Keys to Continued SDR Progress

Several key technical areas are making important contributions to the continued evolution of SDR. This section identifies and describes a set of these key technical areas, including specific SDR Forum contributions via the Technical Committee.

1 SDR Architectures

SDR architectures include transceivers that perform up-conversion and down-conversion between baseband and RF, exclusively in the digital domain, reducing the RF interface to a transmit-channel power amplifier, low-noise amplifier for the receive path, and minimal analog filtering. Waveforms are generated as sampled digital signals, converted from digital to analog via a wideband digital-to-analog converter (DAC) and then possibly up-converted from IF to RF. The receiver, similarly, employs a wideband analog-to-digital converter (ADC) that captures all of the relevant RF channels. The receiver then extracts, down-converts and demodulates the designated channel waveform by the use of software. An SDR will perform significant amounts of signal processing in a general-purpose processor, or a reconfigurable piece of digital electronics.

As shown in Figure 1, each processing element in a multiprocessor SDR has its own set of software modules to implement the two parallel paths. The set for each processor has a layered structure, with hardware at the bottom, processor-specific software in the middle, and software modules that implement the application on the top. The software application modules contain all of the processing capability to implement the radio's control and information functionality. This leads to the fundamental benefit of a software defined radio: the attributes of the system can be changed by bringing in new software applications without any change, replacement, or modification of hardware. The underlying middleware remains in place to load, unload, and support the application software by providing a variety of radio platform applications.

The software-defined HOOK2™ AN/PRC-112G® Combat Search and Rescue radio provides downed aircrew with direct communications with rescuers for terminal area guidance, two-way encrypted messaging, and automatically updated global positioning. Photo courtesy of General Dynamics.

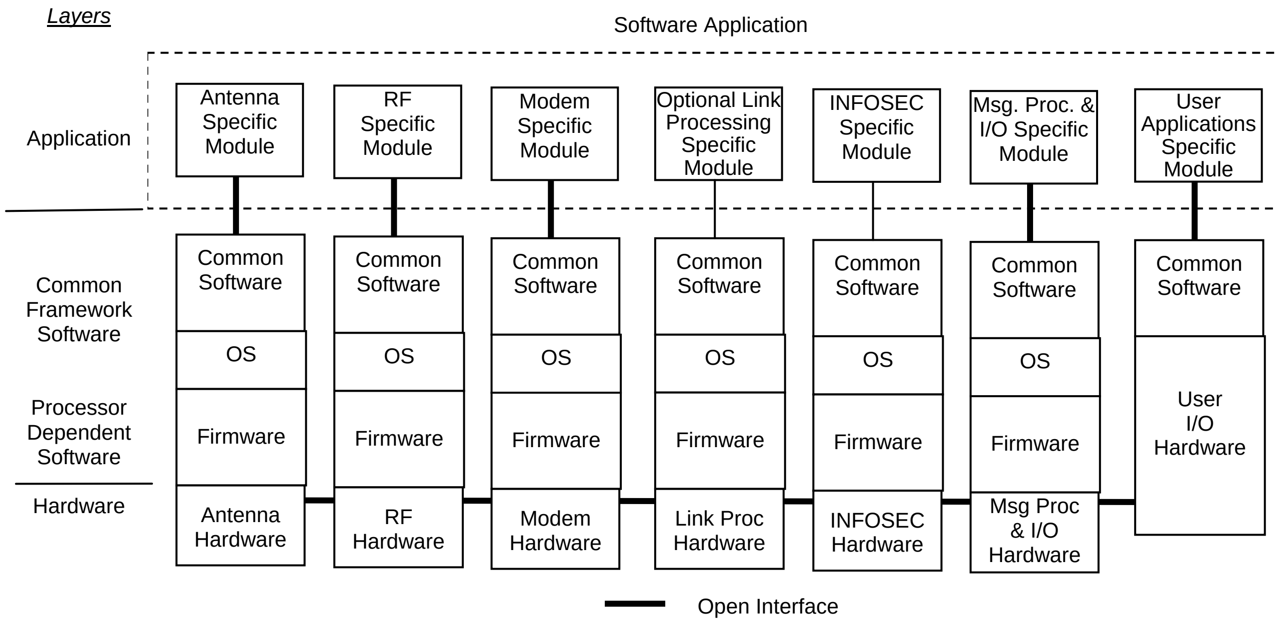

Figure 1 Example Implementation of SDR Forum Software and Hardware Open Architectures
(*Source:* SDR Forum Technical Report 2.1, "Architecture and Elements of SDR Systems as Related to Standards," 1 Nov. 1999)

