Developing SCA-Next based OMAP hosted Waveforms for Emerging Communication Standards

November 2011

Point of Contact:
Amy Corman
SDR Business Development Mgr.
DataSoft Corporation
1275 W. Washington St., Suite 106
Tempe, AZ 85281
Phone: (480) 763-5777 x447
email: amy.corman@datasoft.com

DataSoft

1 Executive Summary

The commercial, public sector and military wireless communications systems demand radios with greater functionality and capacity at lower costs for interoperability, technology evolvement, waveform development and system upgrades. Software Defined Radio (SDR) technology promises to revolutionize the communication industry by delivering flexible radio platform(s) that can host a variety of software solutions for wireless communication protocols at a lower-overall cost. SDR technology enables the exploration of concepts such as dynamic spectrum access, cognitive radio networks, emerging standards and IP development.

The transition of communications systems from static, hardware based platforms to configurable, flexible software defined platforms has been a focus and priority in wireless communications for the past decade. In commercial systems, SDR is useful for demonstrating IP cores and new waveforms, for PHY-MAC development and for next generation base station technology where one installation must support several waveforms based on both legacy and future cellular standards. The military has fully embraced SDR technology through the Joint Tactical Radio System (JTRS) program. The JTRS radios are a family of interoperable, modular, software-defined radios that operate as nodes in a network that provides secure wireless communication and networking services for mobile and fixed forces including U.S. allies, joint and coalition partners and disaster response personnel [1]. The current JTRS system will use up to 9 waveforms that will interoperate with 13 SDR-based hardware platforms. The Software Communications Architecture (SCA) specification is the basis for all JTRS software that was created to maximize software application portability, reusability and scalability while providing flexibility to address specific applications [2].

The SCA specification is inviting to waveform (WF) developers as it is a *standards-based framework* and is a basis for evolving software reuse and portability in SDRs. The ability to develop and port waveforms to different hardware platforms is the main purpose for the SCA specification and SCA-compliance. This capability benefits the WF engineer by providing the opportunity to use a low-cost HW and SW system for development, research and investigation and then the ability to efficiently port to the final hardware platform which may be unavailable, difficult to acquire or too expensive to use in development.

This paper will present a platform developed to assist the modern SCA/SDR/Modem engineer to rapidly develop new waveforms for cognitive networks, white spaces and other emerging standards. The paper is organized as follows: an overview of the capabilities of Software Defined Radio (SDR) to develop new waveforms and protocols, to create new IP cores or to investigate existing waveforms, followed by a summary of SDR enabling technologies, and finally a method to extend a non SCA-compliant waveform, or an existing IP core to SCA-compliance using the Spectra DTP4700E SDR Platform – the latest offering in the affordable, wide-band, high performance baseband and RF development family from PrismTech and DataSoft, available January 2012. For non SCA-compliant waveform development, the SDR hardware platform, Thunder is available from DataSoft. Figure 1 details the software stack for each system.

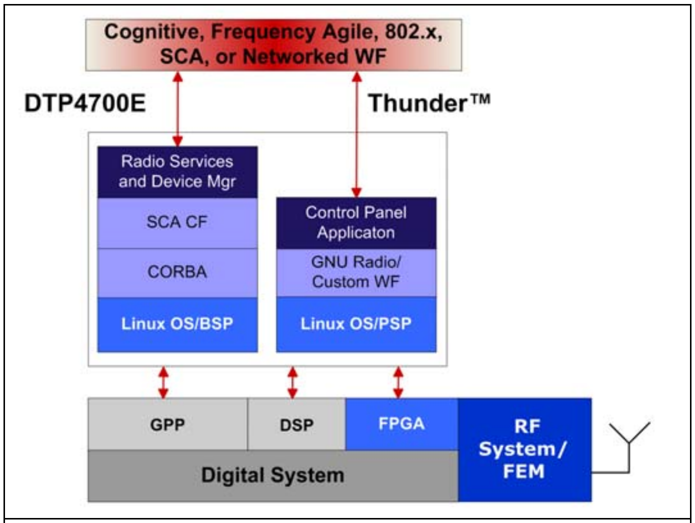

Figure 1: Thunder and Spectra DTP4700E platform software stacks

DataSoft

2 The SDR Technology Advantage

A SDR system has a generic hardware platform on which software runs to provide functions including modulation and demodulation, filtering, signal bandwidth, carrier frequency and frequency hopping. With a change to a software parameter – either in real-time or as a new software load - the performance of the radio can change. In order for the radio to re-configure itself, the software runs on programmable hardware processors. Typically present in a SDR system are a general purpose processor (GPP), a digital signal processor (DSP) and a field programmable gate array (FPGA). SDR reconfigurability enables the exploration of concepts such as dynamic spectrum access, cognitive radio networks, emerging standards and IP development.

In addition to SDR reconfigurability, another major advantage of SDR technology is waveform portability. For a waveform to truly be interoperable with military and civilian systems, it must be SCA-compliant. The SCA specification was created by the JTRS program and industry partners to ensure waveform interoperability between platforms.

