Lessons Learned in the First Year Operating Software Defined Radios in Space

David Chelmins[a], Dale Mortensen[b], Mary Jo Shalkhauser[c], Sandra K. Johnson[d], and Richard Reinhart[e]
NASA Glenn Research Center, Cleveland, Ohio, 44135

Operating three unique software defined radios (SDRs) in a space environment aboard the Space Communications and Navigation (SCaN) Testbed for over one year has provided an opportunity to gather knowledge useful for future missions considering using software defined radios. This paper provides recommendations for the development and use of SDRs, and it considers the details of each SDR's approach to software upgrades and operation. After one year, the SCaN Testbed SDRs have operated for over 1000 hours. During this time, the waveforms launched with the SDR were tested on-orbit to assure that they operated in space at the same performance level as on the ground prior to launch to obtain an initial on-orbit performance baseline. A new waveform for each SDR has been developed, implemented, uploaded to the flight system, and tested in the flight environment. Recommendations for SDR-based missions have been gathered from early development through operations. These recommendations will aid future missions to reduce the cost, schedule, and risk of operating SDRs in a space environment. This paper considers the lessons learned as they apply to SDR pre-launch checkout, purchasing space-rated hardware, flexibility in command and telemetry methods, on-orbit diagnostics, use of engineering models to aid future development, and third-party software. Each SDR implements the SCaN Testbed flight computer command and telemetry interface uniquely, allowing comparisons to be drawn. The paper discusses the lessons learned from these three unique implementations, with suggestions on the preferred approach. Also, results are presented showing that it is important to have full system performance knowledge prior to launch to establish better performance baselines in space, requiring additional test applications to be developed pre-launch. Finally, the paper presents the issues encountered with the operation and implementation of new waveforms on each SDR and proposes recommendations to avoid these issues.

I. Introduction

Conducting on-orbit checkout, waveform updates, and comprehensive performance characterization on three software defined radios onboard ISS has provided an immense amount of knowledge on the processes, architectures, and approaches for using SDRs in space. This paper captures, at a high level, some of this knowledge for future SDR platform and waveform developers.

This paper is comprised of a brief overview of the SCaN Testbed system and a description of each SDR in Section II. Section III covers example lessons learned during the initial year operating the three SDRs in space including: 1) Test the SDR hardware independent of the software, 2) Working with obsolete hardware and software, 3) Need for command flexibility, 4) Need for telemetry flexibility; 5) Architectures to assist on orbit diagnostics, 6) Importance of high-fidelity engineering models, and 7) Third-party software development. Section IV provides a summary of the paper and conclusions. Section V captures future work envisioned to add more complex waveforms on the flight system and incorporate additional third-party users.

[a] Experiments Lead, SCaN Testbed Project, 21000 Brookpark Rd. MS 54-1, AIAA Non-Member.
[b] Waveform Developer, SCaN Testbed Project, 21000 Brookpark Rd. MS 54-4, AIAA Non-Member.
[c] Waveform Developer, SCaN Testbed Project, 21000 Brookpark Rd. MS 54-4, AIAA Non-Member.
[d] Co-Principal Investigator, SCaN Testbed Project, 21000 Brookpark Rd. MS 54-1, AIAA Non-Member.
[e] Principal Investigator, SCaN Testbed Project, 21000 Brookpark Rd. MS 54-1, AIAA Non-Member.

II. SCaN Testbed System Overview

The objective of the SCaN Testbed is to study the development, testing, and operation of Software Defined Radios and their associated applications in the operational space environment to reduce cost and risk for future space missions. SDRs offer the promise of in-flight reconfigurability, autonomy, and an evolution to cognitive operation. The SCaN Testbed was launched on July 20, 2012 to the ISS on a Japanese H-II Transfer Vehicle (JAXA HTV3), and transferred and installed via Extravehicular Robotics on the ExPRESS Logistics Carrier-3 in the inboard, Ram-facing, Zenith-facing payload location on an exterior truss of the ISS, as shown in Figure 1. The SCaN Testbed flight system contains three SDRs capable of operating at S-, Ka- and L-band, along with the accompanying radio frequency (RF) and antenna systems.

Figure 1. SCaN Testbed on the International Space Station.

In addition to the flight payload, the SCaN Testbed system also consists of a control center located at NASA's Glenn Research Center in Cleveland, Ohio. The control center supports data flow on two paths: the primary path and the experiment path. The data path for commanding the SCaN Testbed flight system and receiving Health and Status (H&S) telemetry from the flight system is through the standard ISS command/telemetry system (the "Primary Data Path"). The second data path exists when the SCaN Testbed radios and RF system communicate directly with NASA's Tracking and Data Relay Satellite System (TDRSS)[1]; this is referred to as the "Experiment Data Path" (see Figure 2). The primary path is used for SDR commanding and H&S telemetry as a requirement of operating on ISS. The primary path also provides an additional benefit for experimental testing of new waveforms on the SDRs by enabling a redundant file transfer path if new waveform updates encounter unexpected behavior.

