DEVELOPMENT ENVIRONMENT FOR AN ATC EXPERT SYSTEM

David A. Spencer
Massachusetts Institute of Technology
Lincoln Laboratory

This presentation concerns a development environment for an expert system to control air traffic. The expert system itself will not be discussed, as work on that system is just starting. This effort is sponsored by the Federal Aviation Administration (FAA) under a contract that started in 1983 to look into artificial intelligence (AI) applications in air traffic control (ATC). Two projects have been emphasized, an air traffic control project and a weather radar data interpretation project. This presentation will discuss the ATC project, and the talk by Steven Campbell will discuss the weather project.

Overview

The presentation by Paul Neumann of the FAA described the need for increased automation in ATC. This presentation will briefly address that issue and then discuss AI techniques, particularly expert system techniques, and why they might be applicable to ATC automation. The presentation will then examine the system which the MIT Lincoln Laboratory has created to support the development of an ATC expert system. Its function is to provide an environment in which an air traffic expert system can be tested against traffic scenarios. Finally, several snapshots of one of these traffic scenarios will be shown. This will demonstrate the development system's current capabilities and also give an indication of the traffic control situations which the expert system must eventually handle.

The Need for ATC Automation

The motivation for additional automation is that the amount of traffic is expected to increase significantly, perhaps by as much as a factor of 2, over the next 20 years. It is too expensive to increase the number of controllers proportionally. Furthermore, adding controllers implies reducing the size of the sector controlled by each controller. At some point the increased intersector coordination workload takes away much of the benefit of the reduced traffic load per controller. It is hoped, therefore, that additional automation will increase the individual controller's productivity in the sense that he will be able to safely handle larger traffic loads without any increase in perceived workload or stress.

The FAA has been implementing automated controller aids for precisely this purpose at least since the late 1960s or early 1970s. The current automated systems in fact allow controllers to handle much more traffic than they could have under earlier non-automated systems. However, the current ATC computers perform mainly clerical functions. There are some exceptions, but by and large they perform clerical operations that assist the controller by performing calculations and providing information to him. The motivation for looking at AI is the belief that there is going to be more automated decision making in ATC.

Artificial Intelligence Technology

AI technology can usefully be looked at from at least three viewpoints. One viewpoint is system or application oriented. From this viewpoint AI technology can be divided into the areas of knowledge-based or expert systems, speech and natural language, vision and image understanding, and robotics (by which is meant the mechanical aspects of AI not included in the other areas). Another viewpoint emphasizes the basic underlying techniques on which all AI systems are based. The major categories here are representation of knowledge, reasoning (which includes search, planning and problem solving, as they are all closely related), pattern recognition (although numerical pattern recognition techniques are usually deemed not to be AI), and learning.

The emphasis of the MIT Lincoln Laboratory air traffic control work is on knowledge-based expert systems. There is some interest in speech and natural language as ways of providing an interface to such an expert system, but the Lincoln Laboratory is currently not doing any research in that area. The techniques of interest are representation of knowledge, reasoning/planning/problem solving, and to some extent pattern recognition. No work involving learning techniques is currently planned.

A third viewpoint on AI technology emphasizes the software development methods used by AI researchers. These include both specialized hardware, such as Lisp machines, and powerful software tools. These represent important developments for software engineering as a whole, not just for AI.

Expert Systems

Expert systems have an expert level of problem-solving ability within a narrow domain. They incorporate knowledge of human experts in a form that, ideally, is uniform and easily modifiable. The reason for wanting these characteristics is that the expert system development process involves presenting problems to the system, having a human expert criticize the system's solutions to those problems and then quickly localizing and modifying the items of information that caused an erroneous conclusion to be drawn. This can be contrasted with a more typical software development process where the program is represented in a flowchart-like fashion, and where a particular piece of knowledge about the world may be represented diffusely throughout the flowchart. In an expert system this piece of information is represented as one rule or one fact, one piece of knowledge. If you modify that one piece of knowledge you modify its use throughout the system.

