Open Architecture Standard for NASA's Software-Defined Space Telecommunications Radio Systems

Richard C. Reinhart and Sandra K. Johnson
Glenn Research Center, Cleveland, Ohio

Thomas J. Kacpura
ZIN Technologies, Cleveland, Ohio

Charles S. Hall
Analex Corporation, Cleveland, Ohio

Carl R. Smith and John Liebetreu
General Dynamics–Advanced Information Systems, Scottsdale, Arizona

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to *help@sti.nasa.gov*

- Fax your question to the NASA STI Help Desk at 301–621–0134

- Telephone the NASA STI Help Desk at 301–621–0390

- Write to:
  NASA Center for AeroSpace Information (CASI)
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TP—2008-214941

Open Architecture Standard for NASA's Software-Defined Space Telecommunications Radio Systems

Richard C. Reinhart and Sandra K. Johnson
Glenn Research Center, Cleveland, Ohio

Thomas J. Kacpura
ZIN Technologies, Cleveland, Ohio

Charles S. Hall
Analex Corporation, Cleveland, Ohio

Carl R. Smith and John Liebetreu
General Dynamics–Advanced Information Systems, Scottsdale, Arizona

Acknowledgments

Open Architecture Standard for NASA's Software-Defined Space Telecommunications Radio Systems

Richard C. Reinhart and Sandra K. Johnson
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Thomas J. Kacpura
ZIN Technologies
Cleveland, Ohio 44142

Charles S. Hall
Analex Corporation
Cleveland, Ohio 44142

Carl R. Smith and John Liebetreu
General Dynamics–Advanced Information Systems
Scottsdale, Arizona 85257

Summary

NASA is developing an architecture standard for software-defined radios used in space- and ground-based platforms to enable commonality among radio developments to enhance capability and services while reducing mission and programmatic risk. Transceivers (or transponders) with functionality primarily defined in software (e.g., firmware) have the ability to change their functional behavior through software alone. This radio architecture standard offers value by employing common waveform software interfaces, method of instantiation, operation, and testing among different compliant hardware and software products. These common interfaces within the architecture abstract application software from the underlying hardware to enable technology insertion independently at either the software or hardware layer. This paper presents the initial Space Telecommunications Radio System (STRS) Architecture for NASA missions to provide the desired software abstraction and flexibility while minimizing the resources necessary to support the architecture.

Software-Defined Radios in Space

Software-defined radio (SDR) has emerged as a revolutionary approach to developing and operating communication radios for a broad range of domains including commercial, military, and public service. Advances in digital signal processing and computing power have evolved radio implementations from being primarily electronically (i.e., hardware) based to being firmware- and/or software-based. Each of these application domains has leveraged reconfigurable hardware to satisfy the specific desired operating requirements (refs. 1 to 4). In many of these domains, engineers have pursued standards or other sets of rules to establish the policies required to achieve a desired degree of commonality within the domain for development and operations (refs. 5 to 8).

As NASA enters the future of space science and exploration, the growth of reconfigurable electronics offers the opportunity to improve the way space missions develop and operate space transceivers for communications and navigation. Reconfigurable SDRs with communications and navigation functions implemented in software (e.g., firmware) provide the capability to change the functionality of the radio during mission development or after launch. Changing the operating characteristics of a radio after it has been deployed to space offers the flexibility to adapt to new science opportunities. By adapting generic space platforms to meet specific mission requirements, SDRs can recover from anomalies within the science payload or communication system and potentially reduce development cost and risk.

Traditional approaches to radio development have been exemplified by proprietary or custom implementations in application specific integrated circuits (ASICs), meeting a specific set of mission requirements. The advent of software-based radio functionality offers NASA the opportunity to separate the software proliferation and its associated complexities from the underlying and evolving reprogrammable hardware technology by adopting an open architecture standard. An SDR architecture may be defined as "…a comprehensive, consistent set of functions, components, and design rules according to which radio communication systems may be organized, designed, constructed, deployed, operated, and evolved over time. A useful architecture partitions functions and components such that (a) functions are assigned to components clearly and (b) physical interfaces among components correspond to logical interfaces among functions" (ref. 9).

NASA is developing the STRS Architecture (refs. 10 to 12) as an open architecture standard. If adopted, this architecture will reduce dependence on ad hoc implementations, which have inherent risks. These implementations leave NASA dependent upon the proprietary developments, and they provide unique operation and test interfaces, varying hardware descriptions,

software development artifacts, and documentation standards. These developments are typically not extensible or scalable, and do not enable design reuse or other economies of a standard, open architecture approach.

