J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

AIRCRAFT COMPUTERS

The aircraft industry and the computer industry are relative newcomers in two centuries of technical innovation. It is only natural that these powerful industries have merged to provide continuous improvements in capabilities and services for aircraft customers. Landau (1) defines an aircraft as any structure or machine designed to travel through the air. He then defines a computer as a person who computes or a device used for computing. From these definitions, an aircraft computer is a device used on (or in association with) any air-traveling machine or structures used to make computations. Computers can be found in every aspect of the aircraft industry. On the aircraft, there are computers for flight control and display, computers monitoring and regulating flight functions, computers recording and processing flight activities, computers providing passenger entertainment, and computers providing communication and navigation. Equally important are the ground-based computers at airports, maintenance depots, and air traffic control stations that provide services for all aspects of flight.

Figure 1 shows a typical aircraft central computer (*CC*) used in modern fighter aircraft. This particular computer is also referred to as a fire-control computer (*FCC*), because it directs the delivery of weapons in conjunction with the aircraft's sensor systems.

Aircraft Analog Computers. Early aircraft computers were used to take continuous streams of inputs to provide flight assistance. Examples of aircraft analog inputs are fuel gauge readings, throttle settings, and altitude indicators. Landau (1) defines an analog computer as a computer for processing data represented by a continuous physical variable, such as electric current. Analog computers monitor these inputs and implement a predetermined service when some set of inputs calls for a flight control adjustment. For example, when fuel levels are below a certain point, the analog computer would read a low fuel level in the aircraft's main fuel tanks and would initiate the pumping of fuel from reserve tanks, or balancing fuel between wing fuel tanks. Some of the first applications of analog computers to aircraft applications were for automatic pilot applications, where these analog machines took flight control inputs to hold altitude and course. The analog computers use operational amplifiers to build the functionality of summers, adders, subtracters, and integrators on the electric signals.

Aircraft Digital Computers. As the technologies used to build digital computers evolved, digital computers became smaller, lighter, and less power-hungry, and produced less heat. This made them increasingly acceptable for aircraft applications. Digital computers are synonymous with stored-program computers. A stored-program computer has the flexibility of being able to accomplish multiple different tasks simply by changing the stored program. Analog computers are hard-wired to perform one and only one function. Analog computers' data, as defined earlier, are continuous physical variables. Analog computers may be able to recognize and process numerous physical variables, but each variable has its unique characteristics that must be handled during processing by the analog computer. The range of output values for the analog computer is bounded as a given voltage range; if they exceed this, they saturate. Digital computers are not constrained by physical variables. All the inputs and outputs of the digital computer are in a digital representation. The processing logic and algorithms performed by the computer work in a single representation of the cumulative data. It is not uncommon to see aircraft applications that have analog-to-digital and digital-to-analog signal

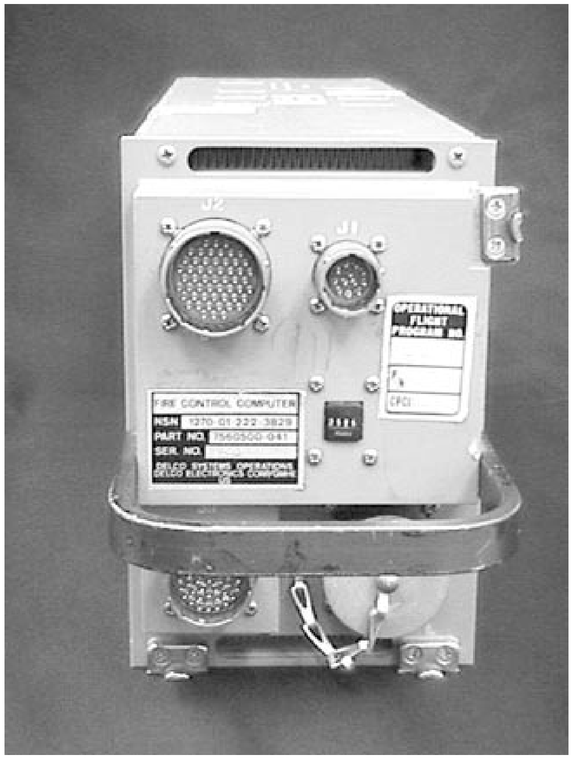

Fig. 1. Typical aircraft central computer.

converters. This is more efficient than having the conversions done within the computers. Analog signals to the digital computer are converted to digital format, where they are quickly processed digitally, and returned to the analog device through an digital-to-analog converter as an analog output for that device to act upon. These digital computers are smaller, more powerful, and easier to integrate into multiple areas of aircraft applications.

Landau (1) defines a digital computer as a computer for processing data represented by discrete, localized physical signals, such as the presence or absence of an electric current. These signals are represented as a series of bits with word lengths of 16, 32, and 64 bits. See *microcomputers* for further discussion.

Wakerly (2) shows number systems and codes used to process binary digits in digital computers. Some important number systems used in digital computers are binary, octal, and hexadecimal numbers. He also shows conversion between these and base-10 numbers, as well as simple mathematical operations such as addition, subtraction, division, and multiplication. The American Standard Code for Information Interchange (*ASCII*) of the American National Standard Institute is also presented, which is Standard No. X3.4-1968 for numerals, symbols, characters, and control codes used in automatic data-processing machines, including computers.

Microcomputers. The improvements in size, speed, and cost through computer technologies continually implement new computer consumer products. Many of these products were unavailable to the average consumer until recently. These same breakthroughs provide enormous functional improvements in aircraft computing. Landau (1) defines microcomputers as very small, relatively inexpensive computers whose central processing unit is a microprocessor. A microprocessor (also called *MPU* or central processing unit [*CPU*]) communicates with other devices in the system through wires (or fiber optics) called lines. Each device has a unique address, represented in binary format, that the MPU recognizes. The number of lines is also the address size in bits. Early MPU machines had 8-bit addresses. Machines of 1970–1980 typically had 16-bit addresses; modern MPU machines have 256 bits.