The expert system development process, therefore, is a form of rapid prototyping. You can quickly develop a program that produces results. The expert can criticize these results, and in a matter of a few years to a few days it is possible to try several problems, criticize the answers, make necessary modifications and quickly converge to a system that performs properly. Contrast this with the typical development process for any large military or air traffic software system. This is a promising approach for the initial, more experimental stage in the development of any large software system. While such a prototype may not be operationally qualified, due to slow response times, or excessive resource usage, or lack of some functions, it can be reimplemented knowing that the functional requirements and algorithms are well understood.

Some expert systems have the ability to explain their reasoning to the human expert in a format that is easily understood. This may be natural language text, or a graphical representation of the rules or facts and their relationships. The domain expert does not also have to be a software expert in order to understand the expert system's behavior. If done well, the explanation is in a form that helps locate particular rules or facts that are incorrect.

Approach To An Air Traffic Expert System

The approach being followed is to incorporate an air traffic controller's expertise into an automated system using the expert system methodology. A feasibility demonstration is now being developed and consultations have been held with a retired controller. Some standard training scenarios have been obtained as test problems. At each of the en route centers there is a training simulator called the dynamic simulator (DYSIM) and standard sets of test problems. These problems are localized to particular air traffic environments. In other words, Boston Center has Boston Center training problems, not "National Standard" training problems, although the general nature of the problems, the types of situations that arise, and the level of difficulty are constrained by guidelines from the FAA Academy.

For purposes of a feasibility demonstration, many of the real aspects of ATC are being simplified. Concentration is on the simpler environment of high altitude en route traffic control. And in fact the initial focus is on these training scenarios, not on real traffic data. No account is being taken of weather conditions, radar outages and other less common occurrences that do appear in the training scenarios. For further development, of course, these simplifying assumptions would have to be removed.

An eventual operational version of this system could potentially be used as a controller's assistant, similar in concept to the pilot's assistant discussed in another presentation. There are a lot of operational concerns with that concept, but the operational issues are not being addressed at the moment by the MIT Lincoln Laboratory. What is being focused on are the technical issues of whether automated traffic control can be performed, how it might be implemented, and how it would behave. To this end the problem of how such a system would interface to the human controller is not now being addressed. Instead the focus is on a totally autonomous system.

It would also be possible to adapt such a system for use in training controllers. This would provide a useful service in a non-safety critical area while allowing the system development process to continue by exposure to a wide variety of traffic situations.

Potential Benefits

There are several potential benefits of an expert system approach to an automated controller's assistant. One is that the system may be more understandable to the controller. One problem with some decision aids is that they are based on mathematical procedures that are not intuitive to a controller and do not correspond to the controller's problem-solving methods. When a recommendation is made, the controller does not know how to evaluate that recommendation as it is in a different "coordinated space" from his own.

A system based on expert knowledge from air traffic controllers should behave in a way understandable to controllers and should be able to give understandable explanations. A controller could then immediately see whether proposed actions were reasonable or not. He could evaluate the basis for these actions. If he was dissatisfied, he could ask the system to explain its reasoning. This would probably be done off-line, later, in a playback mode.

Controllers have their own individual control styles. It would be desirable to have a system that could be adapted to this style. The system is going to be the controller's assistant, and he would like it to adopt his style, not enforce its own. A simplified analogy would be the calculators that can be adapted to display using a preferred number representation. Numbers can be displayed in scientific notion or fixed point, and with the decimal point in European or American format (i.e., comma versus period). This does not affect the basic functionality of the calculator, but it makes it a lot easier for a person to adapt the calculator to his preferred notation. Another potential benefit of the expert system approach may be to provide this adaptability.

Another aspect of this is that it is unlikely that the expert system development process will stop, that there will be some point where the design of the air traffic automation system is completed, never to be changed. Conditions change and systems evolve. Furthermore, there is a large component of ATC problem solving that is site dependent. It is not just that there are local map information and particular minimum en route altitudes, and so forth, that must be learned when a controller learns a new sector. The problem-solving techniques themselves depend on the particular traffic environment. Controllers learn by on-the-job training in their local environment. Furthermore, they are only certified on particualr sectors, not on all sectors in the en route center. In discussions with controllers they often mention special procedures that have evolved to deal with frequently recurring traffic problems in their sector.

