Raytheon

Customer Success Is Our Mission

AIR
LAND
SEA
SPACE
CYBER

SCA 4.1 Perspective and Product Transition

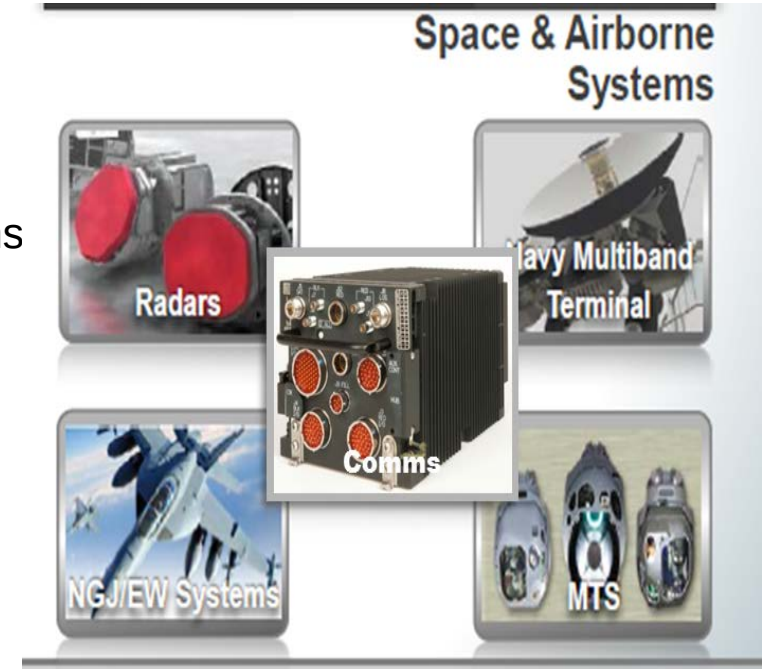
Jerry Bickle
Jimmie Marks
March 26, 2015

Agenda

- Raytheon Software Communication Architecture (SCA) Usage
- SCA 4.1 Development Approach
  - Scalable 2.2.2 Device Manager
  - Scalable 2.2.2 Domain Manager
  - SCA 4.1 Device Manager Transition
  - SCA 4.1 Domain Manager Transition

Raytheon SCA Usage

- The SCA specifies an industry standard Software Architecture to facilitate reuse and portability between products
  - Customer requirement for various Raytheon programs
  - Internal Raytheon company requirement
- Raytheon utilizes the SCA in our Communications and Electronic Warfare (EW) business areas
  - Multiple product lines
  - Multiple products



- Increasing Internal emphasis on:
  - Common architecture frameworks within and across product lines
  - Architecture and Technology Neutral Implementations to support reuse and evolution
  - Software Bus beyond GPP by communication middleware
  - DSP software and firmware portability

Raytheon SCA Usage

- SCA Inherently provides:
  - Establish DOD, Industry and International Architecture Standard
  - Provides architecture framework for Reuse, Portability, Modularity, Interoperability and Testability
  - Allows for technology neutral implementations across different products
  - Allows for Multi-Function capabilities within a single product
    - Communications Waveforms, EW Techniques, Identification, Friend or Foe (IFF), and other applications
  - Capability for Product Line Reuse and portable Software components
    - Reusable / Portable Comms Waveforms and Services
    - Reusable / Portable EW Techniques and Services
      - Technique Libraries (i.e. Jamming Algorithms) and distributed processing (e.g. Multi-pod communications) Developed EW "Techniques" e.g. JAM, Search, Reactive – JAM, DF algorithms
    - Generic Multi-channel System and Resource Management

- Multiple Raytheon SCA Products across EW, SATCOM, and Airborne product areas

SCA 4.1 Development Approach

- Started with Raytheon SCA certified 2.2.2 Core Framework baseline
- Re-factored SCA 2.2.2 implementation into a Scalable SCA 2.2.2 that mimics SCA 4.0 OE Profiles
  - Operating Environment Medium and Full Profiles along with components Units of Functionality (UoF)
  - CF 2.2.2 IDL refactored to mimic SCA 4.x IDL files.
    - Application Base Interfaces
    - Device Base Interfaces
    - Device Manager Interfaces
    - Domain Manager Interfaces
- Development OE tooling
  - OS: GHS Integrity, VxWorks, Linux
  - ORBs: ACE/TAO, PrismTech SPECTRA ORB, OIS ORBexpress
  - Processors: ARM

Raytheon

Scalable 2.2.2 Device Manager

- Utilized UNCHECKED_NARROW for
  - LifeCycle
  - PropertySet
  - DomainManager
  - ExecutableInterface
  - LoadableInterface
  - AggregateDeviceAttributes
- Optional UOFs Supported: Log Producer and Log Capable
- Supported UOFs:
  - CORBA Provider
- Deployed Devices and Services with various UoFs
  - Testable, Configurable, Controllable, Connectable
  - Aggregatable, Allocatable

SCA 4.1 Device Manager

- Reuse code from Scalable 2.2.2 Device Manager
- Minor Impacts
  - Register and unregister operations
  - Registered Components attribute
  - Registration to Domain Manager
  - Base Interface Name Changes
- DCD Deployment logic the same except for the addition of ComponentFactory
  - PlatformComponentFactoryDeployment UOF