2 Hardware Architectures

The hardware architectures that support SDRs have two defining characteristics—the signal is sampled as close to the air interface as possible, and the entire data path from that point runs exclusively through a programmable processing chain. This processing chain typically consists of a combination of field-programmable gate arrays (FPGAs), digital signal processors (DSPs) and general-purpose processors (GPPs). It is the reprogrammable nature of this chain that enables the modulation format or other processing characteristics to be changed on a somewhat dynamic basis.

Beyond these few common hardware characteristics, however, considerable variation exists, depending on specific application and frequency band. Several common hardware architectural patterns have emerged to date. One of these is a "slice" architecture pattern, in which multiple identical hardware sets are used to implement multi-channel radios. Another pattern is a "pooled resource architecture," by which multiple radio channels and applications can be simultaneously spread across multiple processing resources in a distributed "using available capacity" manner. Finally, in the military domain, certain security-based architectures have taken hold that segment the hardware into "plaintext" and "cipher-text" processing elements, and employ programmable cryptographic modules and "core" control processors.

3 Software Architectures

The hardware platform is just an enabling technology for an SDR—the software architecture hosted on this platform truly forms the SDR per se. Here, the primary defining characteristic for an SDR is a clear, formal separation from the "platform" software and the applications (or waveforms) that run on top of this platform. In this way, a waveform can be developed and deployed onto a platform without the need to simultaneously code the waveform "into" the platform or the platform device drivers. Figure 2 shows the software focus areas of the SDR Forum.

In most cases "application frameworks" tend to be employed in SDRs to realize this separation of concerns in the context of an open, standards-based solution. These frameworks put an emphasis on standard interfaces through which application components and platform "core frameworks" can communicate in a standardized way, as well as providing standardized platform services. The most developed framework at this time is the U.S. Government's Joint Tactical Radio System (JTRS) Software Communications Architecture (SCA). The SCA defines a base platform offering Portable Operating System Interface (POSIX)–based OS services, a middleware layer (Common Object Request Broker Architecture, or CORBA) supporting distributed object processing, and a standard core framework for deploying platform devices, services, and installable applications.

The commercial wireless market is similarly developing frameworks for that application domain, with the first SDR basestations now becoming available. The initial versions of these frameworks are currently more proprietary in nature, but as the market matures, more open standards are expected.

The standardization of interfaces to SDR hardware functions is of key importance in realizing application portability between platforms. These interfaces are referred to as application programming interfaces (APIs). APIs should ideally start at the information source, moving through the control of the platform hardware by the waveform, to eventually control the antenna functions. As described in

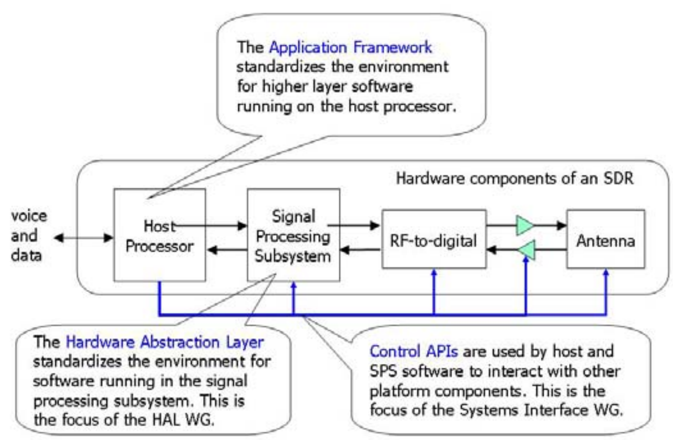

Figure 2 Software Architecture Focus Areas of the SDR Forum
(Courtesy of John Chapin, Vanu, Inc.)

more detail in subsequent sections, the SDR Forum System Interface Working Group, and HAL Working Group have been doing significant ongoing work in this area. HAL stands for the Hardware Abstraction Layer for the signal processing subsystem of software defined radios.

