JTRS INFRASTRUCTURE ARCHITECTURE AND STANDARDS

Donald R. Stephens (CommLargo, Inc.: St. Petersburg, FL, USA; don@commlargo.com);
Brian Salisbury (Space and Naval Warfare Systems Center: San Diego, CA, USA;
brian.salisbury@navy.mil); Kevin Richardson (The MITRE Corporation: McLean, VA, USA;
kwrich@mitre.org)

ABSTRACT

In March 2005, the USD (AT&L) appointed a Joint Program Executive Officer (JPEO) for JTRS to provide an overarching management structure across all Department of Defense (DoD) JTRS programs. The JPEO JTRS was given full directive authority for all waveform, radio, and common ancillary equipment development, performance and design specifications, standards for operation of the system, and JTRS systems engineering. The JTRS program objective is to develop and integrate a family of Software Defined Radios which maximize software and hardware commonality and reusability. To achieve this goal, the JPEO is facilitating the establishment of a standardized infrastructure for Joint Tactical Radio (JTR) sets.

1. INTRODUCTION

Earlier DoD radio programs such as Speakeasy [1], Global Mobile Information Systems (GloMo) [2], Digital Modular Radio (DMR) [3], and others have attempted to achieve interoperability and reusability by specifying Application Program Interface (API)s and radio infrastructures with varying degrees of success. Many of the APIs and resulting infrastructure were proprietary and not open to the entire DoD radio community.   In contrast to these previous programs, the JPEO JTRS has accumulated a code base of 20 APIs and over 3.5 million lines of code compliant to the Software Communications Architecture (SCA).   This software code base with Government Purpose Rights (GPR) represents a foundation for future JTRS radio development.

2. RATIONALE FOR THE INFRASTRUCTURE

Three primary goals of the JPEO JTRS are to provide the DoD with needed communications capabilities while reducing the time to field those capabilities, ensuring interoperability among all JTRS products, and maintaining overall system affordability. These goals are realized within the JTRS program through the incorporation of software reuse and portability strategies.

Software reuse and software portability are often misunderstood concepts because their definition is dependent on the context in which they are used. Throughout the code base within the JPEO JTRS Information Repository, the reuse of code across the JTRS enterprise provides an effective means of accomplishing each of the three stated objectives. During the development of software for the JTRS program, the ability to reference or reuse software from other JTRS developments, either in whole or part, avoids duplication of effort and assists in the propagation of sound coding practices. The distribution of software post-development also avoids duplication of effort and improves interoperability by providing functional, tested code as a basis for porting efforts rather than requiring the initiation of a new development activity. However, full interoperability is only established through rigorous testing to a documented standard.

The effectiveness of the software reuse within the JTRS program is governed by the ability to create highly portable software – that is, a software base that can be migrated easily to different processing environments.   Without building software with portability in mind, the benefits of software reuse at any level can be overwhelmed by the cost and effort required to leverage software written for specific environments or software that is poorly documented and thus only readily understood by the original developer.

Analogous to software reuse, there are many levels of and thus definitions of portability. Within the JTRS program, portability may be generalized to include two main focus areas. The first and lowest level of portability is generating code such that an implementation may be moved from one specific host or development environment (i.e. a specific set of compilers, linkers, libraries, OS, chipsets, etc.) to another with a minimal amount of changes to the original code. Achieving portability in this sense requires the use of standardized libraries, coding languages (and their compliant toolsets), as well as the standardization of critical interfaces and complete documentation.   These requirements are captured within the JTRS infrastructure by the definition of standardized APIs (including the SCA Application Environment Profile (AEP)) as well as by the JPEO JTRS participation in architecture and design reviews. These characteristics also form the basis of the JTRS Portability Assessment which is conducted on each acquired waveform.

Even if code is developed in this manner, the portability of the application is limited if each target platform has a different method for instantiating and controlling

applications within its domain. The rework required to adapt to each individual platform can quickly outweigh the benefits of using common software in the first place. To address this issue the JPEO requires each JTR Set and application to be compliant with the SCA, which defines a protocol for the installation, instantiation, and control of software artifacts within JTR Sets.

