AN SCA-COMPLIANT, MICROPHONE TO ANTENNA SOFTWARE RADIO

Stephen P. Reichhart (Air Force Research Laboratory; Rome Research Site, New York, reichharts@rl.af.mil); Bruce Youmans (PAR-RRC; Rome, New York, Bruce.Youmans@rl.af.mil)

ABSTRACT

In 1998, the Air Force Research Laboratory (AFRL) Information Grid Connectivity Branch (IFGC, Rome Research Site) teamed with PAR Rome Research Corporation (RRC) to design, develop and implement a software reprogrammable, hardware reconfigurable, wireless communications testbed. The purpose of this testbed was to develop and evaluate new communication techniques and algorithms. The Software Radio Development System (SoRDS) is a fully programmable and reconfigurable software radio, based on a parallel computing platform using the Linux operating system, and capable of transmitting and receiving voice, data, video, or network traffic at RF frequencies from HF through L-band. SoRDS is a portable platform that enables rapid development and demonstration of wireless communication applications. SoRDS was programmed with the Future Multiband Multiwaveform Modular Tactical Radio (FM3TR) frequency hopping voice waveform, and has demonstrated interoperability with other FM3TR-programmed software radios on numerous occasions both in the US and Europe. SoRDS is also used as a testbed for other waveforms, such as VHF FM, SATURN, and streaming video.

To maintain AFRL-IFGC's expertise in software radio technology, a transition of SoRDS capabilities to a Joint Tactical Radio System (JTRS) Software Communications Architecture (SCA) compliant architecture is in progress. AFRL-IFGC and RRC have teamed with L-3 Communications Analytics Corporation (LAC) who developed a JTRS SCA version 2.2 compliant Core Framework (CF) for the JTRS Technology Laboratory (JTeL) supporting the JTRS Joint Program Office (JPO). The CF is hosted on a Linux operating system running on a general purpose PC in conjunction with The ACE Orb (TAO) from Washington University. The team will transition the SoRDS' RF capability to a Linux based PC platform. Then the team will integrate the RF capability with an SCA compliant operating environment and demonstrate an SCA-compliant FM waveform.

This paper will describe the process, procedures and performance in transitioning RF-capable technology from a field-proven software radio to an SCA-compliant PC based system. We will discuss the development of a complex SCA-compliant FM3TR waveform application operating from microphone/speaker to antenna, and the advantages of using the Linux operating system for implementing this complete, JTRS SCA-compliant radio system.

1. INTRODUCTION

This article focuses on the effort to advance the field of software reprogrammable radios and the building of a software radio testbed to develop advanced radio waveforms and techniques. The Air Force Research Laboratory (AFRL) has performed research on software reprogrammable radios since 1989 when the SPEAKeasy program was begun to determine if a multiband, multimode software radio was technically feasible. The AFRL testbed aids in the development of advanced waveforms and radio techniques regardless of the processors or system architecture used by the intended software radio platform.

The testbed (called SoRDS) was developed by a team comprised of AFRL and PAR-RRC personnel. SoRDS is a fully programmable and reconfigurable software radio, based on a parallel computing platform and is capable of transmitting and receiving voice, video or network traffic. SoRDS is a portable platform that enables rapid development and demonstration of wireless communication applications. It transmits or receives RF over a frequency range of 30 MHz to 2500 MHz and supports frequency hopping in excess of 2000 hops per second.[i]

2. JOINT TACTICAL RADIO SYSTEM

The Joint Tactical Radio System (JTRS) will combine the functionality of numerous single function radios among the services into a single, Joint-interoperable family of radios. The JTRS will first meet Joint, then Combined, then Alliance / Coalition interoperability by providing tactical radio sets that may include routers, switches, and other networking components/functions integral to the set and configured to meet the diversity of host platforms. The JTRS satisfies requirements common to the three domains that coincide with operational missions and environments: Airborne, Maritime/Fixed, and Ground. The radio sets in the JTRS will be software-reprogrammable, multi-band/multi-mode capable, network able, and provide simultaneous voice, data, and video communications.

The functionality and expandability of the Joint Tactical Radio System is built upon the Software Communications Architecture (SCA). The SCA is an open architecture framework that tells designers how elements of hardware and software are to operate in harmony within the JTRS. It governs the structure and operation of the JTRS, enabling programmable radios to load waveforms, run applications, and be networked into an integrated system. Design engineers use the SCA definition document just as an architect or planner uses a local building code to design and build homes.

Through adherence to standards detailed in the SCA definition document, both hardware and software designers know what equipment and programs to design. The SCA does not tell designers how to design their equipment and programs. Thus, JTRS compliant radios and networked systems, when designed in compliance with the SCA, will meet JTRS standards for interoperability, just as properly designed plumbing or electrical systems meet local codes for construction and safety.