In short the benefits of using SDR Technology include:

- **Cost savings:** Waveforms for emerging commercial, military and public safety communications take many engineering months to develop. The ability to re-use waveforms on new and different platforms significantly reduces project start-up costs and could offer communication engineers both time and budget for new innovations. Changes to standards, system upgrades and fixes can occur via software – eliminating the need and cost of replacing fielded hardware systems. Various agencies, whether military or public safety can minimize the many different types of radios required to operate on legacy waveforms and converge on a few common platforms for multiple radios based on the application need (hand held, basestation, etc.)
- **Obsolescence mitigation:** Since SCA-compliant waveforms are interoperable between hardware platforms, part obsolescence on a particular hardware device becomes a minimal issue since other platforms can operate the same software architecture. SDR platforms become "future proof" through flexible architectures.
- **Interoperability:** Public Safety, Military and Commercial wireless devices benefit from interoperability gained from SDR technology. Interoperability implies the waveforms can be used across all platforms with minimal modification, legacy equipment can interact with new SDR interoperable radios and radios designed for specific applications (handheld, airborne, basestation, maritime) have the same baseline architectures while satisfying the specific demands of the application. JTRS leads the effort of promoting interoperability through the SCA architecture; however as the technologies used for all wireless communications and data continue to emerge, interoperability allows technology to evolve while maximizing value of legacy systems. SDRs can adapt to future protocols as they are adopted, eliminating the need for costly equipment upgrades.
- **Spectrum Efficiency and Spectrum Sharing**: The demand for spectrum for wireless communications and data exceeds the capacity with no resolution in sight. The current spectrum allocations of fixed frequency assignments for one purpose are inefficient and will quickly lead to a "spectrum crises". By using SDR technology in a cognitive application like Dynamic Spectrum Access - a method of secondary spectrum use on an opportunity basis, or in an unlicensed band application, wireless systems can continue to support growth and changing demands in traffic while still supporting critical public safety, military and emergency operations.

DataSoft

2.1 SDR Enabling Technology Component: SCA and Software

To support global standardization and interoperability of military radios, the U.S. Department of Defense JTRS program and the industry created the Software Communications Architecture (SCA).

The SCA is an open architecture framework that tells designers how elements of hardware and software operate within the SDR platform. It governs the structure and operation of the JTRS or other SDR platform, enabling programmable radios to load waveforms, run applications, and be networked into an integrated system.

The SCA defines standard interfaces that allow waveform applications to run on multiple hardware sets. The SCA operating environment (OE) is a basic set of core services and component interfaces for waveform execution and portability. The OE combines the set of Core Framework (CF) services, interfaces, board support packages, operating system and middleware services to host an application.[4] Figure 2 describes the architectural structure of the SCA (version 2.2.2 and SCA-next).

SCA compliance ensures hardware and software from one vendor will work with the hardware and software from another vendor. Per the JTRS SCA-next specification, “an application consists of multiple software components that are loaded onto the appropriate processing resource. These components are managed by an implementation of the Framework Control Interfaces. The application components communicate either with each other or with the services and devices provided by the system through extensions of the SCA-defined PortAccessor interface. Similarly, communications between the application and the Framework Services Interfaces are accomplished through the Transfer Mechanism (CORBA in version 2.2.2)”. [5]

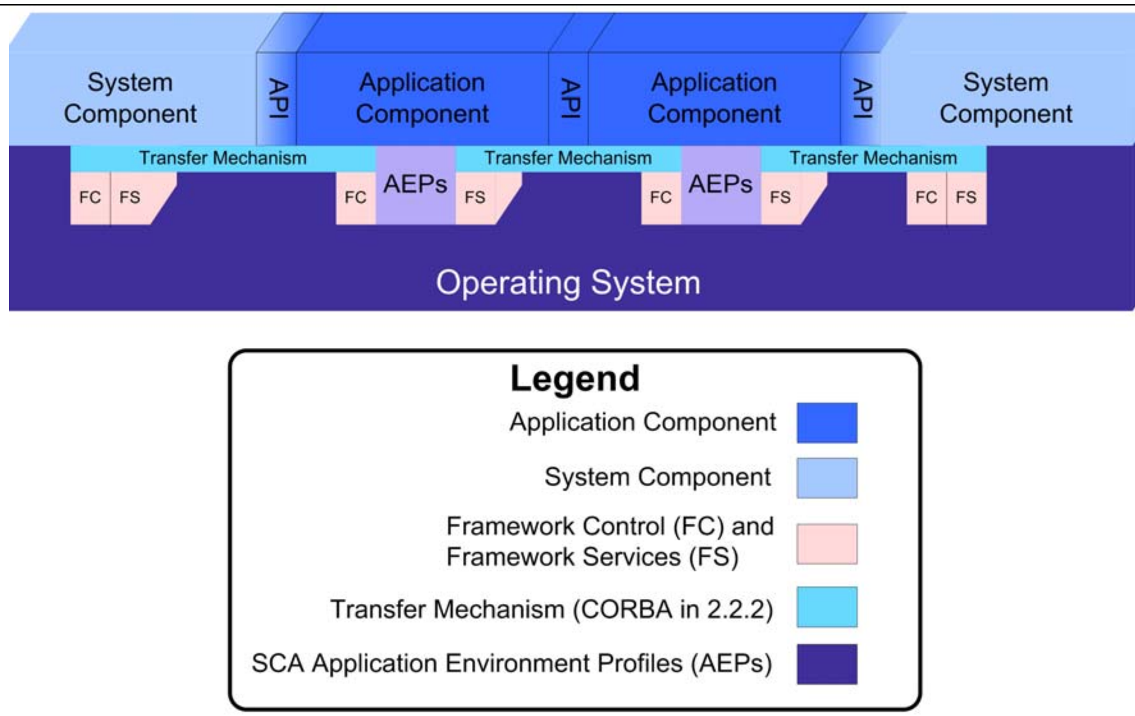

Figure 2: SCA architecture layer diagram

2.2 SDR Enabling Technology Component: Hardware Architecture

To realize the JTRS vision, a SDR platform, capable of implementing complex communications waveforms is required. The SDR platform has unique design challenges and, unlike the familiar

DataSoft

low-power wireless handsets, must cover wide frequency bands and remain linear over wider bandwidths to support networking waveforms required for both military, homeland security and commercial SDRs.