4 Application Programming Interfaces

The specification and utilization of APIs is a key tenet toward the objective of achieving the portability and reuse of software applications across SDRs. The SDR Forum System Interface Working Group is responsible for identifying radio, management, network, and physical services and their interfaces. It also maintains and promotes system interface standards and develops and promulgates new standards when necessary. The group began by defining a set of general APIs. These were meant to be applicable to any SDR, but were probably an 80 percent solution for any particular implementation. The overarching plan was to provide this set to the SDR community at large but also specifically to the JTRS JPO as a jumping-off point for a true set of APIs applicable to JTRS.

The Object Management Group's Software-Based Communications Domain Task Force (DTF) also began work in a similar area, so a liaison relationship was created between the OMG and the SDR Forum to combine this work. The final document from this joint effort is the Platform Independent Model (PIM) and Platform Specific Model (PSM) for Software Radio Components. This document uses the Model Driven Architecture approach of the OMG, creating a PIM that can be applicable to any implementation and a PSM that is applicable to a CORBA–based implementation, which is currently the JTRS implementation. Other PSMs can be mapped from the PIM and have been encouraged within the SDR Forum.

As of June 2004, the OMG SWRadio final submission was accepted by the OMG Architecture Board. It is now a publicly available adopted specification and can be found on the OMG web site. Interested members can subscribe to the srsupport@yahoogroups.com mailing list or check their website at http://sbc.omg.org. The Forum's System Interface Working Group is also planning to work on developing APIs to other areas of the SDR, such as RF internals and smart antennas. This activity will focus on collecting the publicly available information that has already been developed for various radio programs (e.g., JTRS Cluster programs).

The SDR Forum has issued an RFP for an Open Source Reference Implementation of an SCA-compliant waveform using the OMG waveform APIs from the PIM/PSM specification. Figure 3 shows the API subset of the PIM agreed to at the April 2004 meeting of the SDR Forum's System Interface Working Group for this purpose:

PIM Level

Common Layer Facilities
- Flow Control Facilities
- Protocol Data Unit (PDU) Building Block*
- Stream Facilities *

Link Layer Control (LLC) Facilities
- Local Link Management Package
- Connectionless Link Package *
- Connection Link Package *

Medium Access Control (MAC) Facilities
- Flow Control
- Scheduling of Transmission

Physical Layer Facilities
- RF/IF Facilities (simulated)
- Modem Facilities
- I/O Facilities (simulated)

** If connectionless LLC is chosen, PDU building block will be implemented; if connection LLC is chosen, stream facilities will be implemented.*

Figure 3 API Subset of the PIM Level

The Forum's System Interface Working Group plans to continue working on new API definitions, first focusing on Device APIs. In addition, API-related Change Proposals to the JTRS SCA specification will be reviewed. There will also be joint meetings with other SDR Forum working groups to coordinate individual API definitions being worked on in other groups such as the Smart Antenna API.

5 Software Communications Architecture

The SCA is an open architecture framework developed for the use in software defined radios. Originally developed in the late 1990s (with significant input and feedback provided by the SDR Forum Mobile Working Group) to meet the needs of U.S. government and military radio systems, the SCA has evolved to become a de facto international standard, with SCA-based implementations and activity occurring on all continents.

The SCA specification consists of a main document and three supplements, each with a different purpose within the standard. The main document describes the software operating environment (OE) onto which waveforms can be deployed and details the required interfaces that waveforms must support. The OE itself can be further divided into operating system (POSIX), middleware (CORBA ORB), and framework interfaces. The framework supports the creation and management of platform devices and services and well as the interfaces with which installed waveforms are registered. Appendices to the SCA formally specify the framework interfaces in CORBA Interface Design Language, define a POSIX profile, and specify the format and content for the XML files that an application must provide in order to be managed by the OE.

An API supplement specifies a number of "building blocks" for the common layers a waveform typically implements, as well as a standardized I/O device. These building blocks serve as interface templates in that rather than identify specific data types, they identify generic control mechanisms that a given waveform specializes to make a concrete interface. Also included is a data item description (DID), which describes how a waveform must document its APIs is included.

The security supplement to the SCA details, in tutorial and requirements form, the security requirements that a U.S. Government SCA-compliant radio must implement. This material was separated from the base SCA to enable non-government implementations to follow the "core" SCA when no such security requirements exist.

