Common Tactical Controller

Presentation to NDIA Robotics Division
Bill Thomasmeyer
August 25, 2015
Current Paradigm

One proprietary, operator control station per RAS platform

Historically Acceptable

- Minimized program risk
- Operationally not an issue with a single RAS fielded per military unit
The Problem

One proprietary, operator control station per RAS platform

Historically Acceptable

Increasingly Problematic with the Proliferation of Heterogeneous Systems

- Greater Physical Burden on Warfighter
- Growing Logistic Complexity (e.g. incompatible batteries, radios)
- Unnecessary Program Costs for Redundant Equipment
OSD/JGRE Assigned Task

Phase I Effort

- Investigate and report on current approaches and efforts to develop a common controller for Group 1 UAS and ground RAS

- Determine a potential path forward that leverages current investments
Common OCS Hardware Platform

Tactical Robot Controller (TRC)

Addresses the Weight, Cost, and Logistic Issues

- SWaP still an issue
- Radio comms still an issue
Common OCS Hardware Platform

**Tactical Robot Controller (TRC)**

Addresses the Weight, Cost, and Logistic Issues

Running Silos of Proprietary Control Software

- Cognitive Burden of Different User Interfaces
- Operational Limitations of Not Being Able To:
  - Effect Collaborative Behaviors
  - Share Data
  - Request Services
  - Dynamically separate / distribute control of the platform and payloads
Government-Owned OCS Software

Modular Operator Control Unit (MOCU)

SPAWAR R&D Controller Software
Being Adapted & Customized for Programs of Record

- Addresses being locked into a proprietary vendor
- Enables Commonality and Interoperability *within* a PoR
Government-Owned OCS Software

Modular Operator Control Unit (MOCU)

SPAWAR R&D Controller Software
Being Adapted & Customized for Programs of Record

Lacks a True Open Architecture

- Doesn’t Allow for Interoperability outside a PoR
- Difficult to Integrate Proprietary Software: limits competition and stifles innovation
- Increasingly Difficult over Time to Enhance, Extend, and Support
  - Add-on capabilities need to be customized for each variant
  - Entire software needs to be re-tested with each release
Open OCS Software Architecture

**UAS Control Segment (UCS)**

- Developed for Group 2 – 5 UAS
- Significant DoD Investment and Growing Interest
- Enables plug-n-play, component-level software compatibility
  - E.G. seamless integration of Government-owned and proprietary components
- Key Characteristics
  - Data-Centric, Service-Oriented, and Model-Driven
- Reduces software development costs
  - Enables writing software one time
  - Unbundles testing and debugging
- Enables true, services and data interoperability locally …
Open OCS Software Architecture

**UAS Control Segment (UCS)**

- Developed for Group 2 – 5 UAS
- Significant DoD Investment and Growing Interest
- Enables plug-n-play, component-level software compatibility
  - E.G. seamless integration of Government-owned and proprietary components
- **Key Characteristics**
  - Data-Centric, Service-Oriented, and Model-Driven
- Reduces software development costs
  - Enables writing software one time
  - Unbundles testing and debugging
- Enables true, services and data interoperability locally as well as across the network
UCS Open Architecture
Conceptual Overview

- Master Library controlled by the standards committee
- Contains a master list of defined platform independent services
- Separated into domains and sub-domains

Information
Architecture
Standards Body (SAE)
UCS Open Architecture

Conceptual Overview

- Master Library controlled by the standards committee
- Contains a master list of defined platform independent services
- Separated into domains and sub-domains

- “Master” repositories can be established to contain descriptions, assumptions, supported services, and other characteristics for a specific system
- One or more other, “subordinate” repositories contain representations (description, attributes, metadata, source code, and/or object code) of software components, including UIs, available to be integrated for a specific system implementation
UCS Open Architecture

Conceptual Overview

- Master Library controlled by the standards committee
- Contains a master list of defined platform independent services
- Separated into domains and sub-domains

- “Master” repositories can be established to contain descriptions, assumptions, supported services, and other characteristics for a specific system
- One or more other, “subordinate” repositories contain representations (description, attributes, metadata, source code, and/or object code) of software components, including UIs, available to be integrated for a specific system implementation

