AEROSPACE EXPERT SYSTEMS

Since the Wright brothers first took flight, the progress of mankind in the domain of flight has been nothing short of spectacular. The progress of technology, faster aircraft, instrument flight, and increased air traffic resulted in establishing a governmental infrastructure to control air traffic. During the late 1920s air traffic control was established as a profession. Rules, procedures, standards, and facilities were designed to provide for safe and orderly traffic patterns and required separation between the aircraft (1). Stakeholders of the airspace system include airline operators, airport authorities, weather specialists, and air traffic controllers (ground, tower, en-route, flow), as well as the passengers. In addition to civilian users, including the airlines and general aviation, the system must accommodate military and space flight activities.

The demand for air transport has steadily increased since introduction of jet aircraft during the late 1950s. The safety and effectiveness of the national airspace system (NAS) depends on the performance of the air traffic management (ATM) personnel—the employees of the Federal Aviation Administration (FAA). The increasing complexity of the system and proliferation of computing equipment have generated an urgent need to explore the possibility of supporting the human component of the system with tools and techniques based on the concepts and methods of artificial intelligence (AI) (2).

Intelligence is defined as "the ability to learn or understand from experience, ability to acquire and retain knowledge" (3). Applied AI is about programming computers to perform tasks previously assumed to require human intelligence. The usefulness of AI is the measure of its success. The key issue is realization that knowledge must be represented explicitly in terms of its nonalgorithmic contents. The computer program acts upon it by deduction and reasoning applying various search algorithms. There is a need to create software products representing an artificial expertise—a container for limited-domain knowledge. This development is particularly important in the case when the pool of available experts is limited (or about to be limited in the future).

As has been noted in other works, "an AI system must be capable of doing three things: (a) store knowledge; (b) apply the knowledge stored to solve problems; and (c) acquire new knowledge through experience. An AI system has three key components: representation, reasoning, and learning" (3a).

ARTIFICIAL INTELLIGENCE—CONCEPTS AND APPROACHES

The difficulties associated with learning have led people to use other methods for augmenting knowledge bases. Expert systems (ES) are such an example, where the compiled knowledge and experience of a human expert are used in lieu of having the system develop its own experience, duplicating that of the expert. This is not to say that learning cannot be part of an ES.

The methods and techniques of AI are well suited for applications not amenable to standard, procedural, problem-solving techniques. Examples of such applications are where the available information is uncertain, sometimes erroneous, and often inconsistent. In such a case, using quantitative algorithmic calculations may not lead to the solution, whereas use of plausible and logical reasoning may. The approach taken by AI leads generally to a nonoptimal, but acceptable, solutions reached by using rules of thumb and logical inferencing mechanisms.

For such an approach, the system is represented by a factual description in the form of chunks of meaningful data (knowledge) related to the system state and by the relationships among those data. An external, domain-independent inferencing mechanism makes it possible to draw new conclusions from existing knowledge resulting in changes and updates of the knowledge base. The AI discipline concerned with these issues is called knowledge representation. There are various paradigms of how to represent human knowledge: predicate calculus, production rules, frames and scripts, and semantic networks. The selected representation scheme must express all necessary information, support efficient execution of the resulting computer code, and provide a natural scheme for the user. AI is concerned with qualitative, rather than quantitative, problem solving. Thus the selected knowledge representation and the used tools must be able to (a) handle qualitative knowledge, (b) allow new knowledge to be created from a set of facts, (c) allow for representation applicable to not only a specific situation but also to general principles, and (d) capture complex semantic meaning and allow for meta-level reasoning (reasoning about the knowledge itself, not just the domain).

A distributed-intelligence system (DIS) is the concept of a system operated by a machine and managed by a human. The human operator is involved in planning, making decisions, and performing high-level functions, whereas the machine portion of the system executes most of the system's regular operational functions, collects and stores data, and handles routine decision situations with a limited number of options (4). Such an approach requires further research in the area of man–machine interface and physiopsychological aspects related to the stress and anxiety factors.

Decision support systems (DSS) are computer systems that advise human operators or are automated systems that make decisions within a well-defined area. The systems are used where similar decision processes are repeated, but where the information to decide upon may differ. Some DSS are known as expert systems (ES): they imitate human expert behavior. Decision procedures of an expert are analyzed and transformed into rules and subsequently implemented into the system. The ES is a computer program providing solutions to problems normally requiring a human expert with an appropriate domain knowledge and experience. The experts are employed to solve problems requiring planning or decision making. They frequently use rules of thumb—heuristics based on experience, analogies, and intuitive rationale to explain the behavior associated with their area of expertise. Development of ES requires identification of the thought process describing how the human expert solves a specific problem. There are three steps in this identification (1) knowledge representation, (2) knowledge acquisition, and (3) knowledge processing (5).

To imitate a human expert, a successful ES must reach the solution based on the predefined body of knowledge, despite incomplete or uncertain data; explain and justify how the solution was reached; and communicate with the user and/or environment, thus acquiring new expertise. Some other advanced properties include the ability of natural language communication to support an effective user interface and easy knowledge acquisition. The ability to determine relevance, by either giving a referral or degrading gracefully at the expertise boundary, is another important human characteristic, as are common sense reasoning and breaking rules. The issue of effective acquisition of new knowledge and reorganization of expert knowledge base is closely related to machine learning—another important AI discipline closely associated with cognitive science.

Besides the aerospace industry, other successful implementations of ES in such areas as medical diagnosis, manufacturing, and military command and control have shown potential for further expansion. However, the analysis of such systems is, in most cases, time-neutral. That is, most expert systems are not able to look into the future and predict future system behavior. Reasoning on a given problem is applied in the current time instant. Only an integrated DSS, combining the potential of simulation and the reasoning ability of ES, can be used in time-dependent applications (6).

Various knowledge acquisition schemes have been developed to extract expert knowledge. Interview techniques and observation of humans performing expert tasks have been used to elicit knowledge. If the problem domain is well defined, the approach leads to a rule-based system, where the expertise is represented as a set of if–then rules. An inferencing mechanism searches the rule base, drawing inferences and asserting/retracting the facts about the problem at hand. For the expert operating a complex system, it is easier to explain general experiences rather than identify specific rules. This observation led to emergence of case-based reasoning, where various operational cases are acquired and modeled into the system knowledge base. Using the process of induction the ES infers new rules based on the current facts and the experiences from previous similar cases. The system then produces advice and modifies its knowledge base (7).