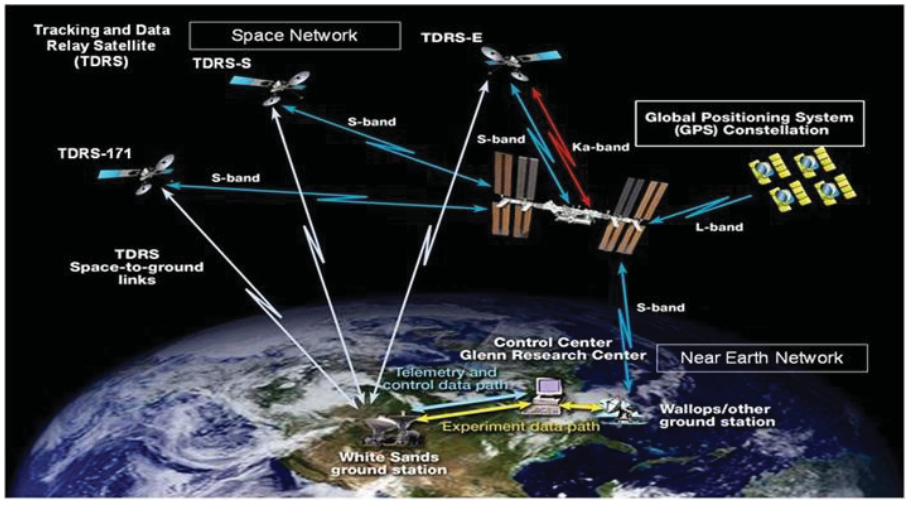

Figure 2. SCaN Testbed System Architecture.

At the core of the flight system (Figure 3) are three unique software defined radios provided by government and industry partners. Each of the three SDRs has an Operating Environment (OE), which includes an operating system and infrastructure services to applications and waveforms in accordance with the Space Telecommunications Radio System Standard (STRS). The OE middleware (compliant with the STRS Architecture Standard[2]) abstracts the SDR hardware from the waveform application software. In addition to the OE, each SDR runs STRS-compliant waveform applications, which implement the unique capabilities of the radio to receive and transmit information via RF signals.

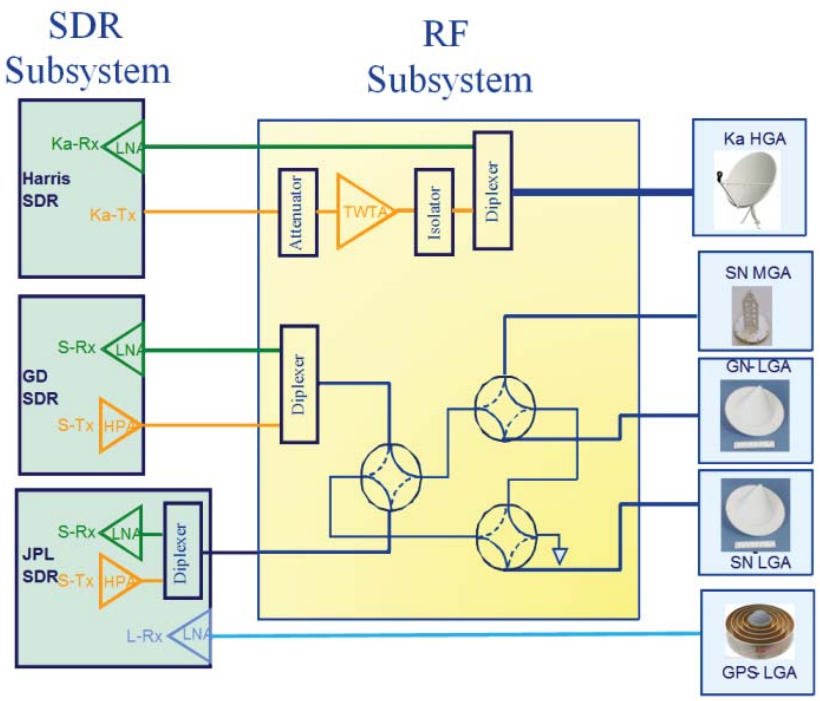

Figure 3. SCaN Testbed Flight System Block Diagram

The three SDRs in SCaN Testbed have varying capabilities. The General Dynamics (GD) Corporation SDR is capable of full-duplex, TDRSS-compatible, STRS-compliant S-band communications. The GD SDR leverages developments of the TDRSS fourth-generational transponder. The Jet Propulsion Laboratory (JPL) SDR is capable of full-duplex, TDRSS-compatible, STRS-compliant S-band communications and receive-only Global Positioning System (GPS) L-band for navigation. It leverages development of the Electra[a] radio, and the S-band and GPS portions can be operated simultaneously[3]. Both S-band radios can support communication via a gimbaled medium gain antenna (MGA), a space-facing low gain antenna (LGA), or a ground-facing LGA. The Harris Corporation SDR is NASA's first TDRSS-compatible flight Ka-band transceiver[4] for space operations, and it communicates through a space-facing gimbaled high gain antenna (HGA). The Ka-band travelling wave tube amplifier (TWTA) is a separate module from the Harris SDR.

The GD and Harris SDRs were developed under a cooperative agreement with NASA, where each company funded a portion of the development costs. Due to the schedule and funding constraints, useable capabilities in leveraged systems were implemented directly without a strong focus on size, weight, and power.

III. Examples of Lessons Learned

Lessons learned in the process of procuring, developing, and testing the SDRs were documented in previous work[5,6]. The following lessons capture the knowledge gained in the first year of on-orbit SDR operations.

A. Test the SDR hardware independent of the software

The primary advantage of an SDR is the ability to change functionality and signal processing capabilities after deployment. However, in order to make these changes effectively, the platform must be well-understood. The translation of a RF input signal to voltage by the analog-to-digital converter (ADC), as well as the hardware processing chain that leads up to the ADC, must be well-characterized. On the transmit side, the translation of waveform drive levels sent to the digital-to-analog converter (DAC) and the corresponding output power into a nominal load must be well-understood.