NASA can successfully apply SDR technology in a common way across missions while preserving the unique implementations necessary to meet specific mission needs by developing radios that are based on an open architecture standard. An open architecture standard promotes the use of published, well-defined interfaces that enable different vendors to provide radios that conform to the interface standard, thus providing commonality among different implementations at the operation and test interfaces and enabling interoperability between providers of different hardware and software elements. Standard interfaces provide flexibility for component replacement, technology insertion, and risk reduction through standard hardware and software component reuse. Standards promote the growth of a large base of domain experts—Agency personnel, software and hardware providers, and the user and operations communities—all knowledgeable of the common standard. NASA's goal in developing the STRS standard is to simultaneously capture the benefits of SDR technology and the economies and benefits of an open architecture standard.

While the STRS architecture is applicable to both space and ground domains, NASA tailored the architecture definition to be responsive to the unique constraints and challenges posed by operating in the space environment.

Space environment.—Space components are designed to withstand space effects, single event upsets, total ionizing dose, and ionizing dose rate. The performance of radiation capable processors is a generation or two behind their terrestrial equivalents, which limits their capability.

Spacecraft resources.—Size, weight, and power (SWaP) are premium space resources. The limited availability of these resources constrains the radios conforming to the architecture.

Reliability and availability.—Software-based radios and changes to on-orbit configurations must meet high reliability requirements, especially for safety-critical applications such as crewed spacecraft.

Static waveform deployment.—Waveforms will deploy on-orbit in a predicted manner based on careful and verified ground testing for the intended platform configuration. Waveform changes during flight will be made in controlled conditions, through a commanded sequence from ground control.

Long mission development times.—The development time of space radios is often much longer than terrestrial equivalents. Late changes or evolving requirements must be accommodated with software once hardware is fixed early in the design cycle.

The STRS architecture, designed to be responsive to the uniqueness of the space environment, ensures that NASA will benefit from systems that can be reprogrammed with new software years into the future, and will enable NASA to capture the natural economies that arise in an open-standard environment.

STRS Architecture Description

The STRS provides commonality to develop, operate, and maintain space- and ground-based reprogrammable assets. The STRS is composed of the architecture standard, compliance and certification laboratory, repository of hardware, software components and artifacts, an acquisition model tailored to maximize the benefits of an open architecture, reference implementations, and associated technology programs to advance the components of space radios.

The STRS architecture standard defines both internal and external radio interfaces as illustrated in figure 1. Internally, the architecture describes the connections between radio components, beginning with the operating environment separating the waveform application from the underlying hardware (on the general purpose processing module, defined later) and the interconnections among different radio components. The STRS open architecture definition identifies interfaces and applies rules for the hardware and software to realize the benefits of SDR.

The radio functions are distributed among various modules to organize platform services and waveform functions within the radio. Modules are a logical division of functionality (with specific physical implementation of the architecture left to the designer) to maintain common interface descriptions, terminology, and documentation among SDR developments. Figure 1 shows several modules. The general purpose processing module (GPM) provides the basic software execution process based on general purpose processors. The GPM is a required module within the STRS radio to support the execution of the software-based operating environment responsible for waveform instantiation and execution, certain radio services, and hardware abstraction. The signal processing module (SPM) conducts high-speed data and signal processing and clock distribution and may provide an external interface to the payload data. Radiofrequency module(s) (RFM) provide radiofrequency (RF) front-end functions for waveforms anticipated to operate in the ultra high frequency (UHF, S, X, Ku, and Ka frequency bands, which are allocated for use by NASA in space. Other modules not explicitly shown include a security module or optical module as required by the transceiver.