Common terminology for an MPU is *random-access memory* (*RAM*), *read-only memory* (*ROM*), *input–output*, clock, and *interrupts*. RAM is volatile storage. It holds both data and instructions for the MPU. ROM may hold both instructions and data. The key point of ROM is that it is nonvolatile. Typically, in an MPU, there is no operational difference between RAM and ROM other than its volatility. Input–output is how data are gotten to and from the microcomputer. Output may be from the MPU, ROM, or RAM. Input may be from the MPU or the RAM. The clock of an MPU synchronizes the execution of the MPU instructions. Interrupts are inputs to the MPU that cause it to (temporarily) suspend one activity in order to perform a more important activity.

An important family of MPUs that greatly improved the performance of aircraft computers is the Motorola M6800 family of microcomputers. This family offered a series of improvements in memory size, clock speeds, functionality, and overall computer performance.

Personal Computers. Landau (1) defines personal computers as electronic machines that can be owned and operated by individuals for home and business applications such as word processing, games, finance, and electronic communications. Hamacher et al. (3) explain that rapidly advancing very large-scale integrated circuit (*VLSI*) technology has resulted in dramatic reductions in the cost of computer hardware. The greatest impact has been in the area of small computing machines, where it has led to an expanding market for personal computers.

The idea of a personally owned computer is fairly new. The computational power available in hand-held toys today was only available through large, costly computers in the late 1950s and early 1960s. Vendors such as Atari, Commodore, and Compaq made simple computer games household items. Performance improvements in memory, throughput, and processing power by companies such as IBM, Intel, and Apple made facilities such as spreadsheets for home budgets, automated tax programs, word processing, and three-dimensional virtual games common household items. The introduction of Microsoft's Disk Operating System (*DOS*) and Windows has also added to the acceptance of the personal computers through access to software applications. Improvements in computer technology offer continual improvements, often multiple times a year. The durability and portability of these computers is beginning to allow them to replace specialized aircraft computers that had strict weight, size, power, and functionality requirements.

Avionics

In the early years of aircraft flight, technological innovation was directed at improving flight performance through rapid design improvements in aircraft propulsion and airframes. Secondary development energies went to areas such as navigation, communication, munitions delivery, and target detection. The secondary functionality of aircraft evolved into the field of avionics. Avionics now provides greater overall performance and accounts for a greater share of aircraft life-cycle costs than either propulsion or airframe components.

Landau (1) defines avionics [avi(ation) + (electr)onics] as the branch of electronics dealing with the development and use of electronic equipment in aviation and astronautics. The field of avionics has evolved rapidly as electronics has improved all aspects of aircraft flight. New advances in these disciplines require avionics to control flight stability, which was traditionally the pilot's role.

Aircraft Antennas. An important aspect of avionics is receiving and transmitting electromagnetic signals. Antennas are devices for transmitting and receiving radio frequency (*RF*) energy from other aircraft, space applications, or ground applications. Perry and Geppert (4) illustrates the aircraft electromagnetic spectrum, influenced by the placement and usage of numerous antennas on a commercial aircraft. Golden (5) illustrates simple antenna characteristics of dipole, horn, cavity-backed spiral, parabola, parabolic cylinder, and Cassegrain antennas.

Radiation pattern characteristics include elevation and azimuth. The typical antenna specifications are polarization, beam width, gain, bandwidth, and frequency limit.

Computers are becoming increasingly important for the new generation of antennas, which include phased array antennas and smart-skin antennas. For phased array antennas, computers are needed to configure the array elements to provide direction and range requirements between the radar pulses. Smart-skin antennas comprise the entire aircraft's exterior fuselage surface and wings. Computers are used to configure the portion of the aircraft surface needed for some sensor function. The computer also handles sensor function prioritization and deinterleaving of conflicting transmissions.

Aircraft Sensors. Sensors, (the eyes and ears) of aircraft, are electronic devices for measuring external and internal environmental conditions. Sensors on aircraft include devices for sending and receiving RF energy. These types of sensors include radar, radio, and warning receivers. Another group of sensors are the infrared (*IR*) sensors, which include lasers and heat-sensitive sensors. Sensors are also used to measure direct analog inputs; altimeters and airspeed indicators are examples. Many of the sensors used on aircraft have their own built-in computers for serving their own functional requirements such as data preprocessing, filtering, and analysis. Sensors can also be part of a computer interface suite that provides key aircraft computers with the direct environmental inputs they need to function.

Aircraft Radar. Radar (radio detection and ranging) is a sensor that transmits RF energy to detect air and ground objects and determines parameters such as the range, velocity, and direction of these objects. The aircraft radar serves as its primary sensor. Several services are provided by modern aircraft radar. These include tracking, mapping, scanning, and identification. Golden (5) states that radar is tasked either to detect the presence of a target or to determine its location. Depending on the function emphasized, a radar system might be classified as a search or a tracking radar.

Stimson (6) describes the decibel (named after Alexander Graham Bell) as one of the most widely used terms in the design and description of radar systems. The decibel (dB) is a logarithmic unit originally devised to express power ratios, but also used to express a variety of other ratios. The Power ratio in dB is expressed as $10 \log_{10} P_2/P_1$, where P_2 and P_1 are the power levels being compared. Expressed in terms of voltage the gain is $(V_2/V_1)^2$ dB provided the input voltage V_1 and output voltage V_2 are across equal resistances.

Stimson (6) also explains the concept of the pulse repetition frequency (*PRF*), which is the rate at which a radar system's pulses are transmitted: the number of pulses per second. The interpulse period T of a radar is given by $T = 1/\text{PRF}$. For a PRF of 100 Hz, the interpulse period would be 0.01 s.

The Doppler effect, as described by Stimson (6), is a shift in the frequency of a radiated wave, reflected or received by an object in motion. By sensing Doppler frequencies, radar not only can measure range rates, but can also separate target echoes from clutter, or can produce high-resolution ground maps. Computers are required by an aircraft radar to make numerous and timely calculations with the received radar data, and to configure the radar to meet the aircrew's needs.

Aircraft Data Fusion. Data fusion is a method for integrating data from multiple sources in order to give a comprehensive solution to a problem (multiple input, single output). For aircraft computers, data fusion specifically deals with integrating data from multiple sensors such as radar and infrared sensors. For example, in ground mapping, radar gives good surface parameters, while the infrared sensor provides the height and size of items in the surface area being investigated. The aircraft computer takes the best inputs from each sensor, provides a common reference frame to integrate these inputs, and returns a more comprehensive solution than either single sensor could have given.

