THE USE OF ARTIFICIAL INTELLIGENCE TECHNIQUES IN THE FEDERAL AVIATION ADMINISTRATION SYSTEMS

David A. Spencer, Massachusetts Institute of Technology, Lincoln Laboratory

Software techniques and development methodologies from artificial intelligence have a role to play in the National Airspace System as this system becomes more automated. However, the applications must be carefully chosen. Artificial intelligence (AI) techniques must always be compared with more conventional "algorithmic" methods for achieving the required behavior. Where an algorithmic approach is known, it is generally preferable to an AI approach. At the opposite end of the spectrum of complexity, there are behaviors which may be desirable but which cannot be achieved with any current technology. Both the "algorithmic" boundary and the "impossible" boundary are fuzzy and change over time. Research is often needed to determine where a specific task lies. AI techniques appear to have advantages in areas where the necessary flow of control is very complex, or not well understood, or where the system is likely to be extensively modified during its lifetime in unforeseen ways, or where it is desirable that the reasons behind the system's behavior be understandable to users. The latter might include giving the system the ability to answer questions about why it took a certain action.

Software Development Environments and Methodologies

The software development environments and methodologies used by AI researchers are important in their own right, whether or not the system being developed uses AI techniques. These include powerful personal work stations, such as LISP machines and the highly integrated environments created by the LISP machine software systems. Awareness of and experience with these capabilities should be encouraged within the Federal Aviation Administration (FAA). Although the FAA normally does not develop its own software, it would benefit from use of similar techniques by its contractors.

One interesting software development methodology is that used in creating expert systems. This involves quickly creating a prototype design, using some representation such as production rules that is modified. This prototype is then tried out on test problems and the results criticized by an expert. If errors are found it is relatively easy (compared with conventional implementation in a procedural language) to find the offending rule or fact and modify it to get the desired behavior. This approach has two advantages. First, it exposes the design to criticism by the eventual users or other knowledgeable persons early in the design phase. Second, this criticism is based on actual performance of the prototype on test problems, rather than on a system description in a specification or design review document. Over time, these test problems are made harder and more realistic, ultimately becoming the actual production environment.

Functional Specifications

While such a prototype may not satisfy all the system requirements in terms of real time performance, capacity, or resource usage, it does provide the basis for a functional specification of the system that is known to work as desired. This functional behavior can then be reimplemented, if necessary, to achieve performance goals. This approach explicitly recognizes that a certain amount of debugging of the requirements, the specification and the design always occurs, and that it is better to do this in a low overhead situation rather than at the point where dozens of contract personnel are trying to implement the operational version of the system.

This approach can be contrasted with two alternatives. In the first the requirements are very broadly stated, the contractor is given considerable latitude in how to meet those requirements, and it is not always clear whether or not a particular approach will ultimately satisfy the requirements. A rapid prototype effort would resolve the requirements in more detail and settle critical aspects of the technical approach before the detailed design is begun.

The second alternative is that a large, detailed specification is written by one contractor and another is hired to implement it. The specification may not be validated against realistic scenarios until the production system enters acceptance testing.

Rapid prototyping such as is proposed here is particularly efficient when done in an environment such as a LISP machine work station, and is facilitated by techniques developed by AI researchers for representing data and procedures in an easily modified manner. This combination of capabilities provides a software environment where rapid implementation and modification of a prototype are feasible, and provides a flexible inteactive graphics interface which can be programmed to simulate functionally any necessary displays and controls. The gain in flexibility is paid for by greater resource usage, by lower system capacity or slower response time, and by lack of mechanical (as opposed to functional) realism in the man-machine interface.

In summary, there are beneficial side effects from AI research in the area of software engineering. These are independent of whether or not AI techniques are used in implementing the operational software.

ARTIFICIAL INTELLIGENCE AND AIRSPACE MANAGEMENT

Paul C. Leonard, Air Transport Association of America

Artificial intelligence (AI) introduction into airspace management must not attempt to automate the controller. Just as the flight crew's functions have changed with the introduction of more sophisticated flight management systems in current and future aircraft, airspace managers will most certainly function differently than controllers do today. The flight management systems on aircraft will need to interface with the ground airspace management system with pilots and airspace managers making virtually none of the routine decisions.

Given that airport capacity problems will be addressed by more efficient use of reliever airports, improvements in the utilization of existing airports, and development of some new airports, unconstrained operation should be the rule rather than the exception. Current Federal Aviation Administration (FAA) planning documents such as the National Airspace System Plan and the Research, Engineering and Development Plan do not presently reflect this notion. Rather, they reflect a design philosophy of continued constraint to the airspace users. This