In most cases, the SDR platform characterization can only occur using a special type of waveform software that is designed not to influence the underlying platform behavior. On SCaN Testbed, this has been accomplished using a test waveform that can record and transmit arbitrary samples at the ADC and DAC, respectively. The platform requires a sufficient amount of memory to capture data at high rates on the forward link, as well as file storage capability to transmit an arbitrary pulse shape.

Test waveforms add time and cost to the development schedule, but they also allow independent development of different applications that can run on the platform post-deployment. The first year of operational experience on SCaN Testbed has shown that the operational waveform (the waveform responsible for data communication)

[a] "Electra – Mars Reconnaissance Orbiter," NASA Jet Propulsion Laboratory [online website], URL: http://mars.jpl.nasa.gov/mro/mission/instruments/electra/ [cited 3 March 2014].

generally is less useful for platform characterization. This waveform tends to be developed to operate at a certain frequency, modulation, and data rate, which leads to a platform performance characterization that is implementation-dependent. Characterization using an operational waveform may be useful for future improvements and bug fixes, but it tends to not be useful in the general case.

Characterization with a test waveform yields information about platform RF response and performance. On the forward link, the test waveform can be used to characterize the receiver gain and noise figure across a variety of temperatures and power levels. The conversion of input power to voltage is another important characteristic for development of future waveforms. On the return link, the waveform can be used to test transmitter gain flatness (see Figure 4) and compression, which is useful for modulation types that require well-balanced I and Q components. The modulator and oscillator temperature response will affect the output signal and is simple to characterize by generating a sine wave with the test waveform.

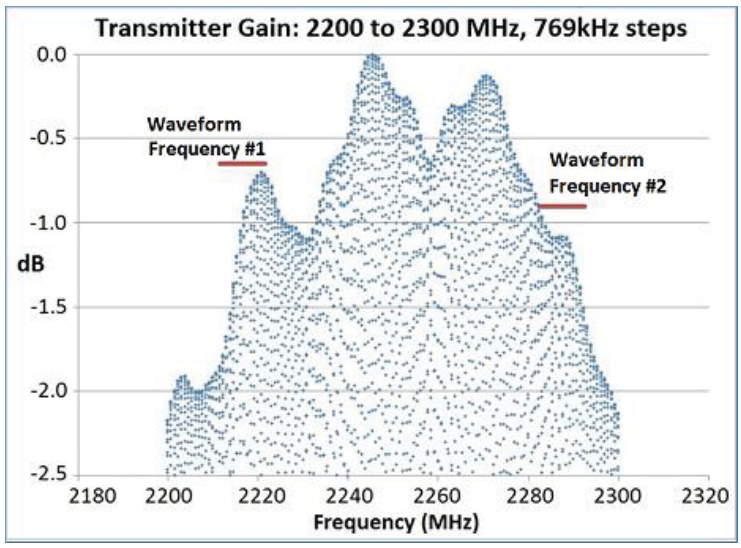

Figure 4. Sample Transmit Gain Measurement

Test waveforms are most applicable on the ground prior to deployment. It is much easier to characterize a platform in a controlled environment than in space. Characterization in space requires a well-known transceiver to characterize against, and often this is a more difficult requirement than characterization of the radio platform itself.

B. Working with obsolete hardware and software

The SCaN Testbed GD and JPL SDRs both rely on Xilinx Virtex-2 field-programmable gate arrays (FPGAs). Xilinx now refers to these FPGAs as "Mature and Discontinued Products,"[a] however they will continue to operate in space for many more years.

The challenge of using mature hardware products in a design is that the development environments tend to become obsolete sooner than the hardware will fail. The Virtex-2 was last supported by the Xilinx Integrated Software Environment (ISE) Design Suite 10.1, released in 2008, so present-day developers must use much older design software than the current release.

In itself, software and hardware obsolescence is not a substantial issue; however it presents two larger challenges. First, new waveforms must be designed with old software libraries that are supported by the old development environment. Many bugs that are present in these libraries have been fixed in later releases, so manufacturers are reluctant to devote resources to retroactive bug fixes. The second challenge is that it becomes more difficult to purchase development boards for obsolete hardware. There is not much demand for starting new development on 10-year-old hardware, so correspondingly there is little incentive to sell development boards.

The response to this situation is not simple. Many times, older hardware is flown in space because it is proven, low-cost, low-risk technology. Mission designers may be reluctant to include new, unproven hardware – especially in a communications system – because a failure would endanger a much larger and more expensive science mission. In this case, the only option may be to reserve a sufficient amount of development boards and software environment licenses to account for later development. The mission designer must recognize that the low upfront procurement cost may be offset by a large development cost later. On the other hand, flying newer hardware ultimately will enhance the potential capabilities of a mission. Newer hardware tends to support improved features as well as provide a size, weight, and power savings. In this case, risk can be reduced through redundant systems (at greater cost). Ultimately the future waveform development cost will be lower with new hardware since obsolecense will not occur so soon.

[a] "Support – Virtex-II", Xilinx [online], URL: http://www.xilinx.com/support/index.html/content/xilinx/en/supportNav/silicon_devices/mature_and_discontinued_products/virtex-ii.html [cited 3 March 2014].