Aircraft Navigation. Navigation is the science of determining present location, desired location, obstacles between these locations, and best courses to take to reach these locations. An interesting pioneer of aircraft navigation was James Harold Doolittle (1886–1993). Best known for his aircraft-carrier-based bomber raid on Tokyo in World War II. General Doolittle received his master's and doctor of science degrees in aeronautics from Massachusetts Institute of Technology, where he developed instrumental *blind flying* in 1929. He made navigation history by taking off, flying a set course, and landing without seeing the ground. For a modern aircraft, with continuous changes in altitude, airspeed, and course, navigation is a challenge. Aircraft computers help meet this challenge by processing the multiple inputs and suggesting aircrew actions to maintain course, avoid collision and weather, conserve fuel, and suggest alternative flight solutions.

An important development in aircraft navigation is the Kalman filter. Welch and Bishop (7) state that in 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) implementation of the least-squares method. The filter is very powerful in several aspects: it supports estimation of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.

The Global Positioning System (*GPS*) is a satellite reference system that uses multiple satellite inputs to determine location. Many modern systems, including aircraft, are equipped with GPS receivers, which allow the system access to the network of GPS satellites and the GPS services. Depending on the quality and privileges of the GPS receiver, the system can have an instantaneous input of its current location, course, and speed within centimeters of accuracy. GPS receivers, another type of aircraft computer, can also be programmed to inform aircrews of services related to their flight plan.

Before the GPS receiver, the inertial navigation systems (INS) was the primary navigation system on aircraft. Fink and Christiansen (8) describe inertial navigation as the most widely used "self-contained" technology. In the case of an aircraft, the INS is contained within the aircraft, and is not dependent on outside inputs. Accelerometers constantly sense the vehicle's movements and convert them, by double integration, into distance traveled. To reduce errors caused by vehicle attitude, the accelerometers are mounted on a gyroscopically controlled stable platform.

Aircraft Communications. Communication technologies on aircraft are predominately radio communication. This technology allows aircrews to communicate with ground controllers and other aircraft. Aircraft computers help establish, secure, and amplify these important communication channels.

Aircraft Displays. Displays are visual monitors in aircraft that present desired data to aircrews and passengers. Adam and Gibson (9) illustrate F-15E displays used in the Gulf War. These illustrations show heads-up displays (*HUD*s), vertical situation displays, radar warning receivers, and low-altitude navigation and targeting system (*Lantirn*) displays typical of modern fighter aircraft. Sweet (10) illustrates the displays of a Boeing 777, showing the digital bus interface to the flight-deck panels and an optical-fiber data distribution interface that meets industry standards.

Aircraft Instrumentation. Instrumentation of an aircraft means installing data collection and analysis equipment to collect information about the aircraft's performance. Instrumentation equipment includes various recorders for collecting real-time flight parameters such as position and airspeed. Instruments also capture flight control inputs, environmental parameters, and any anomalies encountered in flight test or in routine flight. One method of overcoming this limitation is to link flight instruments to ground recording systems, which are not limited in their data recording capacities. A key issue here is the bandwidth between the aircraft being tested and its ground (recording) station. This bandwidth is limited and places important limitations on what can be recorded. This type of data link is also limited to the range of the link, limiting the aircraft's range and altitude during this type of flight test. Aircraft computers are used both in processing the data as they are being collected on the aircraft and in analyzing the data after they have been collected.

Aircraft Embedded Information Systems. *Embedded information system* is the latest terminology for an embedded computer system. The software of the embedded computer system is now referred to as embedded information. The purpose of the aircraft embedded information system is to process flight inputs (such as sensor and flight control) into usable flight information for further flight-system or aircrew utilization. The embedded information system is a good example of the merging of two camps of computer science applications. The first, and larger, camp is the management of information systems (*MIS*). The MIS dealt primarily with large volumes of information, with primary applications in business and banking. The timing requirements of processing these large information records are measured in minutes or hours. The second camp is the real-time embedded computer camp, which was concerned with processing a much smaller set of data, but in a very timely fashion. The real-time camp's timing requirement is in microseconds. These camps are now merging, because their requirements are converging. MIS increasingly needs real-time performance, while real-time systems are required to handle increased data-processing workloads. The embedded information system addresses both needs.

Aircraft and the Year 2000. The year 2000 (*Y2K*) has been a major concern for the aircraft computer industry. Many of the embedded computers on aircraft and aircraft support functions are vulnerable to Y2K faults, because of their age. The basic problem with these computers has been that a year is represented by its low-order two digits. Instead of the year having four digits, these computers saved processing power by using the last two digits of the calendar year. For example, 1999 is represented as 99. This is not a problem until you reach the year 2000, represented as 00. Even with this representation, problems are limited to those algorithms sensitive to calendar dates. An obvious problem is when an algorithm divides by the calendar date, which is division by 0. Division by 0 is an illegal computer operation, causing problems such as infinite loops, execution termination, and system failure. The most commonly mentioned issue is the subtraction of dates to determine time durations and to compare dates. There problem is not that the computer programs fail in a very obvious way (e.g., divide-by-zero check) but, rather that the program computes an incorrect result without any warning or indication of error. Lefkon and Payne (11) discuss Y2K and how to make embedded computers compliant.

Aircraft Application Program Interfaces. An application programming interface (*API*) is conventionally defined as an interface used by one program to make use of the services of another program. The human interface to a system is usually referred to as the user interface, or, less commonly, the human–computer interface. Application programs are software written to solve specific problems. For example, the embedded computer software that paints the artificial horizon on a heads-up display is an application program. A switch that turns the artificial horizon on or off is an API. Gal-Oz and Isaacs (12) discuss APIs and how to relieve bottlenecks of software debugging.

Aircraft Control. Landau (1) defines, a control as an instrument or apparatus used to regulate a mechanism, or a device used to adjust or control a system. There are two concepts with control. One is the act of control. The other is the type of device used to enact control. An example of an act of control is when a pilot initiates changes to throttle and stick settings to alter flight path. The devices of control, in this case, are the throttle and stick.

Control can be active or passive. Active control is force-sensitive. Passive control is displacement-sensitive.