In our view an air traffic expert system would continue to be adapted to local environments and controller preferences by this process of posing problems and making modifications to the knowledge base. This process would have to be constrained so that certain global safety requirements could not be violated. It is not known at this time how to accomplish that, or in general how to operationally certify such a system.

An Artificial Intelligence Development System

In order to demonstrate these ideas an AI development system (Figure 1) has been put together. There are two Symbolics Lisp machines. One has a large disk and acts as the file server. They talk to each other over an Ethernet. Each system has a monochrome display and a color display. One of the systems has an attached camera so that color screen pictures can be taken.

Why were Lisp machines purchased, and not some more conventional system? One factor was the powerful software development environment they provide. Why is it so good? There is no single most important factor. Instead, it represents the successful integration of a large number of hardware and software capabilities centered on the Lisp language and its unique features.

For our applications, it is important to have good displays. Both the color and monochrome displays have high resolution (1000 percent 900 pixels or better) and they are well supported by the software. It is very easy to develop graphics, multiple window interfaces, and menus on these systems.

Finally, Lisp is the basis for most AI programming, Lisp machines are currently the most powerful AI processors, and they are becoming the standard

AI work station. Their speed and the ability to obtain AI software from other groups were other motivations for going this way.

The Expert System Development Environment

On this equipment, a development environment for an ATC expert system (Figure 2) has been created. A set of interfaces has been provided for the expert system that make available the same information as a real controller would have: position reports from radar, flight strip data (which give the route of flight and desired cruise altitude), radio messages to and from aircraft, and interphone messages to and from adjacent sector controllers for coordination purposes. In order to allow a human expert to view what is going on, the information flowing across these interfaces is displayed on the monochrome screen. This multi-window display (Figure 3) provides menus that allow the operator to control the system's operation, and also provides three windows for displaying flight strip information, for displaying controller-pilot (radio) messages, and for entering input parameters from the keyboard. Menu selection is by means of a mouse.

The aircraft position data is shown on the color screen as a traffic situation (map-like) display, along with airways, VORs, airports, and sector boundaries. Aircraft positions are represented by a dot surrounded by a 5 nmi diameter circle to provide a distance reference. A track history (previous track positions) is provided, the length being controlled from the observer's display. One setting prevents erasure of any of the history, providing a long term map of the paths flown. Associated with each aircraft symbol is a data tag, similar to those on standard ATC displays, giving the flight identity, altitude, cleared altitude, and ground speed. The display can be zoomed in on specific areas, and airway and sector maps can be turned on and off, by means of menu items on the monochrome display.

The flight strip information is mouse-sensitive, that is, flight strips act as menu items. When the flight strip for an aircraft is selected, a first level menu appears that allows the operator to issue ATC commands to the aircraft, move the aircraft's data tag on the situation display, and delete this tag. If ATC commands are selected, a second level menu appears showing the commands currently accepted by the simulated aircraft. The following commands are currently available:

Report aircraft heading, altitude, or airspeed

Fly a given heading (magnetic)

Turn left or right to a given heading
  (forcing a particular direction of turn)

Turn left or right by some number of degrees

Resume own navigation
  (puts the aircraft back on its flight plan
  after a period of vectoring)

Climb/descend and maintain a given altitude

Increase/reduce speed to a given value
  (indicated airspeed in knots)

Increase/reduce speed by a given amount.

Figure 1. Artificial intelligence development system.

