THE WIRELESS INFORMATION TRANSFER SYSTEM (WITS) ARCHITECTURE FOR THE DIGITAL MODULAR RADIO (DMR) SOFTWARE DEFINED RADIO (SDR)

Byron Tarver
Eric Christensen, Ph.D.
Annamarie Miller
Motorola SSG
Scottsdale, Arizona

ABSTRACT

The Software Defined Radio (SDR) is a rapidly emerging concept in wireless communications, brought about by increased preformance in general purpose processing and digital signal processors. The combination of the advances in computing with those advances in networking and radio technology makes the SDR a reality today. The market for SDRs is focused on user application services and the commercial and military need to upgrade and maintain user services with legacy systems without 'a forklift upgrade'. Additionally, the SDR provides an affordable migration path to new systems and services without requiring users to immediately abandon billions of dollars of investment in existing systems. The user needs include seamless interconnectivity to voice and data networks and flexible interoperability with legacy systems. The users need systems that are scalable, allow graded services and security to meet mission objectives, in-field upgrades, and reduced total ownership cost. The needs are fulfilled by SDR services and applications including wireless and wireline communications, routing and bridging dissimilar systems, remote configuration, control and operation, and reprogrammability. The architecture that supports such a system must live and evolve through use of open standards architecture, high reliance on standard hardware modules, and software programmability. Motorola System Solutions Group's Wireless Information Transfer System (WITS) is an SDR selected as the U.S. Navy's Digital Modular Radio (DMR) offering the services and fulfilling the needs of the communications community. WITS/DMR products share a common architecture designed to support hardware and software evolution and applications independence. This paper discusses the DMR/WITS architecture and its ability to meet the challenges of the communications customers and support an orderly evolution from today's wireless systems to the advanced networks of tomorrow.

INTRODUCTION

The Department of Defense (DoD) customers initially defined requirements for an SDR through the SPEAKeasy, Joint Tactical Radio System (JTRS), and the Navy's Digital Modular Radio (DMR) programs. Motorola SSG as a SPEAKeasy contractor, successfully developed and demonstrated a first generation SDR in the field as a part of the U.S. Army's Task Force XXI in March 1997. Numerous focus groups and interactive briefings with both commercial and government customers and field support personnel and participation in the Software Defined Radio Forum (SDRF) set a direction for the development of Motorola's Wireless Information Transfer System product family. The WITS product family development plan includes platforms and services for varying form factors including maritime/fixed, airborne, mobile, manpack and handheld versions. The Navy's DMR is a modified WITS 6004 maritime/fixed platform.

USER AND SERVICE PROVIDER NEEDS

At the onset of Motorola's SDR development, the focus was on the technology and equipment. As we progressed, proved the viability of the SDR, and technology continued its advancement in both the hardware and software arenas, the need began shifting from equipment or hardware to services. This shift continues to gain momentum driven by the compelling need of individuals and organizations to be informed, to communicate across traditional RF Protocol boundaries, and to be connected anywhere, at anytime. The market for SDRs is focused on user application services and a service provider's ability to upgrade and maintain user services without 'a forklift upgrade'.

SOFTWARE DEFINED RADIO DEFINITION

Traditionally, radios provided capability through hardware with some software control. A Software Defined Radio is a communications system where applications, *e.g.*, waveforms, are defined in software and operations are controlled through software. *Figure 1* illustrates an SDR reference model.