Uncertainty is an inseparable part of any decision process. It may be caused by insufficient understanding of the problem area, by missing or uncertain measurements and observations, or by nondeterministic causal relations. The emergence of Bayesian networks allows researchers to model the problem area with its built-in uncertainties. This approach can result in a directed graph, where the vertices represent the variables of the problem area, and a directed edge between two vertices means the state of one variable influences the state of the other. The magnitude of this influence is represented by a conditional probability (8). Another approach to uncertainty is the application of fuzzy set theory, which allows researchers to assign a range of quantitative variables to create qualitative entities subsequently handled by the ES (9).

NATIONAL AIRSPACE SYSTEM

The FAA has been modernizing the NAS in a massive project initiated in the early 1980s. Several papers (see reading list) related to the application of AI methods and techniques to air traffic control (ATC) in the NAS of the twenty-first century describe attempts to formalize the ATC knowledge base and build systems assisting controller operations and training. Most of the systems use a simulation component designed to predict the future state of the system.

The FAA provides ATC and the traffic flow management (TFM) functions, collectively referred to as ATM, designed to maximize air traffic throughput and minimize delays while maintaining a safe operating environment. The overall ATM system is designed to give equitable access for all NAS users, while ensuring safe separation of aircraft from each other and from terrain/physical obstacles and restricted airspace. Physical and operational constraints of the NAS include availability of airport runways, severe weather conditions, equipment outage, and heavy traffic demands. These constraints limit the options to accommodate user preferences for flight times, routes, and altitudes.

TFM is the ATM function that focuses on managing NAS resources, mitigating contention for scarce capacity, and disseminating information about the anticipated restrictions and demand predictions. TFM is intended to allocate capacity to NAS users in an equitable fashion and allow them to maintain operational control.

The ATC functions of the ATM include real-time separation and sequencing of aircraft en-route, during arrival and departure at the terminal, and ground control on the airport surface. En-route controllers in 20 centers of the continental United States provide services for aircraft between the departure and arrival phases of flight at air route traffic control centers. Services include separating aircraft, monitoring traffic flow, implementing traffic management initiatives provided by TFM, issuing traffic and weather advisories, coordinating special use airspace, and providing emergency assistance. The terminal controllers in more than 50 larger metropolitan airport areas provide ATC services to aircraft traveling between an airport ground control and the en-route environment. The terminal area includes one or more airports controlled out of a terminal radar approach-control facility. The services include aircraft separation, sequencing, traffic advisories, and alerts, significant weather advisories, and radar vectoring for arriving, departing, and through traffic. Airport ground control is responsible for all vehicles operating on taxiways, aprons, and gate areas.

Advanced ATM functionality is being developed to support real-time information exchange between ATC controllers and airlines, facilitating collaborative decision making among airspace users. Advanced ATC decision support tools are also being developed to increase safety, efficiency, and flexibility as the NAS system evolves toward free flight. These advanced functions must be integrated into the flow, en-route, terminal, and tower/ground facilities as part of a coherent national system in order to deliver their full benefits (see Web resources).

AIR TRAFFIC FLOW MANAGEMENT OPERATIONS

The Air Traffic Control System Command Center (ATCSCC) is located in Herndon, Virginia. The current role of the center is to manage the allocation of NAS resources and limit airborne delays. These objectives are accomplished by implementing TFM initiatives: ground delay program (GDP), ground stop program (GSP), miles/minutes-in-trail restriction (MITR), and severe weather avoidance program (SWAP). The center is staffed by experienced specialists with an extensive knowledge of ATC procedures and familiar with the impact of weather conditions and possible TFM initiatives on NAS performance (10).

There are a wide variety of computer tools based on available aircraft data to support the specialist operations. Data can be displayed in both character and graphic formats, showing, for instance, all aircraft scheduled in a specific sector within a specific timeframe, or all aircraft scheduled to arrive at a specific airport. The hourly arrival demand for an individual airport may be displayed and printed. Weather conditions are displayed graphically, including areas of limited ceiling and visibility, precipitation, expected thunderstorms, and jet streams. There is easy access to alphanumeric local ground weather reports, altitude profiles, and briefings from radar observation.

The traffic management specialist responds to the weather situation and to requests from the major airports in the cluster. In cases when an airport acceptance rate is anticipated to decline (deteriorating weather conditions, airport configuration change) the flow controller may consider implementation of the GDP for that airport. The program can be implemented for any combination of the en-route centers, from adjacent to the entire system. The scope of the program, in terms of duration and affected areas, is based on the current situation and determined as the result of the controller's knowledge and experience. The GPD software recomputes departure times, and estimates predicted delays. When the computation predicts acceptable delays in the system, the specialist sends the new schedule to the centers and the airlines for implementation. In the case when an airport is unable to operate or experiences severely reduced capacity with already long delays and surplus traffic, the specialist may order the GSP for the flights destined to the affected airport. Both GDP and GSP affect only the aircraft scheduled for later departure. Any action is coordinated with all interested parties before implementation. The shift supervisor has the final authority on whether or not the proposed plans are implemented.

A regional traffic management unit may request MITR in cases of reduced acceptance rate of the arrival sector caused by weather, traffic volume, or staffing problems. The situation is analyzed and coordinated by the area cluster specialist, and the outcome is conveyed to the affected centers. The role of flow control is limited to a mediation between two adjacent centers.

Severe weather conditions en-route may force a center to request more forceful measures as a significant rerouting of traffic. A separate cluster of flow personnel manages the implementation of SWAP rerouting. The position is equipped with an additional workstation with a database of airport configurations under different weather conditions and the preferential routes among all major airports. The main role is to provide coordination for the new routing.

There is a significant amount of domain knowledge involved in TFM activities. For example, some of the airports require implementation of a GDP/GSP for the entire system, whereas others may restrict the program to the adjacent centers only, based on incoming traffic patterns and past system responses. Rerouting must consider the capacity of the affected sector to handle the additional traffic. Decisions result from the anticipated situation and knowledge of the system operation and its behavior. Therefore, most decisions are based on the flow specialist's personal experience.

Expert Systems for Air Traffic Flow Management

The FAA Operations Research Service developed SMARTFLO—a knowledge based system. This prototype computer program provides the TFM specialists with a user-friendly support tool capable of suggesting a set of required traffic management initiatives for a specified time horizon (11). The system reasoning capability was based on the available information (inquiring for more information if necessary), and the base for decision and the line of reasoning and was presented to the user. The system includes encyclopedic knowledge about NAS operations.