The Supplemental Hardware Supplement (SHS) constitutes the final and most recent supplement. Whereas the core SCA specifies the environment for running waveforms on (CORBA-hosting) "general purpose" processors, the SHS standardizes the environment for processor DSP chips and FPGAs. This area has become increasingly important as the upper frequency range covered by the SCA moves into the Gigahertz frequency range and into the >100 M bits per second (Mbps) data rate range.

As the SCA has matured, there have been several areas of evolution, and the SDR Forum has been heavily involved in defining the direction of this movement. The initial revision of the SHS was drafted by the Forum's HAL Working Group, and then passed on to the JTRS Joint Program Office with subsequent approval into the SCA.

Recently, a joint venture between the government JTRS JPEO and the SDR Forum has resulted in the formation of the Forum's SCA Working Group, which held its first meeting during the SDR Forum's 44th General Meeting in June 2005. The SCA Working Group will serve as an industry "users group" to provide input and feedback to the JPEO and Tactical Assault Group (TAG) bodies on future directions related to the SCA.

In summary, the SCA architecture has grown out of early SDR Forum activities, and the SDR Forum continues to be the foremost industry group ensuring that this government standard fulfils not only its government commitment, but also commercial promise.

The Communications Research Centre Canada (CRC)—in partnership with Defence Research and Development Canada (DRDC)—with support from the SDR Forum, developed a Java-based, open-source SCA reference implementation. Available to the industry, the SCA reference implementation has been regularly updated and has received more than 10,000 downloads.

6 Hardware Abstraction Layer

The HAL Working Group of the SDR Forum Technical Committee was active from 2003 until late 2004. In summer 2004, the JTRS JPO began work on SCA version 3.0, which directly overlapped with the HAL Working Group focus area. HAL Working Group work was suspended in fall 2004, and members transferred to participation in the System Interface Working Group, which developed the SDR Forum input to the JTRS SCA 3.0 process.

The Hardware Abstraction Layer for the signal processing subsystem (SPS) encompasses standards, interfaces, tools and techniques that improve software portability and performance across radio platforms with diverse signal processing subsystems. The SPS provides high-speed computation for the OSI layer 1 (modem, spreading, coding) and application level functions (audio, video) of a software defined radio. The SPS is commonly implemented by using some combination of DSPs, FPGAs, or other flexible processing engines. Figure 4 illustrates the role of the SPS and the focus areas of the HAL Working Group.

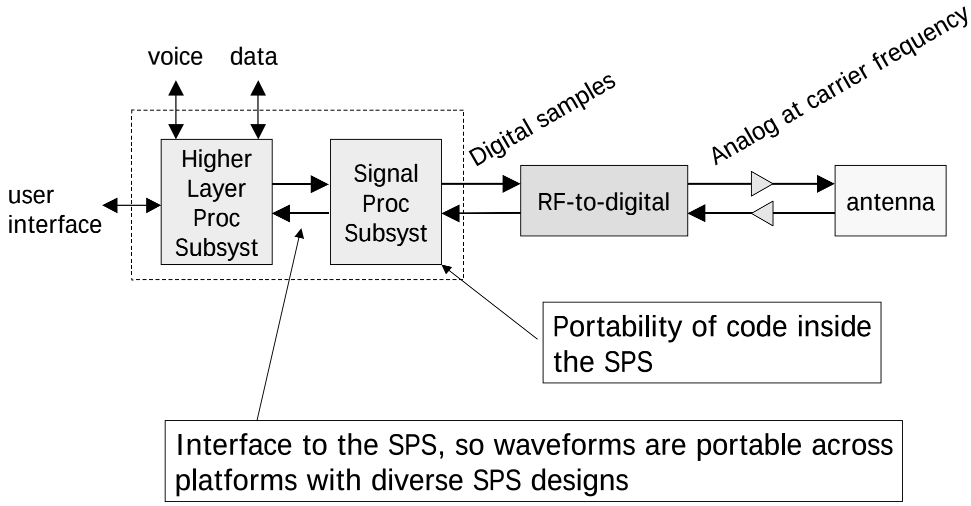

Figure 4 SDR Architecture Model and Areas of HAL WG Interest

Work by the Forum and other organizations such as OMG, OMA, and JTRS has resulted in proposed standards for various aspects of software defined radios. However, there was widespread agreement that software interactions with and software inside the SPS had not yet been adequately addressed in an open forum before the SDR Forum began its HAL Working Group work. Figure 5 shows where the HAL Working Group focused on APIs and tools to improve software portability.

