automate—especially because of the thousands of rules involved.

The current system—a collection of thousands of pages of manuals that contain procedures that call up other procedures that in turn call up other procedures—is nonextensible and hard to modify. "We need a system that can explain what it's trying to do," Georgeff said. Standard programming languages fail to do that because verification across procedure calls are difficult.

If you use traditional expert-system tools, you have little control structure, essentially because nothing keeps track of past events that may have an effect on the solution of a problem. Such control conditions become so cumbersome and complex that you lose understanding and efficiency, he explained.

For fault evaluation and correction, an expert system must also make a decision in a reasonable amount of time and be able to put problems in perspective. One floor participant illustrated this point by describing an expert system that was given a chess problem to solve while the room it was in was set afire. Faced with this situation, an expert system, the storyteller warned, had better not wait until it solved the chess problem before deciding to leave the room or put out the fire.

Georgeff said he is exploring a space shuttle fault-correction automation system that has no rules. Instead, the system would use generalized procedures and specifications. The context of the problem would be part of the procedures rather than part of a long string of unwieldy if-then statements that invoke a procedure.

Essentially, he explained, the system would be a set of executable specifications with associated triggers or demons that call up the procedures. Such a design could circumvent the large, cumbersome lists of rules used in traditional systems, Georgeff said.

Fuzzy logic. Lofti Zadeh of the University of California at Berkeley dealt with the issue of logic and meaning in AI rather than with specific applications. He described the calculus of his fuzzy logic, which tries to quantify degrees of usuality.

"Many in AI feel classical logic can deal with common-sense reasoning. I disagree. Facts and rules are imprecise." Zadeh said that most facts and rules have elastic meanings. For example, a hotel that tells its guests "checkout time is 11 a.m." might mean anything from "checkout time is any time before 11 a.m." to "checkout time is any time before noon."

The rule is context-dependent, subject to local differences from hotel to hotel and from clerk to clerk. What rule-based system should do, Zadeh argued, is "try to describe the meaning in a context rather than legislate it." He used the term dispositional evaluations to mean a collection of unrigid dispositions (a disposition is a "fact" that is preponderantly, but not always, true).

A person, he said, interprets the phrase "young men like young women" to mean "(most) young men like (mostly) young women." What AI systems must do is figure out how often the strict interpretation of "young men like young women" is true. Knowing the degree of usuality allows expert system to handle—indeed, expect—deviations from the norm.