C. Need for Command Flexibility

SDRs have unique commanding needs compared to non-software defined radios. The inherent versatility and re-programmability of SDRs to change the on-orbit operating capabilities of the radio will add additional complexity to the commanding of the radio. The challenge to SDR designers is to develop command and telemetry protocols that are efficient and flexible, allowing upgrades to launch waveforms and facilitating new commands associated with the development of new waveforms.

Traditional radios are typically hard-wired to use a specific waveform with few variations to its configuration. Commanding of traditional, non-software defined radios is generally limited to commands that execute a single operation. By contrast, SDR commands are complex, including single operation commands, multiple operation commands, and command scripts that may automatically submit a dozen or more commands in a predefined manner.

An important command requirement unique to SDRs is the generic command – a command that calls another command that may not be part of the original command set. The communication system infrastructure (ground and avionics software) must accommodate a generic command so launch waveforms can be modified and new waveforms can be uploaded without immediately requiring augmentation of the radio command interface. A generic command enables operators to instruct the radio to exercise a new or modified capability, which is also essential during ground-based development of new waveforms or waveform upgrades.

The three radios in the SCaN Testbed each have a different command implementation and structure[7]. The choice of implementation largely was based upon the previous experience of each radio manufacturer and the ability to re-use existing software. In general, the interface requirements were left vague in order to evaluate design tradeoffs. Commanding on the GD radio uses a fixed binary command format. Commands are sent from avionics to the GD radio in a 256-byte packet over a MIL-STD-1553 serial data bus. Although a generic command was not specifically implemented by GD, the commanding protocol for the GD radio easily accommodates a generic command with no impact on the payload avionics processor. The generic command would allow the user to enter the specific binary value of a new command. The JPL radio uses a text command format over the MIL-STD-1553 serial data bus. The text-based commands are byte intensive, but are variable in length, and thus the commands are easily expandable with no impact on the avionics processor. The commands are entered in a format understandable to human operators and easily parsed by the JPL radio. Generic command capability is provided by default. The Harris radio utilizes the SpaceWire (SpW) link for command and telemetry communication with the avionics processor. This radio uses the remote memory access protocol (RMAP) command format which is designed to support SpW applications. The RMAP format contains a command identifier, a waveform application selection, the number of command properties that will be sent, the property names, and the property values. This format is very flexible, allowing an large number of properties to be sent in text format. It is also very flexible in content allowing new commands to be sent for waveform changes. The SCaN Testbed ground software has implemented a generic command for the Harris radio to accommodate waveform changes and updates.

The widely varied approaches of the three SCaN Testbed radios gives NASA an opportunity to work with different approaches and determine their strengths and weaknesses and recommend the best approach for future SDR development. The first year of experience with SCaN Testbed shows that future SDR development should define a command structure that is free-form like the JPL and Harris radios with perhaps a more byte-efficient implementation. This approach enhances usability and upgradability with minimal impact to the rest of the system.

Minimizing the effect of human-caused commanding errors is important in a complex communication system like an SDR. Pre-defined commands in reviewed and tested procedures minimize the chance of commanding errors. However, special parameters for some commands (e.g., file system operations) can be difficult to predict and must be entered at the time of operation, which increases the chance of text entry errors during commanding. Scripts that automatically execute multiple commands also make commanding easier and error-resistant., with the caveat that command scripts increase the chance of unexpected consequences if the wrong script is commanded accidentally.

The complexity, re-configurability, and re-programmability of the software defined radios require radio-knowledgeable mission operators. Procedures reduce the chance of human error, but often become a "crutch" for the operators; operators may not understand the system well enough to react to issues. The complexity of the SCaN Testbed makes it difficult for the operators to be experts on each of the radios because of the differences in waveforms, commands, and telemetry. Mission-based SDRs that are not part of a testbed will be less complex, but nevertheless the re-programmability of SDRs adds complexity over traditional radios. Additional operating training is recommended so that operators of SDRs can more effectively respond to unexpected behavior, malfunctions, off-script commanding, etc. Operators may need to be able to diagnose and debug, especially when a radio subject matter expert is not available. This is particularly important for changes to waveforms or new waveforms that impact commanding, telemetry, and/or the functionality of the radio. Operators will need to be knowledgeable about the radio and be flexible to future changes.

D. Need for Telemetry Flexibility

Telemetry is another characteristic that has unique challenges when applied to SDRs. During development, thought should be given to the flexibility of the telemetry implementation. As new waveforms are developed, and telemetry is added or changed, the ground and avionics software impact needs to be minimized. The existing ground and avionics software should not need to be changed to accommodate new waveform telemetry.