The SCA thus provides a development rule set focused on the detailed radio set, waveform and software development standards and specifications that describe what to build to make the system interoperable and to have interchangeable equipment, software and other network components. The JTRS is composed of loadable waveforms, radio sets, and network management software. Radio sets are tied together into a system using network management procedures and waveforms forming the JTRS. All components of the JTRS (i.e., waveforms, hardware, network management software and networking hardware) will be SCA compliant.

The SCA is not a system specification, as it is intended to be implementation independent, but a set of rules that constrain the design of systems. The Operating Environment (OE), consisting of the CF, Common Object Request Broker Architecture (CORBA) middleware, and Operating System (OS), imposes design constraints on waveform and other applications to provide increased portability of those applications from one SCA-compliant radio platform to another. These design constraints include specified interfaces between the CF and application software, and restrictions on waveform usage of OS Application Programming Interfaces (API). The SCA also provides a building block structure (defined in the API Supplement) for defining application software component APIs. The building-block structure for API definition facilitates component-level reuse and allows significant flexibility for developers to define waveform-specific APIs.[ii]

3. AFRL'S SCA-COMPLIANT TESTBED

The AFRL software radio testbed was developed and implemented before the JTRS SCA guidelines were developed. To become SCA compliant SoRDS had to be reworked with the JTRS SCA in mind.

3.1. AFRL Linux Core Framework

AFRL and L3-Comm have developed the Linux Core Framework (LCF) for the JTRS Technology Laboratory (JTeL). AFRL's LCF is currently the only fully JTRS-certified Core Framework on any operating system. During the certification process, the LCF was tested and verified with the JTRS Test Application (JTAP) under the supervision of JTeL engineers. AFRL's LCF was developed under Linux Red Hat version 7.3, and has been tested and verified through version 9.0.

The LCF is based on the Adaptive Computing Environment (ACE), a freely available, open-source, object-oriented (OO) framework that implements many core patterns for concurrent communication software. ACE provides a rich set of reusable C++ wrapper facades and framework components that perform common communication software tasks across a range of OS platforms. ACE is targeted for developers of high-performance and real-time communication services and applications. It simplifies the development of OO network applications and services that utilize inter-process communication, event de-multiplexing, explicit dynamic linking, and concurrency. In addition, ACE automates system configuration and reconfiguration by dynamically linking services into applications at run-time and executing these services in one or more processes or threads.

The CORBA middleware used was "The ACE Orb" (TAO). TAO is a freely available, open-source and standards-compliant real-time implementation of CORBA that provides efficient, predictable, and scalable quality of service (QoS) end-to-end. Unlike conventional implementations of CORBA, which are inefficient, unpredictable, non-scalable, and often non-portable, TAO applies the best software practices and patterns to automate the delivery of high-performance and real-time QoS to distributed applications. The flexibility of the features provided by the TAO ORB and its CORBA services simplified the development of the LCF.

4. THE WAVEFORM APPLICATION: FM3TR

The Future Multiband Multiwaveform Modular Tactical Radio (FM3TR) digital communication waveform was developed by AFRL engineers as part of the FM3TR multinational consortium. This waveform has features generically similar to military waveforms including frequency hopping and frame synchronization, but without the features that would require classification.

AFRL and its agents have experience in developing software implementations of FM3TR for SDRs, building RF-capable SDR development systems and developing the LCF. Since AFRL is recognized as a leader in SDR and communication waveform development, the JTeL tasked AFRL to develop and demonstrate a JTRS SCA-compliant FM3TR software waveform on a PC-based platform. This waveform and the JTRS Operating Environment (JOE) on which it is demonstrated includes no classified information or Intellectual Property (IP) rights which prevent sharing the waveform with foreign nations and foreign corporations. The FM3TR waveform will be compliant with version 2.2 of the JTRS SCA. The waveform application is designed such that it can be eventually ported to a TBD target SDR platform to support the JTRS JPO's international objectives.

This SCA-compliant FM3TR Software Development Project software package will include: a) An Operating System (Linux) per SCA v2.2, Section 3.1.1. b) Middleware and Services (ACE/TAO) per SCA v2.2, Section 3.1.2. c) A Core Framework per SCA v2.2, Section 3.1. d) An FM3TR Waveform Application per SCA v2.2, Section 3.2. e) FM3TR APIs per SCA v2.2, API Supplement.