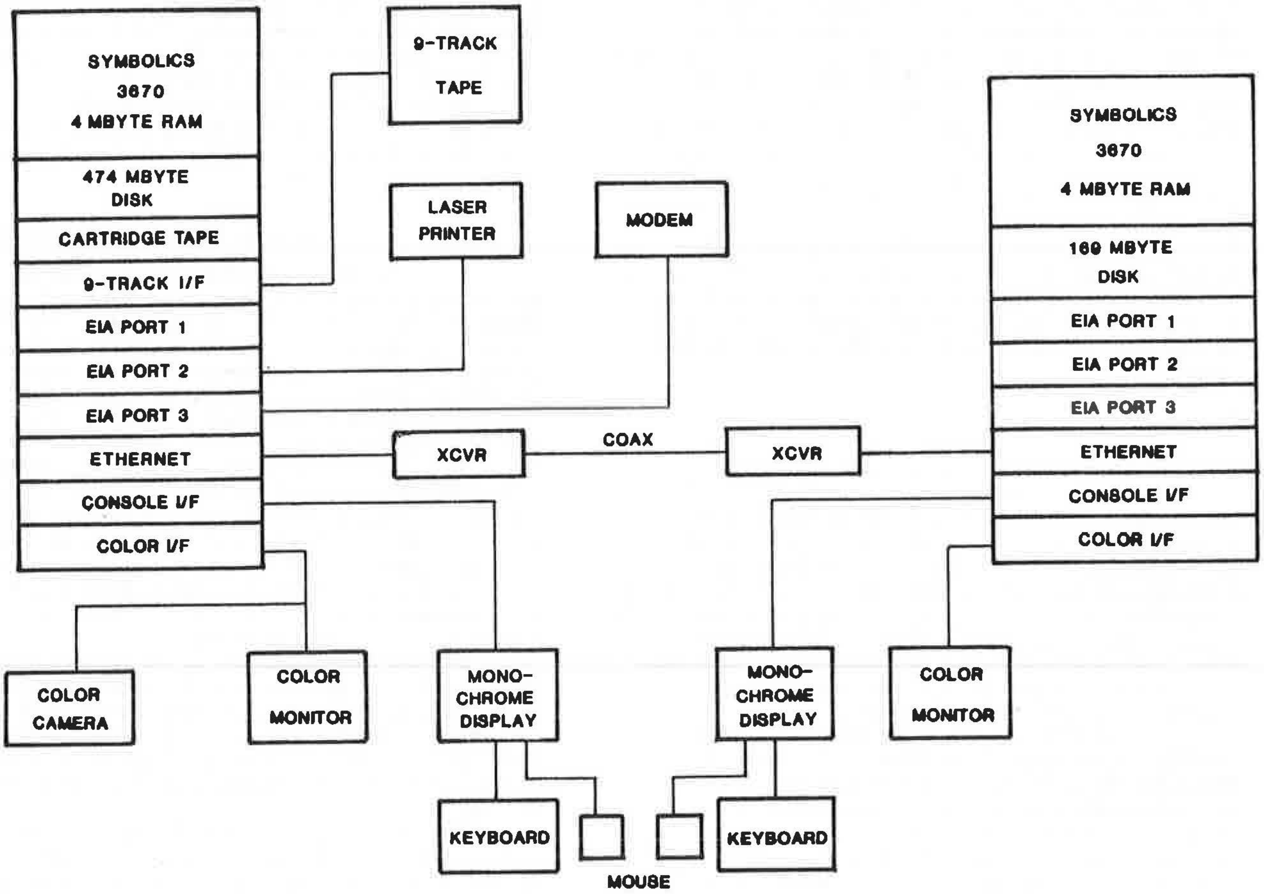
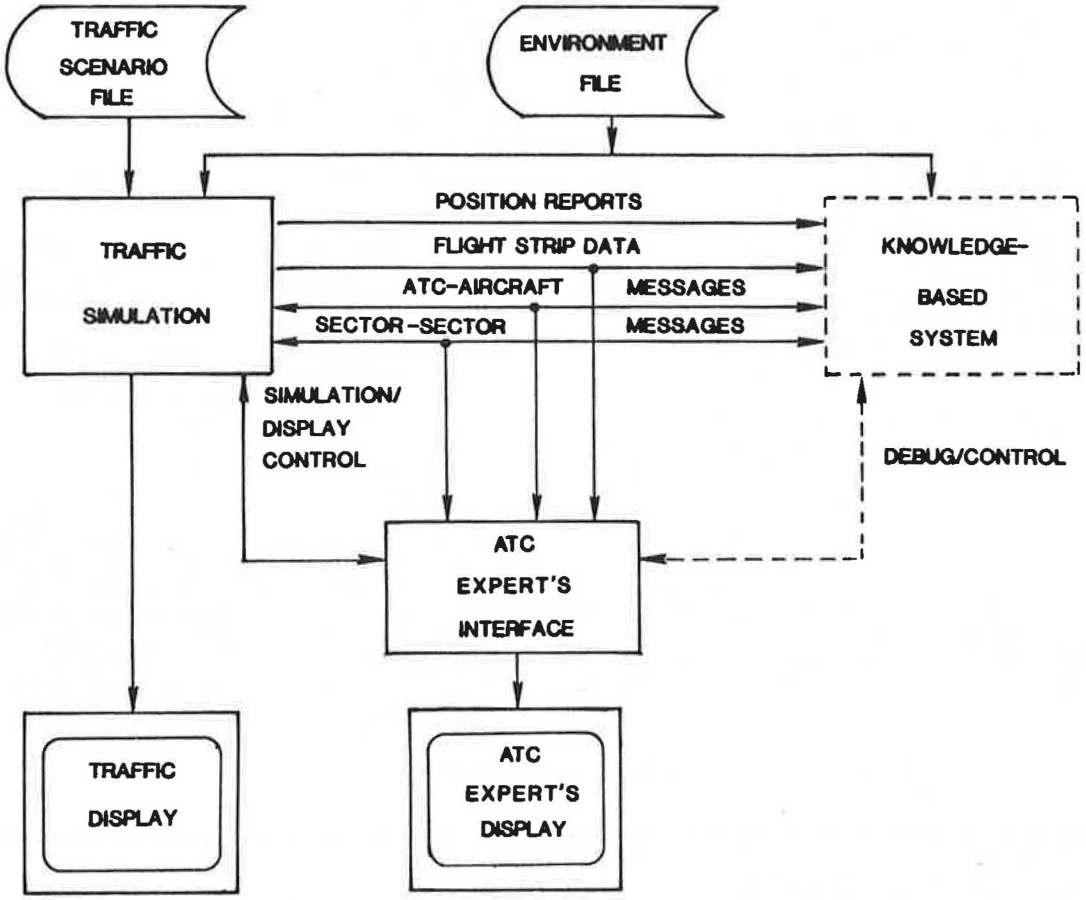