Table 1 captures lessons learned from the first JTRS waveform ported to a JTR set. While the SCA provides an excellent framework for General Purpose Processor (GPP) software, it does not specify the radio-domain specific interfaces for the waveform software to access or control radio hardware. The absence of JTR Set interfaces resulted in additional waveform modifications because the waveform developer and JTR Set developer did not have common interfaces.

Table 1 Lessons Learned from Initial Waveform Porting

Porting Issue	Corrective Action by JPEO for Future Waveforms
Absence of JTR Set APIs	JTRS Standards has approved 22 APIs and Specifications
Missing Files and Documents	JTRS Software Standard specifies required documents and directory structure
Missing Functionality	JTRS Software Standard specifies required delivery.
Absence of DSP and FPGA Specifications.	MHAL API standardized and additional extensions in development.
Waveform Connection Complexity	Developer education at the JPEO software conferences
Security Compliance	Developer education at the JPEO software conferences and participation by NSA

During the establishment of the JPEO's Information Repository, it was noted that a coding standard was necessary to specify the files required in a waveform delivery and the build structure such that independent organizations can reproduce the waveform's software build. In addition to the file and build directory specification, the JTRS Software Standard provides expected coding practices for both C++ and VHSIC Hardware Description Language (VHDL) code deliveries.

The SCA defines the base interfaces for software components deployed upon a Common Object Request Broker Architecture (CORBA) enabled processing environment, but does not specify communication or messaging for waveform components deployed upon most

Digital Signal Processors (DSPs) or Field Programmable Gate Arrays (FPGAs). The JPEO JTRS has standardized the Modem Hardware Abstraction Layer (MHAL) API to define a messaging protocol for components located on either DSPs or FPGAs.

The last two issues of Table 1 concern the education of the JTRS developer community in best practices for reducing the required processing resources for waveform implementation and observing good security practices in a distributed processing environment. JPEO JTRS has hosted a series of software conferences to address the education of the JTRS development community.

From analysis of the software code base and waveform porting, the JPEO determined that a radio infrastructure which provides a host environment for radio waveform and applications for all DoD JTRS sets was needed. The objectives of the infrastructure are to provide a consistent operating environment for waveforms independent of physical form factor or mission. The radio infrastructure is being defined for the first increment of JTRS. Future increments of the JTRS product line will incorporate infrastructure changes for advances in technology and missions.

3. SPECIFICATION OF THE INFRASTRUCTURE

JTRs are expected to provide the functionality of legacy single-function radios in addition to emerging transformational networking capabilities. By inference, a JTR set is thus expected to possess the low-latency and operator response of older hardware-intensive radios but also provide the flexibility and configurability of a computer network. In deployment, JTR sets vary in mission from large multi-channel radios to a single channel small form factor radio for smart ordnance. Such expectations require the overarching architecture framework to be scalable and extensible.

The JTRS infrastructure must support configurable, distributed processing where it is most applicable and also provide the low latency signal messaging necessary for the physical layer of high bandwidth waveforms. As expected, existing technology constrains the simultaneous fulfillment of flexibility/configurability and latency/ response.

The concept of an infrastructure is to define a host environment for the execution of waveforms and applications as shown in Figure 1. The GPP provides a known environment for waveform components. Developers are guaranteed a specific set of real time operating functions, distributed messaging through CORBA, and radio domain-specific interfaces such as GPS, Ethernet,