The hardware abstraction layer (HAL) is among the SPM interfaces that provide the library of functions that offer a platform-independent view of the specialized hardware (e.g., field programmable gate array (FPGA)) algorithm implementations by abstracting the physical hardware interfaces. It implements the software that is directly dependent on the underlying hardware. The proposed standard requires that developers publish hardware interfaces (i.e., HAL application programming interface (API)) such as FPGA data and control interfaces. The HAL API must include a description of each method and/or function used, including its calling sequence, return values, and functionality explanation. This
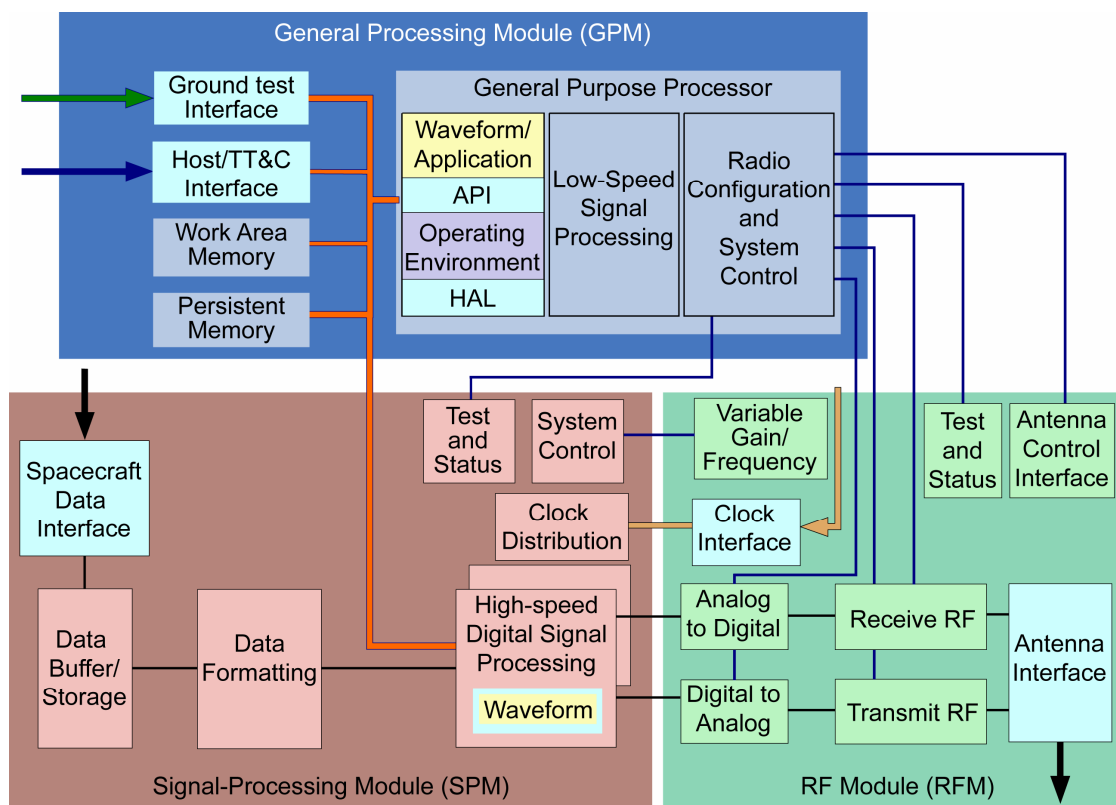
Figure 1.—STRS modular architecture.

permits NASA to access the developer's proprietary, intellectual property associated with the waveform algorithms by exposing the interfaces used in the FPGA or other hardware for subsequent developments or corrections without relying on the continuing involvement of the original developer, as traditional radios require.

The hardware interface definition (HID) requires developers to provide a description of the physical hardware interfaces used in the implementation and a mapping of the control interfaces to each of the modules. These HIDs are provided by the module developers to allow NASA to augment or replace modules from in-house and outside sources, possibly from other than the original vendor. Competing existing radio modifications or additions instead of relying on sole source contracts to legacy providers offers an opportunity for cost reduction.

Examples of the HID include interface type, transfer speeds, signal definition, addressing, data width, timing, control signals, messages, interrupts, hardware and software boundary (model, drivers, custom interfaces, and operating environment), and implementation summary (size, weight, power consumption, radiation level, and reliability).

Figure 2 illustrates internal software interfaces. Represented as multiple layers, the software interfaces provide applications access to the hardware platform or platform services through the APIs and STRS operating environment (OE).

OE elements include STRS infrastructure, real-time operating system (RTOS), HAL, board support packages (BSP), and Portable Operating System Interface (POSIX). The OE abstracts the platform hardware (e.g., processors and specialized hardware) and makes the software portable to a new platform. To comply with the STRS architecture, the radio developers must provide an STRS infrastructure that supports the interfaces and radio services used to execute and control the STRS applications as specified in the architecture standard. Platform services provided by the infrastructure to the application or waveform include waveform control, spacecraft timing, file management, and others. Services may include those provided by the operating system and/or custom services provided by the infrastructure. Other elements of the OE such as the RTOS provide access to processor operating system (OS) services and functions and a minimum POSIX profile for the allowed OS services.