Figure 2. An ATC expert system development environment.



A major component of this environment is a traffic simulation. It is driven by a traffic scenario file which is basically the aircraft flight plan information. These data are derived from the DYSIM problems. Data have been obtained for the easiest ten (out of 18 total) training scenarios from the Boston Center. There is also an environment file which specified the map information: VORs, airports, airways and so forth.

This traffic simulation was developed using some ideas and code from a Lisp-based simulation developed at the MIT Flight Transportation Laboratory by Professor Antonio Elias and Dr. John Pararas. Work was started in September 1984 and a working version similar to the current one was available in April 1985. During the period two people worked about half time on design, coding and debug of the software, one person on the traffic situation display and the other on the simulator and the operator's display. The entire operator's interface, including the multi-pane operator's display, was implemented in two weeks in essentially the current form, using system utilities for creating multi-pane windows and menus.

Purpose Of The Development Environment

The purpose of the development environment is two-fold. First, with the knowledge-based system turned off it is possible for a person, such as a real controller, to control the simulated traffic. The necessary man-machine interface is provided by a combination of menus and text input. The controller can be observed doing this, and his behavior incorporated into rules in the knowledge-based system. Then the controller is able to criticize the performance of those rules when the knowledge-based system controls the simulated traffic. Thus, facilities are provided to first find out what the necessary ATC knowledge is and then to demonstrate that it has been correctly implemented.

A Traffic Scenario

[At this point of the presentation the first and easiest of the DYSIM scenarios was presented. The technical difficulty of reproducing color images and the loss of resolution when screened images are reduced to publication size have made it necessary to delete this portion of the presentation from the paper.]

Comments On The Scenario

In this first DYSIM training scenario the simulations that arise are of a few simple types:

1) Arriving aircraft for airports underneath the sector or in nearby sectors must be allowed to descend to appropriate altitudes.

2) Departing aircraft must be allowed to climb to their cruise altitudes, as specified in their flight plans.

3) There are a number of cases where flight paths cross. These can lead to conflicts

depending on the altitude behavior of the aircraft. However, there are no built-in conflicts where two aircraft in the scenario are co-altitude on conflicting paths. Thus, this acts more as a constraint on solutions to 1) and 2).