audio, etc. This promotes reuse and portability of waveform components. The hardware dependencies are isolated from the applications by the APIs defined for radio devices.
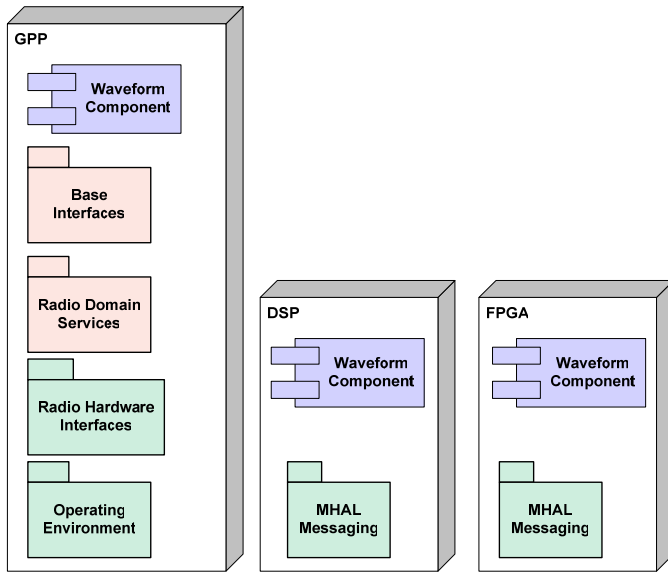

Figure 1 (U) Infrastructure Defines a Known Host Environment for all JTR Sets

Waveform components residing on a FPGA or DSP are provided a messaging environment that guarantees messaging between waveform components located on different processing elements.

The foundation for the JTRS family of radios is the SCA which is an architecture framework, specification, and guidance document for software defined radios. As part of the specification, the SCA defines a set of base interfaces known as the Core Framework, which isolate the system's applications from the underlying hardware. These interfaces require the collaboration of implementations provided by both the JTRS Set provider and application developers.

To support waveform portability and code reuse, the SCA specifies operating system APIs that must be provided by the JTR set's Real Time Operating System (RTOS). The AEP of the SCA specifies a subset of the Portable Operating System Interface (POSIX) Realtime and embedded application profile that every JTR Set must support and to which each application is limited. The JTR Set typically provides additional capabilities than those specified in the AEP for Set-specific functions, but by providing a consistent and minimally constrained set of operations; all applications can depend on their RTOS requirements being supported regardless of form factor or mission. This allows independent waveform developers to generate software that is guaranteed to execute upon the JTR set.

To further enhance portability, the SCA specifies the CORBA as a middleware to provide communication between application components as well as between the application and the hardware devices of the JTR Set. Connections are made between components across the ORB (Object Request Broker) using the SCA-defined interfaces. Once deployed and connected, any software component loaded onto a JTR Set is provided a common execution environment. This architecture is depicted in Figure 2.
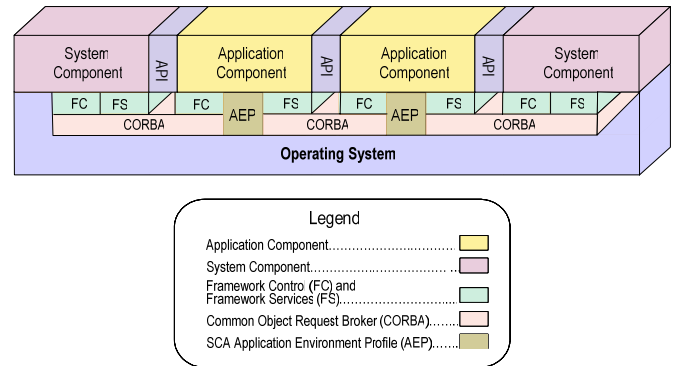

Figure 2 (U) SCA Architecture Layer Diagram

Within this architecture, a generic JTR Set is depicted by a variable collection of hardware elements, which include processing, input and output devices that need to be connected together to form communication pathways (channels) based on the requirements of the specific software loaded onto the system – and it is this software that will determine the RF output of that channel. The role of the SCA in this instance is to provide a common infrastructure for the lifecycle management of the software and hardware elements, a common method of ensuring that their requirements and capabilities are commensurate, and a common scheme for component interconnection.