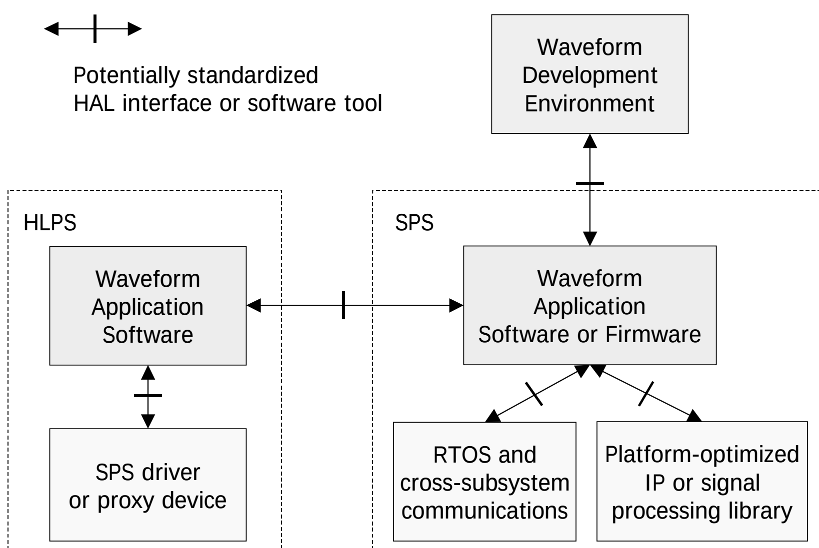

Figure 5 SDR Software Model and Potential Roles for an SPS HAL

The HAL Working Group developed and distributed an RFI to assess the state of the art in tools, techniques, and end-user requirements for HAL-related technologies. Twenty-four responses were received from 17 different U.S. and European organizations with multiple products or technologies to describe. Nine of these organizations gave presentations on their views at the SDR Forum's General Meeting in Mainz, Germany, in April 2004.

The submissions were carefully analyzed, and a report on the RFI results was prepared by the HAL Working Group and approved by the SDR Forum. Concluding recommendations from the report were as follows:

- The SDR Forum should form a study group on design flows and tools. This will increase clarity of discussions, provide a framework for developing SDR industry-specific requirements, and provide a roadmap for vendor tool improvements.
- The HAL Working Group should develop a consensus HAL reference model and/or layer definitions to clarify the categorization of the various functions performed by a HAL.
- The SDR Forum should wait for the results of the JTRS JPO SCA 3.0 extension effort before pursuing further investigation of low-level HAL functions.
- The HAL Working Group should interact further with organizations proposing high-level HAL standards and assure their efforts are presented to SDR Forum members for consideration.
- The SDR Forum Technical Committee should consider sponsoring a reference implementation of a high-level HAL at some point in the future.
- The SDR Forum should exploit its close ties to OMG, JTRS, and its new partnership with E2R to promote harmonization and reduce duplication of effort regarding configuration and control of signal processing subsystems.

7 Security

Security is an essential element for the successful deployment and operation of SDRs. Security is a broad technical area that permeates the SDR architecture and includes the following key capabilities:

- Maintaining the privacy of user traffic over-the-air, including up to high levels for military applications.
- Protecting access to a user's personal information stored on an SDR device, through access control mechanisms
- Protecting the integrity of software downloaded to an SDR, as well as protecting the device from loading malicious or harmful software (i.e., viruses).

The SDR Forum's Security Working Group is chartered to promote the economical application of SDR security concepts in the framework of an overall system architecture. For the past year, the activities of the Security Working Group focused in three primary areas.

1. Application of the important security concepts presented in the SDR '02 conference paper "Security Concepts for Software Defined Radios" to the work and activities of the SDR Forum Regulatory Committee, the Download Working Group and the Public Safety Special Interest Group. The results of these efforts have led to the publication of the SDR Forum Download Working Group document DL-SIN (Document SDRF-02-W-0005-V1.0*), Security Considerations for Operational Software for Software Defined Radio Devices in a Commercial Wireless Domain*, as well as the preparation and publication of formal Requests for Information and Comment related to DL-SIN. These documents were issued as SDRF-04-W-0005-V1.0, "Request for Information on Security for Wireless Reconfigurable Devices" and SDRF-04-W-0006-V1.0, "Request for Comments on SDR Forum Document DL-SIN."