4) There is one instance of two arriving aircraft for Boston converging at the Albany VOR and staying together for their remaining time in the sector. Again, there is no direct conflict initially, but this complicates solutions to 1) as the aircraft will be in conflict if they are both allowed to descend to the same altitude as required by the arrival procedures. One solution involves sequencing the two aircraft.

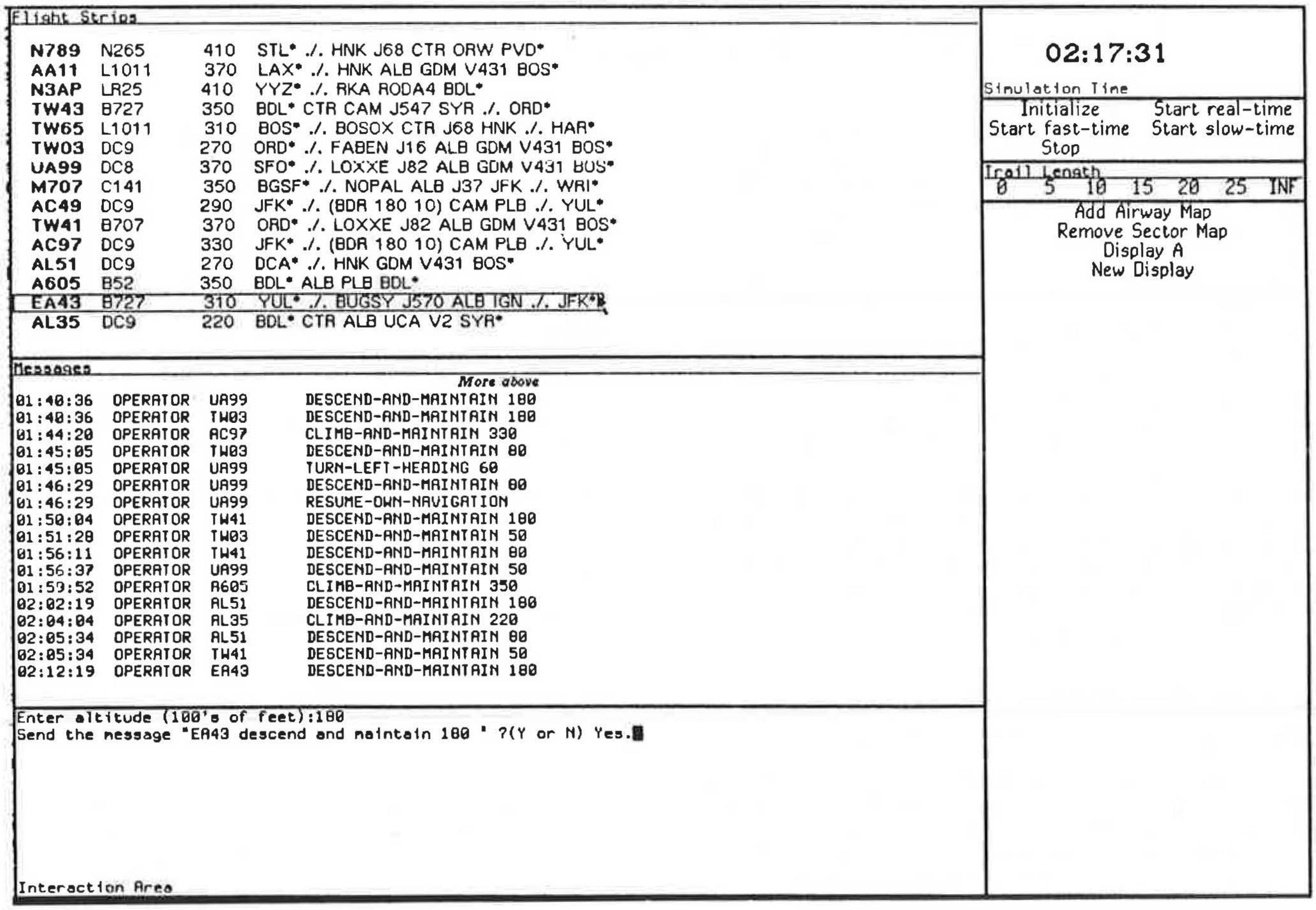
5) There is at least one opportunity to expedite flow by giving a direct routing, possibly involving a radar vector.