As with commanding, the telemetry of each SDR in the SCaN Testbed was implemented with a different approach, all based upon previous implementations. The GD radio has twelve, 16-bit status words to communicate periodic telemetry status to avionics over the MIL-STD-1553 bus in a fixed binary packet (see Figure 8). The packet is pushed out by the radio to avionics and it must be read before it is overwritten. The telemetry status words contain a limited amount of spare bits that can be used for new telemetry. Use of the spare bits or re-purposing of previously used bits is possible for waveform changes, but the new values are difficult to interpret without ground software changes. The GD approach is efficient, but inflexible to changes. In the JPL radio, the periodic telemetry size is limited to the size of a MIL-STD-1553 packet, minus overhead. The waveform telemetry inside this packet is returned using American Standard Code for Information Interchange (ASCII) characters, which can be inefficient depending on the native data type. An operator can query the radio for specific additional telemetry or do a "query all" to get all the available telemetry (the periodic telemetry is a subset of all available telemetry). The results of this on-demand query are returned in a console window, which is embedded in a MIL-STD-1553 packet and reassembled as a text display on the ground. The queried telemetry is displayed on a serial console window and thus is not as simple to view and parse as the other telemetry displayed in a graphical user interface (see Figure 7). The JPL radio approach is flexible at the expense of a high level of overhead for interpretation. The Harris radio uses a telemetry packet that is filled by reading selected parameters from a configuration file. The telemetry is returned in response to an avionics query and the configuration file determines which telemetry to poll. The Harris telemetry is contained within Extensible Markup Language (XML) structure with name/value pairs and additional information including min and max values. Generic (new) telemetry can be displayed by ground software through a name/value pair display. The Harris approach is a compromise approach that has fixed packet space, but easily modifiable contents (see an example configuration file in Figure 6).

Bit Position															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PRG EE ERR	PRG EE	RAM CS STS	RAM PWR STS	EE PWR STS	PRM PWR STS	NV CS STS	MEM DMP	FILE UPLD	FILE DNLD	FPGA STS	SP	SP	SP	SP	SP
MSB															LSB

Figure 7. Example of GD SDR Telemetry Packet

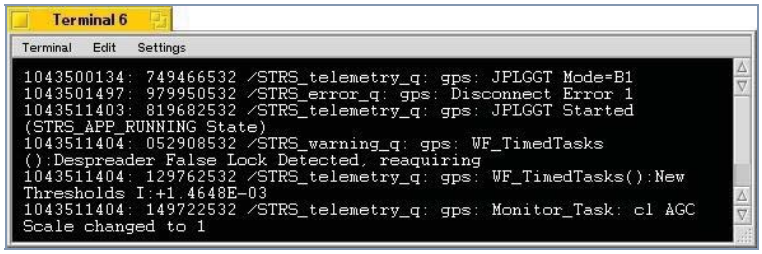

Figure 6. Example of JPL SDR Text Telemetry

```
<?xml version="1.0" ?>
<HarrisWaveFormPropertySettings>
  <Property name="STRS_APP_STATE" value="" />
  <Property name="APQM_DB" value="" />
  <Property name="RX_CENTER_FREQ" value="" />
  <Property name="RX_ATTEN_VAL" value="" />
  <Property name="USE_XML_DEFAULTS" value="" />
  <Property name="WF_GPP_INFO" value="" />
</HarrisWaveFormPropertySettings>
```

Figure 5. Example of Harris Name/Value Pairs in XML

Pre-defined fields are rigid and inflexible, and not well-suited for SDR telemetry reporting. Use of name/value pairs is recommended for the primary telemetry downlink, with the caveat that flexible-length generic string-based telemetry may be useful for diagnostics or other telemetry that does not need to be parsed easily. Thought must be given to the ability to display new telemetry in an understandable format on existing ground software graphical user interfaces. Complete SDR documentation is also important for the ability to expand telemetry and develop new waveforms with new telemetry.

E. Architectures to Assist On-orbit Diagnostics

If the SDR experiences an anomaly on orbit, determining the cause and solution can much more complicated than debugging the issues when the system is still physically accessible. SDRs typically have interfaces for loading

software and viewing diagnostic information available during the development stage that are not accessible when in the operational stage. The command and telemetry set available via the avionics interface may also be reduced from the full set available during development. Limiting the set of in-flight commands is necessary to perform reasonable verification and validation, where all commands must be tested. But this limited set may eliminate seldom-used diagnostic commands.

The SDRs on the SCaN Testbed each have two interfaces with the avionics: one for command and telemetry and a second for the transfer of data (see Figure 9). For the two S-band SDRs, the command and telemetry link is via MIL-STD-1553. For the Ka-band SDR, the command and telemetry interface uses SpW. All three SDRs use SpW for the data interface. Each SDR has additional development interfaces (shown in the clouds in Figure 9). The JPL SDR has an RS-232 interface used to bypass Avionics and directly interface to the SDR, a Joint Test Action Group (JTAG) interface for programming the FPGA directly, and access to various pins of the FPGAs. The GD SDR has a RS-422 input which is reconfigurable by the waveform, JTAG for FPGA development, and digital input/output (I/O) pins. It also provides a Universal Asynchronous Receiver/Transmitter (UART) for a command line interface to a VxWorks shell running on the GD general purpose processor (GPP). The UART interface was used during development and test to view and modify memory contents, which is useful for troubleshooting. Some digital I/O lines remain directly connected to Avionics in the flight configuration, but some are disabled. The Harris SDR has an RS-422 interface that provides VxWorks shell access, and the SDR also provides direct access to the internal "card cage" backplane signals, which could include JTAG with slight modification.

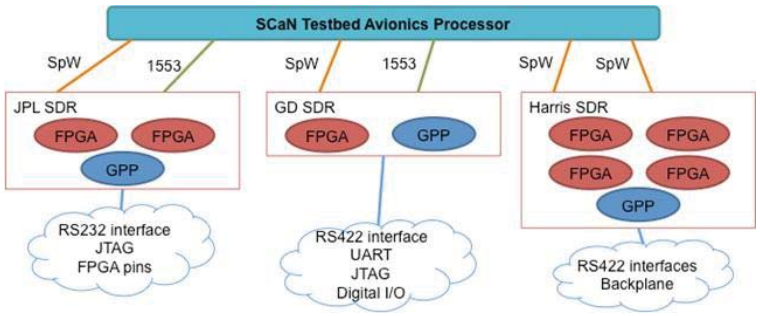

Figure 8. SCaN Testbed SDR Data Interfaces

It is recommended that the avionics processor be given the capability to access all available diagnostic commands and diagnostic interfaces in the flight state. Use of these commands and interfaces would be reserved for off-nominal conditions.

