Architectural Overview of the SPEAKeasy System

Peter G. Cook and Wayne Bonser

***Abstract*— SPEAKeasy is a successful implementation of a software-defined radio (SDR) for military applications. It permits general-purpose digital hardware to communicate over a wide range of frequencies, modulation techniques, data encoding methods, cryptographic types, and other communication parameters. The background of SDR's for military and commercial needs is discussed, and the SPEAKeasy architecture is defined.**

***Index Terms*—Analog–digital conversion, CYPRIS, digital signal processors, joint tactical radio system, SDR Forum, software-defined radio (SDR), SPEAKeasy, waveform applications.**

I. Introduction

THIS paper describes the background of the SPEAKeasy system, and provides an overview of its architecture.

A. Background

Wireless communication using radio frequency links is a key element of increased mobility in military applications. Emerging technology has permitted increased reliability and functionality with reduced size, weight, and power.

With current levels of embedded computational power, the concept of digital signal processor (DSP)-based modems with software-defined capability has become a reality. Analog-to-digital converters (ADC's) are moving closer and closer to the antenna as conversion rates and the dynamic range increase. Similar advances in cryptography, multimedia processing, networking, and operator interfaces improve other aspects of the system. As digital processing elements take on more and more tasks previously accomplished in less capable programmable and single-purpose hardware devices, system flexibility is enormously increased. The potential of this technology was first described in the open literature as a result of the SPEAKeasy I program [1]. Software-defined radios (SDR's) offer a wide range of capabilities defined in software running on common hardware. This yields advantages over the less flexible implementations of previous technology. They allow for improvements or enhancements without altering the circuitry. SDR capabilities enable customers to acquire common hardware, and to satisfy individual requirements with software that fits their specific application, as is done in the personal computer marketplace [2].

In the late 1970's, the U.S. Air Force began work on a system known as the Integrated Communications, Navigation, Identification, and Avionics system (ICNIA). This was one of the first systems to use DSP-based programmable modem and control to obtain a fully integrated capability for airborne platforms rather. Prior technology required a set of independent federated hardware modules. ICNIA was based on three devices: a 25-MHz 6800-series low-latency processing element, a general-purpose processing element, and an enhanced security unit. The enhanced security unit interfaced to an external federated communications security (COMSEC) module. This technology laid the foundation for systems like SPEAKeasy and the Advanced Tactical Fighter's (F22) Communications, Navigation, and Identification system that evolved over the next two decades.

B. SPEAKeasy Phase-1

Building on ICNIA technology, in 1989, the Air Force initiated the Tactical Anti-Jam Programmable Signal Processor (TAJPSP) with the objective of developing a modular, reprogrammable modem. The military projected that it would need robust electronic counter-counter measures (ECCM) and low probability of intercept (LPI) waveforms that would necessarily evolve. An adaptable, software-defined platform, in the form of a programmable modem, appeared to be key to fielding successive waveforms as they became mature. The TAJPSP captured the attention of others to become the triservice programmable radio program called SPEAKeasy.

The primary objective of SPEAKeasy was to develop a modular, reprogrammable, modem with an open architecture. It was to demonstrate this SDR core on the air. A secondary objective was to develop a generic software architecture to facilitate the addition of new waveforms. SPEAKeasy exploited the ICNIA technology, but also adopted a more tactically acceptable security architecture and improved DSP—the TMS320-C40.[1] Technology examined under the initial program[2] was crucial to eventual military products. This included advanced designs for:

- fast fourier transforms (FFT's);
- ADC's with high-speed sampling and large dynamic range;
- a quad-DSP module (four TMS320-C40's);

Manuscript received November 10, 1997; revised March 12, 1998, June 3, 1998, and August 11, 1998. This work was supported by the U.S. Air Force Rome Laboratories under Contract F30602-95-C-0115.

P. G. Cook is with the Systems Solutions Group, Motorola, Inc., Scottsdale, AZ 85257 USA.

W. Bonser is with the Air Force Laboratories, USAF, Rome, NY 13441-4505 USA.

Publisher Item Identifier S 0733-8716(99)02977-7.

[1] The TMS320-C40 (Texas Instruments: www.ti.com), 1998.

[2] The initial SPEAKeasy project has been reported in the literature as SPEAKeasy I; its formal designation at this time is SPEAKeasy Phase 1.

- RF up- and downconversion with wide instantaneous bandwidth;
- a programmable information security (INFOSEC) module architecture based on a 40 MHz reduced instruction set computer (RISC) known as CYPRIS.

The Phase-1 program conducted from 1992 to 1995 was to demonstrate a four-channel, wide-band architecture for high-speed frequency-hopped and pseudorandom spread-spectrum waveforms. It was to operate over the military HF, VHF, and UHF frequency bands. RF studies established that the RF operating range from 2 MHz to 2 GHz could not be implemented in a single wide-band RF channel. The RF section was therefore based on available technology using three RF bands. The low band ranged from 2 to 30 MHz; the midband ranged from 30 to 400 MHz, and the high band ranged from 400 MHz to 2 GHz. Only the midband was implemented in the Phase-1 feasibility demonstration. The hardware included a VME-chassis in a 6 ft rack with the VME bus for control and a unique high-speed ring bus for data. The user interface included a Sun SPARC workstation to download script files into a terminal controller unit. The Phase-1 equipment did not have sufficient FFT-based processing capacity to demonstrate the wide-band waveforms. The system was demonstrated on the air in June 1995 at Hanscom Air Force Base, MA, during the Joint Warrior Interoperability Demonstration (JWID-95). The primary objective, the development of the modem, was achieved. In addition, the Phase-1 software performed well enough that it was released on request to other agencies. Unfortunately, the modem software, user interface, and waveform development environment did not achieve the objective of ease of use.

C. Current Deficiencies and Mission Needs

During the early 1990's, the Air Force evolved its approach to command, control, communications, and computer (C4) systems to support operations that integrate multiple military services (e.g., Army, Navy, and Air Force, called "joint" operations[3]). These documents outline the basic military needs that the SPEAKeasy program addresses.

The military employs hundreds of communications systems that were developed when the services executed missions more or less autonomously. Many of these systems were never meant to work together ("interoperate"). Thus, they currently preclude the military from achieving the seamless communications it needs for an integrated battlefield. Law enforcement, public service, and emergency service organizations also are disadvantaged due to the proliferation of noninteroperable radios at federal, state, and local levels [4]. Interoperability can be increased by SDR's capable of bridging these incompatible systems. SDR's also permit the incremental deployment of new capabilities.

The dramatic pace of advances in communications technology, coupled with the military's traditionally long systems acquisition cycles have resulted in the technological obsolescence of new systems before they are fielded. Costs have prohibited retrofitting old systems with improved capability, resulting in reduced military readiness. Current radio systems cannot be technologically updated cost effectively. Software radios provide the opportunity for "future proofing" via preplanned-product improvement.