2. Development of a security model that allows defining the communications channels with the framework of a system of systems architecture, the perceived threats against both the channel and the system, as well as identification of the security measures employed against the threats. This model has been included in the SDRF Download Working Group published document DL-SIN identified above. The use and application of this model was also presented to the Public Safety Special Interest Group to assist them in addressing security issues affecting their operating domain. Finally, the Security Working Group drafted a comprehensive response to the Public Safety Request for Information on the topic, *How Software Defined Radio Technology Can Meet the Communications and Interoperability Requirements of Public Safety.*

3. Development and application of a Business Model that supports policy based management of a communications system and network and allows trade-off and evaluation of the economic benefits gained by the appropriate and proper introduction of security features. This effort has involved direct participation within the activities of the SDR Forum Commercial Wireless Working Group.

8 Tools and Development Processes

The design process is closely tied to the tool chain. Creating best practice techniques is crucial for technology developers and tool vendors working with waveforms, platform infrastructure, platform hardware, core technology, and systems within an SDR context. As the degree of freedom in process and tool choice becomes increasingly limited when mapping SDR applications to platforms, the need for clearly defined best practice techniques becomes essential. As a result, the SDR Forum created the Design Process and Tools Working Group to research the best practices for design flows and tools used by industry in creating systems, software, waveforms, and components for SDR.

The identified design flows will target specific development scenarios in the commercial, civil, and defense sectors. A key part of the recommendations will be the identification of classes of tools necessary to support the development stages associated with these design flows and the attributes and features required in those development tools, including the necessary interoperability among tools.

The recommendations will cover four key areas:

1. *Terminology:* Develop a common terminology that can be used to describe the design flows and tools.
2. *Technology Push:* Identify and recommend reference design flows that are utilized by industry in the development of SDR technology given the current state of the art in tools.
3. *Market Pull:* Identify design flows that are desired by industry in the development of SDR technology as well as the classes of tools necessary to facilitate each state within those design flows.
4. *Gaps:* Identify the gaps that exist between technology push and market pull, and make recommendations to industry toward closing those gaps.

The success of this activity will center on the ability of the group to attract active participation from all the industry groups involved, including waveform developers, component developers, platform vendors, tools vendors, and so forth, and as such a key part of this charter is to provide outreach to these various groups.

The Design Processes and Tools Working Group is working to bridge any gaps that may exist between the design processes that are used in the creation of SDR technology and the tools that are available to support those processes. One of the first activities of the group was the development and distribution of an industry-wide Request for Information to gauge the current market, challenges, and key issues in this area. The results from this RFI will be evaluated and presented during the SDR Forum's Annual Technical Conference and Product Exposition in November 2005 in Orange County, California.

9 Smart Antenna

Smart antenna radios are radio systems that use multiple antenna elements with smart signal processing that helps improve the range, capacity, and reliability of a wireless system. Smart antenna technology has seen adoption in wireless military applications and in commercial cellular base stations. Most recently, smart antennas have also been considered for wireless local area networks (WLANs) using a type of smart antenna technology called multiple input, multiple output (MIMO) technology.

Smart antenna implementations include beamforming, space-time coding, and multi-antenna multi-user detection, among many others. Software defined radios are designed to have multi-mode capability with support for multiple standards and waveforms. This philosophy of design needs to be extended to radios with multiple antennas, which, in turn, are capable of operating under different types of smart antenna technology. Hence the standardization and identification of API interfaces for use of smart antennas lends itself to an SDR design effort.

The Smart Antenna Working Group within the SDR Forum is working toward standardization of the APIs for SDR systems to include smart antennas. With the wide variety of smart antenna systems available today, and the growing number of standards each having its own implementation and interface issues, it is difficult to create APIs that cover every possible scenario. At the same time, it is also important to determine how introduction of smart antenna systems would affect the current SCA radio framework.

Toward this goal, the Smart Antenna Working Group, led by Virginia Tech and Hanyang University, is collecting use-case scenarios for different types of smart antenna implementations. These use-case scenarios will help identify generic modules required for different types of implementations as well as specific ones. This will then help set a framework for smart antennas within a SDR. Once the framework is available, it will become easier to specify the interfaces for smart antenna modules that are general purpose in nature but can be extended to specific implementations.