- User Interfaces and software components are selected from the repositories and integrated to create the control software that runs on a hardware platform
OSD/JGRE Assigned Task

Path Forward that Leverages Current Investments

Open Architecture

Information Architecture
UCS Architecture

Product Line Architecture
Repository
Repository
Repository

Deployment Architecture

Common UI
Component
Component

Software
Government Owned
MOCU
WMI
Proprietary

Hardware
UCS

• Transitioning to SAE
• Agreement with the Open Group to align with the FACE standard
• Information architecture being extended to the ground domain
• Software development tools are starting to become available

MOCU 4

• UCS compatible system for developing and maintaining a graphical user interface, plug-n-play modular components that connect over a DDS communications bus, and vehicle/device interfaces
• Runs on Windows, Ubuntu, and Red Hat platforms
Pursue an Open Architecture and Open Business Model based on UCS that leverages Government-owned software (e.g. MOCU 4) in order to meet upcoming program of record (e.g. CRS-I) needs in the near term; and serve as the basis for the proposed common controller in the long term.
OSD/JGRE Assigned Task

Phase II Effort

- Track UCS transition to SAE and the development and adoption of UCS Ground Domain services
- Work with SPAWAR to vet MOCU V4.0 and whether/how the software could be transitioned into a UCS repository
- Develop the framework for an open business model
- Work with the Government to determine whether/how the proposed strategy might align with the CRS-I PoR and identify milestones that would need to be met
- Vetting proposed approach with PM-SUAS to socialize future migration of TOGA controller software to common controller architecture
- Develop a detailed, path forward implementation plan & proposal
- Identify and resolve other details that arise
**Path Forward**

*Potential Multi Stage Approach*

**Stage 1** – CRS-I PoR defines a UCS PLA for the CRS-I controller, compliance with which would be a CRS-I KPP or requirement

**Stage 2** – if/when a common controller PoR is established, the CRS-I PLA can be seamlessly transitioned to be the initial PLA for the common tactical controller

**Stage 3** – the common tactical controller PLA can be extended and/or new common controller PLAs defined to support:

- current PoR platforms (e.g. MTRS Increment 2, TOGA, AEODRS, etc.), enabling their controller software to be later upgraded for compliance
- new PoRs whose platforms and payloads will be developed from the get-go to be compliant with the common controller PLAs
Stage 0 – NAMC, in partnership with the Government, develops and supports a prototype PLA for a tactical handheld controller to operate MT/1* platforms and payloads

Stage 1 – CRS-I PoR defines a PLA for the CRS-I controller, compliance with which would be a CRS-I KPP, requirement, or objective

Stage 2 – if/when the common controller PoR is established, the CRS-I PLA can be seamlessly transitioned to become the initial PLA for the common tactical controller

Stage 3 – the common tactical controller PLA can be extended and/or new common controller PLAs defined to support:
   • current PoR platforms (e.g. MTRS Increment 2, TOGA, AEODRS, etc.), enabling their controller software to be later upgraded for compliance
   • new PoRs whose platforms and payloads will be developed from the get-go to be compliant with the common controller PLAs

* Man-transportable, ground RAS, platforms and payloads compliant with the open, MT/1 standardized baseline instantiation developed and supported by NAMC in partnership with the Government
Prototype Tactical Controller PLA

“Strawman” SOW

- Define and establish a master repository specifying the system description, operating system and middleware, and other system level requirements, characteristics, etc.
- Use a core version of MOCU 4 to implement a Government-owned software repository
- Develop procedures for registering and describing available proprietary software components in a repository
- Make GFE TRCs available to software system integrators for prototype development and demonstration purposes
- Develop a Web portal & provide baseline support functions
Prototype Tactical Controller PLA

Benefits to Government & Industry

• Informs Government decision whether/how to specify UCS compatibility for the CRS-I PoR controller
• Provides a baseline for defining the CRS-I PoR PLA
• Provides an interim standard to develop and test against
• Facilitates innovation and partnering
• Grows the industry base
• Increases competition
• Lowers risk and cost
Questions / Discussion