4. STANDARDIZING THE INFRASTRUCTURE

The JPEO JTRS has over 3.5 million source lines of code in its software repository. This SCA-compliant software has been developed by the JTRS community for the different programs of record. To protect this software base, it was important that the APIs and standards developed for the infrastructure be compatible with the code already present in the repository. This represented a challenge for creation of enterprise-wide standards, because, although the code was written for an open architecture system, there were still implementation and mission-specific interfaces and behavior.

Figure 3 illustrates the process used to define the initial JTRS APIs. The initial drafts of the APIs were derived from the software and documentation in the JPEO software

repository. Implementation and mission-specific functionality was removed and the APIs were generalized for applicability across the JPEO enterprise. Three primary rules were observed in the generalization: 1) Backward compatibility to the software base was required, 2) Where necessary, base and extension APIs were generated to provide minimum capability for small form factor radios and extended capability for the larger JTR sets, 3) Only waveform-to-JTR set interfaces were defined – JTR set interfaces were allowed to be defined by the set provider.
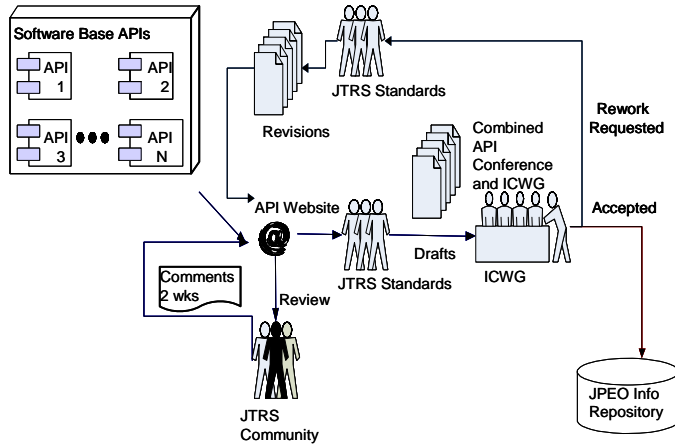

Figure 3 (U) Process for Initial Definition of JTRS APIs

After review and comment by the JTRS community, comments were adjudicated and draft APIs were submitted to the Interface Control Working Group (ICWG) for a decision. If accepted, the draft was converted into a final version and if not accepted, the API was scheduled for rework and resubmission.

It is recognized that missions and external requirements may conflict with the JTRS Standards. To allow arbitration of such circumstances, the JPEO ICWG has a waiver process as depicted in Figure 4. The ICWG is composed of decision makers who have visibility across the entire JTRS enterprise and can determine when resources should be applied to achieve a specific standard or API, or whether a new API/waiver should be granted.
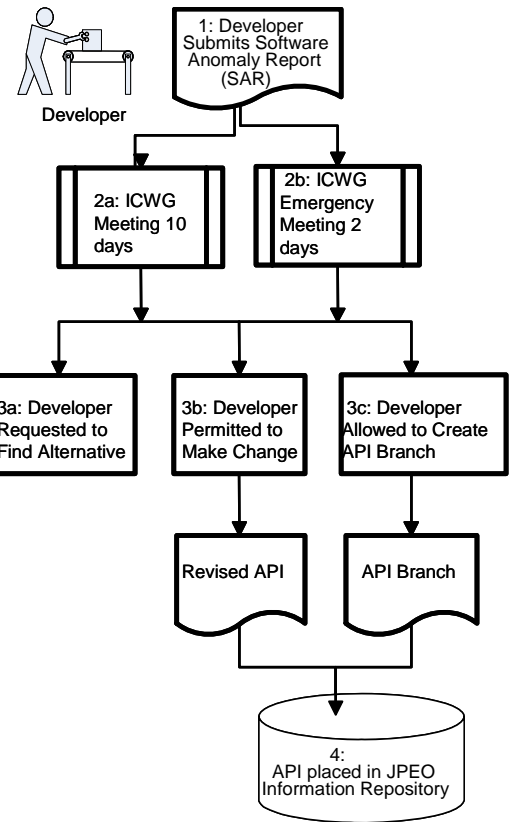