APIs, a key aspect of the software architecture, are used to facilitate software configuration and control of the target platform. The philosophy on which the STRS architecture is based avoids the conflict between open architecture and proprietary implementations by specifying a minimum set of APIs to execute waveform applications and deliver data and control messages to hardware components. Application-specific control and data message contents are encapsulated in data elements, which are transported to destination components that

have the knowledge to use the contained data. In this manner, STRS is decoupled from the intellectual property rights of platform, waveform, and module developers. The definition of APIs is based on a set of sequence diagrams derived from the use cases identified in the STRS Software Architecture Concepts and Analysis document (ref. 12).

The STRS software subsystem in figure 3 is a view of the software elements and logical connections of the OE and STRS applications. The STRS applications include communication waveforms and supporting dedicated services that inherit the POSIX, STRS, and waveform application programming interfaces.
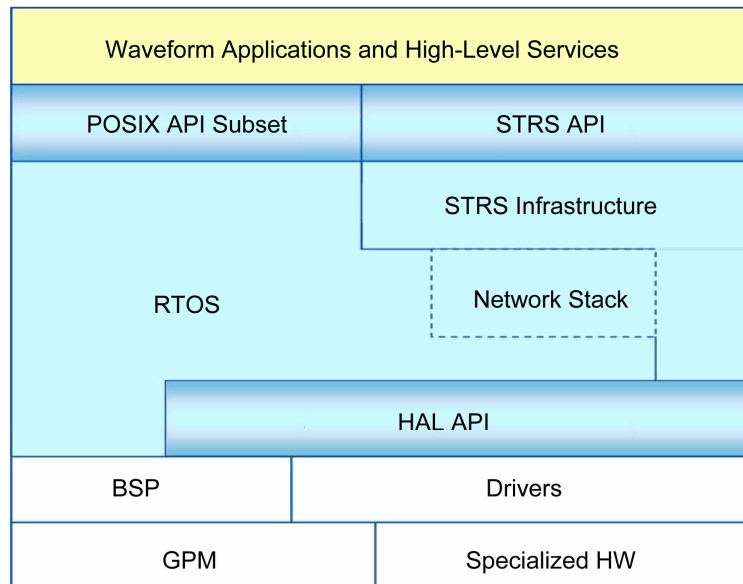

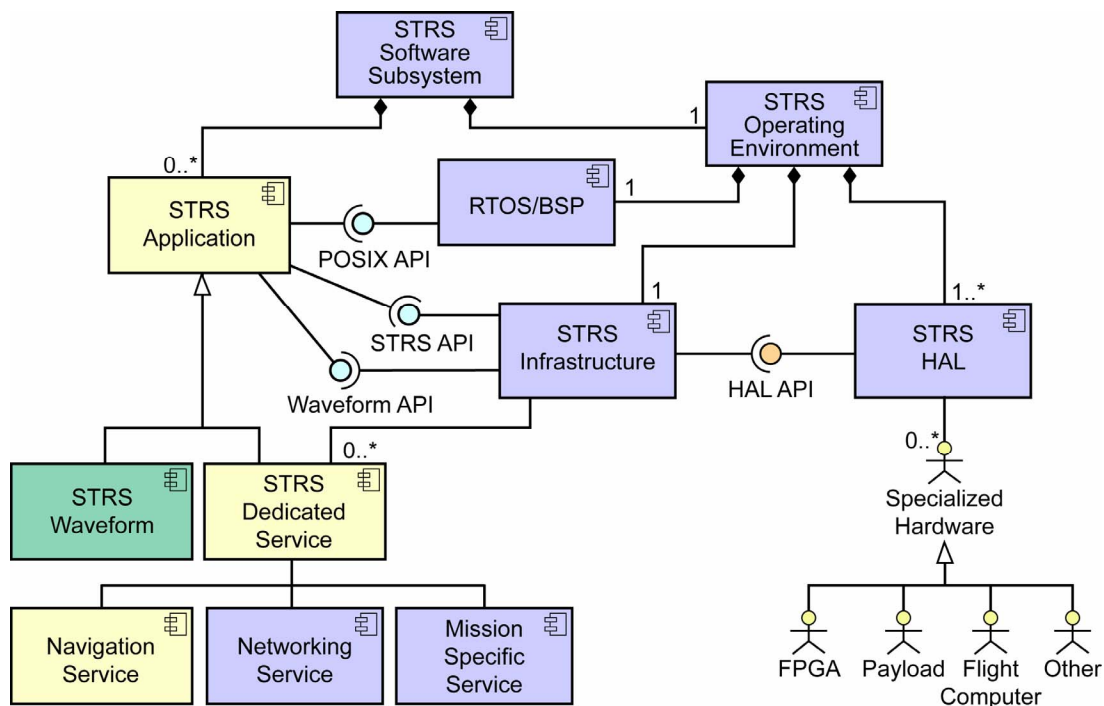Figure 2.—STRS internal interfaces.