Mechanical control is the use of mechanical devices, such as levers or cams, to regulate a system. The earliest form of mechanical flight control was wires or cables, used to activate ailerons and stabilizers through pilot stick and foot pedal movements. Today, hydraulic control, the use of fluids for activation, is usual. Aircraft control surfaces are connected to stick and foot pedals through hydraulic lines. Pistons in the control surfaces are pushed or pulled by associated similar pistons in the stick or foot pedal. The control surfaces move accordingly.

Electronic control is the use of electronic devices, such as motors or relays, to regulate a system. A motor is turned on by a switch, and quickly changes control surfaces by pulling or pushing a lever on the surface. Automatic control is a system-initiated control, which is a system-initiated response to a known set of environmental conditions. Automatic control was used for early versions of automatic pilot systems, which tied flight-control feedback systems to altitude and direction indicators. The pilot sets his desired course and altitude, which is maintained through the flight control's automatic feedback system.

To understand the need for computers in these control techniques, it is important to note the progression of the complexity of the techniques. The earliest techniques connected the pilot directly to his control surfaces.

As the aircraft functionality increased, the pilot's workload also increased, requiring his (or his aircrew's) being free to perform other duties. Additionally, flight characteristics became more complex, requiring more frequent and instantaneous control adjustments. The use of computers helped offset and balance the increased workload in aircraft. The application of computers to flight control provides a means for processing and responding to multiple complex flight control requirements.

Aircraft Computer Hardware. For aircraft computers, hardware includes the processors, buses, and peripheral devices inputting to and outputting from the computers. Landau (1) defines hardware as apparatus used for controlling spacecraft; the mechanical, magnetic, and electronic design, structure, and devices of a computer; and the electronic or mechanical equipment that uses cassettes, disks, etc. The computers used on an aircraft are called *processors*. The processor takes inputs from peripheral devices and provides specific computational services for the aircraft.

There are many types and functions of processors on aircraft. The most obvious processor is the central computer, also called the mission computer. The central computer provides direct control and display to the aircrew. The federated architecture (discussed in more detail later) is based on the central computer directing the scheduling and tasking of all the aircraft subsystems. Other noteworthy computers are the data-processing and signal-processing computers of the radar subsystem and the computer of the inertial navigation system. Processors are in almost every component of the aircraft. Through the use of an embedded processor, isolated components can perform independent functions as well as self-diagnostics.

Distributed processors offer improved aircraft performance and, in some cases, redundant processing capability. *Parallel* processors are two or more processors configured to increase processing power by sharing tasks. The workload of the shared processing activity is distributed amongst the pooled processors to decrease the time it takes to form solutions. Usually, one of the processors acts as the lead processor, or master, while the other processor(s) act as slave(s). The master processor schedules the tasking and integrates the final results. On aircraft, this is particularly useful in that processors are distributed throughout the aircraft. Some of these computers can be configured to be parallel processors, offering improved performance and redundancy. Aircraft system redundancy is important, because it allows distributed parallel processors to be reconfigured when there is a system failure. Reconfigurable computers are processors that can be reprogrammed to perform different functions and activities. Before computers, it was very difficult to modify systems to adapt to their changing requirements. A reconfigurable computer can be dynamically reprogrammed to handle a critical situation, and than returned to its original configuration.

Aircraft Buses. Buses are links between computers (processors), sensors, and related subsystems, for transferring data inputs and outputs. Fink and Christiansen (8) describe two primary buses as data buses and address buses. To complete the function of an MPU, a microprocessor must access memory and peripheral devices. This is accomplished by placing data on a bus, either an address bus or a data bus, depending upon the function of the operation. The standard 16-bit microprocessor requires a 16-line parallel bus for each function. An alternative is to multiplex the address or data bus to reduce the number of pin connections. Common buses in aircraft are the Military Standard 1553 Bus (Mil-Std-1553) and the General-Purpose Interface Bus (GPIB), which is the IEEE Standard 488 Bus.

Aircraft Software. Landau (1) defines software as the programs, routines, etc. for a computer. The advent of software has provided great flexibility and adaptability to almost every aspect of life. This is especially true in all areas of aerospace sciences, where flight control, flight safety, in-flight entertainment, navigation, and communications are continuously being improved by software upgrades.

Operation Flight Programs. An operational flight program (*OFP*) is the software of an aircraft embedded computer system. An OFP is associated with an aircraft's primary flight processors, including the central computer, vertical and multiple display processors, data processors, signal processors, and warning receivers. Many OFPs in use today require dedicated software integrated support environments to upgrade and maintain them as the mission requirements of their parent aircraft are modified. The software integrated support environment [also called avionics integrated support environment (*AISE*), centralized software support activity

(*CSSA*), and software integration laboratory (*SIL*)] not only allows an OFP to be updated and maintained, but also provides capabilities to perform unit testing, subsystem testing, and some of the integrated system testing.

Assembly Language. Assembly language is a machine (processor) language that represents inputs and outputs as digital data and that enables the machine to perform operations with those data. For a good understanding of the Motorola 6800 Assembler Language, refer to Bishop (13). According to Seidman and Flores (14) the *lowest-level* (closest to machine) language available to most computers is assembly language. When one writes a program in assembly code, alphanumeric characters are used instead of binary code. A special program called an assembler (provided with the machine) is designed to take the assembly statements and convert them to machine code. Assembly language is unique among programming languages in its one-to-one correspondence between the machine code statements produced by the assembler and the original assembly statements. In general, each line of assembly code assembles into one machine statement.

Higher-Order Languages. Higher-order languages (HOLs) are computer languages that facilitate human language structures to perform machine-level functions. Seidman and Flores (14) discuss the level of discourse of a programming language as its distance from the underlying properties of the machine on which it is implemented. A low-level language is close to the machine, and hence provides access to its facilities almost directly; a high-level language is far from the machine, and hence insulated from the machine's peculiarities. A language may provide both high-level and low-level constructs. Weakly typed languages are usually high-level, but often provide some way of calling low-level subroutines. Strongly typed languages are always high-level, and they provide means for defining entities that more closely match the real-world objects being modeled. Fortran is a low-level language that can be made to function as high-level by use of subroutines designed for the application. APL, Sobol, and SETL (a set-theoretic language) are high-level languages with fundamental data types that pervade their language. Pascal, Cobol, C, and PL/I are all relatively low-level languages, in which the correspondence between a program and the computations it causes to be executed is fairly obvious. Ada is an interesting example of a language with both low-level and high-level properties. Ada provides quite explicit mechanisms for specifying the layout of data structures in storage, for accessing particular machine locations, and even for communicating with machine interrupt routines, thus facilitating low-level requirements. Ada's strong typing qualities, however, also qualify it as a high-level language.