Other architecture and behavior modifications could be added to the SDR-based system to improve diagnostics and repairability. The avionics software could be developed to add autonomy. If certain link or telemetry conditions exist, the avionics processor could be programmed to enable an action within the SDR. For example, if the SDR status shows that the radio is not properly responding to commands, avionics could reset the SDR or force entry into a failsafe state to retrieve diagnostic files such as warning and error logs. The reverse situation is also true: the SDR can also be configured to diagnose and resolve errors occurring within the critical avionics system.

Reducing the risk of new software loads is necessary for SDRs, since a mission is unlikely to be able to afford the additional weight or cost of a duplicate SDRs. Methods such as providing a "gold code" failsafe state are important to allow recovery of a SDR following a failed waveform load.

F. Importance of High-Fidelity Engineering Model

Most NASA missions have a version of the flight system kept on the ground, which is usually an engineering model, but may be a flight spare or only a prototype. Flight missions that utilize SDRs have a strong motivation for the highest fidelity model on the ground because any software revisions or new waveform applications will need to be tested and characterized in a controlled environment first. Uploading a new waveform to a space asset without relevant verification and validation testing may put significant risk on a mission, thus driving the importance of a high-fidelity payload model. The ground integration unit (GIU), as it's called for SCaN Testbed, can have multiple uses throughout the life of a project. In the case of SCaN Testbed, the engineering model SDRs and other subsystems were assembled into the GIU and tested before the flight hardware assembly, helping define integration and test procedures. Since launch, the GIU has been used for new waveform development, pre-upload software verification and validation activities, and debugging anomalies that occur on-orbit.

The SCaN Testbed GIU is a flight-like ground system, but some compromises were made due to cost and schedule. The differences between the GIU and flight system are as follows:

- The GIU has the Engineering Model (EM) Versions of the SDRs. There are subtle timing differences between the Flight SDRs and EM SDRs due to the EM Units having many commercial components instead of

radiation hardened components rated for space flight. This prevents thermal vacuum testing with the GIU to simulate the space environment.

- The GD SDR in the GIU does not have a Power Amplifier (PA) slice. Testing can still be accomplished with cable connections and no antennas. However, transmit effects specific to the PA are lost.
- The GIU only has the Operational Heater installed; survival heaters are not present. More importantly there is no means to cool the GIU. On-orbit operational temperatures have typically been below room temperature.
- The L-Band, S-Band and Ka-Band RF Signals that go to antennas on the Flight System are terminated at coax connectors on the GIU, to which users can connect their test equipment.
- Two Ka-Band TWTAs are installed on the GIU. These are a Commercial Ka-Band TWTA which is used for Harris SDR Waveform development and the Spare Lunar Reconnaissance Orbiter TWTA which is used for final validation of the Harris SDR Waveform prior to upload to the Flight System. Both TWTAs will be available for the life of the project.
- A TWTA pre-amplifier is installed between the Harris up-/down-converter and the input to both TWTAs. The pre-amplifier in conjunction with the variable attenuator allows the signal levels to the TWTAs to be adjusted to place them in the correct saturation points.
- The GIU can transmit lower power Ka-Band RF by bypassing the two TWTAs.

The development of a new waveform for the Harris SDR brought limitations of the SCaN Testbed GIU to the forefront. It was the first new waveform development for the Harris SDR, and the waveform passed all of its verification and validation testing on the GIU. Upon beginning operations on the flight system there were issues with command and telemetry between the SDR and avionics. It was not immediately traceable to the new waveform application because some successful tests were conducted. The issue was significant in that it required a reboot of the SDR to recover. After a few days of this intermittent behavior, new waveform operations were suspended until the root cause could be found.

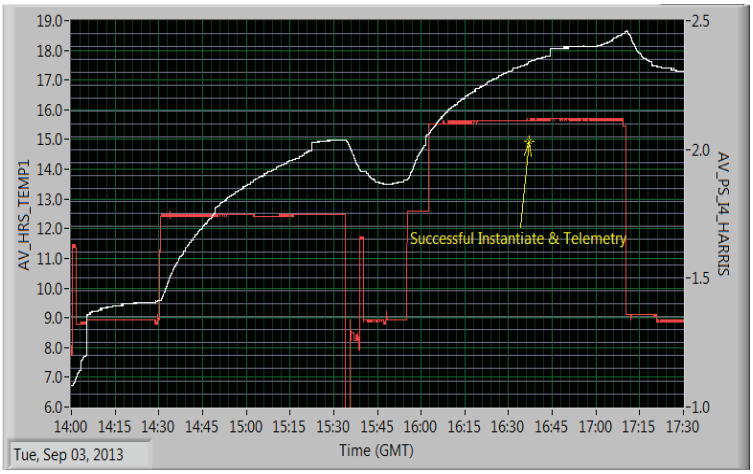

Figure 9. New Waveform Testing Temperature Correlation.