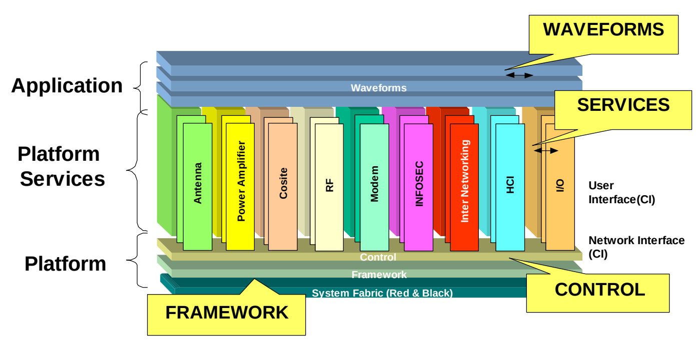


Figure 1. SDR Reference Model

0-7803-6521-6/00/$10.00 (c) IEEE

There are four software architecture elements that provide foundations for rapid and remote upgradability. These elements are *Framework, Control, Services, and Waveforms*. The *Framework* element is middleware that provides mechanisms for writing portable software objects and provides “glue” for the platform services and applications to bind to the radio platform. The *Control* element manages the radio operations and provides services for radio robustness. The *Service* elements provide specialized processing for waveform signal flow including security and networking. The *Waveforms* are the SDR applications that use the *Services* provided by the platform. What makes the SDR so adaptable is its use of general purpose processing resources such as DSPs, FPGAs, and general purpose microprocessors, *e.g.*, PowerPC computers or Pentium™ processors instead of application specific ICs (ASICs).

The capabilities and features of an SDR to enhance the functionality and insure against obsolescence are numerous. They include:

- Reconfiguration and redefinition
- Modular structure with open interface standards
- Multi-Channel interoperability features supporting transcoding, transcrypting, transmodem, and crossbanding
- Control and configuration
- Maintain Quality of Service (QoS)

SYSTEM REQUIREMENTS FOR SDRS

The requirements for an SDR architecture to meet the user and service provider’s needs include but are not limited to:

- Reprogrammability to support control and configuration
- Portable design, reconfigurability and upgradability to support reconfiguration and redefinition
- Flexibility to support multi-channel interoperability such as simulcasting, bridging, and routing
- Scalability, extendibility, and modular design to support growth over time.
- Ability to allow third party application/waveform development to support new features and services without hardware upgrades or intimate familiarity with the internal software and hardware structures
- Affordability to promote commercially available personal and industrial computer software and hardware products and component standards.

A WITS node offers seamless interoperability in a single equipment. WITS is capable of wireless communications from 2 MHz to 2 GHz, on up to four simultaneous channels per equipment, with wireline communications via standard protocols. A WITS node as depicted in *Figure 2* may be instantiated as a single Information Transfer Unit (ITU) or a combination of units to provide wireless and wireline communications. The WITS system channels are fully configurable and reprogrammable in the field by the operator or remotely from control consoles providing flexibility in fielding.

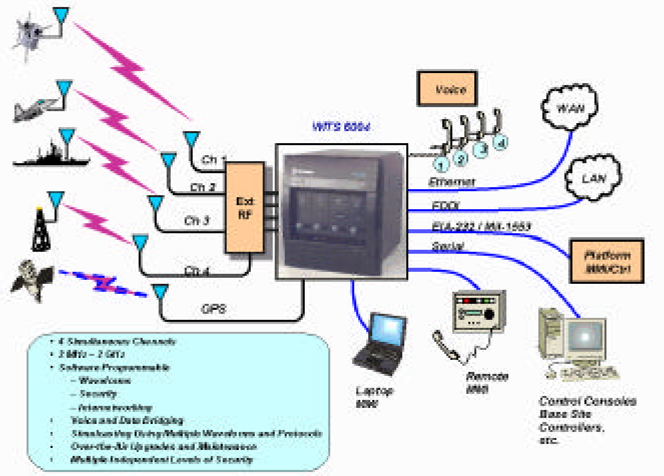


Figure 2. WITS Wireless and Wireline Connectivity

The functional architecture for WITS takes two market-driven forms, one for the commercial and the other for the military market. The primary difference is the required level of security. The military architecture requires strict separation of encrypted (black) and unencrypted (red) signals and physical regions, with higher complexity and costs, while many commercial solutions need security only as an additional processing service. It is important to note that the same equipment (hardware and software) is used in implementations of both forms. The forms differ by the insertion of the proper security design for the application.