It should also be noted that this is not a complete representation of this problem as it is used at the Boston Center. In addition to the basic aircraft flight paths, which are automatically simulated by both DYSIM and this environment, there is also a set of manual inputs to the DYSIM environment. These are indicated in notes that are given to the instructors running the DYSIM. In the DYSIM the instructors play the roles of simulated pilots and adjacent sector controllers, manually handling verbal communication and some decision making that is difficult or impossible to simulate automatically. This also allows them to vary the scenario from run to run. The notes for the first training scenario indicate the following complications are to arise:

1. Two aircraft request radar vectors to fixes.
2. One aircraft requests to descend below positive control airspace, to cancel IFR, and requests traffic advisories.
3. Radar outages occur at different times in two adjacent sectors requiring re-identification of aircraft entering from those sectors.

In addition, there are the normal requirements for coordination with adjacent sectors. This can be straightforward when aircraft crossing the boundary are in radar contact and on their flight plan route. It is more complex if the adjacent sector has had a radar failure, or if an aircraft has been vectored off its route or is otherwise not conforming to the standard procedures. The development environment does not currently support any of these activities, although a rudimentary form of handoff will be added soon.

Figure 3. Monochrome display of an AI traffic scenario system showing menus, flight strip information and controller-pilot messages.



```
N789  N265    410  STL* ./. HNK J68 CTR ORW PVD*
AA11  L1011   370  LAX* ./. HNK ALB GDM V431 BOS*
N3AP  LR25    410  YYZ* ./. RKA RODA4 BDL*
TW43  B727    350  BDL* CTR CAM J547 SYR ./. ORD*
TW65  L1011   310  BOS* ./. BOSOX CTR J68 HNK ./. HAR*
TW03  DC9     270  ORD* ./. FABEN J16 ALB GDM V431 BOS*
UA99  DC8     370  SFO* ./. LOXXE J82 ALB GDM V431 BOS*
M707  C141    350  BGSF* ./. NOPAL ALB J37 JFK ./. WRI*
AC49  DC9     290  JFK* ./. (BDR 180 10) CAM PLB ./. YUL*
TW41  B707    370  ORD* ./. LOXXE J82 ALB GDM V431 BOS*
AC97  DC9     330  JFK* ./. (BDR 180 10) CAM PLB ./. YUL*
AL51  DC9     270  DCA* ./. HNK GDM V431 BOS*
A605  B52     350  BDL* ALB PLB BDL*
EA43  B727    310  YUL* ./. BUGSY J570 ALB IGN ./. JFK*
AL35  DC9     220  BDL* CTR ALB UCA V2 SYR*
```

```
01:40:36  OPERATOR  UA99    DESCEND-AND-MAINTAIN 100
01:40:36  OPERATOR  TW03    DESCEND-AND-MAINTAIN 100
01:44:20  OPERATOR  AC97    CLIMB-AND-MAINTAIN 330
01:45:05  OPERATOR  TW03    DESCEND-AND-MAINTAIN 80
01:45:05  OPERATOR  UA99    TURN-LEFT-HEADING 60
01:46:29  OPERATOR  UA99    DESCEND-AND-MAINTAIN 80
01:46:29  OPERATOR  UA99    RESUME-OWN-NAVIGATION
01:50:04  OPERATOR  TW41    DESCEND-AND-MAINTAIN 100
01:51:20  OPERATOR  TW03    DESCEND-AND-MAINTAIN 50
01:56:11  OPERATOR  TW41    DESCEND-AND-MAINTAIN 80
01:56:37  OPERATOR  UA99    DESCEND-AND-MAINTAIN 50
01:59:52  OPERATOR  A605    CLIMB-AND-MAINTAIN 350
02:02:19  OPERATOR  AL51    DESCEND-AND-MAINTAIN 100
02:04:04  OPERATOR  AL35    CLIMB-AND-MAINTAIN 220
02:05:34  OPERATOR  AL51    DESCEND-AND-MAINTAIN 80
02:05:34  OPERATOR  TW41    DESCEND-AND-MAINTAIN 50
02:12:19  OPERATOR  EA43    DESCEND-AND-MAINTAIN 100
```

```
Enter altitude (100's of feet):100
Send the message "EA43 descend and maintain 100 " ?(Y or N) Yes.
```