SDR hardware must be flexible and broadband with software control of frequency, modulation, channel bandwidth, security and other waveform requirements such as linearity, frequency hopping and power output. The typical SDR hardware architecture has three basic components: The baseband processor(s), the digital conversion section and the RF front end. The baseband generates waveforms to transmit and analyzes waveforms received by the RF front end, and digitized by the converter. The ultimate SDR has the digital conversion right at the antenna, allowing for complete software control of the radio; however the speeds of the converters are not yet viable for this. Direct digital conversion is a method which converts the RF to baseband I/Q signals, eliminating the analog IF stage, allowing most of the signal processing to occur at baseband. The architecture DataSoft features for the digital conversion provides a very low phase noise frequency source to the receive components and can be digitally modulated to produce all transmitter modulation types. Figure 3 shows the block diagram of the DataSoft SDR hardware platform used for the Thunder and the Spectra DTP4700E SCA-compliant waveform development platforms.

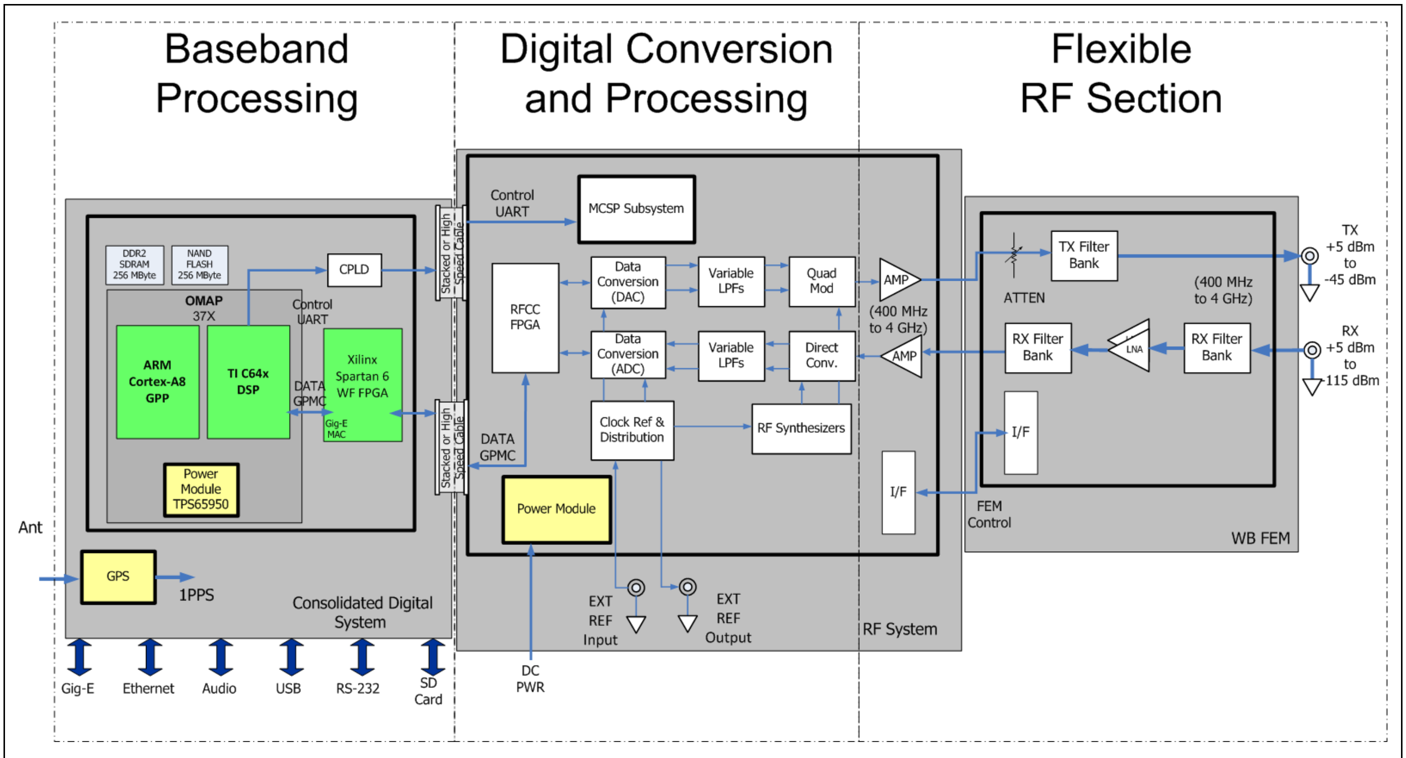

Figure 3: DataSoft SDR hardware system used for Thunder and for Spectra DTP4700E waveform development platforms

DataSoft

Enabling technologies:

RF Front End: For received signals, the front end functions are to receive the signal, perform pre-selection, image rejection and harmonic filtering functions. The receiver functions are set and controlled by software and achieved with tunable filters, programmable attenuators, switches and detectors. The transmitter uses the same techniques in the opposite direction.