The U.S. Defense Department's emphasis on reducing life-cycle costs and the shift toward an integrated commercial and military national industrial base motivated technical change [5], [6]. For the military, postdeployment costs are by far the largest portion of a system's total cost [7]. These concerns are not unique to the military, and have resulted in the formation of the Software Defined Radio (SDR) Forum, formerly the Modular Multifunction Information Transfer System (MMITS) Forum. The SDR Forum includes among its members the SPEAKeasy team, The MITRE Corporation, the Federal Aviation Administration (FAA), and BellSouth [8]. It was created to facilitate the adoption of radio standards for software-defined communications systems with open system architectures. Such systems are required by military communicators and commercial wireless service providers [9]. Today, the SDR Forum is an international not-for-profit corporation that includes members from academia, the military, component manufacturers, wireless service providers, and regulatory bodies. Together, they are addressing critical issues such as software downloading, hardware and software module interfaces, and protocols [10], [11].

Single-purpose radios have created size, weight, and power problems for command and control centers that require interoperability across multiple military services. Platforms that require increased radio resources include the Airborne Battlefield Command and Control Center (ABCCC),[4] the Joint Surveillance and Target Acquisition Radar System (Joint-STARS),[5] the Airborne Warning and Control System (AWACS),[6] Army mobile command centers, and Navy ships. Increased numbers of radios have been used to ensure that all potentially required waveforms and channels are available, even if the number of radios simultaneously used is only a small fraction of the number carried. A reconfigurable radio ameliorates size, weight, and power requirements without compromising the availability of radio channels. It also reduces the number of types of spare parts.

Military doctrine[3] states that C4 systems should be "interoperable, flexible, responsive, mobile, disciplined, survivable, and sustainable." Commonality and standardization are the core mechanisms that support these attributes. Equipment and systems are common when:

1) they are compatible;
2) each can be operated by personnel trained on the others without additional specialized training;
3) spare parts are interchangeable;
4) consumable items are interchangeable.

In the military, standardization is the process by which commonality is achieved. One standardization approach consolidates procurements across military departments. If

[3] Joint Pub. 6-0 (Washington, DC: U.S. Department of Defense), 1996.

[4] See http://www.dm.af.mil/ec130e.htm.

[5] See http://www.af.mil/news/factsheets/E_8C_Joint_Stars.html.

[6] See http://www.af.mil/news/factsheets/E_3_Sentry_AWACS_.html.

SPEAKeasy technology were deployed through such a consolidated procurement, the costs of unique equipment training, spare parts, unique maintenance, and unique documentation would be reduced. This SPEAKeasy vision is embraced by U.S. military communications organizations. For example, the U.S. Office of the Secretary of Defense's (OSD) Programmable Modular Communications System Integrated Product Team (PMCS-IPT) issued a Guidance Document [11] that embodies this vision. This Guidance Document follows the functional partitioning of SPEAKeasy. It is also the basis for the current Joint Tactical Radio System (JTRS) procurement. All systems developed or procured after October 1, 1999 are to be JTRS compliant. SPEAKeasy hardware and software documents have been used by both the JTRS joint program office and the SDR Forum in support of standards processes.

D. SPEAKeasy Phase-2

The on-going SPEAKeasy Phase-2 program was initiated in 1995. Its objective was to expand beyond the modem to encompass an open, modular, reprogrammable architecture for the entire radio system, from user input and output (I/O) to RF. To lower the system life-cycle costs, the Phase-2 design emphasized commercial off-the-shelf (COTS) modules and commercial standards [3]. Its capabilities were to include several different types of I/O, internetworking, reprogrammable security services, programmable wide-band modems, and continuous RF coverage at a power output of 2 W from 2 MHz to 2 GHz.

Motorola, the prime contractor, designed a special wide-band RF transceiver unit for Phase-2. It reduced intermodulation distortion of intermediate frequency (IF) conversion through its homodyne-based design. It employs TMS320-C40 DSP's augmented with field-programmable gate arrays (FPGA's). The program initially continued the development of the CYPRIS INFOSEC module. Advanced waveforms, however, cannot tolerate the long context-switching delays of CYPRIS. High-speed switching among data encryption, generation of hop sequences for transmission security (TRANSEC), and generation of pseudorandom codes for spectrum spreading is necessary. This requires rapid context switching among encryption and TRANSEC states. The programmable Advanced INFOSEC Module (AIM)[7] consists of three distinct 32-bit data RISC machines with 100 MHz clocks that provide 1200 million instructions per second (MIPS) of processing power in a 3 V package. AIM is expected to redress the context-switching problems identified using CYPRIS. As built, the current Phase-2 units are comprised of about 70% COTS modules. They employ the industry standard PCI bus. A commercial palm-top computer with Windows 95 provides the user interface. They operate in three bands from 20–400 MHz (HF, VHF and UHF).

SPEAKeasy Phase-2 was to be a four-year research and development program consisting of four sequential "model-year" development cycles. Model-year-3 and model-year-4 units were to be sufficiently robust to be tested in the field. However, model-year-1 units were enhanced for field demonstrations in the Army's Task Force-XXI Advanced Warfighting Experiment (TF-XXI-AWE) at Ft. Irwin, CA, in March 1997, approximately 15 months into the program. The Air Force Tactical Air Control Parties controlled supporting aircraft, bridging Air Force HAVE QUICK UHF to Army Single Channel Ground Air Radio System (SINCGARS) VHF radios. SPEAKeasy also bridged voice channels between aircraft using HAVE QUICK and soldiers using a commercial handheld land mobile radio (LMR). The LMR waveform was developed in less than two weeks, and was downloaded to the SPEAKeasy units in California from the laboratory in Arizona by telephone. This extended SPEAKeasy's capability during the exercise. One unit was repaired in the field using low cost-parts from a commercial computer. This experience somewhat validates the expected low cost of replacement parts for COTS-based implementations. It also raises questions about long-term availability when the commercial sector no longer stocks the parts.

Due to the success of SPEAKeasy at TF-XXI-AWE, a decision was made to immediately enter production rather than invest in further research and development. Consequently, Phase-2 had no opportunity to implement AIM and the full RF range of 2 MHz–2 GHz. The model-year-1 units also did not include wide-band waveforms, networking, or data–gateways. These units were limited in RF to the range from 4–400 MHz. The only waveforms were AM and FM voice. A small development program plans to demonstrate secure voice and data modes. The program demonstrated the benefits of an open, modular, COTS-based, reprogrammable system architecture.