Figure 3.—STRS software subsystems.

An STRS waveform comprises the end-to-end functionality (e.g., modulation, coding, frequency conversion, and filtering) and bidirectional transformations applied to information content that is transmitted over the air. Dedicated services are used to support STRS waveforms and provide mission-specific services such as navigation (ref. 13), networking link monitoring, data compression, and others. As the architecture matures and functional boundaries expand, higher-level waveform function interfaces will be added to the architecture to support common interfaces to networking and security applications.

Dedicated services may be used to extend the capabilities of the STRS infrastructure by creating new platform services.

This provides a mechanism to both evolve and scale the architecture based on mission requirements (mission-specific services). For example, missions with unique network requirements could create dedicated services to provide advanced networking capabilities such as routing packets between multiple STRS nodes.

The waveform applications can be distributed upon various processing elements and specialized hardware (e.g., FPGA and/or digital signal processor (DSP)) as determined by the radio developer. The waveform developer can make this allocation choice depending upon the waveform requirements and the capabilities of the target radio platform.

STRS waveform applications use platform services in the form of specific APIs defined by the architecture. The platform services are designed to reduce the time to port waveforms from one platform to another since the same set of interfaces and services are provided by each platform. The architecture promotes waveform application code reuse by providing a library of compliant, reusable software routines.

STRS applications use three sets of interfaces to interact with the STRS OE, as depicted in figure 3. The architecture strives to use standard interfaces already defined and available. (1) The minimum POSIX interface (ref. 14) to the RTOS is the Institute of Electrical and Electronics Engineers (IEEE) PSE51 profile that provides a uniform mechanism to access operating system services. Larger profiles PSE52 and PSE53 are also supported by the architecture, with upward compatibility. (2) The STRS API provides the interface to the STRS infrastructure for radio control, system management, device control, and messaging. (3) The waveform API specifies the set of interfaces and functions that STRS application developers must implement. These interfaces are used by the STRS infrastructure to control and configure the application. In addition, the HAL API defined by the platform developer presents an interface to the lower-level device drivers required to support specialized hardware. It also defines the physical and logical interfaces for intermodule and intramodule integration.

Specialized hardware represents interfaces to the physical layer hardware modules and external systems existing on the spacecraft.

Finally, STRS configuration files contain platform and waveform specific information to customize waveform installation using eXtensible Markup Language (XML), validated by a unique XML schema for either platform or waveform configurations. Platform configuration files provide the STRS infrastructure with information about the devices and modules installed in the system and the device drivers required to perform radio operations.

STRS Reference Implementations

Multiple locations are developing STRS reference implementations using different platforms and operating systems.

Glenn Research Center (GRC) is developing a reference implementation on a Spectrum Signal Processing, SDR–3000, a commercial off-the-shelf SDR platform. The GRC operating environment used on the SDR–3000 consists of the Wind River VxWorks RTOS and vendor-supplied board support package. The STRS infrastructure is developed in C++ and will support the deployment of STRS waveform applications written in either C or C++.

The reference implementation has matured the application, device, and infrastructure APIs. Application APIs have been grouped similarly to the Joint Tactical Radio System Software Communications Architecture (JTRS SCA) and Object Management Group (OMG)/SWRadio resource component model interfaces such as "ControllableComponent," "LifeCycle," "PropertySet," "TestableObject," and "ComponentIdentifier" (ref. 15).

Other APIs categories include Messaging/Interprocess Communications, Device Control, Platform Services (Infrastructure), System Management, and Application/Waveform Control.

