Guest Editors' Introduction

The Federal Aviation Administration's Advanced Automation Program

Valerio R. Hunt, US Department of Transportation, FAA
Gregory V. Kloster, Knowlex Technology Corporation

Engineering the world's largest command and control system is a complex undertaking. This issue communicates the authors' experiences in the design of the next-generation air traffic control system.

We started working on this issue with the objective of capturing our collective experiences in adapting and innovating modern computer science and system engineering techniques and methods to the problem of air traffic control systems design. We recognize that the Advanced Automation Program is a complex technological undertaking that requires government and industry to address issues and problems in system safety, reliability, long system life, architecture design, software engineering, and man-machine interface, or MMI, design in very new and innovative ways. We look at this special issue as a vehicle for technology transfer to readers that are concerned not only with the development of a safe air traffic control system, but with the technologies used in the design, development, and implementation of large complex, highly available, and interactive systems.

Computer automation was introduced into en-route and terminal radar air traffic control centers in the early to mid-1970’s. In the en-route centers flight data and radar data processing represent the key automation functions operating on IBM 9020 computers. Automated radar data processing and limited flight data readout and input capabilities are operational in systems known as automated radar terminal systems, or ARTS, II and III. The
ARTS II system was implemented on Burroughs computers and the ARTS III/IIIA systems operate on Sperry Univac computers.

The Advanced Automation Program consists of two major Federal Aviation Administration, or FAA, system acquisitions. The first is the host program, which involves the replacement of current IBM 9020 computer systems at all en-route air traffic control centers. This acquisition was awarded (after a design competition) to IBM in July 1985. The second is the Advanced Automation System, or AAS, which replaces both the en-route and terminal approach control air traffic control, or ATC, computer systems over the next decade. The AAS will provide the computers, software, and controller workstations (called sector suites) for integrated en-route and terminal radar ATC. The AAS will also include a tower control computer complex for use by the controllers in the tower cabs. This new system will replace the en-route and terminal ATC computers, flight strip printers and display equipment, and the backup radar display system.

The AAS program consists of a two-phased government acquisition where the first phase includes a competitive design and prototype "fly-off" between two industrial teams lead by GM-Hughes Aircraft and IBM. The second phase consists of a winner-take-all production and implementation contract. (Because of the nature of the design competition we have taken great pains to avoid publication of technical data that might compromise the competition and proprietary rights of the competitors.) All of the articles in this special issue, with the exception of one, deal with the AAS program.

The FAA has accumulated a legacy of experiences in both ATC operations and software maintenance and test. The consensus has been that the new advanced ATC automation system must

- provide a time-responsive set of ATC services (to the controller) that are highly available and feature inherent protection mechanisms to detect, contain, and recover from faults;
- provide for evolution in ATC functions and capabilities without affecting safety and required redesign of the basic system; and
- provide a user/computer interface that facilitates the evolution of the system and accommodates the basic manner in which the controller performs his or her ATC information processing tasks today.

Based on our past experiences and this consensus, we feel that risks that might result in either unsafe, unsuitable operations or costly high-risk technology are both real and must be eliminated or greatly reduced. We recognize that the AAS poses several challenges to the engineer and designer.

Since the inception of the AAS requirements definition effort over the past six years, significant progress has been made in MIMI (also termed computer-human interface) design and human engineering methods; computer workstation display technology; software engineering practices; tools; computer system performance modeling; and computer system fault tolerance. We have felt strongly that these practices, tools, and methods must be used in order to mitigate the critical technical risks associated with either a system design that inhibits system evolution or a user interface that is not acceptable to air traffic controllers.

Of paramount concern is the development of a safe system architecture that provides adequate coverage and protection from faults. In response to this latter concern the FAA has not only levied on the AAS stringent reliability and availability requirements, but has adopted standards and practices aimed at the development of a highly reliable system.

The strategy selected by FAA to meet these unique requirements resulted in extensive focus on precontract requirements definition activities (prior to the design competition phase award) and prime contractor system design in the early part of the program. The articles in this issue address many of those activities undertaken prior to the start of the AAS Design Competition Phase in August 1984. Postcontract award activities have been described in the AAS Statement of Work. Some of these efforts represent further refinement and implementation of FAA initiatives in MIMI requirements engineering and controller workstation design. Others require the prime contractors to refine and develop innovative approaches to engineering a fault-tolerant computer architecture and distributed software processes.

**Background and overview of the AAS program**

The first article in this issue examines the challenges and hurdles faced by government engineers and industry developers. In this article, *The FAA’s Advanced Automation System: Strategies for Future Air Traffic Control Systems*, Valerio Hunt and Andres Zellweger describe the system acquisition and transition strategies for the FAA’s Advanced Automation Program. The article also provides a background discussion (in vignette form) of the US air traffic control system to familiarize the reader with this application. In particular, the authors examine the technological demands that the AAS requirements place on the system designer.

The investment in time and money (estimated $2-3 billion) to design and implement such a system dictates balancing the factors of safety, reliability, and long system life. The current ATC automation systems will have been in service over 20 years when they are replaced by the Advanced Automation System, and there is no reason to suspect that the basic AAS system design and software will not have a 20- to 30-year life span.