E. Challenges

SPEAKeasy addresses multiple technical challenges. Open architecture, modularity, and reprogrammability objectives affect the technical aspects of each module of the system. For example, SPEAKeasy developed new methods to control a tactical radio. With the assistance of military operators, a COTS-based user interface was developed. Operation uses a touch-screen graphical user interface and the Windows 95 operating system. The granularity of the architecture had to be adjusted so that capabilities could be extended and units could be repaired in the field. INFOSEC is a fundamental military design issue not fully addressed by the commercial sector. Therefore, the use of COTS-based products in a secure ECCM-capable radio was a challenge. An INFOSEC module that would handle multiple encryption and TRANSEC algorithms simultaneously, context switching between these at various rates depending on the waveform, is still a challenge. The wide RF spectrum coverage (2 MHz–2 GHz) in a linear front end was also a critical technology driver. The exciter must deliver continuous linear output to synthesize spectrally pure waveforms defined by software and digital-to-analog converters (DAC's). Spread-spectrum (ECCM and LPI) waveforms typically require such high linearity.

[7]The Advanced INFOSEC Module (AIM) (Scottsdale, AZ: Motorola Corp.), 1998.

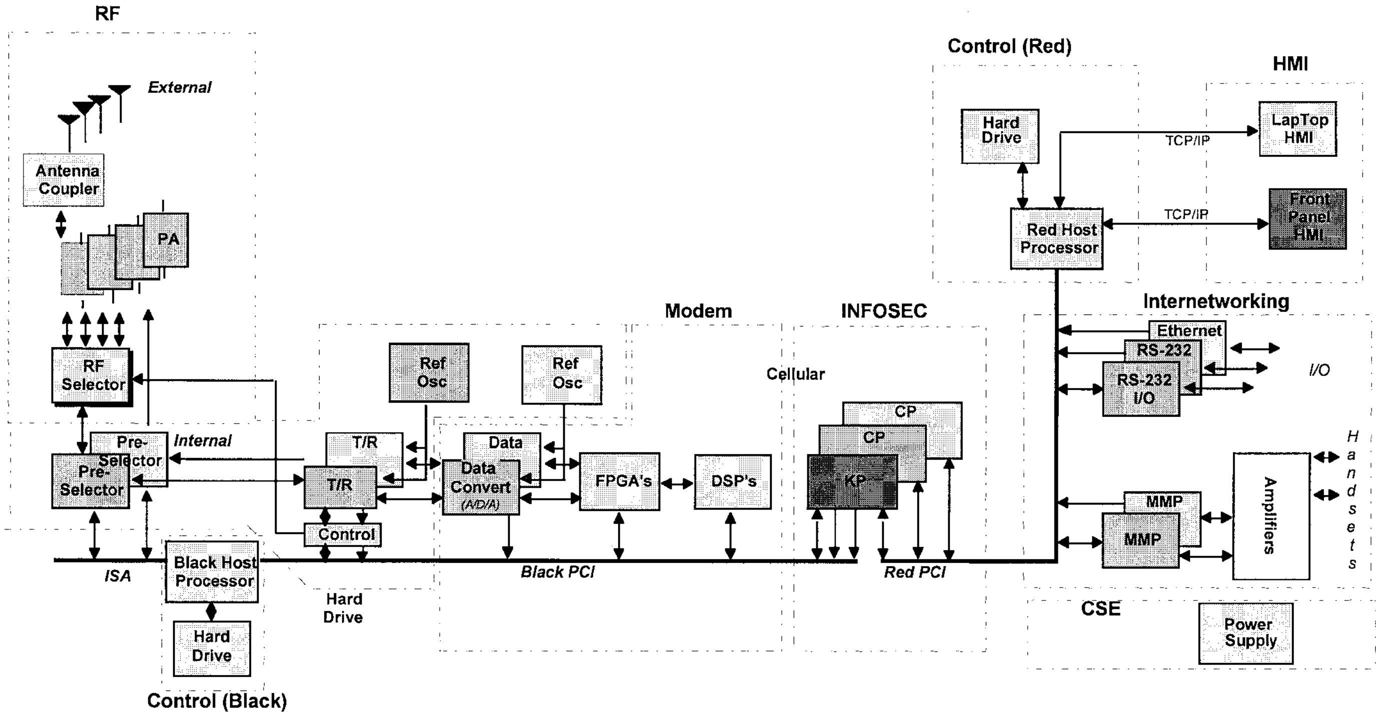


Fig. 1. SPEAKeasy block diagram.

F. The Future

Future proofing is the inclusion in current systems of facilities for the incremental addition of additional capability and new technology to avoid obsolesce. Software modularity and associated applications programmer interfaces (API's) are critical to future proofing. As an SDR, its multiprocessor architecture enables multiple RF channels, each of which employs a waveform determined by dynamically loaded application software. The SDR Forum provides a venue for technical exchange on these issues. This consortium of government, industry, and academia has the objective of defining standards to achieve the potential of SDR's, avoiding the proliferation of competing proprietary standards.

SPEAKeasy demonstrated that the technology of SDR's is feasible. Future military and commercial requirements will be met through SDR technology brought to the marketplace as commercial off-the-shelf (COTS) equipment. Special mission requirements may necessitate nonstandard hardware configurations, but open software architecture will permit both the substantial reuse of software and the introduction of mission-specific application software.

II. Architecture

The following sections describe the hardware and software architecture of SPEAKeasy Phase-2.

A. Hardware Subsystems

Fig. 1 is the hardware block diagram of SPEAKeasy Phase-2. System resources are attached to either the red (unencrypted) or the black (encrypted) peripheral component interconnect (PCI) bus. System resources are initially in a pool of available elements. They are logically assigned to virtual channels as they are instantiated. Each virtual channel is independent, executing any waveform for which there is software in the system archive. The two buses are implemented separately, with INFOSEC services and interbus communication provided by the key processor (KP) and crypto processors (CP). A CP is required for each virtual channel, but information flow does not require KP intervention once the channel is set up.

The external RF section, to the left in the figure, consists of antennas, power amplifiers, couplers, and selectors external to the unit. Modulation and conversion of digital data into analog RF takes place in the chain of transmit/receive (T/R) modules, data converters, FPGA's, and DSP's. This chain operates in half-duplex mode, reversing the flow of information between transmit and receive modes. Full-duplex operation uses two channels, one receiving and the other transmitting. To the right in the figure are facilities for voice operation, network interconnection, and user interface.

Fig. 2 shows the corresponding subsystem organization with the tasks allocated to each element. A description of the function of each subsystem follows.