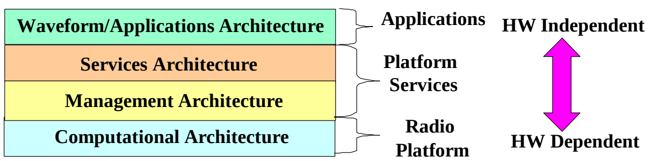
Current development of the WITS Products includes the WITS 6004, the WITS 5004, and the WITS 3002. An instantiation of the WITS architecture embodied as a 6004 completed evaluation by the Navy DMR program. The 6004 is a 4 channel full-duplex system, the 5004 is a 4 channel half-duplex system, and the 3002 is a transportable two channel half-duplex system. All include embedded reprogrammable INFOSEC for all algorithms and key management. The INFOSEC uses the Motorola Advanced INFOSEC Machine (AIM) that has received NSA endorsement. Waveforms successfully implemented in software and executing on the WITS 6004 to date include UHF SATCOM (181A), DAMA (183), HaveQuick I/II, SINCGARS, Link 11, VHF for ATC, and UHF LOS.

DMR/WITS SOFTWARE ARCHITECTURE

The driving requirements for the DMR/WITS software architecture of reprogrammability, reconfigurability, multi-channel interoperability features, and flexibility require state-of-the-art software technology. The relationship between the hardware and software must be one that

0-7803-6521-6/00/$10.00 (c) IEEE

isolates the radio platform hardware and software from the waveform application software. *Figure 3* illustrates a simplified view of the DMR/WITS software architecture strategy that supports long term usability of the waveform application independent of the evolutions in the radio platform. The separation of the Computational Architecture (Hardware and Framework) allows hardware to change independent of applications offering longevity and access to evolutions in hardware and software.



Note: Adapted from Overall Concepts and Principles of TINA, Feb. 17, 1995

Figure 3. Simplified WITS/DMR Architecture

The WITS Products Common Software Architecture depicted in *Figure 4* maintains independence of the Radio Platform from the Radio Application. This independence allows the radio platform hardware and software to change without modification to the radio applications (*e.g.*, SINCGARS waveform) extending the product life and enhancing software portability. The software architecture uses distributed object technology based on minimum Common Object Request Broker Architecture (CORBA) accommodating real-time requirements, Application Programming Interfaces (APIs) implemented in Interface Definition Language (IDL) for internal interfaces and Abstract Syntax Notation 1(ASN.1) for external interfaces.

Our Object Request Broker (ORB) is a middleware implementation, which provides the bus for the software to communicate across heterogeneous COTS hardware creating a distributed operating system for all the processors. The varied processors' operating systems and connectivity protocol elements are encapsulated in POSIX compliant APIs. The platform services are then accessible without COTS-specific calls allowing hardware modification without application software impact. This allows hardware technology insertion with little or no impact to the software and accommodates the different rate of change for hardware and software.

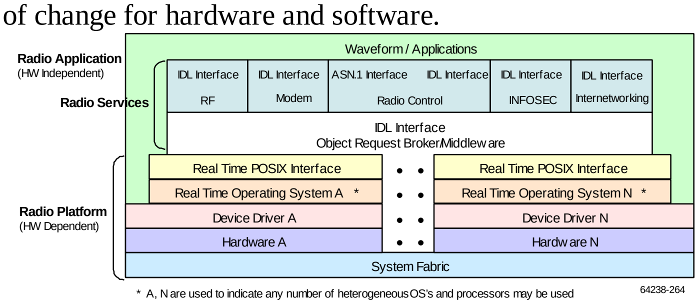


Figure 4. DMR/WITS Software Architecture Diagram

Radio application software including waveforms are decomposed into objects. These objects are reduced to an elemental functional level with an API in IDL to allow reusability. Objects that perform various waveform functions can be distributed to run on different nodes. The dynamically linked objects create a virtual channel. Our approach enables software portability between hardware versions as well as enabling participation by our teammates and third party software and waveform developers. Our approach provides for a software programmable, scaleable and reconfigurable design.