The basic activities in ES development are knowledge representation, knowledge acquisition, and feasibility prototype. The TFM system is described in terms of its concepts and activities. Among the concepts defined are pacing airport, affected area, weather, volume/demand, TFM initiatives, preferred route, rerouting, and arrival distribution to the airport. TFM activities include acquiring information about the system and weather and subsequently processing the information generating the required TFM initiatives. A GDP initiative is proposed, using heuristics based on the flow controller's knowledge. The basis is the same: The action is considered when anticipated demand exceeds the expected capacity. However, the numerical values used (e.g., rate, time interval), the area involved, or freedom to use other options (GSP, MITR) are left to the specialist.

The initial knowledge-acquisition phase included observation of daily operations and interviews with ATCSCC personnel. The analysis of daily reports including specialist logs, weather reports, listing of traffic management initiatives for the day, and estimates of system performance (delays) provided an additional source of knowledge. As the result of these activities, a knowledge representation scheme (facts and rules) was selected and a prototype was implemented. CLIPS, an ES tool developed by NASA, was used as the inferencing mechanism. Subsequently, the Sun workstation-based graphic interface was added to complete the initial implementation. The advanced prototype of the system, implemented on a Hewlett Packard HP9000 workstation, used TAE+ graphic interface tool with active agents and the object-oriented paradigm. The objects contain information about their relationships with other objects, identification, source information, definition and description, lists of relations, rules, and constraints defining their behavior. The classes of object in SMARTFLO are jobs, agents, resources, operations, models, events, systems, devices, sites, and environment.

The SMARTFLO advanced prototype graphic windows allow the user to select the airport/center, enter data, query for system information, and watch suggested TFM initiatives being generated. A user-friendly consultation allows the user to ask "What should we do?", and "Why?". In addition to generating the suggested TFM initiatives and playing what-if scenarios, the system includes an electronic encyclopedia with extensive information about ATCSCC operations.

INTELLIGENT TRAINING

Computer-aided instruction (CAI) is a common use of computers in education and training. CAI tools incorporate well-prepared course materials and lessons plans into routines optimized for each student. However, conventional CAI tools are limited to either electronic page-turners or drill-and-practice monitors, severely limiting the overall effectiveness of the system in a situation where declarative knowledge is sought.

The incorporation of AI techniques into CAI spawned the creation of intelligent tutoring systems (ITS) capable of modeling the student learning process, drawing conclusions from student problem solving behavior, and modifying the sequence in which material is presented to the student (12). ITS is intended to help individual students identify their specific weaknesses and rectify them effectively and to be sensitive to the student's preferred style of learning. The objective of some researchers is to produce entirely autonomous ITS based on pedagogical expertise and the principles in the domain knowledge.

The major blocks of a modern simulation-based ITS are (a) simulator, (b) domain expert, (c) student model, (d) evaluator, (e) scenario generator, (f) training manager, and (g) user interface. The simulator represents the real-life system for which the student is being trained. The domain expert contains the body of knowledge that should be presented and taught to the student. It is also used for evaluation of student performance and the overall learning progress. To achieve these objectives, most systems generate and store all feasible solutions to the problems in the same context as the student, so that their respective answers can be compared. The student model contains knowledge about the student's understanding of the material. This knowledge is extremely important in the decision making process affecting the choice of subsequent tutoring strategies. The evaluation module is used to evaluate the student performance based on the situation assessment derived from the simulation status. The scenario generator is used to generate realistic training scenarios appropriate for the student. The core of the system is the training manager, containing the knowledge about teaching methods. The training manager, based on the current evaluation, selects the next scenario component from the scenario generator monitoring the student's performance. The module uses a decision making process based on teacher experience and assessment of past training sessions. Finally, the success of an ITS depends significantly on user-interface quality. All of the components are coordinated by a communication entity referred to as a *blackboard*. Any ITS module can place information on the blackboard making it available to all other modules.

An ITS is often in the form of a computer-based problem solving tutor, a coach, a laboratory instructor, or a consultant. For the development of an ITS for aerospace (ATC specialists, pilots, astronauts, and airline dispatchers), the most suitable tutoring strategy seems to be a combination of coaching and guided-discovery learning. The student is in full control of the activity for which the tutoring is provided—a simulated version of a real system. Simulation is used because it provides an attractive motivational context for discovery learning. The coaching task of the ITS is to foster the learning inherent in the activity itself by emphasizing existing learning opportunities and by transforming failures into learning experiences.

The potential of coaching is evident when ITS is embedded in an interactive, real-time simulation. It must be noted that many simulation-based ITS are strong in their simulation capabilities, but rather weak in delivering an effective tutoring strategy.

Air Traffic Control Intelligent Training System

The ATC-ITS prototype system (13) was designed to support ATC training by providing the student with an interactive, simulated ATC domain by creating effective training scenarios with exercises matching the student performance level, by coaching the student through the scenarios, and by evaluating student performance.

The first major task is to build a knowledge base representing the domain expertise of an en-route air traffic controller. The principal methods used are observations, informal and formal interviews, operational task analysis, and simulations. A key element is the advice provided by air traffic controllers with both operational experience and en-route training expertise (14). One aspect of ITS development focuses on the expert system and its interface with a realistic simulation. The other aspect deals with the training methods, controller characteristics, student capabilities and skills, development of training strategies, and assessment techniques for student performance evaluation.

The ATC-ITS provides training for en-route ATC students. It is assumed that the trainee has control over aircraft within a specified training sector. The trainee is responsible for hands-off procedures including receiving incoming and transferring outgoing aircraft. The transfers and clearances must conform to the sector standard operating procedures. The route through the sector must closely follow the flight plan. The students must conform to ATC communication standards and phraseology, secure an efficient and orderly movement of aircraft through the sector, and provide appropriate responses to pilot requests. The trainee is primarily responsible for ensuring proper aircraft separation within the controlled sector. All the activities are monitored and evaluated by the ITS.

The simulation, implemented in object-oriented `MODSIM` language, provides a realistic, real-time, en-route air traffic sector simulation displaying aircraft targets with data blocks, flight paths, and flight strips, as well as controller-to-pilot and controller-to-controller communications. ATC-ITS graphics represents an en-route radar scope. It displays sector boundaries, airways, fixes, restricted areas, targets, data blocks in different forms (limited, full, separation violation, handoff initiated, handoff accepted, radio failure, emergency, etc.). The graphics display responds to mouse operations such as handoff initiation, handoff acceptance, data block move, and altitude or route change for selected aircraft. The communication window simulates the radio communication between the controller and simulated pilots.