SCA 4.1 Device Manager Transition

- Device Manager needs to have the requirement to ensure platform allocation properties are set before registering to domain manager.
- Clarification of Component Late Registration to Domain Manager
- How to know if a Service Component is suppose to register? (SCA type supporting interface only like LogService versus non-SCA type)
- Allow Device Manager to give deployment information out for its deployed components like Application Manager component does

SCA 4.1 Device Manager Transition, cont'd

- Static Link Image
  - Full Device Manager OE Profile with UOFs: CORBA Provider, Log Producer, Configurable, Connectable, Interrogable, Management Releasable, Platform Component Factory Deployment
    - 19% Size Savings in comparison to SCA 2.2.2
    - Without Interrogable – Push registration only
      - 22% Size Savings in comparison to SCA 2.2.2
  - Medium Device Manager OE Profile with UOFs: CORBA Provider, Log Producer, Configurable, Connectable, Platform Component Factory Deployment
    - 26% Size Savings in comparison to SCA 2.2.2

Scalable 2.2.2 Domain Manager

- Utilized UNCHECKED_NARROW for
  - LifeCycle
  - Port
  - PortSupplier
  - PropertySet
  - ExecutableInterface
  - LoadableInterface
- Optional UoFs Supported:
  - Application Installable, App Deployment Data, App Releasable, Event Channel, Log Producer, CORBA Provider, Event Channel, Event Provider
- Deployed Application component with various UoFs
  - Testable, Configurable, Controllable, Connectable
- Full and Med Device Managers registered to Domain Manager

Raytheon

SCA 4.1 Domain Manager Transition

- Reuse code from Scalable 2.2.2 Domain Manager
- Minor Impacts
  - Register and unregister operations
  - Domain Manager Interface attributes and interface return types
  - Base Interface Name Changes
- SAD Deployment logic the same except for
  - ResourceFactory versus ComponentFactory
  - Naming Services versus ComponentRegistry
  - Connections

SCA 4.1 Domain Manager Transition

- DomainManagementObjectAddedEventType
  - Give out a ComponentType
- Application Deployment Types
  - Remove and replace with ComponentType
- ApplicationDeploymentAttributes
  - Replace types and readonly attributes with Deployment Information that can be given out as part of each registered component.
  - Inherited by DeviceManagerAttributes
- DomainManager
  - Change return type for applications and applicationFactories to be CF::Components.
- Late Device Manager Registration requirements for Domain Manager needed

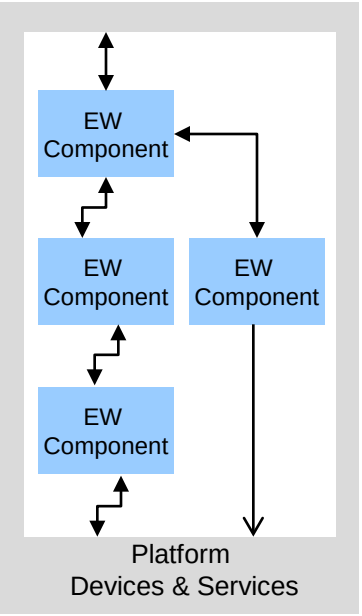
SCA 4.1 Domain Manger Transition

- Similar size reductions as Device Manager Component in regards to medium and full OE profiles with UOFs: CORBA Provider, Log Producer, Installable, Event Channel, Configurable, Connectable, and Interrogable

Summary

- The SCA 4.1 specifies a scalable technology neutral architecture
- Straight forward development methodology for 2.2.2 transition to 4.1
  - Reduction in memory footprint size, increase performance

- SCA 4.1 Features
  - Architecture Profiles (Scalability)
    - Operating Environment Profiles
    - Components Profiles
  - Standardized Scalable Platform Manager Types and Interfaces
    - Device and Domain Manager components
  - Standardized Scalable Component Types and Interfaces
    - Application components
    - Platform Components
  - Continued leverage of commercial standards to reduce development cost
    - UML, IDL, XML, Communication Middleware, POSIX
  - Standards for Operating System Interaction for embedded systems
    - GPP and DSP POSIX profiles (Lightweight)
  - Compatibility with v2.2.2 Applications



Raytheon

Backup

Backup

Title: SCA4.1 Perspective and Product Transition

Submitted by:
Gerald Bickle, Raytheon Space and Airborne Systems
Jimmie Marks, Raytheon Space and Airborne Systems

The Software Communication Architecture (SCA) facilitates a strategy for the acquisition of affordable software systems that promotes innovation and rapid integration of portable capabilities across global defense programs. The SCA 4.1 specification is an update to the SCA standards to address international government & industry specification coordination, while maintaining backward compatibility with existing products produced to earlier versions (i.e. SCA 2.2.2). This presentation describes Raytheon's view of SCA 4.1 and transition approaches for SCA 4.1 migration. Raytheon's views and transition approaches reflect Raytheon's usage of the SCA in multiple business areas and products such as tactical radios and Electronic Warfare systems.