Troubleshooting the problem was very difficult because it could not be duplicated on the GIU. The new waveform ran nominally on the GIU using the exact same configuration and command sequences that failed on the flight system. Log files and telemetry from the flight system were carefully studied revealing the problem was in the SDR's internal communications between the processor and FPGAs. Still there was no indication as to why the problem didn't occur on the GIU as well, until a study of the SDR's temperature correlation with the problem yielded an explanation. The problem only occurred when temperatures were colder than 14 Celsius, a situation that does not occur with the room temperature GIU.

Figure 10 shows telemetry data over time of one of the flight system tests with the new waveform. The lighter trace that generally ramps up is the SDR internal temperature with its scale on the left in Celsius. The darker discrete level trace shows the SDR's current draw in amperes with the right side scale, an overall indication of operations on the SDR. When temperatures were below 14 C initially the new waveform would not instantiate and run properly, as indicated by the current draw being only about 1.75 A. However, after the SDR had warmed and a second attempt was made the waveform instantiated and ran nominally. Other similar tests on different days supported this temperature correlation. Other telemetry indicators were used to align events and confirm the analysis.

If building an EM that can be temperature cycled routinely for new software verification testing is unreasonable, then there needs to be a way to insure reliability of new waveform code for space environment operations. For the new waveform SCaN Testbed issue described in the preceding paragraphs there was, in fact, already proven code in the launch waveform that works over all flight system temperatures. Unfortunately at the time of the new waveform development the proven source code was not yet available. One goal of STRS is facilitating waveform code reuse, especially on non-identical SDR platforms. The STRS Waveform Application Repository is building up a database of NASA mission waveform applications that missions can reuse and to which new code can be contributed. If proven flight waveform applications and associated components are available then the need for a higher fidelity engineering models can be reduced (see Figure 11).

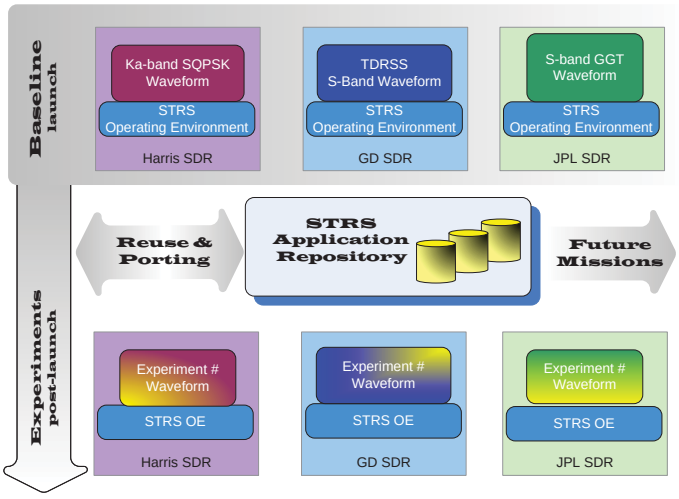

Figure 10. STRS Application Repository Approach

G. 3rd Party Software Development

One of the goals of the SCaN Testbed Project is to conduct experiments that produce applications for the Testbed's software defined radios, compliant with the STRS Architecture. These applications demonstrate the functionality and value of the SDRs and become part of the STRS Repository, to be available for future uses on other STRS compliant platforms.

Development of new software for SDRs launched into space pose new challenges for both NASA and platform providers. For NASA, future missions must either obtain software (i.e. new applications or updates) from the original platform and software provider, or develop new software itself (or have developed) using documentation and information from the platform provider. The former approach creates a dependence on the original platform provider and risks to both NASA and the company for projects that need new software years after launch. To have the original platform developer provide new software, the company needs to maintain the product line and have developers on hand and available to develop the new software for the original platform. A second approach would be for NASA to maintain a maintenance support contract with the platform provider for years after delivery as a sort of insurance in case new software is needed, but this obviously raises a significant cost to the mission. For the platform provider, these approaches also present risk to maintain a software product line for hardware which may have become obsolete. Even with the benefit of a maintenance contract, the company would have to dedicate employees to support future software development, possibly at the expense of a more profitable and current product line.

STRS was developed to relieve these risks for both NASA and the platform developer. STRS enables 3rd party application software development to reduce dependence on unique, proprietary architectures and dependence on specific companies and even specific developers, while still protecting original platform developer information. With each software application submission to the STRS repository, NASA seeks a license to use, distribute, and develop derivative works for U.S. Government purposes and having official business (e.g. contract, Space Act Agreement) with NASA with suitable protections for the developer. This allows the original developer to do whatever they wish commercially with the software, and yet prevents others from commercially benefiting from having access to the software. These arrangements are common and are typically considered or covered under Government Purpose Rights. This approach allows NASA to use the software without further charge and maintain a relationship with the original developer in case new features are needed which were not delivered with the original software. NASA could also make changes to the software itself and then provide a copy of the changes back to the original developer. Therefore, there is a mutual benefit for access to any work based on the original waveform submission. Also, there seems more and more opportunity for platform or software developers to provide servicing to those companies that use develop new software or use software from the repository. The complexity of the radios and the software are such that there still is opportunity for the original platform or software developer to provide consultation if one needs to get the last bit of performance from a radio platform.