Figure 4 (U) Waiver Process for APIs and Standards

As the JTRS family of radios has developed, the users are requesting a new set of external interfaces for the JTR sets as shown in Figure 5. Common APIs for software loaders, network managers, etc, will promote additional software reuse within the JTRS family of radios and simultaneously reduce support equipment.
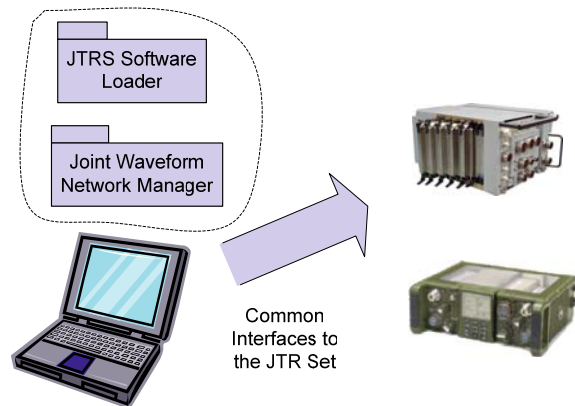

Figure 5 (U) External APIs for the JTRS Family of Radios

5. FUTURE INCREMENTS OF THE INFRASTRUCTURE

The JPEO JTRS has reaffirmed its commitment to the SCA and its role in the evolution of DoD software defined radio programs. For the JPEO strategy to remain relevant it will need to continue to evolve to meet the needs of future waveforms and acquisitions. In addition, it is imperative for the framework to take advantage of technological developments within industry as a whole and provide a mechanism for incorporating those innovations.

The SCA portion of the infrastructure will be restructured to allow it to be tailored to accommodate a wider range of deployment environments. The framework will provide opportunities for extension that will allow for the incorporation of platform technologies (e.g. CORBA) other than those that are currently identified within the infrastructure. As observed earlier, the current infrastructure is oriented towards a GPP environment. A future definition will characterize an approach which specifies a common approach for an "SCA model" on a wider set of operating environments. The approach must incorporate technologies that are complementary to the existing SCA model, yet reflect the capabilities of widely available technologies and meet the needs of the next generation of program acquisitions. Candidate approaches for such extensions include ORB utilization in non-GPP processing environments, the introduction of a FPGA/DSP container model or leveraging specifications targeted for system on chip design. Another area for evolution of the infrastructure is to expand its support for services within the scope of its component-based framework. Any modifications within this area will focus on better integrating the service concept within the overall management approach.

The JTRS APIs will also evolve to provide a wider array of platform interfaces. The initial API development approach provides a bridge between the initial phase of JTRS waveform development and the initial product deployments. This work must continue in order to provide an environment that facilitates application development that is consistent with the original design tenets of the JTRS program. However, once that set of interfaces has been identified and developed then the program will be able to specify common APIs to support JTRS platforms such as system management and control services.

6. SUMMARY

The JPEO JTRS contains a standards organization that has been chartered with the responsibility of creating architectural framework and infrastructure specifications that allow the JTRS program to facilitate the development of waveform, radio, and common ancillary equipment. The specifications that the group has developed and refined have been focused on meeting the program objectives. The initial development approach has focused on leveraging the body of work developed in the early portion of the JTRS program while establishing a flexible and extensible platform that will serve the current and future needs of system and application development.

10. REFERENCES

[1] P.G. Cook and W. Bonser, "Architectural Overview of the SPEAKeasy System", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, April 1999, pp. 650 - 661

[2] B.M. Leiner, R.J. Ruth and A.R. Sastry, "Goals and challenges of the DARPA GloMo program", *IEEE Personal Communications*, vol. 3, no. 6, December 1996, pp. 34 - 43

[3] B. Tarver, E. Christensen, A. Miller and E.R. Wing, "Digital modular radio (DMR) as a maritime/Fixed joint tactical radio system (JTRS)", *MILCOM 2001 - IEEE Military Communications Conference*, no. 1, October 2001, pp. 163 - 167