As depicted in *Figure 4*, the WITS software architecture is designed to support portability, through use of clearly defined interfaces and layering to insulate the applications from changes in operating system and hardware implementations. For example, using the software architecture shown in *Figure 4*, Motorola and its teammates are able to use a Microsoft® Windows NT® development environment using Intel® Pentium® processors and directly move the code developed in the Windows NT environment to Motorola MPC750 Single Board computers using the VxWorks® operating system by simply recompiling the code for the MPC750 target. This allows us to use development tools available for Microsoft Windows NT, which significantly reduces the time required to develop new waveforms and applications for the WITS products. The use of a non-real-time operating systems such as Microsoft NT or Sun Solaris™ can be used for functions that do not require real-time processing.

WITS HARDWARE ARCHITECTURE

Figure 5 illustrates the hardware architecture for the WITS 5004. The WITS products share an architecture designed to be scalable and extensible to support multiple implementations and reconfigurable to support operational variations. As such, the Line Replaceable Unit (LRU) boundaries are well defined allowing the addition of optional hardware and software to support features such as a point solution dedicated GPS channel, a cellular telephone or non-standard I/O connectivity. LRUs include Transmit/Receive (T/Rs), Preselectors, Modem, Networking, and INFOSEC. The WITS hardware architecture provides commonality among product family members ensuring long-term support and upgradability.

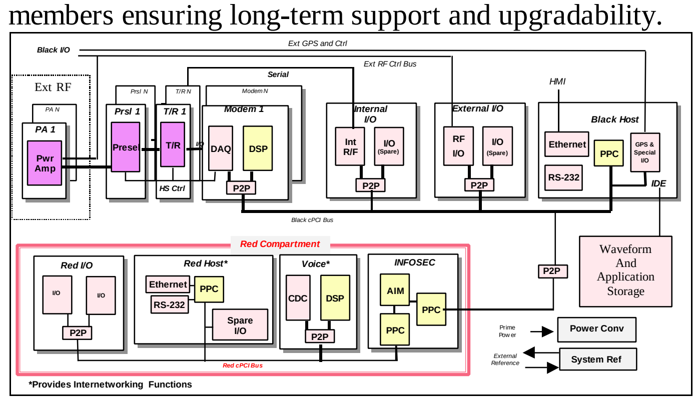


Figure 5. Physical Architecture for 5004

0-7803-6521-6/00/$10.00 (c) IEEE

The driving requirements for the WITS architecture include the ability to support complex waveforms such as Link-16, EPLRS, Project 25, DAMA, HDRLOS, Tetra, *etc*. and to support future defined high data rate mobile dynamic networked waveforms in excess of 10Mbps. Physical modules are depicted as the shadowed boxes. The external RF is platform dependent and varies between installations and therefore kept separate from the unit.

INTEROPERABILITY APPLICATIONS

Most radio communications devices developed in the last decade have operator control systems that are software defined and may also include the capability to configure the active communications channel through software selection. However, to meet user needs, the SDR must go beyond software control capabilities and provide services that allow disparate systems to communicate. It allows communications connectivity between dissimilar radio sources without added equipment. Such is the case when military and civilian units join forces for disaster relief or when one military service resources wishes to communicate with another's resources. For example, military personnel with a radio communicating with a SINCGARS waveform cannot directly speak to Emergency personnel using Project 25. *Figure 6* depicts SDR functions that offer the user interoperability. These include crossbanding, transcoding, transcryption, and transmodem.