The RF subsystem transmits and receives RF signals, converting signals from the analog modem into propagating RF waves, and conversely. RF subsystem functions include upconversion, downconversion, filtering, amplification, preamplification, diversity combining, T/R switching, cosite interference cancellation, antenna coupling, multicouplers, and antennas. The RF subsystem transmits and receives multiple channels simultaneously in the frequency range from 2 MHz to 2 GHz. It includes the antennas, couplers, cosite filters, power amplifiers, transmit/receive switching system, T/R

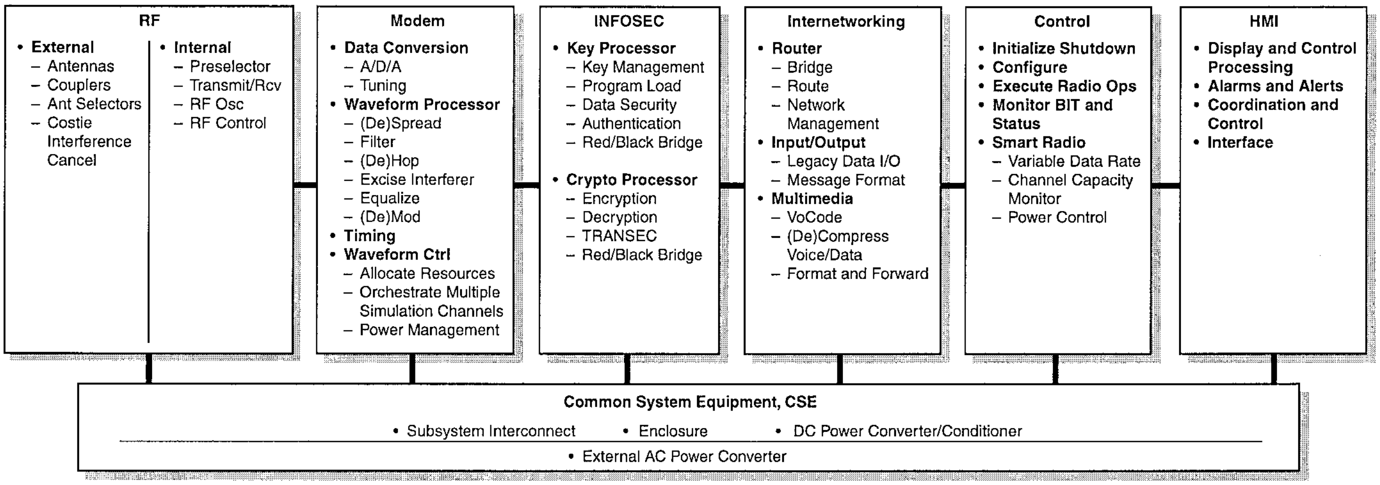


Fig. 2. SPEAKeasy system architecture and functional allocations.

modules, and control functions for half-duplex operation. It switches between receive and transmit states within milliseconds.

The modem subsystem converts between analog and digital format, modulating and demodulating the waveforms. The subsystem can transmit and receive multiple channels simultaneously, receiving analog signals and producing demodulated digital information during receive mode. During transmit mode, the input digital information is modulated and converted to output analog signals to excite power amplifiers at the specified transmitter power level. Frequency-hopping software in the modem generates frequency changes from hop sets maintained by the INFOSEC module. The modem subsystem includes signal processing functions, such as ADC's and DAC's, modulation and demodulation, forward error correction, interference suppression, carrier tracking, bit interleaving, and data framing. Modem resources may be allocated to any of the installed channels based on the waveform selected when the channel is assigned. Simple, low-speed waveforms place less demand on available resources than wide-band, complex waveforms. Changing parameters and reallocating channels may be performed without interrupting the operation of concurrently established channels.

The INFOSEC subsystem provides cryptographic security and TRANSEC. It contains one or more crypto-processor modules to encrypt baseband data and to create TRANSEC bitstreams. The INFOSEC subsystem also manages keying material to support these functions. The COMSEC and TRANSEC services enable the system to interoperate with legacy communications equipment and to provide communications at the same security level as that equipment.

The internetworking subsystem provides routes Internet protocol (IP) packets, accesses I/O ports, and processes channelized voice using multimedia processors (MMP's). These processors compress and restore voice, bridge voice and data from one RF mode to another, condition the signals, and deliver voice to the network or to the user. The subsystem simultaneously supports multiple independent half-duplex radio channels of voice or data in transmit and receive modes. It also supports multiple wireline voice and data circuits. In addition, it interconnects arbitrary data channels via IP routing to any other IP host or network. The network-level features efficiently route packet-switched traffic to the intended destinations. The router function has primary responsibility for routing data packets between appropriate links, and for determining which of the available paths in an attached network is used. It also provides I/O data bridging services, gateway services, and data-gram services.

The control subsystem generates commands that initialize, operate, and shut down the system. Initialization includes configuration of the radio, router, and I/O; downloading of waveforms and algorithms; built-in test (BIT); and diagnostics. It manages networks, subsystem configurations, files, and databases for the other subsystems. It receives commands from the user interface. It also provides long-term storage, including waveform software, cryptographic keys, status, logs, and other data. Operational functions include channel and waveform management; default system configuration controls, user authorization and levels of control, smart radio, and internetworking functions. Smart radio functions use software to improved performance through variable data rates, channel monitoring, and power control. The shutdown function releases resources, updates persistent storage, and prepares for reactivation.

The human–machine interface (HMI) subsystem provides the system-level user interface. An embedded palm-top computer provides the primary user interface for the local operator using the Windows 95 operating system. The HMI also includes remote control on a data circuit through a communications port.

The common systems equipment (CSE) subsystem houses the system, interconnects the subsystems, and supplies power, thermal management, and environmental protection. The enclosure addresses physical security, electromagnetic interference (EMI), and radio frequency interference (RFI) shielding and isolation from the environment (temperature, shock, and vibration). The degree of protection allows commercial com-