High-level languages have far more expressive power than low-level languages, and the modes of expression are well integrated into the language. One can write quite short programs that accomplish very complex operations. Gonzalez (15) developed an *Ada Programmer's Handbook* that presents the terminology of the HOL Ada and examples of its use. He also highlights some of the common programmer errors and examples of those errors. Sodhi (16) discusses the advantages of using Ada. Important discussions of software life-cycle engineering and maintenance are presented, and the concept of configuration management is presented.

The package concept is one of the most important developments to be found in modern programming languages, such as Ada, Modula-2, Turbo Pascal, C++, and Eiffel. The designers of the different languages have not agreed on what terms to use for this concept: package, module, unit, and class are commonly used. But it is generally agreed that the package (as in Ada) is the essential programming tool to be used for going beyond the programming of very simple class exercises to what is generally called software engineering, or building production systems. Packages and package like mechanisms are important tools used in software engineering to produce production systems. Feldman (17) illustrates the use of Ada packages to solve problems.

Databases. Database are essential adjuncts to computer programming. Databases allow aircraft computer applications the ability to carry pertinent information (such as flight plans or navigation waypoints) into their missions, rather than generating them in route. Databases also allow the aircrew to collect performance information about the aircraft's various subsystems, providing a capability to adjust the aircraft in flight and avoid system failures.

Elmasri and Navathe (18) define a database as a collection of related data. Data are described as known facts that can be recorded and have implicit meaning. (A simple example consists of the names, telephone numbers, and addresses of an indexed address book. A database management system (*DBMS*) is a collection

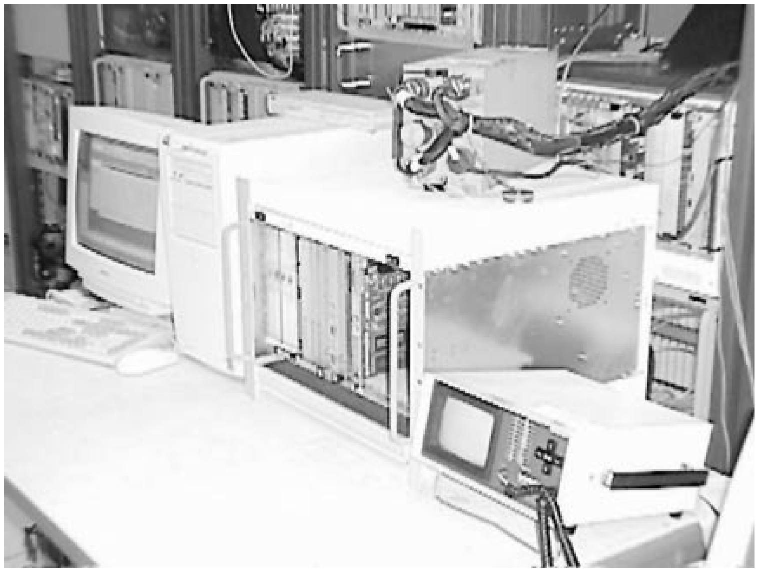

Fig. 2. An aircraft avionics support bench.

of programs that enable users to create and maintain a database. The DBMS is hence a general-purpose software system that facilitates the processes of defining, constructing, and manipulating databases for various applications.

Verification and Validation. A significant portion of the aircraft computer's life-cycle cost is system and software testing, performed in various combinations of unit-level, subsystem-level, integrated-system-level, developmental, and operational testing. These types of tests occur frequently throughout the life of an aircraft system because there are frequent upgrades and modifications to the aircraft and its various subsystems. It is possible to isolate acceptance testing to particular subsystems when minor changes are made, but this is the exception. Usually, any change made to a subsystem affects other multiple parts of the system. As aircraft become increasingly dependent on computers (which add complexity by the nature of their interdependences), and as their subsystems become increasingly integrated, the impact of change also increases drastically. Cook (19) shows that a promising technology to help understand the impact of aircraft computer change is the Advanced Avionics Verification and Validation (*AAV&V*) program developed by the Air Force Research Laboratory.

Sommerville (20) develops the concepts of program verification and validation. Verification involves checking that the program conforms to its specification. Validation involves checking that the program as implemented meets the expectations of the user.

Figure 2 shows an aircraft avionics support bench, which includes real components from the aircraft such as the *FCC* line replaceable unit (*LRU*) sitting on top of the pictured equipment. Additional equipment includes the buses, cooling, and power connection interfaces, along with monitoring and displays. On these types of benches, it is common to emulate system and subsystem responses with testing computers such as the single-board computers illustrated.

Figure 3 shows another verification and validation asset called the workstation-based support environment. This environment allows an integrated view of the aircraft's performance by providing simulations of the aircraft's controls and displays on computer workstations. The simulation is interfaced with stick and throttle controls, vertical situation displays, and touch-screen avionics switch panels.

Fig. 3. A workstation-based aircraft avionics support environment.

Object-Oriented Technology. Object-oriented (*OO*) technology is one of the most popular computer topics of the 1990s. OO languages such as C++ and Ada 95 offer tremendous opportunities to capture complex representations of data and then save these representations in reusable objects. Instead of using several variables and interactions to describe some item or event, this same item or event is described as an object. The object contains its variables, control-flow representations, and data-flow representations. The object is a separable program unit, which can be reused, reengineered, and archived as a program unit. The power of this type of programming is that when large libraries of OO programming units are created, they can be called upon to greatly reduce the workload of computer software programming. Gabel (21) says that object-oriented technology lets an object (a software entity consisting of the data for an action and the associated action) be reused in different parts of the application, much as an engineered hardware product can use a standard type of resistor or microprocessor. Elmasri and Navathe (18) describe an object-oriented database as an approach with the flexibility to handle complex requirements without being limited by the data types and query languages available in traditional database systems.

