



Joint Program Executive Office Joint Tactical Radio System

SCA Next Rollout Overview



24-25 August 2010
JTRS SCA Working Group

JPEO JTRS

Distribution A- Approved for public release; distribution is unlimited (06 August 2010)



SCA Next





SCA Next Scope Overview

- **Focus is on the Software Communications Architecture (SCA) standard**
 - Baseline is SCA v2.2.2
 - JTRS Standard Application Program Interfaces (APIs) are not included at this time

SCA Next is Evolutionary not Revolutionary



SCA Next Process

- **Backwards Compatibility**

- Preserve existing investments
- When change will “break” existing implementation
 - Rationale must be compelling
 - Cost
 - Schedule
 - Performance





SCA Next Overview

- **Objective**

- Reduce development resources
 - Budget
 - Schedule
- Reduce test and certification time
 - Reduce number of requirements
 - Increase use of automated testing
- Improve performance
 - Reduce boot up latency
 - Reduce memory footprint
- Incorporate lessons learned

Internationally Developed Open Specification



SCA Next Overview

- **Expected Benefits**

- Reduce time to market for SCA compliant products
- Facilitate improved integration of systems engineering and software design and development practices
- Address recurring issues raised by the SCA product developers
- Support innovation and optimization in design and development of compliant products

Internationally Developed Open Specification



SCA Next Process

- **Establish clear objectives**
- **Establish a “dedicated” Team of volunteers**
 - Collect/evaluate/prioritize candidate enhancements
 - “Dedicated” volunteers lead task execution
 - Task lead coordinates resources to develop a recommended resolution
 - Team votes on proposed resolution

Backwards Compatibility is an overarching tenet



Change Request Prioritization

- **Approximately 60 changes submitted**
 - 23 selected for resolution
 - 2 Closed for redundancy with other changes
 - High return on investment
 - Strong volunteer support
 - Within JTRS Community
 - Within Wireless Innovation Forum (WInnF)
- **Further changes may be initiated**
 - Strong rationale
 - Strong volunteer support



SCA Next Change Request Categories

- **Migrate SCA to Platform Independent Specification**
 - Remove CORBA specific references
 - Develop technology specific mappings
 - Make domain profile platform neutral
- **Provide support for more compliance points**
 - Define SCA profiles
 - Introduce configurable capability within SCA constructs
- **Provide component/interface separation**
 - Define SCA component model
- **Improve integration of systems engineering and software design and development practices**
 - “Compliance by inheritance” validation
 - Support for automated testing
 - Integrate Naming Service and Domain Finder
- **Address recurring issues raised by SCA product development community**
 - Integrate changes into SCA specification
 - Decompose CF.idl into a collection of files
 - Remove unnecessary requirements
 - Define a static SCA profile
 - Define programming language specific guidance on libraries, exception handling, and runtime typing
 - Expand Application Environment Profile (AEP) to include Networking operations
 - Make Application Factory deployment and configuration more deterministic
 - Develop Appendix D (Domain Profile) requirements
 - Develop equivalent SCA extensions for Devices
 - Remove file operations
 - Formalize Set up and tear down semantics
 - Allow nested applications to be connection endpoints
 - Reorganize SCA so that development responsibilities are more self evident (CF , WF, Device and Service)
 - Bootup performance
 - Ambiguity in requirements



Active SCA Next Change Requests

Task	Summary	Status
SCA Requirements Revisions (including UUID Removal)	Evaluate requirements with respect to waveform portability and internal to SDR Set infrastructure	To be presented
Enhanced Automated Testability	Evaluate SCA requirements to identify requirements that must be tested manually. For those requirements assess whether requirement can be modified to enable automated testing and preserve the intent of the requirement.	To be presented
Deployment Optimization	Investigate enhancing SCA deployment mechanisms to reduce application deployment time (e.g. looking at port connections).	To be presented
Lightweight Components	Explore approaches to improve flexibility of architecture i.e. accommodate various platforms requirements (mobile versus static, single channel versus multiple channels, single waveform versus multiple waveforms, small form factor, etc.) instead of one size fits all architecture	To be presented



Active SCA Next Change Requests

Task	Summary	Status
CORBA Neutral Representation	Develop a SCA Platform Independent Model (PIM); i.e. technology implementation agnostic. Will still retain a mapping of the SCA to a CORBA Platform Specific Model (PSM)	To be presented
CORBA Evolution	Migrate from minimumCORBA to CORBA/e to enable use of Real-time embedded CORBA features; update CORBA Service references to enable use of full or lightweight versions	WInnF Lead To be presented
Architecture Consistency - Factory Model	Explore approaches to improve architectural consistency in requirements and required behavior among SCA components in the SCA specification without requiring major modifications to application code. Focus is on Core Framework (CF) and domain profiles	WInnF Lead To be presented
Application Enhancements - Nested Applications	Revise existing Software Assembly Descriptor (SAD) and SCA application specification to facilitate applications that contain sub-applications	To be presented



Active SCA Next Change Requests

Task	Summary	Status
Application Enhancements - Inter-Application connectivity	Review requirements for component configuration and initialization during application creation. Application factory configures (only) the assembly controller descriptor "configure" properties with values and not all application components.	To be presented
Recommended C++ Features for SCA implementations	Define a set of language feature guidelines for C and C++ based upon best industry practices for embedded real-time software systems that are size, weight, and power constrained. These guidelines will also address application portability as well as performance	To be presented
Interface Definition Language (IDL) Refactorization	Reduce memory footprint of executables by decomposing SCA v2.2.2 CF.idl into a set of IDL files enabling SCA components only to include interfaces that are needed.	To be presented
Lightweight (LW) AEP	Determine whether a LW AEP should be specified for use on non GPP processors (e.g. DSP). Topic is also looking at AEP on GPP to determine whether AEP should be expanded to incorporate socket oriented operations.	To be presented



Active SCA Next Change Requests

Task	Summary	Status
Service Deployment and Initialization	Clarify service deployment for initialization and configuration when Lifecycle and PropertySet are supported along with connection behavior	To be presented
Component Model Introduction	Provide a representation that captures the runtime characteristics of the elements that are instantiated within an SCA compliant design. Provides a reference platform, upon which modeling tools can build, to support design of SCA compliant products.	To be presented
Development of SCA User's Guide	Make development responsibilities more self evident (CF, Waveform/Application, Device and Service Developers)	To be presented



Active SCA Next Change Requests

Task	Summary	Status
Integrate Changes into SCA specification	Integrate SCA Next changes into the main body of the SCA specification.	Volunteers welcomed
Domain Profiles	Revise the Domain Profile to include looking at alternatives to XML for domain profile such as .ini files	Input welcomed
Asynchronous Messaging	Improving performance (encompasses one-ways, Asynchronous Messaging Interface (AMI), etc). How CORBA one-ways or C++ calls without returns should be supported (error handling; flow control)	WInnF Lead Project initiation



SCA Next Way Forward

- **Publish Draft Specification late 2010**
- **Encourage SCA Product Developers to prototype changes**
 - Provide Feedback

SCA Next



Internationally Developed Open Specification