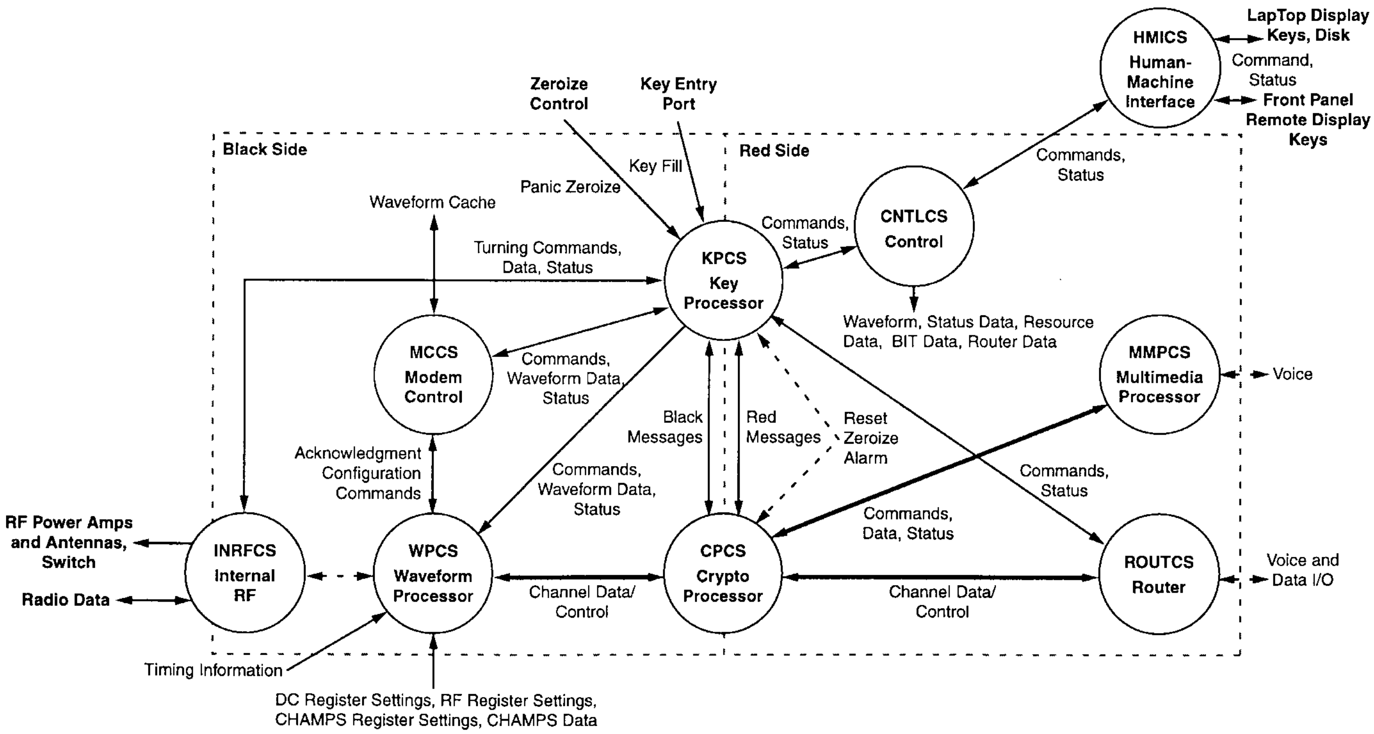


Fig. 3. SPEAKeasy control software (CS) and data flows.

ponents to operate at design capabilities in spite of field conditions.

Hardware was selected through extensive trade studies. Alternate approaches for each of functions to be performed were rated according to the following:

- initial cost
- life-cycle cost
- size
- power
- weight
- risk
- network compatibility
- interface robustness
- availability of alternate form factors
- commercial applications
- preplanned product improvement (P3I) potential
- performance.

The primary acceptance criterion was the composite score for system cost. Procurement cost was weighted by scores assigned to the factors listed above in the composite score.

III. Software Structure

This section provides an overview of the software structure. Fig. 3 indicates the names of the primary system modules and relevant data flows.

Table I indicates the functionality of each software module with its size in executable lines of code.

TABLE I
SOFTWARE MODULES

Module Name	Functions	Language	Lines of Code
INRFCS	RF API RF Control T/R Control	C	2500
WPCS	Common SW SINCGARS SIP UHF SATCOM HaveQuick I & II HF ALE (141A)	C C C C C	4400 3685 3150 3500 8820
MCCS	Allocate Resources Manage Channels Manage Power	C++	2340
KPCS	Key Management Control Red-Black Bridge Program Load SBA Authentication	Ada	33853
CPCS	Encryption Decryption TRANSEC	Assembly	23737
CNTLCS	Startup/Shutdown Waveform Instantiate BIT Management Status Management System Data Management	C++	12000
MMPCS	Vocode (De)Compress VVD Format and Forward	C	3500
ROUTCS	Router I/O MMP Proxy	C	62500
HMICS	Operator Command Process Display process	C/C++	12000

1) Internal RF Control Software (INRFCS): INRFCS is the Internal RF control module. Because the RF stage conditions analog signals, its functionality is limited to control functions. These include interconnection of hardware including antennas, power amplifiers, preselectors, and interference cancellers. It relays commands to those components that have programmable elements. It reconfigures for transmit/receive, sets the power level for the high-speed power amplifier, and sets the local oscillator frequency when used for up- or downconversion.

2) Modem Control Computer Software (MCCS): During initialization, MCCS allocates modem resources to logical channels as needed for the channel waveform. During operation, its state machines control the functions of resources assigned to a specific channel.

3) Waveform Processing Computer Software (WPCS): WPCS executes on the waveform processor DSP's. Its function is determined by the specific waveform and communications application task modules that have been loaded into it. Functions include spreading, despreading, modulation, demodulation, synchronization, interleaving, deinterleaving, equalization, framing, and deframing.

4) Key Processing Computer Software (KPCS): KPCS maintains cryptographic key control as authorized by the cognizant central authority (CA) for the system. Operating on both the red and black buses, it maintains security by monitoring control messages sent across that boundary for authenticity. It is the only entry into the system for messages between the system control module and other modules.

5) Cryptographic Processor Computer Software (CPCS): CPCS is assigned to a specific channel by the KPCS. It provides a virtual connection between the red bus and black bus, and performs simple bus-to-bus operation in plaintext mode. It supplies real-time encryption and decryption to data streams that need that service.

6) Multimedia Processing Computer Software (MMPCS): MMPCS serves as a vocoder to interface analog voice facilities with the digital data carried on an application channel. It also compresses, decompresses, and bridges among voice channels channels.

7) Control Computer Software (CNTLCS): CNTLCS starts the system and shuts it down. Under operator, script, or remote control it controls application instantiation and channel setup. It is the system data manager, and maintains system status. It is also the conduit for information passage between the system and the user interface software.

8) Routing Computer Software (ROUTCS): ROUTCS routes traffic and interconnects protocols to interconnect landline ports. It serves as a networkwide connectivity function for SPEAKeasy. It routes data packets between appropriate links, and determines which of the available paths are used within communications subnetworks. Services provided by the router include I/O data bridging, gateway, and datagram routing. It services external router requests. As the local communications control processor, it supports RS232, Ethernet, FDDI, and MIL-STD-1553. It also routes IP packets and terminates IP protocol stacks.

9) Human–Machine Interface Computer Software (HMICS): HMICS employs a display, keyboard, and other I/O devices for local or remote operator control of the system. Its control panels replicate functions of legacy waveforms in operation. Multiple channels can be in operation simultaneously, and the HMI permits the operator to switch between their control displays.

IV. SPEAKeasy System Operation

A. Overview

When powered up, the red and black host processors scan their respective PCI buses to establish memory maps of the location of available resources. They then communicate those maps to the control subsystem, which sets up static control channels to each resource using a parent–child hierarchical relationship. Agents are also created in the subsystems with responsibility for managing subsystem resources and responding to resource allocation requests. These requests are generated in response to user commands through the user interface software. Dynamic virtual channels are established using control messages communicated over the static control-channel structure. These dynamic channels directly connect resources assigned to the channel at the communications layer level to meet real-time information transfer constraints.