Digital Conversion: In the receive direction, the RF signal is downconverted to baseband by DDS-driven direct conversion. Technical concerns that plagued direct conversion design in the past, such as precise quadrature splitting and amplitude balance of the LO signals, have been resolved by IC manufacturers who support the wide bandwidth cellular data and satellite communications markets.[3] Channel selectivity is accomplished via programmable state variable filtering (SVF). The SVFs have a very high dynamic range so they can remove strong interfering signals while passing weak desired signals. Finally the baseband digital I/Q samples are converted to a 12-bit digital representation of the information with DC offsets for processing in the FPGA, DSP or GPP.

In the transmit direction, the DataSoft SDR utilizes a direct-conversion, quadrature modulator to convert the digital data to RF. The low distortion enables the modulator to achieve high output power levels with minimal adjacent channel leakage. This in turn allows for less gain in subsequent stages of the transmitter. Modulation begins with the filtered baseband signals from the 16-bit DAC which drive the I/Q inputs of the quad mod. The DataSoft transmitter uses careful proprietary methods to control the LO leakage and any non-ideal quadrature splitting or gain mismatch between the I/Q channels.

Baseband Processing: The most flexible SDRs have a General Purpose Processor (GPP), a Digital Signal Processor (DSP) and a Field Programmable Gate Array (FPGA) processor. The GPP controls the radio functions and has sufficient processing capability for simple waveforms. The OMAP processor used in the DataSoft SDR includes an on-chip DSP. Complex computational work can be off-loaded to the DSP while the GPP can focus on the control of the SDR. Complex waveforms like JTRS WNW, MILSATCOM, and LTE–A require the parallel processing of FPGAs to perform signal processing. FPGAs offer higher processing rates and bit resolutions for complex FIR filtering, and FFTs. Splitting processing algorithms between the FPGA and DSP can optimize power consumption and performance.

3 Platform Description

DataSoft has developed a complete SDR development platform populated with the latest component technology, resulting in an optimized, small form-factor platform, with low power consumption, that enables the development, testing and deployment of waveforms. Both Thunder and DTP4700E support the implementation and deployment of current and next-generation SCA-compliant networking waveforms required for military, homeland security and commercial SDRs. The hardware design incorporates innovative components and demonstrates superior power consumption with wide bandwidth, high performance RF selectivity and re-programmability for waveform processing.

The platform provides the entire signal chain hardware from antenna to baseband as well as a software board support package that supports a complete suite of software development tools in a single integrated development platform.

With this platform, developers can easily design SCA-compliant waveforms as well as create and test single or multi-protocol radios for applications in military, public safety, commercial, professional mobile radio (PMR) and land mobile radio (LMR) communication systems. The platform includes application development tools, platform software and sample applications allowing the communications engineer to focus on waveform development without becoming an SCA expert. Figure 4 details the software and hardware configurations of each platform.

DataSoft

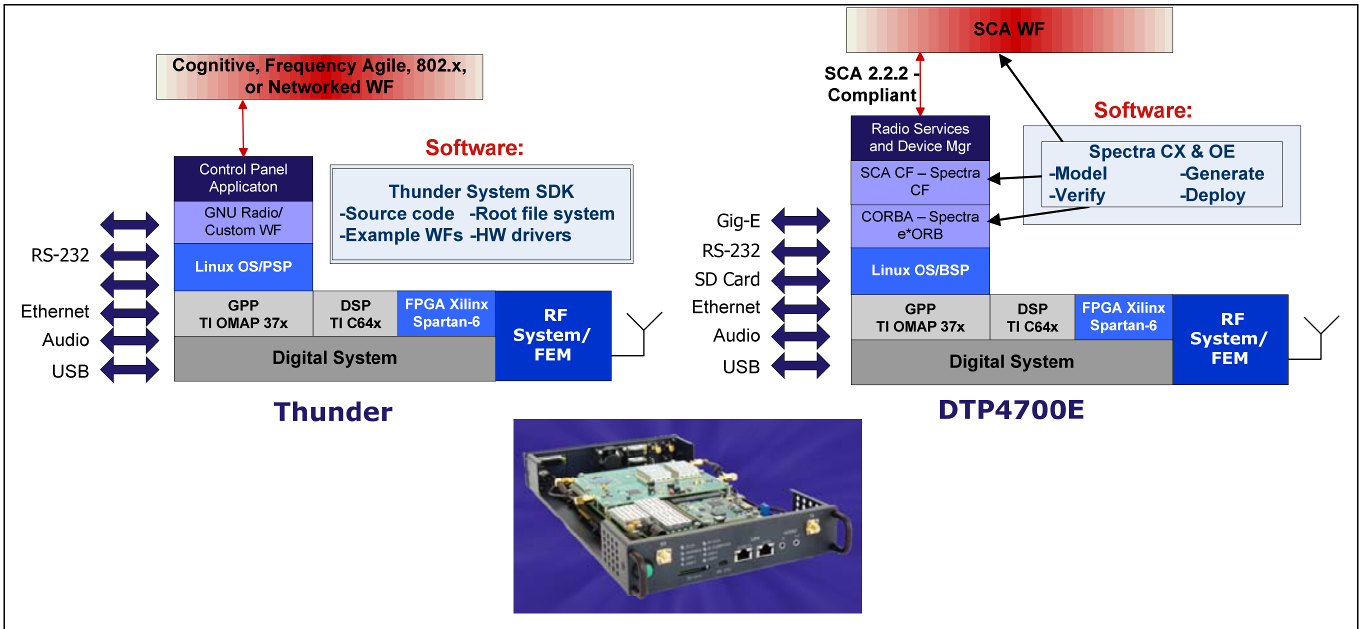

Figure 4: Thunder and Spectra DTP4700E platforms SW and HW stack

4 The Future of SDR Waveform Development Using a Processor Efficient Approach

DataSoft's new development approach of using GNU radio and SCA-next simultaneously to synthesize new waveforms offers unique benefits. First, GNU radio implemented and hosted on the OMAP processor offers a highly efficient signal processing engine. Second, new proposals of SCA-next potentially permit the use of a highly optimized and unique non-CORBA transfer mechanism. This implies that we are no longer shackled to use, from a radio engineer's perspective, inefficient CORBA middleware to produce a communications solution, and instead can focus on solving unique spectral and signal processing problems, PHY-MAC cross-layer design optimizations and finally work on providing wireless MANET style commercial multi-band, and networked waveforms operating in white space bands – the real challenges that SDR can solve.