The air traffic expert controller is an implementation of a rule-based ATC expert model. It can provide expert knowledge relative to ATC decision making. The knowledge base consists of facts and rules. The situation facts are assessed from the simulation module and the user input. The facts represent aircraft status, sector situation, controller workload, pilot and adjacent controller requests, emergency events, and so on. The action facts reflect the required action to be implemented by the controller as a direct response to the given situation. The rules reflect the controller experience in tandem with ATC operational procedures and regulations. The system uses a forward-chaining mechanism with its production rules where the situation facts are the rule premises and the action facts are the rule conclusion. The CLIPS inferencing mechanism uses the Rete algorithm (15) to search the rule base. Assertion or retraction of the facts to or from the knowledge base constitutes the principal knowledge control handling mechanism (16). The ATC rules represented in the knowledge base are used by CLIPS inferencing to arrive at the "best" decision. Firing the rules leaves a trace, which provides the user an explanation about the line of reasoning leading to the specific recommended action. The prototype system was developed on a Sun workstation with a Sunview windowing system.

Expert Systems for Pilots

ES in aerospace are often designed to aid the operator of the system, including piloting vehicles. Typically the use of an ES in an operational environment such as piloting can be oriented to supporting the pilot or assessing the pilot. Often, ES that are capable of assessment incorporate some form of feedback or critique to the pilot, further enhancing their usefulness.

Pilots perform three fundamental activities while flying. The first, and primary task, is control. This relates directly to the pilot's ability to manage the attitude of the aircraft about the three axes of the aircraft. ES that support control include stability and inertial reference systems. The second activity the pilot performs is navigation. This is control of the movement of the aircraft from point A to point B. Pilots are supported by ES such as autopilots and navigation computers. The third activity of the pilot is the need to communicate information relating to the aircraft. ES can provide data link and information fusion based on the pilot's current and future activities. Knowledge acquisition for these types of systems can be very difficult, costly, and time consuming. Techniques to improve the process include multiple model integration, indexing, and multiple input summarization. Examples of systems supporting the pilot are Pilot Associate (17) and Hazard Monitor (18). Both systems are based on real-time data acquisition from the aircraft avionics. The system processes the data to create knowledge chunks, used as knowledge base by ES.

Control

Pilots generally understand that aircraft are dynamically unstable. This is true, even when tendencies toward instability are rendered invisible by avionics designed to override those innate problems. As a result, the human ability to control aircraft attitude is often suspected in control-related incidents. Computing systems supported by expert knowledge about the engineering and dynamics of the vehicle are often employed in these situations. While these may not be purely and exclusively expert systems, they provide an excellent, if somewhat oversimplified, example. Flight stability ES use a theoretical approach, coupled with a dynamic input, to create a response algorithm. The use of an optimal control model provides the system with a baseline of engineered performance. The ES designer would typically use a rule-based approach to implement this model. Dynamic input is provided by both the pilot and the flight computers and by inertial reference computers (if available), allowing a control model to be exercised. Often, a sophisticated formula or Bayesian loop is used to control the limits of the autopilot system. Engineers are concerned with the pilot's induction of out-of-phase or undampened input to the system, and thus, in some aircraft, such as those produced by Airbus Industrie, the autopilot system will actually retain and enforce control input limits made by the pilot.

Navigation

Navigating an aircraft is sometimes simple and sometimes complex. As with control, a number of ES-type technologies exist to ensure that the task of navigating the aircraft is safe and effective. When coupled with the control computers, the autopilot can take on the task of determining position and guiding the aircraft to its intended destination by returning commands to the autopilot system. This would imply that aircraft are comprised of complex ES networks, which is the case in many advanced turbojet and most military aircraft. Some corporate and general aviation aircraft have similar although less expert systems.

When a pilot of any aircraft is operating in conditions that are favorable for seeing and navigating the aircraft by looking out the window, the use of such systems is less critical. In these circumstances, ES are often employed to monitor and assess pilot navigation performance rather than actively navigate the aircraft. In conditions where the pilot relies heavily on ES to navigate the aircraft, such systems are designed to mimic the decision process and control input that would be provided by the pilot. It is common to have an ES that uses both inductive and deductive structures to provide the input to the control algorithms. Most often the system design is reduced to having a known position and a set of control rules derived for the primary task of controlling (discussed previously) which combine to create a desired effect—the future location. This process is implemented in an ES using deductive reasoning.

Communication

When exploring pilot communication activities, a number of different communications take place where ES are employed. Pilots receive information from the aircraft systems in the form of displays, and send information to each other and to others on the ground. A remarkably clear *pattern-of-information needs* exists during a large percentage of the time pilots are flying. Using this pattern, ES designers have implemented systems that anticipate and provide the information needed when it is needed. Typical systems include the automated information and crew alerting systems used to monitor aircraft systems, detect trends and anomalies in the system, and alert the crew to the problem. These are truly ES, in that they gather data and, rather than merely responding to it, they analyze it, consider alternative responses, and *then* initiate action.

These ES are found on most transport and military aircraft and are developed using engineering data to derive functional limits, which in turn support both rule-based and input-driven algorithms. Other forms of ES, which support pilots by managing information, are used to communicate data from the ground to the aircraft. Based on a set of rules developed by industry, a priority is assigned and managed to ensure the timely, accurate, and intelligible delivery of data between the ground station and the aircraft. These ES function on the premise that navigation and environmental information, such as weather reports, may or may not be relevant to the particular aircraft receiving the broadcast. Using rules and real-time information residing in the aircraft itself, the ES monitors, acquires, and posts information for the pilot based on the criticality of the information already in the queue.

Pilot Knowledge Acquisition

Methods of capturing knowledge of both control and navigation activities are varied. Certain things are known about the aircraft that are derived from the engineering process. Other systemic effects, such as the role of the pilot, are less certain, and there is a need to capture expert knowledge. One such method is the use of models that present reasonable representations of the expert.