The operator uses the HMI to initiate operation, and to specify how the system is to be configured. A menu specifies the waveform to be used, the radio frequency, and waveform-specific parameters. Commands from that interaction are passed on to the control subsystem, which translates them into specific instructions to each of the other subsystems. These subsystems allocate resources, initialize, and instantiate channel application tasks to communicate on the specified channel in the desired configuration. Channel instantiation includes the following steps:

allocate devices that comprise a channel
instantiate channel application tasks
interconnect the dynamic information paths
set waveform-dependent parameters
activate the channel.

Channel deactivation reverses this process. Resources freed by channel deactivation may be reassigned to another channel. Resource pools increase reliability through redundancy, with graceful degradation in the event of equipment failure.

B. Communication Protocol Model

Communication among software modules employs messages transmitted over the system PCI bus. A three-layer software model forms a stack of logical layers of abstraction that build on the hardware of the physical layer as shown in Fig. 4. The three software layers are

link layer
communications layer
application layer.

1) Link Layer: The link layer moves data on a byte-by-byte basis between two devices over the physical bus. The link layer API function calls are the following.

Function Call	*Parameters*
`lnkStart`	`(pErrorNotify)`
`lnkStop`	`( )`
`lnkRead`	`(pLocal, Remote, Len_In_Bytes)`
`lnkWrite`	`(Remote, pLocal, Len_In_Bytes)`
`ErrorNotify`	`(ErrorNumber)`

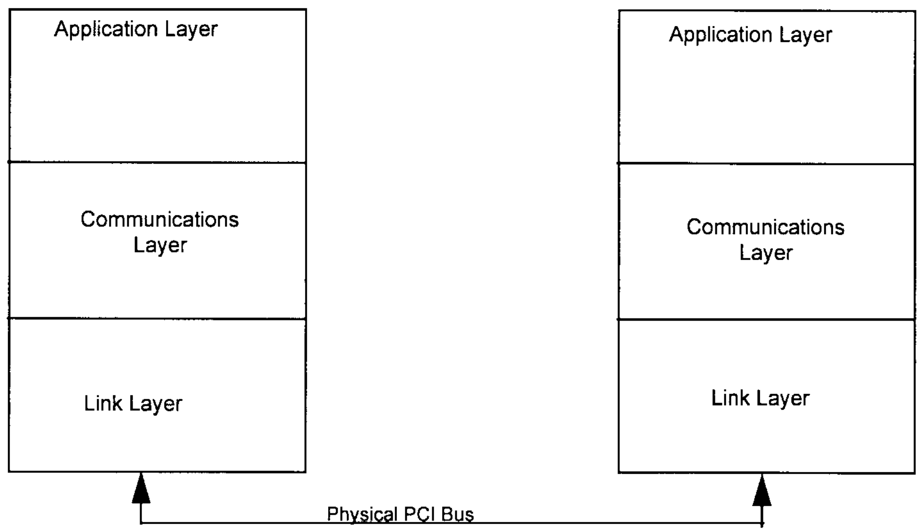


Fig. 4. SPEAKeasy internal protocol stack.

TABLE II
RADIO FREQUENCY CONTROL

No	Name	Phase	No	Name	Phase
1	ACK	All	18	Antenna_Select	Params_&_Mode
2	Buffer_Complete	All	19	DeAllocate_Resources	Params_&_Mode
3	Buffer_Notify	All	20	Define_Child	Params_&_Mode
4	Forward_Message	All	21	RF_Direction	Params_&_Mode
5	NACK	All	22	RX_Calibration	Params_&_Mode
6	BIT_Request	Power_up	23	Software_Version_Request	Params_&_Mode
7	Define_Remote_Child	Power_up	24	Hop_Strobe	Operation
8	Define_Remote_Parent	Power_up	25	Initiate_TX_Calibration	Operation
9	Allocate_Resources	Instantiation	26	Receive_Mode_Ind	Operation
10	Connection_Test	Instantiation	27	RF_Frequency	Operation
11	Define_Remote_Child	Instantiation	28	Set_Gain	Operation
12	End_Download	Instantiation	29	Standard_Data_Msg	Operation
13	File_Download_Complete	Instantiation	30	T/R_Transmit	Operation
14	File_Download_Start	Instantiation	31	Transmit_Mode_Ind	Operation
15	Initiate_Download	Instantiation	32	Destroy_Agent	Teardown
16	New_Agent	Instantiation	33	PA_Power	Teardown
17	Standard_Data_Msg	Instantiation	34	Reset_to_Boot	Teardown

These functions require a hardware bus address to transmit data. The virtual channel address to bus address mapping is handled at the higher level communications layer.

2) Communications Layer: The term "communication layer" is consistent with SPEAKeasy documentation. The term "infrastructure layer" may be used interchangeably. This layer employs link layer facilities to create a message-passing facility with an API used by waveform applications. It establishes links among the installed resources, and automatically identifies the devices on the bus when powered up. It performs the buffering and queueing of data internally, prioritized as necessary to achieve system performance requirements. The communications layer API's are the following.

Function Call	*Parameters*
`CommStart`	`(ErrorNotify)`
`commStop`	`( )`
`commCreateAgent`	`(Agent, SendComplete-Notify, ReceiveNotify, ReceiveCompleteNotify)`
`commDestroyAgent`	`(TAgent_Number_Type)`
`commListen`	`(Listen_Desc)`
`commConnectChildren`	`(Agent, ChildDesc, ChildPCIInterfaceDesc, ParentHint)`

TABLE III
MODEM

1	ACK	All		9	Adjust_RX_Calibration_Response	Params_&_Mode
2	Buffer_Complete	All		10	TX_Calibration_Complete	Params_&_Mode
3	Buffer_Notify	All		11	Crypto_Status	Operation
4	Forward_Message	All		12	Pacing_Indication	Operation
5	NACK	All		13	Receive_Mode_Ind	Operation
6	Connection_Test	Instantiotion		14	Standard_Data_Msg	Operation
7	Standard_Data_Msg	Instantiotion		15	Transmit_Mode_Ind	Operation
8	Activate_Channel	Params_&_Mode				