The architecture does not require deployment of XML but instead allows developers to transform the XML to a compact, platform-specific form that requires fewer resources to store and parse. The GRC implementation uses eXtensible Stylesheet Language (XSL) to transform the XML language to a variation of S-expressions, a data structure found in the LISP programming language (ref. 15).

At General Dynamics–advanced Information Systems in Scottsdale, Arizona, two STRS environments have been developed. The first platform consists of Microsoft's .NET that provides OS services to evaluate the capability of the STRS APIs. This platform enables rapid development of the STRS capabilities without being burdened with an embedded environment. Using this platform, a global positioning system waveform was implemented integrating a bitwise simulation of specialized signal processing with a STRS-compliant waveform. The second environment consisted of Wind River's VxWorks RTOS, executing on the General Dynamics StarLight Space SDR development platform, providing the first integration of specialized signal processing hardware with STRS. The GPS waveform, written using Microsoft .NET, was easily ported to the VxWorks-StarLight platform. The waveform was then validated by tracking multiple on-orbit GPS satellites. In addition, General Dynamics has implemented an STRS-compliant Ranging Crosslink Waveform that is applicable to

formation flying missions providing centimeter resolution between spacecrafts.

NASA's Jet Propulsion Laboratory began a reference implementation on an Electra space platform. Initial work included modifying the RTEMS OS to be POSIX-compliant and developing the API set.

These reference implementations demonstrate the architecture's extensibility and scalability to easily adapt between platforms and operating systems.

Vendor Perspective on STRS

The STRS architecture provides radio vendors with the capability to respond more quickly and effectively to new business opportunities. This capability is a result of three factors. First, the adoption of the STRS architecture as the basis for future radio procurement opportunities provides a consistent framework for requirements and capabilities for NASA to use when specifying mission functionality. Second, radio vendors can make use of a mature and consistent understanding of the core environment of the radio to increase existing hardware and software reuse. Third, new procurements can be structured to stipulate new waveform developments on existing platforms.

STRS platforms may initially have higher costs compared to using the historical approach to space radio development. The extra cost for these new platforms is incurred because implementing the API standards and interface verifications for the first time adds development and test costs. Testing is typically the source of a large fraction of the program costs and unproven designs represent the greatest risk component. The modular STRS architecture directly addresses this by providing the capability to incrementally update or enhance an existing platform with a new module or component, rather than develop an entirely new radio. This approach permits a higher technology readiness level (TRL) (ref. 16) to be achieved for these STRS radios over time, determined by the reuse of STRS platform modules and components. Testing is dramatically reduced compared to a new platform, since testing can be confined to focus on (1) the interactions between the new module and the preexisting components and (2) performance of the new module.

After the initial development, subsequent modifications and reuse of the developed platform can take advantage of heritage test processes and software to reduce both test and development costs. As the STRS technology base grows, stable platforms will emerge, consisting of hardware, STRS components (e.g., APIs and HAL), and waveform components. These stable, high TRL platforms will permit rapid, low-cost, low-risk introduction of new waveforms and services. Noteworthy life-cycle cost reductions are anticipated from the introduction of STRS platforms and the stability of the product line, which permits continuous improvement in manufacturing and test processes, as well as refinement of platform and waveform development processes.

Recommendations made by industry participants have produced a database of specific change proposals to the STRS architecture. Over 100 changes to STRS Rev 1.0 have been proposed thus far. Architecture changes include improved taxonomy, specific API changes, improved API descriptions, HID changes to specify physical interfaces and software interfaces separately, and HAL changes to include specifics on configuration and control, functional behaviors, and input/output abstractions.

Conflicting industry feedback is resolved within the SDR Forum, Space Work Group. Over 20 companies have participated on STRS through the SDR Forum. Topics discussed include leverage of the OMG SWRadio Standard (ref. 17) and JTRS SCA for Space, component-based architecture in addition to the current API-based definition, and improvements to the hardware interface definition and signal processing abstraction (i.e., HAL) definitions.

Application to Deep Space, Exploration, and Science Missions

SDRs have the potential to reduce cost and risk and provide enhanced capabilities for a majority of the NASA missions, from small, size- and power-constrained rovers operating on planetary surfaces, to orbiting space outposts. The goal of the STRS standard is to provide the flexibility to allow the radio developer to optimize their implementation for this range of capabilities but retain the benefits of a common standard.