Multiple model integration is used to reduce the need to explicitly define the knowledge for all cases and creates specific rules that fire in general conditions (environment is defined heuristically). This method employs both detailed and general knowledge acquisition and modeling, while yielding high confidence in the rules that fire. Piloting is well suited for such implementations, because the use of procedural knowledge to induce rules can be used to meet the need for specificity, whereas the general environmental conditions may be described using generalizations. The use of concept mapping is another method of reducing knowledge acquisition problems in complex situations. Concept mapping allows specialized knowledge in the form of heuristics and problem-solving methods to be explicitly associated by the knowledge users with static facts and general knowledge (19). Specific to concept mapping is the use of two techniques: first, the use of combing multiple input whereby the experts have collectively generated a *summary map* of the knowledge required for the particular domain; the second technique is that of indexing, which results in the development of themes and key concepts that emerge from the relationships generated in the summary mapping process.

Evaluation of Pilot Performance

ESs most often emerge as the result of capturing knowledge of one or more experts in a format that facilitates computational analysis. Typically the analysis is focused on directing the system to carry out certain rules and implement routines that have been programmed into the system. Often the goal of the ES designer is to achieve a human-like behavior from the system. One of the more difficult tasks of the ES designer is to implement a system that, in real-time or immediately following use, can assess the performance of the human component of the system. This is different from simply developing an ES that can influence the operator during its use. The ES needed to critique human behavior is required to have a set of analysis capabilities that not only relate to the rules and networks used in the system, but a set of principles that have their roots in the psychology of human actions.

A number of systems exist that provide such an ability to assess pilot performance. These systems are developed to aid in predicting human performance, critique performance, and quantify performance for use in a variety of future activities such as curriculum design, monitoring and controlling quality, or systems engineering. Conducting knowledge acquisition for assessment requires the classification of the "expert" relative to the system user.

Pilots operate in a dynamic environment and the variables that comprise the reasoning and observable behavior of pilots is often very abstract or difficult to define. One of the best examples of an ES used in assessing pilot performance is the U.S. Navy effort to design and implement a model of the pilot in flight-test maneuvers. The pilot model is a combination of skill-based models and rules-based models (multiple model integration) whereby the skill-based portion provides a feedback path and the rule-based portion provides the inference and feed-forward path (20). This ES employs the use of the Rasmussen's commonly recognized model of human error comprised of the hierarchical categories of knowledge-based, rule-based, and skill-based behaviors thought to lead to human error. Using a quasi-linear model, the system is capable of quantifying skill-based pilot behavior. To resolve discrete decision tasks, the use of a fuzzy-logic scheme is employed, which supports rule-based behavioral assessments and, when combined with the skill-based models, results in inference rules that can be used to derive large classes of pilot tasks.

Predicting Performance

In a complex system such as an aircraft, the need exists to select the best possible match of pilot and vehicle. To do this, ES are used in ground-based test environments to provide a dynamic environment that adapts to the behavior of the pilot and challenge the pilot based on the correctness of the prior action. These systems utilize psychological principles of human attention and motor skill to create a multiprocessing requirement in the human mind. ES capable of resolving these challenging environments then operate in the background of the test apparatus to provide a predictive model to the expected behavior of the pilot, who is often behind the ES in resolving the problem. The predictive nature of these systems is implemented using both rules-based and neural network structures. The ES controls the timing and the difficulty of the task based on a Bayesian process that involves the users input and changing heuristic models that form the initial foundation of the models. The ES also uses a series of time-related rules that are derived as the system is exercised to control the tempo of the primary task—flying the aircraft. Secondary tasks are driven by the ES control algorithms and the pilots input. Results are compared to the past historical results of others who have been used to establish the baseline for comparison.

Quantifying Performance

Quantifying performance can be aiding by using ES to standardize measurement and classification of pilot performance. An ES used in quantification of pilots in training uses a rules-based approach and an optimal model to infer a score on a pilot's ability to maneuver the aircraft, given a standard. This standard is used to define the baseline (quantify the optimal performance) and a simple network of all outcome collected in the past performances is used to define the pilots expected performance. Using a deductive process, the data collected in the pilot's maneuvering of the aircraft is then reduced to a set of rules where the antecedent is compared with the optimal performance antecedent and the analysis is performed based on quantitative data. This greatly improves the ability of the ES to acquire new data and weight the data according to the *best known performance* of the pilot prior to the event under study. The resulting analysis can drive a rules-based model that will then narrow the set of variables, and identify the rules that fired, leading to a refined set of maneuvers that need attention in future training.

Another ES used to quantify performance is an induction engine developed for studying accident data. It can be said that most accidents involve pilot error. These accidents, when reduced to data, which has been processed using a qualitative approach, can lead to key rules derived through induction. While this method does not result in immediate ES utility, the rules are necessary for the construction of the network of events that preceded the accident. In applying these rules to the performance of the pilot in a simulated environment, the ES is able to rank the pilot's performance by quantifying the rues which, in the past, had a high probability of firing if certain pre-accident events were present.

Knowledge Acquisition Is Based on the Target User

Since the expertise of the user is a known factor in the successful implementation of an ES, expertise is therefore relative to the design strategy. The modeling and design of the ES must be in concert with the level of expertise to be assessed. Building systems that can be flexible and adapt to such requirements require the ES designer to apply elective automation design theories. Typically ES are limited to use by one of three classes of users. The first, the novice, finds the most system utility at a level where rote or procedural knowledge is sufficient, engages the highest use of automation when available, and seeks simplicity in the interface. Second, the journeyman finds utility in the middle range of the system, partially utilizing manual and automated functions, and will tolerate most levels of complex interfaces with some training and practice. Last, the expert is capable of self-instruction on the system since they operate the system using principles and experience-based knowledge. Experts will not tolerate limited flexibility in the system's functional implementation.

EXPERT SYSTEMS AND SIMULATION IN OTHER AEROSPACE APPLICATIONS

The close relationship between aviation and simulation dates from the very beginning of aviation. The widespread use of digital computers and software-based simulations have broadened the scope of aviation simulation. By analyzing computer simulation in aviation and aerospace, we can identify the following main application areas: (a) flight dynamics simulation, (b) flight simulators for training, (c) simulation of air traffic, and (d) simulation of aerospace systems to support control and decision making.

Computer simulation is a technique that provides for imitation of real-life situations using computers. Models and simulation have been used to facilitate understanding of the real-world phenomena. Computer simulation is the discipline of designing and executing a model of an actual system with the subsequent analysis of the simulation output (21). In the complex world of aviation and aerospace, building the correct model is essential for the simulation designer to have a good grasp of the domain. Conversely, managers responsible for aerospace projects requiring simulation should be aware of the advantages and shortfalls of simulation techniques and methods. A careful approach to the design, implementation, verification, and validation of the model must be taken. The simulation experiment must be well designed, and feedback from the real-world system needs to be used as often as possible. Abundant research proposes using a knowledge-based approach to support the simulation.