The development approach, shown in Figure 5, introduces an added feature in that we do not make use of Python to create the GNU radio signal processing chain, but rather use the SCA framework minus SCA ports and CORBA. This allows for a highly efficient and small C-based waveform and platform solution more akin to the software systems of current 3G, 4G and LTE solutions. This approach dramatically reduces processor utilization in middleware and un-necessary SCA-CORBA infrastructure, and can now redirect that utility for signal processing and working at reducing power consumption overall. This rare and unusual scenario now means waveform engineers and implementers alike are freed up to focus totally on waveform design delving deeper in to algorithms that achieve greater spectral efficiency.

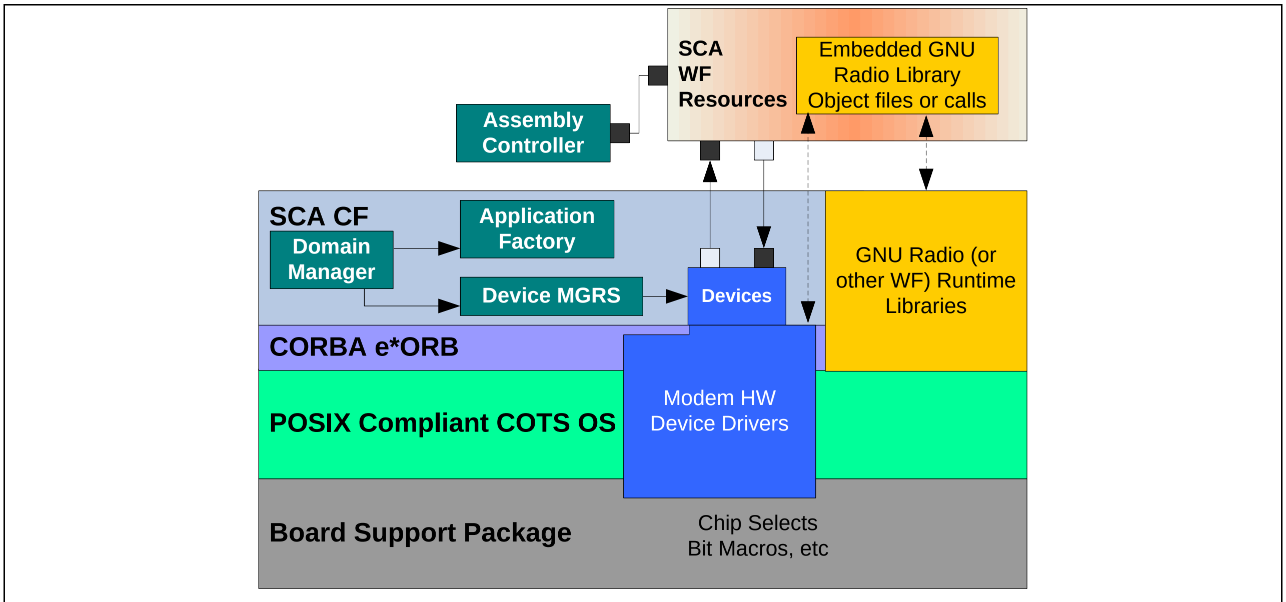

Figure 5: GNU/SCA waveform development approach

Lowering the level complexity of the waveform structural software allows the designer to focus greater attention on improving the spectral efficiencies of the waveform and the data and packet encodings and encapsulations for different classes of environments and with different classes of power constraints.

DataSoft's flexible SDR platform allows the designer to develop a GNU radio based waveform without SCA use on a pure Linux OMAP environment. Once the waveform is stable in that configuration, the next step is to produce a skeleton waveform over a lightweight SCA CF to the Linux OMAP environment. The final steps are to remove all SCA ports, to hardwire the waveform gnu-style and to remove all CORBA ORB use. This produces a highly optimal waveform ready for deployment and testing in rigorous and difficult commercial environments at a very low cost for development, with a fast time to market and at significantly lower total cost of ownership for the carrier-developers of the waveforms.

5 Proposal for SCA Experimental System Using Current Generation Tools and Approaches

In designing and implementing a GNU radio-SCA hybrid waveform implementation using current generation available software, we propose the following approach:

1. Design and test necessary waveform signal processing materiel in a Matlab™, Simulink™, LabView™ tool based environments. Develop any necessary FECs in FPGAs and test on IGLOO equivalent FPGAs for gate count etc.

2. Generate C/and VHDL code for the target processors using the vendor supplied or developer implemented equivalent code for the targets.

3. Look through the GNU radio library for the necessary waveform components that will form part of the signal processing chain of your equivalent SCA waveform.