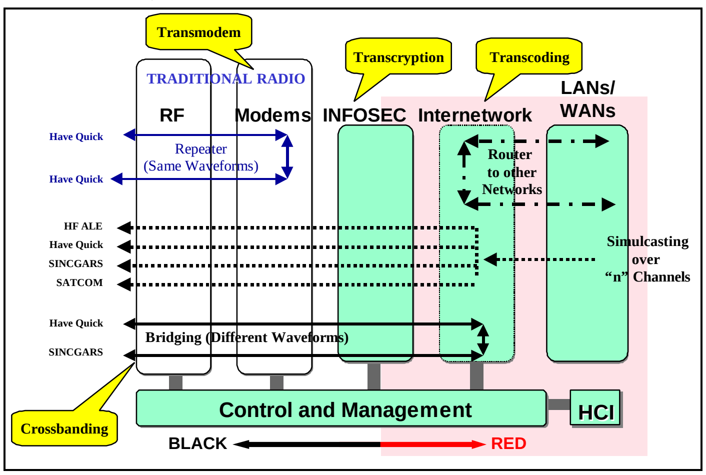


Figure 6. SDR Functions

Crossbanding allows real-time interoperation between different frequency domains. Transmodem, transcryption, and transcoding provide bi-directional links between dissimilar RF communications networks or systems without additional equipment by providing the translations of modulation, encryption, and vocoding from between systems. To enable these translations, waveform applications are decomposed into modular parameterized software components linked at load and/or execution time. Waveform parameters include:

- Modulation type and characteristics
- Digital link, network, and transport protocol(s)
- Source coding
- Security type or algorithm
- Communications channel bandwidth
- Frequency

Local and remote configurability allows an SDR to be configured from anywhere in the world. Users can select a 'preset' for each channel or control the modes and operating condition of any or all RF ports. Unlimited presets, which establish the waveform, crypto algorithm, operating modes, *etc.*, can be stored for each RF port and quickly selected by the operator to configure the system. This feature enables SDR equipment to be configured prior to a mission for specific characteristics that an operator can quickly, without error, set up when needed. With the large number of presets available, numerous complete configurations can be determined in advance and/or saved as field adapted and recalled.

APPLICATIONS DEVELOPMENT

A software defined radio provides applications flexibility in the field and growth over time to meet changing mission requirements and new software technology without hardware changes. As discussed earlier, WITS products are fully software re-programmable and configurable. This increases the lifecycle of the equipment, reduces the cycle time for upgrade, and supports field changes. WITS waveform applications are operator configurable through manual selection or presets. A Windows®-based user interface simplifies radio operation. User manuals and on-line help support the user. New applications may be installed and stored to disc and instantiated as needed.

Modular object decomposition of waveform applications and development of reusable software objects enable software portability and improve maintenance. WITS waveform applications consist of software objects uniquely interconnected and operating on associated hardware LRUs. WITS software objects provide RF programming and control, signal processing, security, networking, voice, and HMI. Objects are parameterizable offering the ability to tune in real-time and select options without software or hardware additions. Our approach allows new capabilities, reconfigurations, and characteristics to be provided—all via software.

Waveform applications may also share software objects. For example, signal processing is decomposed into elements of modulation/demodulation, encoding/decoding, *etc*. When elements are connected into a waveform to meet specific operating requirements, the appropriate modulation scheme is invoked. This approach enables upgrades with minimal development and impact to operational code and provides a generous reuse base for new or modified applications. *Figure 7* illustrates the

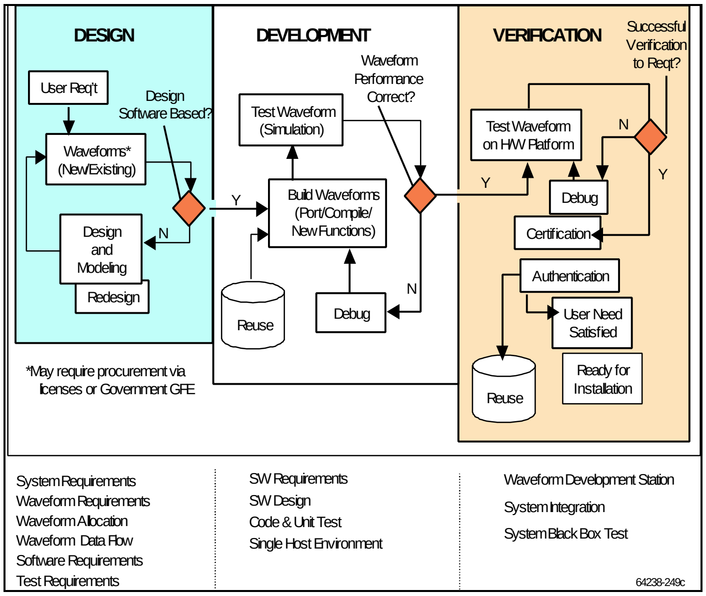


Figure 7. Waveform Development Process

Waveform Development Process used by Motorola. Waveform development is covered in a companion paper "Waveform Application Development Process for Software Defined Radios."