TABLE IV
INFOSEC

No	Name	Phase		No	Name	Phase
1	ACK	All		20	Standard_Data_Msg	Instantiotion
2	Buffer_Complete	All		21	Activate_Channel	Params_&_Mode
3	Buffer_Notify	All		22	Crypto_Algorithm	Params_&_Mode
4	Error_Status	All		23	Define_Child	Params_&_Mode
5	Forward_Message	All		24	Key_Tag	Params_&_Mode
6	NACK	All		25	Load_Key	Params_&_Mode
7	BIT_Request	Power_up		26	Mode_Req	Params_&_Mode
8	Define_Remote_Child	Power_up		27	Plaintext_Confirm_Response	Params_&_Mode
9	Define_Remote_Parent	Power_up		28	Software_Version_Request	Params_&_Mode
10	Agent_Description	Instantiotion		29	Suspend_Channel	Params_&_Mode
11	Connection_Test	Instantiotion		30	Channel_Status	Operation
12	Connect_Children	Instantiotion		31	Deactivate_Channel	Operation
13	Define_Red_Channel	Instantiotion		32	Pacing_Indication	Operation
14	Define_Remote_Child	Instantiotion		33	Receive_Mode_Ind	Operation
15	End_Download	Instantiotion		34	Standard_Data_Msg	Operation
16	File_Download_Complete	Instantiotion		35	Start_Traffic	Operation
17	File_Download_Start	Instantiotion		36	Transmit_Mode_Ind	Operation
18	Initiate_Download	Instantiotion		37	Disconnect_Children	Teardown

Function Call *Parameters*

```
CommDisconnectChildren
                  (Connection)
commSend          (Agent,
                   Buffer_Addr, Size)
commReceive       ( )
commBufferComplete   (Mailbox)
commLengthToSize (Length, Size)
commGetUserDataPtr   (Agent, Addr)
commSetUserDataPtr   (Agent, Addr)
```

3) Application Layer: This layer consists of waveform-specific software. All of the software modules in Fig. 3 reside in this layer, communicating through exchange of messages dispatched through the communications layer. The messages categories include data and control (which includes commands, status, and parameters). Control messages are virtual function calls.

C. Message Passing Architecture

There is no centralized operating system in SPEAKeasy. All modules communicate by cooperatively passing messages across the bus. In general, message passing is asynchronous and concurrent. Multiple communication channels and associated control functions are active concurrently. The messages are shown in Tables II–VII. These tables are organized according to the system module in which the implicit function call is executed. Only the common messages are identified; each waveform has additional messages. Each table indicates the phase of operation to which it applies.

D. Phases of Operation

1) Power Up: During the power-up phase, agents and their child–parent relationships are established. Static control channels are established, and built-in-test (BIT) is performed.

TABLE V
INTERNETWORKING

No	Name	Phase		No	Name	Phase
1	ACK	All		21	Default_Voice_Rate	Params_&_Mode
2	Buffer_Complete	All		22	Default_Voice_Type	Params_&_Mode
3	Buffer_Notify	All		23	Define_Child	Params_&_Mode
4	Forward_Message	All		24	External_Access_Tone	Params_&_Mode
5	NACK	All		25	Loopback_Test_Mode	Params_&_Mode
6	BIT_Request	Power_up		26	Mode_Req	Params_&_Mode
7	Define_Remote_Child	Power_up		27	Mode_Resp	Params_&_Mode
8	Define_Remote_Parent	Power_up		28	PTT_Priority	Params_&_Mode
9	Connection_Test	Instantiation		29	Sidetones	Params_&_Mode
10	Define_Remote_Child	Instantiation		30	Software_Version_Request	Params_&_Mode
11	End_Download	Instantiation		31	Suspend_Channel	Params_&_Mode
12	File_Download_Complete	Instantiation		32	Test_Mode	Params_&_Mode
13	File_Download_Start	Instantiation		33	User_Tone	Params_&_Mode
14	Initiate_Download	Instantiation		34	Channel_Status	Operation
15	New_Agent	Instantiation		35	Deactivate_Channel	Operation
16	Standard_Data_Msg	Instantiation		36	Standard_Data_Msg	Operation
17	Activate_Channel	Params_&_Mode		37	Start_Traffic	Operation
18	Alert_Tones	Params_&_Mode		38	Volume_Level	Operation
19	Bridging	Params_&_Mode		39	Destroy_Agent	Teardown

TABLE VI
CONTROL

No	Name	Phase		No	Name	Phase
1	ACK	All		27	External_Access_Tone	Params_&_Mode
2	Buffer_Complete	All		28	Key_Tag	Params_&_Mode
3	Buffer_Notify	All		29	Loopback_Test_Mode	Params_&_Mode
4	End_Ext_Channel_Control	All		30	Modem_AGC_Loop_Bandwidth	Params_&_Mode
5	Error_Status	All		31	Modem_TX_Power	Params_&_Mode
6	Ext_Channel_Control	All		32	Mode_Req	Params_&_Mode
7	Forward_Message	All		33	Mode_Resp	Params_&_Mode
8	NACK	All		34	Plaintext_Confirm_Request	Params_&_Mode
9	Add_Agent	Power_up		35	Plaintext_Confirm_Response	Params_&_Mode
10	BIT_Request	Power_up		36	PTT_Priority	Params_&_Mode
11	BIT_Response	Power_up		37	Sidetones	Params_&_Mode
12	Define_Remote_Parent	Power_up		38	Software_Version_Request	Params_&_Mode
13	Connection_Test	Instantiation		39	Software_Version_Response	Params_&_Mode
14	Download_Complete	Instantiation		40	Test_Mode	Params_&_Mode
15	Download_Files	Instantiation		41	User_Tone	Params_&_Mode
16	Instantiate_Waveform	Instantiation		42	Channel_Status	Operation
17	Resources_ Allocated	Instantiation		43	Crypto_Status	Operation
18	Security_Level	Instantiation		44	Deactivate_Channel	Operation
19	Standard_Data_Msg	Instantiation		45	Receive_Mode_Ind	Operation
20	Activate_Channel	Params_&_Mode		46	Signal_Strength	Operation
21	Alert_Tones	Params_&_Mode		47	Squelch_Level	Operation
22	Antenna_Select	Params_&_Mode		48	Standard_Data_Msg	Operation
23	Bridging	Params_&_Mode		49	Transmit_Mode_Ind	Operation
24	Channel_Direction	Params_&_Mode		50	Volume_Level	Operation
25	Crypto_Algorithm	Params_&_Mode		51	PA_Power	Teardown