4. Assemble a basic skeleton GNU radio waveform on a Linux host environment using the GNU radio C libraries and assemble for IO and connectivity using Python. At this stage, the designers should also take time to standardize the IO and IOCTL interface definitions of any device drivers that they may use in the GNU radio environment. These interfaces will be recreated later in the SCA environment, in stubbed out form initially, for assembly purposes.

DataSoft

5. Perform a basic test on the waveform implemented in the previous step and validate the waveform.

6. Assemble a basic but 1:1 equivalent skeleton SCA waveform empty shell to that of the GNU radio-waveform built using any popular SCA core framework, and CORBA ORB of choice. For fast time to market, organizations may consider prototyping on Spectra DTP4700E. In this process, it will be necessary to build dummy device drivers for the DTP4700E with the same interfaces as those created in the GNU environment in step 4 above.

7. Test waveform IO using simple test dummy random octet sequences driven through the SCA waveform ports.

8. Commence the iterative transfer to the SCA signal processing blocks any code and calls from the GNU radio waveform block implementation and test IO.

9. Over a sequence of steps it should be possible to transfer nearly all BB block calls into the SCA waveform equivalent blocks, with the exception of any FECs or other IP components developed for FPGAs. Any DSP codecs developed to augment to the GNU radio waveform should be transferable to the SCA system in either a GPP code block or since the DTP 4700E supports DSP activity, can be moved to a TI codec on the OMAPs DSP. FPGA elements developed using other hardware can be moved to the DTP4700E FPGAs easily and this would then complete the waveform port to an OMAP environment.

10. Perform waveform integration testing and IO testing to see if the baseband performance and desired input output of the waveform have the right degree of fidelity.

11. If a transmit and receive waveform was designed, using two DTP4700Es end-to-end in a TX-RX configuration would be recommended to test complete signal processing fidelity, power output dynamic range and behaviors of the waveform.

Figure 6 shows a simplified diagram of the outlined approach. Our approach has the benefit of validating the waveform fidelity and IO at multiple levels and stages. Ahead of SCA time waveform validation, and at multiple points in the development of the waveform, we allow easy validation of comparison and waveform block behaviors at the signal processing level, quickly creating a fully functional waveform. Extensions to the waveform will be easier to implement in later stages as the addition can be added and tested in the GNU radio environment first and validated before an equivalent block need be created in its cousin SCA waveform environment.

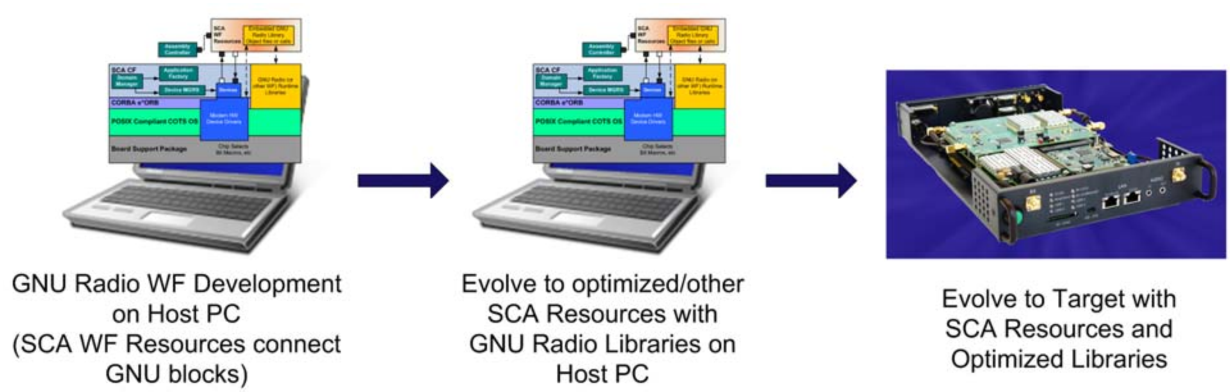

Figure 6: Proposal for SCA experimental system

DataSoft

5.1 Test Bed for System and Waveform Development

A detailed layout of the SDR waveform development environment is shown in Figure 7. The Spectra DTP4700E SDR hardware based on the DataSoft digital and RF platform is the heart of the development test bed and provides the flexibility for SCA-compliant waveform development, waveform porting, interoperability testing and even converting non SCA-compliant waveforms to SCA-compliance.

Some key features of the SDR Platform are:

- SCA-compliant SDR based application development platform to implement and deploy the next-generation of SCA-compliant networking waveforms.
- Full-duplex architecture covering 400 MHz to 4 GHz RF system (and a future 30 MHz to 400 MHz system in Q1-12) interfaced to the digital system providing advanced SDR development on the OMAP platform.
- Fully programmable 20 KHz to 20 MHz signal bandwidth.
- First COTS OE available from a single vendor, prepackaged with a full SDR test platform, RTOS, demonstration waveform and tools.
- Optimized with small form-factor, low power technology that enables the development, testing and deployment of waveforms.

DataSoft

The DTP4700E platform includes example waveforms, a live USB and trial licenses for the core framework; the platform is operational within hours of opening the box. The example waveform is a “coding example”, showing a method to connect a simple waveform in an SCA environment. From this basic waveform coding example, the SCA/SDR/Modem engineer can extend the platform and rapidly develop new waveforms for cognitive networks, white spaces and other emerging standards.