Such factors place difficult demands on the system designer. The functions to be performed by the computer system (and, therefore, the man-machine interface), the external data sources (aircraft surveillance systems, weather sensors, navigation systems, etc.), as well as the traffic levels, air fleet mix, aircraft performance characteristics, and avionic capabilities all change over time. The AAS designers must ensure that this evolution can be accommodated at a reasonable cost and without detrimental impact on system reliability (hardware and software) and on the critical man-machine interface design.

A second demand on the system designer that stems from the long system life expectancy is the need to accommodate, without significant system perturbation, upgrades of hardware technology. This requirement stems from the fact that today’s computer generation for large commercial systems are 4-7 years and even less in the rapidly expanding microprocessor area. Government and industry experience in the last decade has shown that cost-effective computing, from a total life cycle cost viewpoint, is best achieved by a well-defined capacity management program that permits upgrades as demand increases or as a hardware generation is superseded. Without an AAS design that permits hardware upgrades, the initial computer hardware installed only have to be sized to handle the largest projected demand for traffic and functional growth throughout the system life. Maintenance cost would be
unacceptably high by the end of the system's life since it would represent a 20- to 30-year-old technology.

Following the above strategy and applying computer science and software engineering principles of the 1980's will help achieve the extensibility characteristics necessary for a successful long system life, but this is not enough. If a system with the stringent performance and reliability characteristics of the ATC system is to evolve in a cost-effective manner, the system designer must be able to anticipate the most likely course of evolution and the nature of the changes that can be expected. For example, the input/output structure should anticipate future system interfaces and the data structures should anticipate data attributes that might become important as more of the ATC decision-making responsibility is shifted to the computer. The problem for the system designer arises from the fact that no firm requirements for the system interfaces of functions anticipated for the 1990's and beyond exist at this time.

To help add the proper evolutionary characteristics to the system design, FAA has developed a functional and performance description of what, to the best of our current knowledge, a highly automated ATC system would be.3 This description places particular emphasis on how the advanced functions might interface with the ATC computer system that will actually be built by the AAS contractors and that will form the starting point for future evolution. The AAS system designers are required by contract to show FAA, at various stages of system design and development, how their particular design can be extended to accommodate these likely future functions.

### Air traffic control system architecture

In the second article, Evaluating Proposed Architectures for the FAA's Advanced Automation System, Jean-Marc Garot, Thomas Hawker, and Delbert Weathers examine the fundamental methods used by the FAA to evaluate complex distributed system architectures. This article examines the complex issues involved in architecture design, cites lessons learned from other programs, and describes the approach used to evaluate system architectures. This process should prove useful to those interested in both the development and design validation of complex systems. Because of the nature of the AAS design competition, this article has avoided mentioning specifics concerning each prime contractor's architecture. What is noteworthy is that the application of a formal evaluation process is essential to determining that the system design (as proposed) satisfies AAS requirements.

Clearly the basic system design, in terms of both hardware and software structure, will have to remain intact for the total system life. So an important evaluation method must qualitatively examine how the system is expected to be augmented and modified as the system evolves. To make this possible, FAA has required that the system be designed around a local communications network (with the requisite reliability characteristics) based on local area network technology and industry-accepted protocol standards. This will allow the introduction of new hardware technology without major system perturbations. To complement the hardware strategy, software will be functionally distributed and, perhaps more importantly, developed in a single high-order language (such as Ada) that permits easy migration from one processor to another. An important criterion in the evaluation of proposed architectures will be the impact of change on system stability since the ATC application cannot tolerate degradation in system reliability and availability during and after a change.

### Controller MMI and operational suitability

An operative phrase used in connection with the AAS is "operational suitability." While the value of accurately capturing user knowledge and infusing this knowledge into the AAS system development process is clear enough, the process of doing this can be quite a bit more complex than it may seem. Users may be biased by their own limited experience with particular design features, peculiarities of their operations at a given air traffic sector, or the interaction techniques they employ when using the system. Users also tend to offer isolated "fixes" to perceived problems, rather than to regard functions in the context of system design decisions that respond to functional and performance requirements. Lastly, user terminology often differs from engineering terminology. This results in fertile ground for misinterpretation of requirements on both ends. It is for these and other reasons that others have previously noted that noncritical acceptance of user inputs may constitute the first step in the design process that will most assuredly result in an operationally unsuitable design.

The FAA's Advanced Automation Program Office and Air Traffic Service have successfully implemented a program of close cooperation and involvement by "users" in the full cycle of AAS engineering and development. There has been a certain amount of concern that the AAS end-product (in terms of the MMI software and controller workstation) might not be acceptable to the air traffic controller. Also, there was the concern, given the complexity of the controller's job, that there was significant risk in misinterpreting and developing incomplete requirements. The difficulty lies in attempting to develop a system with a new man-machine interface that is expected to be available in the early 1990's. This risk has been mitigated with the formation in 1983 of the FAA's sector suite requirements validation team, or SSRVT, and the subsequent application of comprehensive MMI design and human factors methods. So important is this undertaking two articles in this issue address it.