Conclusion

In conclusion, the author emphasizes the following points:

- The expert system development methodology requires an extensive set of sample problems. It encourages system developers to test their ideas against these problems early in the development life cycle.
- In the ATC setting, this requires a simulated traffic environment, possibly augmented by real-time human inputs in areas difficult to simulate.
- The current simulation is adequate for initial expert system development. Some additions will be needed to demonstrate all aspects of the first training problem.
- Continued development of an air traffic controller expert system will require continued development of the simulation to add to its functionality and to improve the fidelity with which it simulates those functions.

Discussion

George E. Swetnam, Mitre Corporation Does your simulation do anything to help present the conflicts to the controller? What you have done essentially is to replicate the information that is available to him presently on the traffic display. Has any thought been given toward showing him the conflicts in some other form that will help him grasp what the expert system is doing?

David A. Spencer Not particularly. The nice thing about this is that it is a simulator, and you can stop it. If somebody wants to analyze the situation we can stop it at some particular point and look at it, for as long as we want. We do not have to instantly present complex graphics so that a real-time decision can be made.

FAA Comment If I can add to that. Right now we are trying to see if we can get the system to work, but you are exactly right. There are major human factors issues, not only in presenting the conflict but presenting the resolution. These kinds of issues are very crudely understood.

David A. Spencer We are basically working with the problem-solving aspect of air traffic control, defining the problems, defining the solutions, and discovering how to generate reasonable solutions. How to present these to the controller and how he is to use them are complex questions that we are not equipped to handle at this time.

Robert H. Brown, NASA Johnson Space Center
Two questions. How many rules do you have, and are you bothered with garbage collection?

David A. Spencer The second one is easier. No, because we just use a large virtual memory.

Robert H. Brown, NASA Johnson Space Center
You are using the Symbolics machine?

David A. Spencer We are using the Symbolics 3670.

Robert H. Brown, NASA Johnson Space Center
You can run it long enough?

David A. Spencer You can run it long enough, yes. It can run through an entire simulation without garbage collecting.

Robert H. Brown, NASA Johnson Space Center
How long is that?

David A. Spencer An hour of real-time, 15 minutes if you run it in fast time mode.

Robert H. Brown, NASA Johnson Space Center
How many rules?

David A. Spencer We have not really gotten into the expert system. The expert system portion was shown dotted for a reason. At the time the display was made it did not exist. At this point in time it is a shell. The interfaces have been put in. We are now implementing some rules for finding path intersections, lines and line intersection points, that sort of thing. It is at that level at this point. We do not have rules that actually implement air traffic control.

Robert H. Brown, NASA Johnson Space Center
Do you have an estimate of how many rules?

David A. Spencer There is some feel for that. Several people, some here, in fact, have demonstrated that a relatively small number of rules can handle surprisingly complex cases, on the order of 20 to 30 rules.

Curtis A. Shively, Mitre Corporation We have 100 rules.

EXPERT SYSTEM FOR DOPPLER WEATHER RADAR INTERPRETATION

Steven D. Campbell
Massachusetts Institute of Technology
Lincoln Laboratory

Overview

This presentation concerns an expert system being developed for Doppler weather radar interpretation. The objective is to use artificial intelligence (AI) techniques to interpret weather radar displays to recognize wind shear hazards. The reason for developing an automatic recognition capability is that terminal Doppler weather radars will be placed at many airports to detect these hazards and it is not cost effective to put expert radar meteorologists at each of these locations. Thus, an automatic recognition capability is desired which can place a warning on the air traffic controller's screen so that these hazards can be avoided.

The approach being taken is to capture the expertise of a radar meteorologist in recognizing these hazards. Radar meteorologists exist who are very good at picking out microbursts from Doppler radar displays. The goal of this project is to understand what their expertise is, and to try to build it into a computer program so that it can be replicated at many sites.

This presentation will discuss expert systems briefly, summarize the characteristics of wind shear hazards and Doppler radar, outline the design