To provide the scalability required for the range of mission types, STRS-compliant radio implementations will vary. For example, an extra vehicular activity (EVA) radio may not have the SWaP available to support extra memory for upgrades to meet future requirements. This would limit the number of waveform services that must be implemented. At the other end of the spectrum, a planetary outpost will likely be upgraded over time to satisfy evolving requirements for new mission objectives. The planetary outpost radio designs must consider how the new capabilities will be best integrated into the existing implementation. In this case, new RF front ends to support a new frequency band or new waveforms may be added after initial operations. The STRS standard requires that the original radio vendor publishes information about the RF front end and waveform interfaces providing flexibility to NASA to successfully develop and integrate new RF front ends or waveforms into the current system without relying on the original provider.

New capabilities will also be needed to support the number of elements planned for future deep space missions (ref. 18). These elements will require extensive networking capability to meet science needs and provide the needed schedule flexibility. Security concerns to protect data integrity and confidentiality will be increased in the networking, multiuser environment. The STRS architecture anticipates these evolutions and provides an extensible path to support the envisioned capabilities, either as new applications within the radio or as external services.

Table I lists the differences in the characteristics of the NASA missions and how an STRS radio implementation might vary to meet the requirements.

The architecture characteristics scale incrementally from the small to large applications to respond to the mission characteristics.

TABLE I.—MISSION-SPECIFIC ARCHITECTURE CHARACTERISTICS

Example applications	Mission characteristics	Architecture characteristics
Rover and surface elements EVA radios	Highly constrained SWaP Single-frequency band and/or channel 1 to 3 duplex, low to medium data rate links Exposed to harsh space environment Limited or no reconfiguration	Designed for minimal memory footprint and processing capacity One-time programmable FPGAs and ASICs (fixed waveform) HID at radio level (RFM integrated with signal processing module (SPM)) Few or no dedicated services implemented Waveform models and software reusable for future missions Unique, mission-dependent form factor
Orbiting relays Launch vehicles Near-Earth and deep space science spacecraft	Multiple frequency bands and/or channels Multipoint-to-multipoint communication Low to very high data rates Navigation and networking Reconfigurable post-launch	Platform services for post-launch modifications and upgrades HAL for specialized hardware to support data rates API for RF front end control High degree of hardware and software reuse Layered architecture supports new technology insertion
Space station Planetary outpost Ground station	Flexible SWaP Limited exposure to space environment Multiple frequency bands and/or channels Distributed processing Network hub functionality	Higher performance and additional capacity components HID at card level to support physical modularity Dedicated services extend STRS infrastructure capabilities Radio functionality controlled through waveform APIs Supports middleware for interprocess communication

Grouping the mission characteristics into sets leads to distinct radio classes. Within the class, hardware and software subsystems may also become common within the mission set, providing additional opportunities for reuse.

Conclusion

The STRS Architecture Standard will reduce NASA's dependence on ad hoc SDR implementations, which have inherent risks associated with them. STRS provides reliable, flexible, and extensible systems that can be reprogrammed with new software years into the future, and makes the economies that arise in an open standard environment accessible to NASA. This open architecture abstracts software functionality away from specific hardware devices through standard interfaces enabling greater design and software reuse and minimizing the impact associated with parts obsolescence. The open architecture allows NASA to reuse its investment in software radio developments, yet

1. Maintain company proprietary approaches and designs behind the common interfaces
2. Reuse the architecture specification among different developments, different radio classes, and different providers
3. Preserve commonality among designs, development, testing, and space qualification processes
4. Provide radio developers the flexibility to define the radio functionality as necessary during the radio design and development process to meet the specific mission requirements