STRS requirements call for the platform developer to provide a variety of software and documentation to the project. Platform developers provide platform-specific software ("wrapper") for each user programmable FPGA which provides an abstraction to command and data from the processor module or other platform hardware resources available to the application. Documentation on the software interfaces of the platform-specific wrapper describes signal names, signal polarity, format, and data type, timing constraints, clock generation methods, and other aspects. Other software includes the STRS Operating Environment (OE, which is typically not part of the repository unless needed for application developers). Documentation must include a description of the platform hardware and resources available to application developers, STRS OE description and functions, operating system information, and user operation.

Waveform application developers also submit software and documentation to the STRS repository. Software includes applicable system or component software models, or application software sources (e.g., C, C++, header files, VHDL, and Verilog), executable versions and application libraries, if applicable. Documentation includes external software interfaces (e.g., signal names, descriptions, polarity, format, data type), STRS application behavior, version description, and others.

The software delivered with each of the SDRs has functioned well since launch. Each SDR was delivered with a waveform which is compatible with the existing services of the TDRSS Space Network. NASA has made minor modifications to the delivered waveform from GD and developed a small test waveform on the Harris SDR. NASA has also made significant changes to the waveform for the JPL SDR originally developed by the Glenn Research Center (GRC) and the Goddard Space Flight Center (GSFC).

Developing or editing waveforms on the GD and Harris SDR was an excellent experience to assess the delivered software and documentation. While there was a cursory review of the documentation at delivery, the urgency to deliver the flight system on time and test the delivered hardware prevented a more thorough review.

STRS continues to work on standardizing the required deliverables documentation content, but following are a few lessons thus far on software development and delivery.

- Require that a test/sample waveform be delivered with references to the applicable documentation. Show that the test waveform was developed according to the provided documentation. The test application should show how it uses the delivered software wrapper and documentation for access to platform resources.
- Request a template for future waveform applications on the SDR. Describe the process to load and execute additional applications. Ensure that commands and telemetry are not inadvertently limited to the delivered waveform.
- Review software and documentation from the 3rd party developer's perspective. Ensure that the documentation is all that's needed to develop new software. Don't rely on development presentations, or other "insider" information not described in the documentation intended for waveform developers.

To fully assess the documentation's usefulness for future developers, it is recommended that an in-house waveform be developed as soon as practical using only the available documentation. This is the best way to assess if the information in the documentation is sufficient for future waveform development. Developing the waveform soon after delivery will help increase the chances of having the platform developers available to make documentation improvements as needed; waiting too long may result in delays due to the experts being moved to other projects.

IV. Summary and Conclusions

This paper describes knowledge gained operating and modifying three unique SDRs in orbit. These lessons relate to aspects such as the recommendation to test the SDR hardware independent of the software and methods to enable these tests. It describes the complexity working with obsolete hardware and software. The need for more flexibility in the command and telemetry packet exchange is discussed. Suggestions are made for modifying the on-orbit architecture between the SDR and avionics to enhance the ability to assist with diagnostics on-orbit. The need for high fidelity engineering models is important but may be mitigated somewhat by STRS-enabled reuse. The complexities involved in enabling third-party development of new applications on the SDRs are described. SDRs enable enhanced capabilities, from development through on-orbit operation, but have unique challenges described in this paper to assist future SDR missions.

V. Future Work

NASA is soliciting universities, industry, and other Government agencies to propose experiments using the testbed via a Cooperative Agreement Notice and Announcement of Opportunity[a].

Additional, more complex waveforms are in development at this time to be used with the SCaN Testbed. Tests with the new waveforms on the engineering models and their use on-orbit will provide additional knowledge. Third-party developers are initiating design of new waveforms that will be the first independent use of the SDRs and its associated documentation. Limitations with the documentation and test tools will be uncovered, providing additional insight for future SDR systems.

References

[1]"Space Network Users' Guide (SNUG)," NASA Goddard Space Flight Center, 450-SNUG, Revision 9, August 2007.

[2]Reinhart, R., Kacpura, T., Handler, L., Hall, C., Mortensen, D., Johnson, S., et al. "Space Telecommunications Radio System (STRS) Architecture Standard, Release 1.02.1," NASA TM-2010-216809, December 2010.

[3]Mortensen, D., Bishop, D., and Chelmins, D., "Space Software Defined Radio Characterization to Enable Reuse," *30th AIAA International Communications Satellite System Conference*, ICSSC 124, AIAA, Reston, VA, 2012.

[4]Downey, J., Reinhart, R., and Kacpura, T., "Pre-Flight Testing and Performance of a Ka-band Software Defined Radio," *30th AIAA International Communications Satellite System Conference*, ICSSC 124, AIAA, Reston, VA, 2012.

[5]Johnson, S., Reinhart, R., and Kacpura, T., "CoNNeCT's Approach for the Development of Three Software Defined Radios for Space Application," *2012 IEEE Aerospace Conference*, Aerospace Conference, 2012 IEEE, Big Sky, MT, 2012, pp. 1-13.

[6]Chelmins, D., Downey, J., Johnson, S., and Nappier, J., "Unique Challenges Testing SDRs for Space," *2013 IEEE Aerospace Conference*, IEEE, Big Sky, MT, 2013.

[7]Kacpura, T., and Varga, D., "SCaN Testbed Development and Lessons Learned," *2012 International Astronautical Congress*, International Astronautical Federation, Naples, Italy, 2012.

[a] "SCaN Testbed Experiment and System Information," [online], URL: http://spaceflightsystems.grc.nasa.gov/SOPO/SCO/SCaNTestbed/Candidate/ [cited 3 March 2014].