Open System Architecture. Open system architecture is a design methodology that keeps options for updating systems open by providing liberal interfacing standards. Ralston and Reilly (22) state that open architectures pertain primarily to personal computers. An open architecture is one that allows the installation of additional logic cards in the computer chassis beyond those used with the most primitive configuration of the system. The cards are inserted into slots in the computer's motherboard—the main logic board that holds its CPU and memory chips. A computer vendor who adopts such a design knows that, since the characteristics of the motherboard will be public knowledge, other vendors who wish to do so can design and market customized logic cards. Open system architectures are increasingly important in modern aircraft applications, because of the constant need to upgrade these systems and utilize the latest technical innovations. It is extremely difficult to predict interconnection and growth requirements for next-generation aircraft. This is exactly what an open architecture attempts to avoid the need for.

Client–Server Systems. A client–server system is one in which one computer provides services to another computer on a network. Ralston and Reilly (22) describe the file-server approached as an example of client-server interaction. Clients executing on the local machine forward all file requests (e.g. open, close, read, write, and seek) to the remote file server. The server accepts a client's requests, performs its associated operation, and returns a response to the client. Indeed, if the client software is structured transparently, the client need not even be aware that files being accessed physically reside on machines located elsewhere on the network. Client–server systems are being applied on modern aircraft, where highly distributed resources and their aircrew and passenger services are networked to application computers.

Subsystems. The major subsystems of an aircraft are its airframe, power plant, avionics, landing gear, and controls. Landau (1) defines a subsystem as any system that is part of a larger system. Many of the subsystems on an aircraft have one or more processors associated with them. It is a complex task to isolate and test the assorted subsystems.

Another layer of testing below subsystem testing is *unit testing*. A *unit* of a subsystem performs a function for it. For example, in the radar subsystem, the units include its signal processor and its data processor. In order to test a system adequately, each of its lowest-level items (units) must be tested. As the units affect and depend upon each other, another layer of testing addresses that layer of dependences. In the same fashion, subsystem testing is performed and integrated with associated subsystems. It is important to test not only at the unit and the subsystem level, but at the system and operational level. The system level is where the subsystems are brought together to offer the system functionality. *System integration* is the process of connecting subsystem components into greater levels of system functionality until the complete system is realized. The *operational level* of testing is where the subsystem is exercised in its actual use.

Line Replaceable Units. LRUs are subsystems or subsystem components that are self-contained in durable boxes containing interface connections for data, control, and power. Many LRUs also contain built-in test (*BIT*) capabilities that notify air and maintenance crew when there is a failure. A powerful feature of LRUs is that functionality can be compartmentalized. When a failure is detected, the LRU can easily be pulled and replaced, restoring the aircraft to service within moments of detection.

Graceful Degradation. All systems must have plans to address partial or catastrophic failure. System failure in flight controls is often catastrophic, while system failure in avionics can be recovered from. For this reason, most flight-critical systems have built-in redundant capabilities (sometimes multiple layers of redundancy), which are automatically activated when the main system or subsystem fails. Degraded system behavior occurs when the main system fails and backup systems are activated. The critical nature of system failure requires immediate activation of backup systems and recognition by all related subsystem of the new state of operation. *Graceful degradation* is the capability of aircraft computers to continue operating after incurring system failure. Graceful degradation is less than optimal performance, and may activate several layers of decreasing performance before the system fails. The value of graceful degradation is that the aircrew has time to respond to the system failure before there is a catastrophic failure.

Aerospace

Computer technologies have helped provide a continuum of improvements in aircraft performance that has allowed the airspace where aircraft operate to increase in range and altitude. Landau (1) defines aerospace as the earth's atmosphere and the space outside it, considered as one continuous field. Because of its rapidly increasing domain of air and space travel, the United States Air Force is beginning to refer to itself as the United Sates Aerospace Force. Modern air–space vehicles are becoming increasingly dependent on information gleaned from ground stations, satellites, other air–space vehicles, and onboard sensors to perform their mission. These vehicles use signals across the electromagnetic spectrum. Antennas can be found in multiple locations on wings, the fuselage, tails, and draglines. If antennas are located too close together, their signals can interfere with each other; this is called crossed frequency transmission. This interference reduces the efficiency of each affected antenna. Placement of multiple antennas requires minimizing the effects of crossed frequency transmissions. Techniques for this include antenna placement, filtering, and timing. This presents another challenge for aircraft computers to sort and process these multiple signals. Perry and Geppert (4) show how the aircraft electromagnetic spectrum is becoming busy, and thus, dangerous for aerospace communications.

Legacy Systems. Legacy systems are fielded aircraft, or aircraft that are in active use. Probably the only nonlegacy aircraft are experimental or prototype versions. Legacy aircraft are often associated with aging issues, more commonly known as parts obsolescence. A growing problem in these systems is the obsolescence of entire components, including the many computers used on them. Aircraft, like many other systems, are designed with expected lifetimes of 10 to 15 years. Because of the high replacement costs, lifetimes are often doubled and tripled by rebuilding and updating the aircraft. To reduce costs as many as possible of the original aircraft components are kept. Problems arise when these components are no longer produced or stockpiled. Sometimes subsystems and their interfaces have to be completely redesigned and produced at great cost in order to keep an aircraft in service. System architectures and standard interfaces are constantly being modified to address these issues. Aircraft evolve during their lifetimes to a more open architecture. This open architecture, in turn, allows the aircraft components to be more easily replaced, thus making further evolution less expensive.

Unmanned Air Vehicles. Unmanned air vehicles (*UAV*s) are aircraft that are flown without aircrews. Their use is becoming increasingly popular for military applications. Many of the new capabilities of UAVs come from the improved computers. These computers allow the vehicles to have increased levels of autonomy and to perform missions that once required piloted aircraft. Some of these missions include reconnaissance and surveillance. These same types of missions are finding increasing commercial importance. UAVs offer tremendous advantages in life-cycle cost reductions because of their small size, ease of operation, and ability to be adapted to missions.

Man–Machine Systems

An aircraft is an example of a man–machine system. Other examples are automobiles and boats. These machines have the common attribute of being driven by a human. Landau (1) defines man–machine systems as sets of manually performed and machine-performed functions, operated in conjunction to perform an operation. The aircraft computer is constantly changing the role of the human in the aircraft machine. The earliest aircraft required the constant attention of the pilot. Improved flight control devices allowed the pilot freedom for leisure or for other tasks. Modern aircraft computers have continued the trend of making the aircraft more the machine, and less the man system.