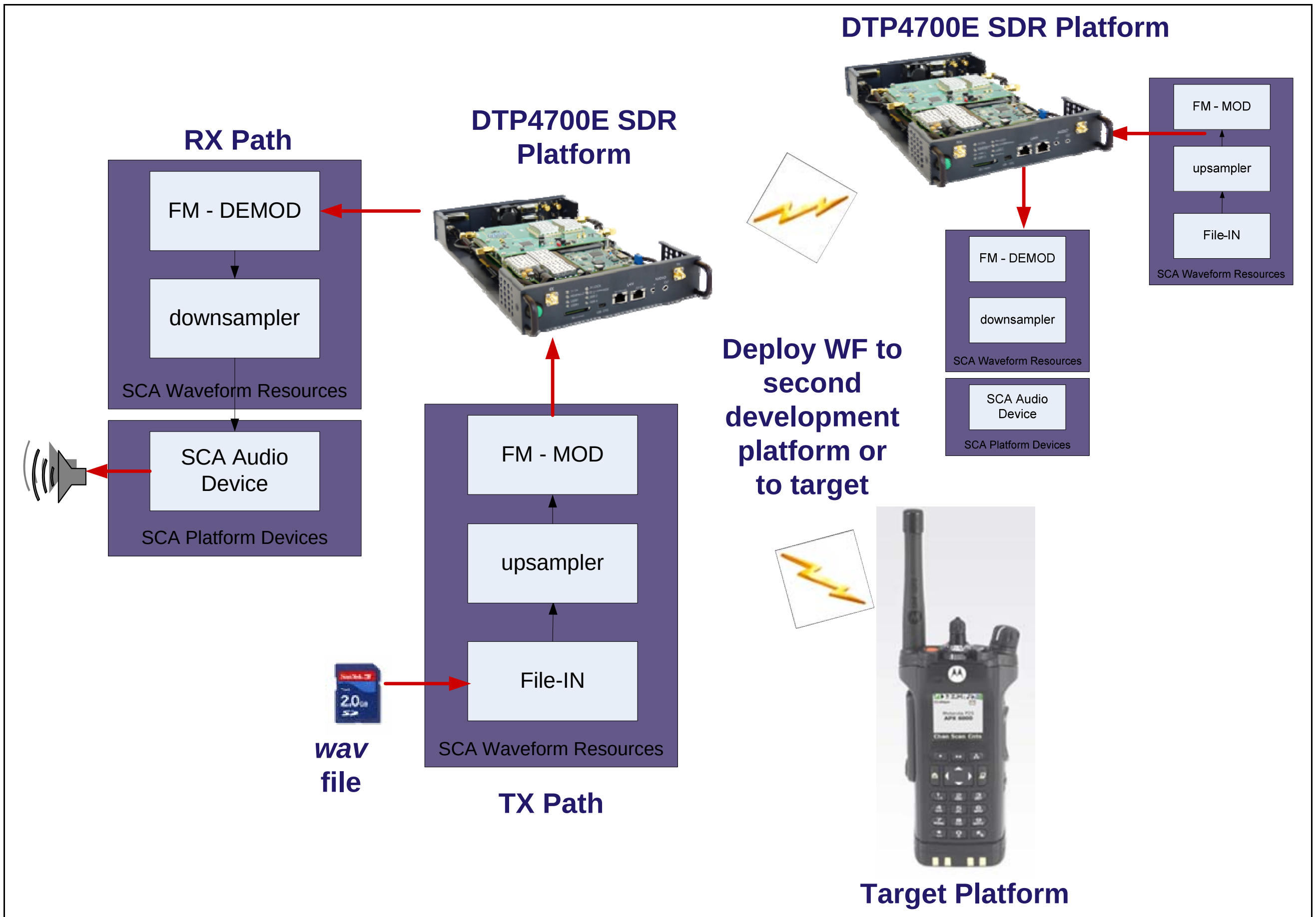

Figure 7: A waveform development environment using the Spectra DTP4700E platform

6 Results and Future Plans

DataSoft started with a simple GNU based FM waveform to demonstrate the approach and to validate the system.

Using a prototype DTP4700E platform, the following SCA components were chosen to develop a simple FM waveform to demonstrate SCA capability on the platform. The components were modeled, connected, and XML descriptors generated with the aid of the PrismTech Spectra CX tool. All of these components run on the OMAP General Purpose Processor (GPP).

AudioPortDevice: The AudioPortDevice is a platform component providing control and data processing for the audio device driver.

Radio Services Device: The Radio Services component provides an HTTP Human Machine Interface (HMI) for the DTP4700E which allows for controlling the SCA FM sample waveform and querying and configuring the platform and waveform attributes.

RF Device: The RF Device component provides for control and data processing for the transceiver module.

FileInput: The FileInput component reads in data from a WAV file via the SCA CF FileManager port parses the file and breaks the data into packets of 16-bit PCM samples which it sends via the dataOut port.

Mod: The Mod component receives sample data from the dataIn port and performs a FM modulation transformation on the data outputting complex I and Q data via the dataOut port.

Demod: The Demod component receives complex I and Q data from the dataIn port and performs a FM demodulation transformation on the data outputting real sample data via the dataOut port.

Resampler: The Resampler component receives sample data from the dataIn port and performs a Finite Impulse Response (FIR) filter transformation for sample smoothing and rate conversion outputting real samples via the audio_sample_stream_uses_port.

The Spectra CX tool was used to model the components specified. A model was created by selecting a component class and specifying attributes and port interfaces and finally connecting the components together. The tool then utilized pre-loaded SCA 2.2.2 compliant IDL to generate XML descriptor files as well as client, server, and application worker stub C++ files. The application code was then plugged into the worker stub files. Figure 8 shows the waveform and platform models.