There are many ways that the simulation and ES can be combined: ES can be used as an intelligent front-end to simulation. ES may invoke the simulation run to generate the required system data. Simulation may be used to update time-related variables in the ES. Finally, simulation can be a technique for ES testing (22).

The AI techniques have enhanced the historically strong relationship between aerospace training and simulation. Using AI may reduce the impact of subjective, impatient, overloaded, and error-prone human instructor links in the training process. Efforts are being made to design systems that incorporate techniques of AI and use the full power of dynamic simulation, including animation and graphics. An extensive application of these techniques gives one possible answer to the problem of effective training in situations resembling real-world conditions.

Flight Dynamics Simulation

The main feature of the flight dynamics simulation is that the aircraft model representing the handling characteristics of the airframe, engines, and the control systems is encoded in the computer. The flight dynamics simulation is based on a rigorous quantitative mathematical model expressed in terms of continuous differential equations. Research on interfacing such quantitative simulation of the aircraft in flight with a qualitative simulation, in an attempt to support decision making, has been presented in (23). The system extracts quantitative data from a mathematical model of aircraft flight dynamics and uses fuzzy inductive reasoning on the qualitative model to recognize the flight accidents.

Fuzzy Reasoning (or Fuzzy Logic) is based on the theory of Fuzzy Sets pioneered by Zadeh (9). It extends the conventional logic introducing the concept of partial truth—truth values between "completely true" and "completely false." Fuzzy Reasoning attempts to mirror the imprecision of the real world by providing a model for human reasoning in which even the truth is not an absolute but rather a matter of degree. Fuzzy Logic has emerged as a key methodology in the conception, design and deployment of intelligent systems.

Flight Simulators

The flight dynamics model and simulation is the first step for creating a ground-based flight simulator, designed to reproduce the behavior of an aircraft in flight. The simple simulators are static flight procedure trainers with the flight instruments driven by the simulation engine. Medium-range simulators use high-fidelity graphics representing a dynamic out-of-the-window view with appropriate terrain database. The high-end simulators use a six-dimensional mechanical platform to imitate the motion cues of the flight. The trainee operating the simulator controls experiences the responses of the model, displayed on the instrument panel and graphic screens similar to those in the real aircraft. Additional elements to be imitated are the communication system (radio), environmental system (e.g., engine noises), weapon system (in the case of military aircraft), etc. The multidisciplinary nature of a flight simulator requires specialized design skills and fundamental knowledge. The problems related to integrating components representing aeronautical, mechanical, electrical, and computer engineering are of primary significance.

Simulation of Air Traffic

Discrete simulation of air traffic has been used extensively to imitate flow of aircraft through the airspace. Applications range from the movement of the aircraft on the ground to the en-route and oceanic flow. The models describe aircraft behavior in all phases of the flight. Using well-established methods of discrete dynamic simulation, the air traffic models can determine a range of system parameters such as time delays, waiting queues, capacity, and occupancy. They may be used to test different ATC procedures, airspace configuration, airport development, and changes in traffic demand and patterns. Combined with animated high-fidelity graphics, the discrete simulation can be used also as a tool for training and human-factor research. Such models are used as tools in airport planning and design in relation to both aircraft and passenger flow.

Often considering individual aircraft is not practical in aerospace strategic models geared toward TFM. The continuous approach where the network of air routes is simulated with flow rate in the links may be used. Techniques of simulation and operation research are used to determine capacities of airways, airports, location of hubs, airline schedules, etc. They are used both for analysis and design as well as for the actual TFM.

One of the comprehensive simulation systems is Total Airspace Airport Modeler (TAAM), developed by The Preston Group, combining real-time simulation and an interactive graphic user interface with AI elements. The system supports the construction of aircraft performance data, airport and airspace elements, procedures and strategies, definition of traffic schedules, and rules for aircraft conflict resolution. TAAM simulates the actual movement of all aircraft through the airspace, collecting data on sequencing, delays, and potential airborne conflicts. A simplistic ES is used to resolve potential conflict between the simulated aircraft (24).

ATCoach, developed by UFA, Inc., and TRACON Pro, developed by Wesson International, are two examples of an off-the-shelf simulation system designed to provide a comprehensive training for air traffic controllers. Both systems include elaborate simulation with user-friendly interface, including speech capability and realistic graphics. The knowledge base of the ATC operation is used to support the training component. In ATCoach, the focus of ES is on monitoring the training session and providing domain guidance to the student (25). In TRACON Pro, the AI techniques support planning of airspace events using procedural techniques as researched by Wesson (26).

EXPERT SYSTEMS IN AVIATION AND AEROSPACE SYSTEMS

Model-based reasoning is another AI approach, which bases system behavior on the behavior of the system subcomponents as represented by the frame-based model. Knowledge-based autonomous test engineer (KATE), developed for the National Aeronautics and Space Administration (NASA) by Boeing Space Operations (27), is a generic software shell for performing model-based monitoring, fault detection, diagnosis, and control. The four subsystems are (1) simulation, (2) monitoring, (3) diagnosis, and (4) control. The system originated in the mid 1980s as a tool to support the operation of the launch processing system. KATE was particularly designed to check sensor operation for the Space Shuttle liquid-oxygen loading system. The system is based on a model of the sensor structure and diagnoses sensor failures. By separation of the system structure from the component functions, a more generic tool was designed. During the early 1990s, the system started its operational application monitoring the tanking data. The system was redesigned and implemented in C++ programming language using popular Motif windowing environment on a UNIX workstation to serve as a part of the vehicle health management system.

Yet another facet of ES application is in the area of planning and scheduling. One example of such application is the automatic cockpit crew scheduling developed by Japan Airlines and NEC (28) The system is designed to prepare monthly schedules for flight crews. The system knowledge is represented in frames and rules. The system's distributed architecture allows it to run inferencing on slave computers, with the master computer serving as a cooperative inference area and the monitor of data integrity. The backtracking technique is used to break a deadlock when the crew assignment can not be found. Another example is an ES tool to support shift duty assignments for an airport staff (29). The rules-based system combines forward-chaining inference and constraints-relaxation techniques. It produces a timetable starting with the initial assignment and continuing through the iterative improvement process. The prototype has been tested in airport operations.