Human Factors of Aircraft Computers. *Human factors* is the science of optimal conditions for human comfort and health in the human environment. The human factors of aircraft computers include the positioning of the controls and displays associated with the aircrew's workloads. They also provide monitoring and adjustment of the aircraft human environment, including temperature, oxygen level, and cabin pressure.

Man–Machine Interface. The *man–machine interface* is the place where man's interactions with the aircraft coordinate with the machine functionality of the aircraft. An example of a man–machine interface is the API, which is where a person provides inputs to and receives outputs from computers. These types of interfaces include keyboards (with standard ASCII character representation), mouse pads, dials, switches, and many varieties of monitors. A significant interface in aircraft comprises their associated controls and displays, which provide access to the flight controls, the sensor suite, the environmental conditions, and the aircraft diagnostics through the aircraft's central computer. Control sticks, buttons, switches, and displays are designed based on human standards and requirements such as seat height, lighting, accessibility, and ease of use.

Voice–Activated Systems. Voice–activated systems are interfaces to aircraft controls that recognize and respond to aircrew's verbal instructions. A voice-activated input provides multiple input possibilities beyond the limited capabilities of hands and feet. Voice–activated systems have specifed sets of word commands, and are trained to recognize a specific operator's voice.

Aircraft Computer Visual Verification. Visual verification is the process of physically verifying (through sight) the correct aircraft response to environmental stimuli. This visual verification is often a testing requirement. It is usually done through the acceptance test procedure (*ATP*) and visual inspections of displays through a checklist of system and subsystem inputs. Until recently, visual verification has been a requirement for pilots, who have desired the capability to see every possibility that their aircraft might encounter. This requirement is becoming increasingly difficult to implement, because of the growing complexity and workload of the aircraft's computers and their associated controls and displays. In the late 1980s to early 1990s, it used to take about 2 weeks to visually verify the suite of an advanced fighter system's avionics. This can no longer be accomplished at all with current verification and validation techniques. Several months would be required to achieve some level of confidence that today's modern fighters are flight-safe.

Air Traffic Control. Air traffic control is the profession of monitoring and controlling aircraft traffic through an interconnected ground–based communication and radar system. Perry (23) describes the present capabilities and problems in air traffic control. He also discusses the future requirements for this very necessary public service. Air traffic controllers view sophisticated displays, which track multiple aircraft variables such as position, altitude, velocity, and heading. Air traffic control computers review these variables and give the controllers continuous knowledge of the status of each aircraft. These computers continuously update and display the aircraft in the ground–based radar range. When potential emergency situations, such as collision, arise, the computer highlights the involved aircraft on the displays, with plenty of lead time for the controller to correct each aircraft's position.

Aircraft Control And Computers

D' Azzo and Houpis (24) give a good explanation of the complexity of what is needed for an aircraft control system. The feedback control system used to keep an airplane on a predetermined course or heading is necessary for the navigation of commercial airliners. Despite poor weather conditions and lack of visibility, the airplane must maintain a specified heading and altitude in order to reach its destination safely. In addition, in spite of rough air, the trip must be made as smooth and comfortable as possible for the passengers and crew. The problem is considerably complicated by the fact that the airplane has six degrees of freedom. This makes control more difficult than control of a ship, whose motion is limited to the surface of the water.

A flight controller is used to control aircraft motion. Two typical signals to the system are the correct flight path, which is set by the pilot, and the level position of the airplane. The ultimately controlled variable is the actual course and position of the airplane. The output of the control system, the controlled variable, is the aircraft heading.

In conventional aircraft there are three primary control surfaces used to control the physical three-dimensional attitude of the airplane: the elevators, rudder, and ailerons. A directional gyroscope is used as the error-measuring device. Two gyros must be used to provide control of both heading and attitude of the airplane. The error that appears in the gyro as an angular displacement between the rotor and case is translated into a voltage by various methods, including the use of transducers such as potentiometers, synchros, transformers, or microsyns. Selection of the method used depends on the preference of the gyro manufacturer and the sensitivity required. Additional stabilization for the aircraft can be provided in the control system by rate feedback. In other words, in addition to the primary feedback, which is the position of the airplane, another signal proportional to the angular rate of rotation of the airplane around the vertical axis is fed back in order to achieve a stable response. A *rate gyro* is used to supply this signal. This additional stabilization may be absolutely necessary for some of the newer high-speed aircraft.

In reading through this example, it should be obvious that as the complexity of the control feedback system of the aircraft increases, there is a need for computer processing to evaluate the feedback and to adjust or recommend flight control adjustments. Additional feedback may come from global positioning, from ground-based navigation systems through radio inputs, and from other aircraft. The computer is able to integrate these inputs into the onboard flight control inputs, and provide improved recommendations for stable flight.

Real-Time Systems

The computers on aircraft are required to perform their functions within short times. Flight control systems must make fine adjustments quickly, in order to maintain stable flight. Sensor suites must detect and analyze potential threats before it is too late. Cabin pressure and oxygen must be regulated as altitude changes. All these activities, plus many others on aircraft, must happen in real time.

Nielsen (25) defines a real-time system as a controlled (by software or firmware) system that performs all of its process functions within specified time constraints. A real-time system usually includes a set of independent hardware devices that operate at widely differing speeds. These devices must be controlled so that the system as a whole is not dependent upon the speed of the slowest device. Hatley and Pirbhai (26) describe timing as one of the most critical aspects of modern real-time systems. Often, the system's response must occur within milliseconds of a given input event, and every second it must respond to many such events in many different ways.

Flight-Critical Systems. *Flight-critical systems* are those activities of an aircraft that must be completed without error in order to maintain life and flight. The aircraft flight controls, engines, landing gear, and cabin environment are examples of flight-critical systems. Failures in any of these systems can have catastrophic results. Flight-critical systems are held to tight levels of performance expectations, and often have redundant backups in case of failure.