References

1. Efstathiou, D.; Fridman, J.; and Zvonar, Z.: Recent Developments in Enabling Technologies for Software Defined Radio. IEEE Commun. Mag., vol. 37, issue 8, 1999, pp. 112–117.
2. Bergstrom, C., et al.: Software Defined Radio (SDR) Special Military Applications. MILCOM 2002 Proceedings, vol. 1, 2002, pp. 383–388.
3. Bagheri, R., et al.: Software-Defined Radio Receiver: Dream to Reality. IEEE Commun. Mag., vol. 44, issue 8, 2006, pp. 111–118.
4. Hasan, M. Sayeed, et al.: Designing Software Defined Small Form Fit Radios for JTRS Networking. MILCOM 2006 Proceedings, 2006, pp. 1–5.
5. Lackey, R.I.; and Upmal, D.W.: Speakeasy: The Military Software Radio. IEEE Commun. Mag., vol. 33, issue 5, 1995, pp. 56–61.
6. Mitola, J.: The Software Radio Architecture. IEEE Commun. Mag., vol. 33, issue 5, 1995, pp. 26–38.
7. Srikanteswara, S., et al.: A Soft Radio Architecture for Reconfigurable Platforms. IEEE Commun. Mag., vol. 38, issue 2, 2000, pp. 140–147.
8. Software Communications Architecture. JTRS Standards, SCA Ver. 2.2.2, May 2006. http://sca.jpeojtrs.mil/downloads.asp?ID=2.2.2 Accessed Aug. 29, 2007.
9. Mitola, Joseph: Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. John Wiley & Sons, New York, NY, 2000.
10. Reinhart, Richard C.; Farrington, Allen; and Israel, Dave: Space Telecommunications Radio System STRS Open Architecture Description, Rev. 1.0, Space Operations Mission Directorate, Apr. 2006. Available from NASA Glenn Research Center's Digital Communications Branch.
11. Reinhart, Richard C.; Farrington, Allen; and Israel, Dave: Space Telecommunications Radio System STRS Open Architecture Standard, Rev. 1.0, Space Operations Mission Directorate, Apr.

2006. Available from NASA Glenn Research Center's Digital Communications Branch.

12. NASA, STRS Software Architecture Concepts and Analysis, Release 1.0, Cleveland, Ohio, December 2006.

13. Ely, T.: Radiometric Tracking and Integrated Navigation with Software Radios. JPL, May 2006.

14. Standard for Information Technology—Portable Operating System Interface (POSIX). Portable Applications Standards Committee of the IEEE Computer Society and The Open Group, IEEE Std. 1003.1, 2004 ed., 2004.

15. Reinhart, Richard C.; Farrington, Allen; and Israel, Dave: Space Telecommunications Radio System STRS Open Architecture Standard, Rev. 1.01, Space Operations Mission Directorate, Jul. 2007. Available from NASA Glenn Research Center's Digital Communications Branch.

16. Mankins, John C.: Technology Readiness Levels. A White Paper, NASA, April 6, 1995. www.hq.nasa.gov/office/codeq/trl/trl.pdf Accessed Aug. 29, 2007.

17. OMG, SWRadio Specification; PIM/PSM for Software Radio Components. 2005.

18. Andringa, J.M., et al.: Mars Human Precursor Mission Concepts for the Decade 2010–2020. Proceedings of the 2005 IEEE Aerospace Conference, 2005, pp. 191–201.

1. REPORT DATE *(DD-MM-YYYY)* 01-05-2008	2. REPORT TYPE Technical Paper	3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE Open Architecture Standard for NASA's Software-Defined Space Telecommunications Radio Systems	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Reinhart, Richard, C.; Johnson, Sandra, K.; Kacpura, Thomas, J.; Hall, Charles, S.; Smith, Carl, R.; Liebetreu, John	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER WBS 439432.04.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191	8. PERFORMING ORGANIZATION REPORT NUMBER E-16174

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	10. SPONSORING/MONITORS ACRONYM(S) NASA
	11. SPONSORING/MONITORING REPORT NUMBER NASA/TP-2008-214941

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category: 32
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301-621-0390

13. SUPPLEMENTARY NOTES

14. ABSTRACT
NASA is developing an architecture standard for software-defined radios used in space- and ground-based platforms to enable commonality among radio developments to enhance capability and services while reducing mission and programmatic risk. Transceivers (or transponders) with functionality primarily defined in software (e.g., firmware) have the ability to change their functional behavior through software alone. This radio architecture standard offers value by employing common waveform software interfaces, method of instantiation, operation, and testing among different compliant hardware and software products. These common interfaces within the architecture abstract application software from the underlying hardware to enable technology insertion independently at either the software or hardware layer. This paper presents the initial Space Telecommunications Radio System (STRS) Architecture for NASA missions to provide the desired software abstraction and flexibility while minimizing the resources necessary to support the architecture.

15. SUBJECT TERMS
Radio; Communication; Software; Architecture; Standards

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON STI Help Desk (email:help@sti.nasa.gov)
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	UU	14	19b. TELEPHONE NUMBER *(include area code)* 301-621-0390