FUTURE TRENDS

ES already plays a vital role in the safety and effectiveness of complex systems. Their future in aerospace includes autonomous vehicles in both military and passenger aircraft; cooperating ES, such as those that would provide separation of aircraft in flight; ATC systems that improve the safety and efficiency of airspace use and of airports; and, to some extent, training systems that deliver individualized lessons to students.

The need to capture knowledge regarding the human operator in the aerospace system is clear; however, the ability to accurately and effectively describe that knowledge in today's complex systems is becoming less practical using old techniques. The future of ES design will focus on practical knowledge engineering techniques that use the target system as a means of collecting information and creating knowledge about the users. In such a systemic approach, knowledge engineering will evolve to include knowledge about the human, human systems interfaces, and the systemic effects on human operators interpretation of the system feedback. The use of such developing technologies as neural networks and ES that adapt will be more prominent than in systems in use today. The adaptive system is capable of both induction and adaptation—principles of elective automation that expose the need for the designer to consider the individual variance of each potential system operator—thereby providing a system that can have various (infinite) combinations of automated support.

As the ES designer elicits knowledge to support the design process, the use of new allocation techniques will arise. This new ES design approach is driven primarily by the need to resolve a design conflict involving the expertise levels of the user. Experts exhibit more bias than novices, declarative knowledge forms a better foundation for self-critique, and experts adopt practices in lieu of formal procedures. This shift in the user's mental attitude creates a need to build an ES that will allow the user to transition from novice to expert over time, while recognizing the shift in bias based on decay of formal knowledge (declarative and procedural) in favor of experience and practices. Failure to allocate functions correctly in this transition-capable system is a breech of the design principle of flexibility in elective automation and will result in substandard human–system performance as both levels.

The need to build transitory systems stems from the computer community itself similar to adaptive design principles. The desire to build small-scale systems has existed for decades; however, practical aspects of building a microsystem, such as complexity of design, cost, and production engineering have been seen as limitations. The microprocessor-facilitated size reduction of computing equipment to a level that made applications practical in typical human-system interfaces. This revolution in size has allowed rapid (and sometimes careless) implementation of technology into human–system interfaces. These implementations have changed the way humans understand and interact with systems and, as the generation of users that has grown up without such aids leaves the design workforce, system boundaries adjust to an artificial, and possibly arbitrary new set of limitations that constrain the next-generation designer. This is the foundation of the need for sound knowledge engineering and functional allocation.

New systems are capable of more and the assumption that humans are therefore not capable or encouraged to do as much in the system as in the past will gradually become a *pseudo-standard* for designers. This will lead to the extinction of human capabilities unless the system design is tempered with an understanding that humans are capable of certain tasks, desire to perform certain tasks, and that allocation of these tasks to ES when humans are capable of them is changing the system dynamics. The correct allocation depends on a number of factors including reliability, cost, and efficacy of the allocation, preference, and systemic risk mitigation.

ES that are currently in use, such as those described in this article, will tend to proliferate in the aerospace community. Synergy among these ES is likely to lead to cooperating ES that seek out and elicit knowledge from each other. This introduces another growing area where the future holds a paradox for designers and implementers. The need to assess the efficacy of information are paramount to human decision making. Primary factors affecting the efficacy of the information is the medium, the source, and the timeliness of that information. Relevance of the information to the problem at hand enters as a secondary criterion when there is ample information pertaining to the problem. The abundance of information (contrasted with the dearth of information), defines the ends of the spectrum in the paradox. The designers job is to foresee the sources, define the timeliness attributes, and deliver the information to the system inference engine so that the process will continue with a high level of confidence and, yet within a reasonable period of time as defined by the knowledge engineers and the users.

At this point it becomes obvious that the designer has set about controlling the ES process and, thus, has limited the system to the boundaries of efficacy that the designer(s) foresee as reasonable. To overcome this contradiction, the ES of the future will use a combination of Bayesian and induction processes to match heuristic models and specific knowledge seeking to reach theoretical *absolute certainty*, while settling for something less to ensure that the process continues. Limits of time will be systemically controlled using system-derived rules. For example, an ES can learn to *tell time* due to the ratio quality of the data when these data are input over a considerable amount of time. ES have high confidence in the longitudinal approach to using ratio data in developing knowledge. ES are considered to be *reasoning* when either the data type or the duration of the data input is less than continuous and extended. The use of quantitative data can improve the use of short-duration inputs, but the less preferred data types (nominal, ordinal, and to some extent interval) create certainty problems for today's ES. This is likely to be the next major improvement in expert systems in aerospace applications—the ability for the system to reason using data that are meaningful to human processing, but not yet reliable for computing.

BIBLIOGRAPHY

1. M. S. Nolan, *Fundamentals of Air Traffic Control,* Belmont, CA: Wadsworth 1993.
2. A. L. Elias and J. D. Pararas, Potential use of artificial intelligence techniques in air traffic control, Transportation Research Circular, TRB, National Research Council, Washington, DC, AI Workshop Report, 1985, pp. 17–31.
3. *Webster's New World Collegiate Dictionary,* 3rd ed. New York: Simon & Schuster, 1996.

3a. S. Haykin, *Neural Networks: A Comprehensive Foundation,* New York: Macmillan, 1994; A. P. Sage (ed.), *Concise Encyclopedia of Information Processing in Systems and Organizations,* New York: Pergamon, 1990.