In the third article, Engineering the Man-Machine Interface for Air Traffic Control, Gregory Kloster and Andres Zellweger not only examine the process used to formulate MMI requirements, but describe the life cycle involvement of the SSRVT in that process. Of note are discussion and examples that show how this method was used to formulate an operations concept for the AAS man-machine interface, derive MMI functional requirements for the controller interface language, and determine requirements for the controller workstation. The origins of this method were developed at the TRW Defense Systems Group and refined at Computer Technology Associates, Inc.4 This we feel represents a dramatic departure from the way systems are typically acquired by the government. The article also describes prime contractor activities after the design competition phase contract award in August 1984.

In the fourth article, The Quantification of Operational Suitability, Mark Phillips describes how the SSRVT and the FAA both derive user requirements and assess the operational suitability of the emerging MMI designs during the design competition phase. This process does not merely consider the users to be passive particip-
pants or make attempts at identifying requirements that are "likely" to result in a "user friendly" design, but shares information on how a team of users and human factors specialists made a joint commitment to ensure the significant and sustained involvement of a representative group of actively working air traffic controllers. Again, one cannot underestimate the importance of applying evaluation methods that offer both quantitative and qualitative evaluations of ergonomic quality. It is here that we offer a comprehensive look at how complex MMI designs can be evaluated in terms of operational suitability and ergonomic quality metrics.

**FAA computer capacity management**

In the fifth article, *Capacity Management of Air Traffic Control Computer Systems*, Sandra Bleistein, Robert Goette, Frank Petroski, and Robert Wiseman describe how performance modeling and analysis is integral to the FAA's strategy of computer capacity management for the current 9020 en-route computer system, host computer system, and the future AAS. Capacity management of these three generations of computer systems consists of five common elements:

1. Specification of performance requirements in terms of computer responsiveness;
2. Projection of traffic workloads;
3. Development of performance and workload measurement capabilities;
4. Modeling and prediction of performance; and
5. Development of a program and related procedures for capacity management.

Each of these elements is discussed in the article. The evolution from basic capabilities in the 9020 system to the anticipated AAS capabilities is described. Results from each system and their influence on the capabilities of the succeeding system are presented.

**AAS dependability**

In the last article, *On the Achievement of a Highly Dependable and Fault-Tolerant Air Traffic Control System*, Al Avizioni and Dan Ball describe the evolution of the AAS reliability, maintainability, and availability (RMA) requirements and discuss issues and concerns in achieving a highly reliable AAS. The authors contrast the reliability problems of the existing system and goals for the new AAS. Of particular interest is a discussion on the evolution of the AAS RMA requirements and the FAA's strategy for achieving a highly reliable AAS. The article concludes with lessons learned so far and guidelines for future acquisitions of complex fault-tolerant systems.

The significance and scope of the FAA's modernization programs for air traffic control should not be underestimated. The technology fallout of the Advanced Automation Program has implications for government, industrial, and commercial organizations engaged in real-time systems, distributed computing, design of fault-tolerant software, design of complex interactive user interfaces, testing, capacity management, and use of new software development tools and Ada. Innovative tools, techniques, and methods will continue to be developed by the FAA, its support contractors, and prime contractors as the AAS moves into the acquisition phase of full-scale development, production, and test. Our view is that additional technical contributions will be forthcoming in this magazine and other IEEE publications. It is in the spirit of promoting technology transfer and communicating our collective experiences that we hope to improve the management and development of complex ultra-reliable systems.

**Acknowledgment**

The authors in this special issue extend to C. V. Ramamurthy their thanks for his guidance and helpful comments.

**References**


**Valerio R. Hunt** served as the director of the Federal Aviation Administration's Advanced Automation Program Office from its creation in 1981 until August 1986. He organized and staffed the office, directed all of the planning and start-up activities leading to the establishment of a technical approach, schedules, financial management plans, budgets, selection of contractors, and management of contractors' work. Currently Hunt is an independent contractor in Bethesda, Maryland, providing consulting services to the government and industrial community.

Hunt received his BS from Lehigh University in 1949 and completed advanced studies in computer science and communications networking in 1957. Honors include Phi Eta Sigma, Phi Beta Kappa, and Tau Beta Pi. His current interests are in the design and management of large computer-based systems.

**Gregory V. Kloster** is currently chairman of the FAA's Advanced Automation System's technical advisory group, or ATAG. The ATAG provides specific high-level industry and university computer science expertise to the FAA in the areas of reliability, maintainability, and availability and system architecture. Prior to this assignment, he was responsible (while at Computer Technology Associates, Inc.) for the technical management of an FAA project that applied an MMI design methodology to engineer the air traffic controller user interface and sector suite workstation requirements.

Kloster holds an MS in management science from West Coast University. He has been president of Knowlex Technology Corp. since 1984. His current interests are in development of tools for system and software engineers. Before that he was an associate and program manager with Computer Technology Associates and an assistant project manager with TRW Defense Systems Group.

Readers can write to Kloster at Knowlex Technology Corp., 301 Dexter St., Denver, CO 80220.