TABLE VII
HUMAN–MACHINE INTERFACE

No	Name	Phase	No	Name	Phase
1	ACK	All	19	Loopback_Test_Mode	Params_&_Mode
2	Buffer_Complete	All	20	Modem_AGC_Loop_Bandwidth	Params_&_Mode
3	Buffer_Notify	All	21	Modem_TX_Power	Params_&_Mode
4	End_Ext_Channel_Status	All	22	Plaintext_Confirm_Request	Params_&_Mode
5	Error_Status	All	23	PTT_Priority	Params_&_Mode
6	Ext_Channel_Status	All	24	Sidetones	Params_&_Mode
7	Forward_Message	All	25	Software_Version_Response	Params_&_Mode
8	NACK	All	26	Test_Mode	Params_&_Mode
9	BIT_Response	Power_up	27	User_Tone	Params_&_Mode
10	Connection_Test	Instantiation	28	Channel_Status	Operation
11	Security_Level	Instantiation	29	Crypto_Status	Operation
12	Standard_Data_Msg	Instantiation	30	Receive_Mode_Ind	Operation
13	Alert_Tones	Params_&_Mode	31	Signal_Strength	Operation
14	Antenna_Select	Params_&_Mode	32	Squelch_Level	Operation
15	Channel_Direction	Params_&_Mode	33	Standard_Data_Msg	Operation
16	Crypto_Algorithm	Params_&_Mode	34	Transmit_Mode_Ind	Operation
17	External_Access_Tone	Params_&_Mode	35	Volume_Level	Operation

2) Instantiation: Communication occurs between agents within the protocol stack. Links between agents are children to a broker agent that instantiates the connection. Once the child agents exchange information, they test the connection to assure correct instantiation of the channel. Operating software uses these channels to load the subsystem processors. Once the load is complete, the agents deliver messages among modules.

3) Parameters and Mode: Waveform parameters are communicated to individual subsystems for proper operation. Typical parameters include receive/transmit state from the handset's push-to-talk switch, transmit power, and antenna selection.

4) Operation: Normal operation of the channel transfers information from source to destination through the internal protocol stack. RF channels and network interfaces may be either sources or destinations, providing transmit, receive, RF channel bridge, and landline switching modes of operation.

5) Teardown: Upon completion of operation, agents are deactivated, channels disconnected, resources returned to the pool, secondary (persistent) storage updated, and the system made ready for reactivation.

V. CONCLUSION

This paper has summarized the background and military needs that led to SPEAKeasy. It has also described the upper levels of the SPEAKeasy architecture. Using current technology, SPEAKeasy has demonstrated the vision of an information transfer system that implements many waveforms using common hardware. It also holds the promise of the economies of scale that are possible with commercial components in volume production.

The SDR Forum is working to satisfy a major need for future development: the adoption, by the industry, of standards comparable to those in effect in the personal-computer market. Such standards could assure vendors of a larger, consolidated market of diverse platforms manufactured in high volumes. The adoption of such standards will also permit the independent development of waveform applications to meet existing and new applications.

REFERENCES

[1] R. J. Lackey and D. W. Upmal, "Speakeasy: The military software radio," *IEEE Commun. Mag.*, vol. 33, pp. 56–61, May 1995.

[2] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, pp. 26–38, 1995.

[3] A. McAuliffe, "Technology drives development of multiband, multimode radio," *Mil. Aerosp. Electron.*, vol. 6, Oct. 1995.

[4] R. E. Hundt, Chairman of the Federal Communications Commission, prepared remarks before the Assoc. of Public Safety Commun. Officers (APCO) Annu. Conf., Aug. 17, 1995.

[5] Under Secretary of Defense for Acquisition and Technology, Honorable P. G. Kaminski, Prepared Remarks before the Res. and Develop. Subcommittee of the House Committee on Nat. Security, Mar. 28, 1995.

[6] ——, Prepared Remarks before the Subcommittee on Defense Technol., Acquisition and Industrial Base of the Senate Committee on Armed Services on Dual Use Technol., May 17, 1995.

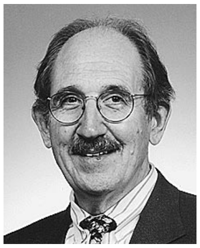
[7] F. J. Schrenk *et al.*, "SPEAKEASY—The universal radio for the 21st century," *MILCOM'95 Classified J.*, Nov. 1995.

[8] S. M. Blust, "Software defined radio: Industry request for information," BellSouth Wireless, Inc., Dec. 1995.

[9] D. M. Climek and E. J. Newman, "The SPEAKeasy software defined radio: An opportunity for transparent wireless telecommunications standards in the Asia-Pacific region," in *Proc. Pacific Telecommun. Council (PTC) Conf.*, Honolulu, HI, Jan. 1997.

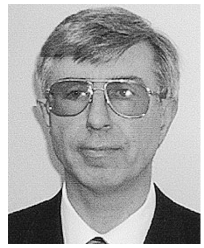
[10] J. A. Hoffmeyer and W. Bonser, "Standards requirements and recommendations development for multiband, multimode radio systems," *MILCOM'97 Unclassified J.*, Nov. 1997.

[11] SDR Forum Tech. Rep. 1.1, SDR Forum, Jan. 1998.



Peter G. Cook received a degree in physics from Williams College, Williamstown, MA.

He is Chief Computer Scientist for the Wireless Information Transfer Systems Business Unit of Motorola's Systems Solutions Group. His current work is concentrated in Software Defined Radios. He has more than 35 years of experience in computer and communications systems, system architecture, and software design and development. Past projects include work on SmartCards, cellular telephones, satellite communications, ATM switches, and multiprocessor system software. He was Motorola Liaison Representative to the MCC research consortium in Austin, TX. Former employers include Four-Phase Systems, Tektronix, and Corning, Inc. He is a retired Air Force Lt. Col., and a former pilot and communications officer.



Wayne Bonser has worked for the Air Force in research and development for 25 years. During his first 11 years, he worked for the Rome Air Development Center's Longwave Communications Section. He was a lead Engineer in the development of VLF/LF communications technology, as applied to the Minimum Essential Communications Network (MEECN). He was also the center of expertise for adaptive HF communications and the Program Manager for the tri-service program known as "New Look" (sponsored by the Defense Communications Agency). He was also involved with meteor burst communications and buried antenna programs. During the period August 1984–August 1989, he was employed as a Principal Scientist at the SHAPE Technical Centre, The Hague, The Netherlands. His principle responsibilities there involved developing and applying interference cancellation techniques to NATO communications problems. He was also involved with adaptive interference cancellation techniques as applied to mobile microwave (*C*-band) and meteoric scatter systems. After his return to Rome Lab in 1989, he was the focal point for Special Operations Forces (SOF). In this role, he was further employed as an alternate AF contact for the Joint Service LPI/D working, and as an INFOSEC development advisor for communications. In 1992, he took the leadership role as Program Manager for the multiband multimode joint service and DARPA program known as SPEAKeasy. He was also responsible for the formation of the international software radio forum known as MMITS. He is currently the Technical Advisor to the AFRL Information Connectivity Branch, and is the Manager of AFRL's Global Grid Integrated Technology Thrust Program. He has chaired symposia, written many technical reports, and made presentations to international audiences. He has taught courses for the IEEE, and has been a Guest Lecturer at the NATO Communications School, Latina Italy. He has several inventions to his credit, and holds an Air Force Patent.