DataSoft

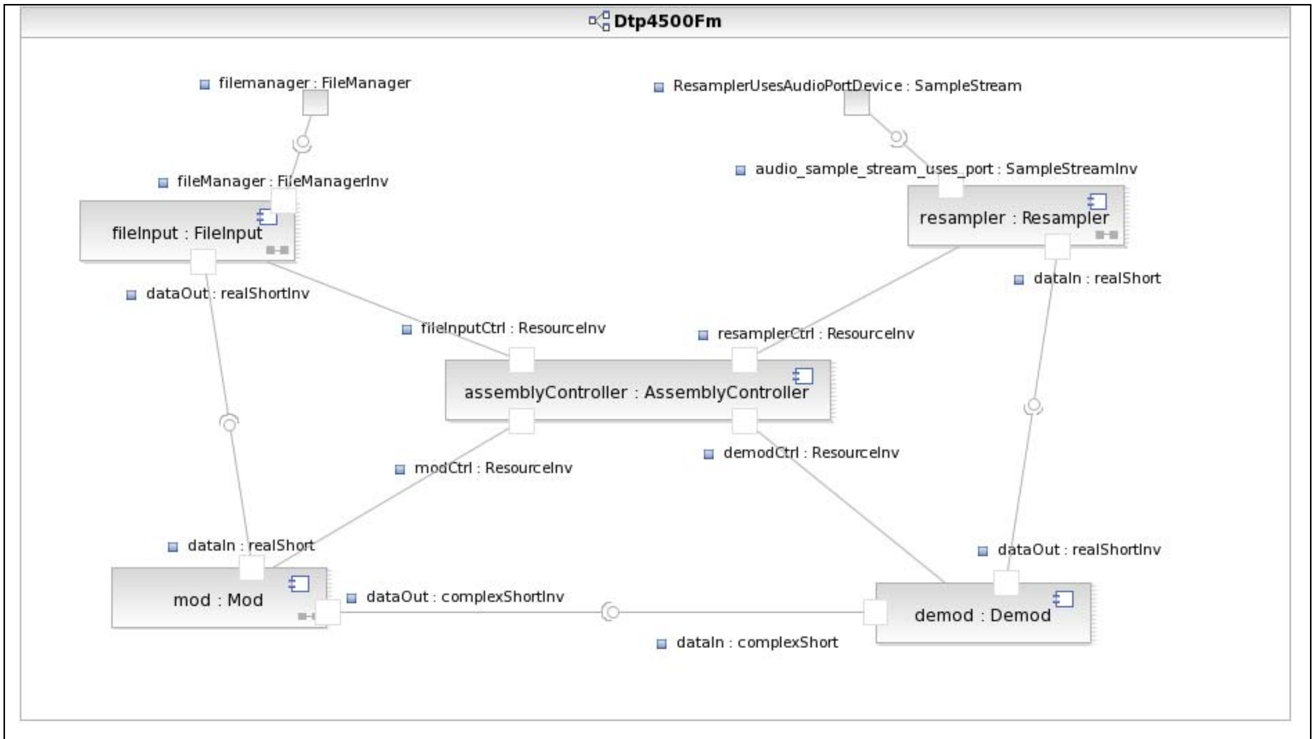

Figure 8: FM SCA WF Model

The Radio Services device HTTP interface is used to install the FM waveform in the ApplicationFactory of the CF. The Radio Services interface is also used to create a waveform application instance and to start and stop the waveform instance.

In the work to date, all waveform components run on the General Purpose Processor (GPP). Analysis shows that the GPP approaches a CPU limitation even with this simple waveform. The overall waveform design was reviewed to see which components were better suited to run on the DSP. The strategy is to move the modulator, demodulator, digital filtering and resampling, and synchronization blocks to the DSP. TI's optimized C64x+ signal processing libraries are being used for the low level optimized filtering. We also intend to port a second waveform that is relevant to cognitive spectrum sensing using the SDR platform and methodology described.

7 Summary

SCA-compliant waveform porting is a well-documented process. As commercial systems migrate to SDR platforms, a framework like SCA-next will be required for interoperability and portability. Many waveforms exist for non SCA-compliant communications; an interesting question arises regarding reuse of the waveform components when introducing SCA-next to the SDR platform. This paper presented a method of converting non SCA-compliant waveform elements to SCA-compliance and highlighted the Spectra DTP4700E SDR development platform for the proposed waveform development approach. PrismTech plans to release the DTP4700E in January 2012 as a commercial product. For non SCA-compliant waveform development, the hardware platform, Thunder is directly available from DataSoft.

[1] JTRS JPEO web site: http://www.public.navy.mil/jpeojtrs/Pages/Welcome.aspx

[2] "Evolution of the Software Communication Architecture Standard" by Kevin Richardson, Chalena Jimenex, Donald Stephens, JPEO JTRS. IEEE Military Communications Conference 2009. (MILCOM09). http://sca.jpeojtrs.mil/references.asp

[3] DataSoft, Internal Proposal for N08-224 Universal Transceiver SBIR

[4] EETimes Design Article, 12/11/2006: "Understanding SDRs and their RF Test Requirements, Part 1" by Eric Hakanson, Anritsu Company. http://www.eetimes.com/design/microwave-rf-design/4018870/Understanding-SDRs-and-their-RF-Test-Requirements-Part-1?pageNumber=1

[5] JTRS Software Communications Architecture Specification, 09 March 2011. http://www.public.navy.mil/jpeojtrs/sca/Pages/scanext.aspx