TRADITIONAL STATION EQUIPMENT VS. SDR STATION EQUIPMENT

How does the traditional station equipment differ with a Software Defined Radio? More importantly, how does the service provider meet user needs in a cost-effective manner? *Figure 8* summarizes the Traditional vs. Software Defined Radio approaches to provide station equipment.

Traditional station equipment typically includes multiple specialized RF transceivers, wireline modems, multiplexers, and routers that are linked through communications services. An experienced systems integrator ensures that a cohesive system of compatible interfaces emerges and that those interfaces function as needed. When upgrades are required, knowledgeable personnel must be retained to determine how the upgrade should proceed.

In contrast, SDR equipment is defined by the number of wireless and wireline interfaces needed at any particular instantaneous time to provide the services regardless of the frequency band, modulation type or format. Software applications loaded on the SDR equipment enable all elements of the needed services. The difference is clear in the amount of complexity reduction afforded by SDR technology. This fact alone can often justify the replacement of traditional racks of "point solution" radios.

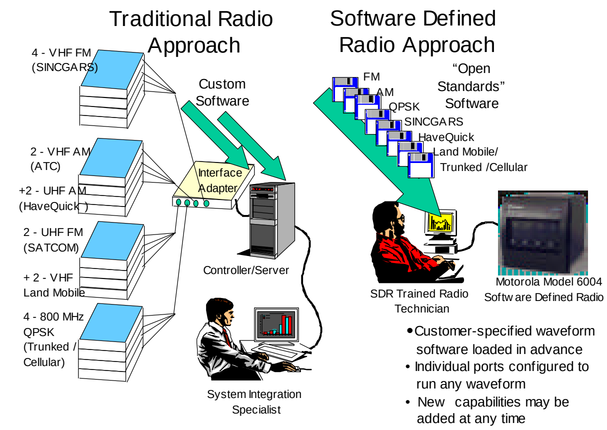


Figure 8. Traditional Radio vs. Software Defined Radio Approaches

SUMMARY

Communications with Software Defined Radios, as applied in complete wireless communications transfer systems act significantly different than traditional radio network implementations. In fact, the revolutionary features noted here and those built from them will probably alter the basic structure of radio networks in ways yet unforeseen. Motorola's Wireless Information Transfer Systems team is working to provide cost effective, flexible solutions where many channels of various legacy equipment are now deployed.

BIBLIOGRAPHY

[1] E. Christensen, A. Miller, B. Tarver, "Object Management and Base Class Interfaces in the Distributed Computing Software Radio Architecture", Submission to the SDR Forum, November 1999.

[2] Programmable Modular Communications System (PMCS) Guidelines, 1997.

[3] "Operational Requirements Document (ORD) for the Joint Tactical Radio (JTR)", March 28, 1998.

[4] P. Cook and W. Bonser, "Architectural Overview of the SPEAKeasy System", May 18, 1998.

[5] TINA, Telecommunications Information Network Architecture, 1995.

[6] P. Cook and M. Glenn, "Software Defined Radio Forum Mobile Working Group Software Framework Architecture", March 30, 1999.

[7] A. Miller, et al., "Technical Summary of the Wireless Information Transfer System", WITS white paper, October 1999.

[8] E. Christensen, A. Miller, E. Wing, "Waveform Application Development Process for Software Defined Radios", April, 2000.

[9] G. Hollingsworth, "The Application of Software Defined Radios", WITS white paper, October 1999.

0-7803-6521-6/00/$10.00 (c) IEEE