The software package follows Logical Device and General Software Rule requirements of SCA v2.2, Sections 3.3 and 3.4, respectively. The Waveform development will include software resources, JOE devices, and associated XML files consistent with the SCA section 3, Appendix D Domain Profile, and the API Supplement.

The FM3TR Waveform includes three SCA resources – Audio I/O, FM3TR Media Access Control (MAC) layer, and Assembly Controller. Three CF Devices are required to execute the waveform – Audio, General Purpose Processor (GPP), and Modem. A standard PC Sound Card will provide microphone input and speaker output; the Human-Machine Interface (HMI) will be a console interface; the Pseudo RF Simulator will simulate the RF for the PC-platform, since an actual RF capability is no as yet defined.

4.1. Three Phases of Development

Phase 1: Creation of the SCA-compliant Audio Device, FM3TR CVSD and FM3TR MAC resources, and the MODEM device software modules. An FM3TR "thread" will then be developed that performs sound recording, and Continuous Variable Slope Delta (CVSD) voice encode and passes unchanged through all the software components, then writes to a file in transmit mode. In receive mode, the CVSD data will be read from the file, passed unchanged through all components, CVSD voice decoded and sound playback performed.

Phase 2: Expansion of the FM3TR thread developed in phase 1 to add preamble transmission and format of CVSD symbols into frames with synchronization in transmit mode. In receive mode added features will include preamble correlation and acquisition, extraction of symbols from the frames, frame synchronization and EOM detection.
The MODEM device was modified to include modulation and demodulation software just prior to or following file access, in transmit and receive modes, respectively.

Phase 3: To validate the LCF and ensure the FM3TR waveform meets real-time criteria; the team will integrate the SoRDS software and hardware, creating a complete, wireless SCA-compliant SDR.

5. COMMUNICATION WAVEFORMS

AFRL and its agents are experienced in developing SDRs and programming them with highly agile, frequency hopping communication waveforms. To validate the LCF and SCA-compliant FM3TR waveform, the FM3TR application will be made wireless by modifying the software to support frequency hopping and the addition of the SoRDS RF interface hardware to the PC platform. The RF interface hardware is supported by software libraries which will be built into the FM3TR application project.

AFRL has learned from their experience in SDR development that two critical tasks must be performed to make frequency hopping concur at the precise points in the waveform symbols in transmit and receive modes:

In transmit mode, the commands that control frequency hopping must be merged with the transmit data symbols, and these commands must be positioned in the data stream. Therefore, the controls words and the data will have equal delays through the principle latency element, the transmit data buffer.

In receive mode, the time delay from when a waveform symbol is received at the antenna and the symbol is processed in software must be measured. Further, the time from the output of a command from software to perform a frequency hop until the RF subsystem performs the hop must also be calculated. These symbol latency and hop command latency time values must be calculated for each new frequency hop and will be used to determine when to send the next frequency hop command to the IF/RF subsystem.

These two critical tasks require that the data used for latency calculations, and the calculation operations be tightly coupled to waveform data processing. The latency calculations can be in units of time, symbols or samples. Symbols or samples can be represented in buffer levels, assuming the buffers are filled or emptied at a constant rate by a physical device. More important, the retune time interval specified in the FM3TR waveform limits the maximum combined size of the transmit / receive buffers and establishes a time constraint on receiver processing response time. The challenge is in implementing these deterministic tasks within the specifications of the SCA.

AFRL's implementation of FM3TR on an SDR is a complete frequency hopping transceiver without employing the too-common practice of pushing deterministic functions to the driver level, risking portability between platforms. AFRL's project goals of an SCA-compliant, frequency hopping, portable FM3TR waveform application executing on an SDR with full RF capability have been met. Demonstrations are planned for January 2004.

6. SUMMARY

The team of AFRL and its agents has completed the SCA-compliant LCF, SCA-compliant FM3TR and SDR implementation. In-lab demonstrations have proven successful FM3TR interoperation between the SCA-compliant SDR and pre-SCA SoRDS 2000. The FM3TR waveform with a frequency hop rate of 250 hops/sec was supported with a table of up to 30 frequencies. Final SCA certification of AFRL's FM3TR waveform application by JTeL is pending but immanent.

The experience obtained in developing AFRL's first SCA-compliant waveform application for an SDR will assist in future JTRS waveform development. These projects will include secure voice waveform applications including SATURN, PLUTO, and public safety waveforms. AFRL will also begin the development of a waveform application test bed with RF capability for use by JTeL in their validation and certification process.

---

[i] "SoRDS: Platform for Voice/Video/Network Radio", by S. Reichhart, D.Benfey, and B. Youmans, MILCOM 2001 Proceedings, October 28-31, 2001.
[ii] http://jtrs.army.mil/