4. L. L. Smith, The distributed intelligence system and aircraft pilotage, in *AI and Simulation,* San Diego: Simulation Councils Inc., SCS, 1985, pp. 26–28.
5. A. Kornecki, Simulation and AI in aviation training, in G. W. Zobrist and J. V. Leonard (eds.), *Progress in Simulation,* vol 2, New York: Ablex Publishing, 1994, pp. 91–122.
6. J. G. Moser, Integration of AI and simulation in a comprehensive decision-support system, *Simulation,* **47** (6): 223–229.
7. Y. Nakatani, M. Tsukiyama, and T. Fukuda, Case-based reasoning and decision aid for engineering design, in *Proceedings of the World Congress on Expert Systems,* New York: Pergamon Press, 1991, pp. 369–376.
8. R. O. Duda, P. E. Hart, and N. Nilsson, Subjective Bayesian methods for rule-based inference systems, in *Proceedings of National Computer Conference,* AFIPS, 1976, pp. 1075–1082.
9. L. Zadeh, The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets Syst.* (11), 199–227, 1983.
10. A. Kornecki, Artificial intelligence for air traffic, *IEEE Potentials,* **13** (3), 11–14, 1994.
11. A. Kornecki, SMARTFLO—Knowledge based system to support central flow operations, in *Proceedings of the 37th Annual Air Traffic Control Association Conference,* 1992, pp. 862–869.
12. T. Diefenbach, D. Carl, and M. Towhidnajad, Intelligent tutoring and air traffic control training, in *Proceedings of the 37th Air Traffic Control Association Conference,* 1992, pp. 489–494.
13. A. Kornecki et al., Intelligent tutoring issues for ATC training system, *IEEE Trans. Contr. Syst. Technol.,* **1** (3): 204–211, 1993.
14. V. Galotti and A. Kornecki, Knowledge engineering for an air traffic expert system, in *Proceedings of the 36th Annual Air Traffic Control Association Conference,* 1991, pp. 207–211.
15. J. Giarratano and G. Riley, *Expert Systems, Principles and Programming,* Boston: PWS-Kent Publishing, 1989.
16. A. Kornecki, Building of an air traffic control expert system, in M. Hamza (ed.),*Proceedings of the IASTED International Symposium on ES,* Anaheim, CA: Acta Press, 1989, pp. 217–219.
17. J. M. Hammer and R. Small, Intelligent interface in an associate system, in W. B. Rouse (ed.), *Human Technology Interaction in Complex Systems,* Greenwich, CT: JAI Press, 1995.
18. E. J. Bass, S. Ernst-Fortin, and R. Small, Knowledge base development tool requirements for an intelligent monitoring aid, in *Proceedings of the 10th FLAIRS,* 1997, pp. 412–416.
19. D. Snyder et al., Knowledge acquisition of tactical air-to-ground mission information using system mapping, in *Proceedings of National Aerospace and Electronics Conference,* New York, IEEE, 1992, pp. 668–674.
20. M. R. Anderson, C. Clark, and G. Dungan, Flight test maneuver design using a skill- and rule-based pilot model, in *Proceedings of International Conference on Systems, Man and Cybernetics,* New York: IEEE, 1995, pp. 2682–2687.
21. P. Fishwick, *Simulation Model Design and Execution—Building Digital Worlds,* Englewood Cliffs, NJ: Prentice Hall, 1995.
22. R. O'Keefe, Simulation and expert systems—a taxonomy and some examples, *Simulation,* **46** (1): 10–16, 1986.
23. A. de Albornoz and F. E. Cellier, Building intelligence into an autopilot—using qualitative simulation to support global decision making, *Simulation,* **62**: 354–363, 1994.
24. Anonymous, *TAAM—Total Airport Airspace Modeller Product Profile,* Richmond, Victoria, Australia: The Preston Group, 1991.
25. J. A. Scardina, P. Y. Ryberg, and A. Gerstenfeld, Future ATC automation aids based upon AI technology, *Proc. IEEE,* **77**: 1625–1633, 1989.
26. R. B. Wesson, Problem solving in the world of an air traffic controller, doctoral dissertation, UT&A, University Microfilms International, Ann Arbor, MI, December 1977.
27. C. O. Pepe et al., KATE—a project overview and software description, report, Boeing Aerospace Operations, Kennedy Space Center, FL, March 1992.
28. K. Onodera and A. Mori, Cockpit crew scheduling and supporting system, in *Proceedings of the World Congress on Expert Systems,* New York: Pergamon Press, 1991, pp. 1–10.
29. K. P. Chow and C. K. Hui, Knowledge based approach to airport staff rostering: a case study, in *Proceedings of the World Congress on Expert Systems,* New York: Pergamon Press, 1991, pp. 46–53.

Reading List

J. Liebowitz, *Introduction to Expert Systems,* Santa Cruz, CA: Mitchell, 1988.

E. Wagner, *Artificial Intelligence and Tutoring Systems,* Los Altos, CA: Morgan Kaufman, 1987.

T. I. Oren, Artificial intelligence and simulation, *AI Applied to Simulation,* **18** (1), 3–8, 1986.

A. M. Wildberger, Integrating an expert system component into a simulation, *AI Papers,* **20** (1), 132–135, 1988.

R. H. Michaelsen, D. Michie, and A. Boulanger, The technology of expert systems, *BYTE,* **10**: 303–312, April 1985.

F. Hayes-Roth, P. Klahr, and D. J. Mostow, Knowledge acquisition, knowledge programming, and knowledge refinement, in P.Klahr (ed.), *The Rand Corporation, R-2540-NSF, 1980,* Reading, MA: Addison-Wesley, 1986, pp. 310–349.

A. Gerstenfeld, Simulation combined with cooperating expert systems: an aid for training, screening, plans and procedures, *J. ATC,* **30**: 33–35, 1988.

D. Spencer, Development environment for an ATC expert system, in *Transportation Research Circular,* TRB, National Research Council, Washington, DC, AI Workshop Report, 1985, pp. 32–37.

C. A. Shively, AIRPACK: Advisor for intelligent resolution of predicted aircraft conflicts, *Transportation Research Circular,* TRB, National Research Council, Washington, DC, AI Workshop Report, 1985, pp. 58–64.

A. Gonzalez et al., Simulation based expert system for training air traffic controllers, in M. B. Fishman (ed.), *Advances in Artificial Intelligence Research,* Greenwich, CT: JAI Press, 1989, pp. 295–308.

R. Steeb et al., Distributed problem solving for air fleet control: framework and implementation, in P. Klahr (ed.), *The Rand Corporation, N-2139-ARPA,* 1984, Reading, MA: Addison-Wesley, 1986, pp. 391–432.

P. McKinnon, Living with artificial intelligence, *J. ATC,* **29**: 23–25, 1987.

Web Sites

Federal Aviation Administration http://www.faa.gov

MITRE Center for Advanced Aviation System Development http://www.caasd.org

National Aeronautics and Space Agency Ames Research Center—Advanced Air Transportation Technologies http://aatt.arc.nasa.gov

Massachussets Institute of Technology—Lincoln Laboratory http://www.ll.mit.edu

AI resources http://www.cs.reading.ac.uk/people/dwc/ai.html

ANDREW J. KORNECKI
JAMES W. BLANCHARD
Embry-Riddle Aeronautical University

AEROSPACE INDUSTRY. See AIR TRAFFIC.