Federated Systems. *Federated systems* are loosely coupled distributed systems frequently used in aircraft system architectures to tie multiple processors in multiple subsystems together. The loose coupling allows the multiple subsystems to operate somewhat autonomously, but have the advantage of the shared resources of the other subsystems. A typical aircraft federated system might include its central computer, its INS, its radar system, and its air-vehicle management system. The INS provides the radar with the aircraft's present position, which is reported to the pilot through displays put forth by the central computer. The pilot adjusts his course through the air-vehicle management system, which is updated by the INS, and the cycle is repeated. These subsystems perform their individual functionality while providing services to each other.

Cyclic Executive. A *cyclic executive* on an aircraft computer provides a means to schedule and prioritize all the functions of the computer. The executive routine assigns the functions and operations to be performed by the computer. These assignments are given a specific amount of clock time to be performed. If the assignment does not complete its task in its allocated time, it is held in a wait state until its next clock period. From the beginning of the clock period to its end is one clock cycle. High-priority functions are assigned faster clock cycles, while low-priority functions are assigned slower cycles. For example, the high-priority executive function might be assigned a speed of 100 cycles per second, while some lower-priority function might have 5 cycles per second to complete its tasks. Sometimes the latter might take several clock cycles to perform a task. An additional feature of cyclic executives is that they are equipped with interrupts, which allow higher-priority systems to break into the executive assignments for system-level assigned tasking.

There are several types of scheduling methodologies that provide performance improvements in cyclic executives. One of the more prominent is rate monotonic analysis (*RMA*), which determines the time requirement for each function and the spare time slots, and then makes time assignments.

BIBLIOGRAPHY

1. S. Landou *Webster Illustrated Contemporary Dictionary, Encyclopedic Edition*, Chicago: J. G. Ferguson, 1992.
2. J. F. Wakerly, *Digital Design Principles and Practices*, Englewood Cliffs, NJ: Prentice-Hall, 1985, pp. 1–48, 53–138.
3. V. C. Hamacher Z. G. Vranesic S. G. Zaky *Computer Organization*, 2nd ed., New York: McGraw-Hill, 1984.
4. T. Perry L. Geppert Do portable electronics endanger flight, *IEEE Spectrum*, **33** (9): 26–33, 1996.
5. A. Golden *Radar Electronic Warfare*, Washington: AIAA Education Series, 1987.
6. G. W. Stimson *Introduction to Airborne Radar*, El Segundo, CA: Hughes Aircraft, 1983, pp. 107, 151–231.
7. G. Welch G. Bishop An introduction to the Kalman filter, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, http://www.cs.unc.edu/˜welch/media/pdf/kalman.pdf, 1997.
8. D. Fink D. Christiansen *Electronics Engineers' Handbook*, 3rd ed., New York: McGraw-Hill, 1989.
9. J. Adam T. Gibson Warfare in the information age, *IEEE Spectrum*, **28** (9): 26–42.
10. W. Sweet The glass cockpit, *IEEE Spectrum*, **32** (9): 30–38, 1995.
11. D. Lefkon B. Payne Making embedded systems year 2000 compliant, *IEEE Spectrum*, **35** (6): 74–79, 1998.
12. S. Gal-Oz M. Isaacs Automate the bottleneck in embedded system design, *IEEE Spectrum*, **35** (8): 62–67, 1998.
13. R. Bishop *Basic Microprocessors and the 6800*, Hasbrouck Heights, NJ: Hayden, 1979.
14. A. Seidman I. Flores *The Handbook of Computers and Computing*, New York: Van Norstrand Reinhold, 1984, pp. 327–502.
15. D. W. Gonzalez *Ada Programmer's Handbook*, Redwood City, CA: Benjamin/Cummings, 1991.
16. J. Sodhi *Managing Ada Projects*, Blue Ridge Summit, PA: TAB Books, 1990.
17. M. B. Feldman E. B. Koffman *Ada Problem Solving and Program Design*, Reading, MA: Addison-Wesley, 1992.
18. R. Elmasri S. B. Navathe *Fundamentals of Database Design*, 2nd ed., Redwood City, CA: Benjamin/Cummings, 1994.
19. R. Cook The advanced avionics verification and validation II final report, Air Force Research Laboratory Technical Report ASC-99-2078, Wright-Patterson AFB.
20. I. Sommerville *Software Engineering*, 3rd ed., Reading, MA: Addison-Wesley, 1989.
21. D. Gabel Software engineering, *IEEE Spectrum*, Vol. **31** (1): 38–41, 1994.
22. A. Ralston E. Reilly *Encyclopedia of Computer Science*, New York: Van Nostrand Reinhold, 1993.
23. T. Perry In search of the future of air traffic control, *IEEE Spectrum*, **34** (8): 18–35, 1997.
24. J. J. D' Azzo C. H. Houpis *Linear Control System Analysis and Design*, 2nd ed., New York: McGraw-Hill, 1981, pp. 143–146.
25. K. Nielsen *Ada in Distributed Real-Time Systems*, New York: Intertext, 1990.
26. D. J. Hatley I. A. Pirbhai *Strategies for Real-Time System Specification*, New York: Dorset House, 1988.

READING LIST

G. Buttazo *Hard Real-Time Computing Systems*, Norwell, MA: Kluwer, 1997.

R. Comerford PCs and workstations, *IEEE Spectrum*, **30**, (1): 26–29, 1993.

D. Dooling Aerospace and military, *IEEE Spectrum*, **35** (1): 90–94, 1998.

J. Juliussen D. Dooling Small computers, aerospace & military, *IEEE Spectrum*, **32** (1): 44–47, 76–79, 1995.

K. Kavi *Real-Time Systems, Abstractions, Languages, and Design Methodologies*, Los Alamitos, CA: IEEE Computer Society Press, 1992.

P. Laplante *Real-Time Systems Design and Analysis, an Engineer's Handbook*, Piscataway, NJ: IEEE Press, 1997.

M. S. Roden *Analog and Digital Communication Systems*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1985.

H. Taub *Digital Circuits and Microprocessors*, New York: McGraw-Hill, 1982.

C. Weitzman *Distributed Micro/Minicomputer*, Englewood Cliffs, NJ: Prentice-Hall, 1980.

CHARLES P. SATTERTHWAITE
Air Force Research Laboratory Embedded Information
System Engineering Branch (AFRL IFTA)