Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 27-31, 2003.

Robotic Architecture Standards Framework in the Defense Domain with Illustrations Using the NIST 4D/RCS Reference Architecture

Hui-Min Huang, James Albus
National Institute of Standards and Technology
Gaithersburg, MD 20899
{hui-min.huang, james.albus}@nist.gov

Jeffrey Kotora
Chair, JAUS Working Group,
Titan Systems, Huntsville, AL, 39824
jkotora@titan.com

Roger Liu
Army Systems Engineering Office (ASEO)
Fort Monmouth, NJ 07703
Roger.Liu@us.army.mil

Abstract

The U.S. Department of Defense (DoD) requires common architectures that enable the collection, processing, analysis, seamless integration, dissemination, and reuse of information and technology in order to achieve its objectives of interoperability. We attempt to provide a high-level overview of some of the key DoD architectural standards and frameworks and illustrate how the NIST 4D/RCS relates to them.

1. Introduction

The combination of taxpayers' demand of high return on their dollars, continued realignment of national budget priorities, and the evolution of the national defense strategy has resulted in the Department of Defense's recognition of and increased reliance on information, technology, interoperability, and joint operations to provide the decisive edge in combat. The U.S. Department of Defense's (DoD) Joint Technical Architecture [1] reflected this view with the following statement: "The nature of modern warfare demands that we fight as a joint team… Full Spectrum Dominance requires Information Superiority… Interoperability is crucial to Information Superiority."

These revealed issues call for common architectures that enable the collection, processing, analysis, seamless integration, dissemination, and reuse of information and technology. While the DoD has a wide spectrum of computing problems that may require vastly different architectural approaches, this paper focuses on the robotic systems architectures. This focus is becoming extremely important as the U.S. Army is well underway with its Future Combat System (FCS) program [2] to transform itself to the next generation objective forces.

At the center of the DoD's overall architectural framework is the Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) Architecture Framework [3]. From there, many architectural approaches have been and are being created at various levels in the DoD hierarchy to support their particular sub-domains. These architectures include the DoD Joint Technical Architecture (JTA) [1], DoD Global Information Grid [4], DoD Technical Reference Model (TRM) [5], Joint Architecture for Unmanned Systems (JAUS) [6], Joint Technical Architecture-Army (JTA-A) [7], the Army Weapon System Technical Architecture Working Group (WSTAWG) [8], U.S. Army Tank-automotive & Armaments Command (TACOM) Vehicle Electronics (Vetronics) Reference Architecture (VRA) [9], and Navy Open Architecture [10].

4D/RCS (Real-time Control System) [11] is a reference model architecture originated at the National Institute of Standards and Technology (NIST). 4D/RCS is a hierarchical control structure. A set of governing rules applies universal controllers to each control level. Controllers are assigned specific capability based on the mission and activity analysis and modeling. A standard interfacing mechanism is used across the architecture. The reference model nature makes 4D/RCS applicable to many defense problems. 4D/RCS also applies a task decomposition process that facilitates both the configuration and the execution of an operational architecture.

We attempt to provide a high-level overview of some of the key DoD architectural standards and frameworks and illustrate how 4D/RCS relates to them.

2. DoD Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) Architectural Framework

The Framework aims at providing a standard and coordinated architectural approach for various DoD agencies and the military Services to develop their particular combat and support architectures, including unmanned systems and other kinds of robotic systems. It is based on the Levels of Information Systems Interoperability (LISI) reference model [12]. C4ISR's objectives are for these individual architectures to be

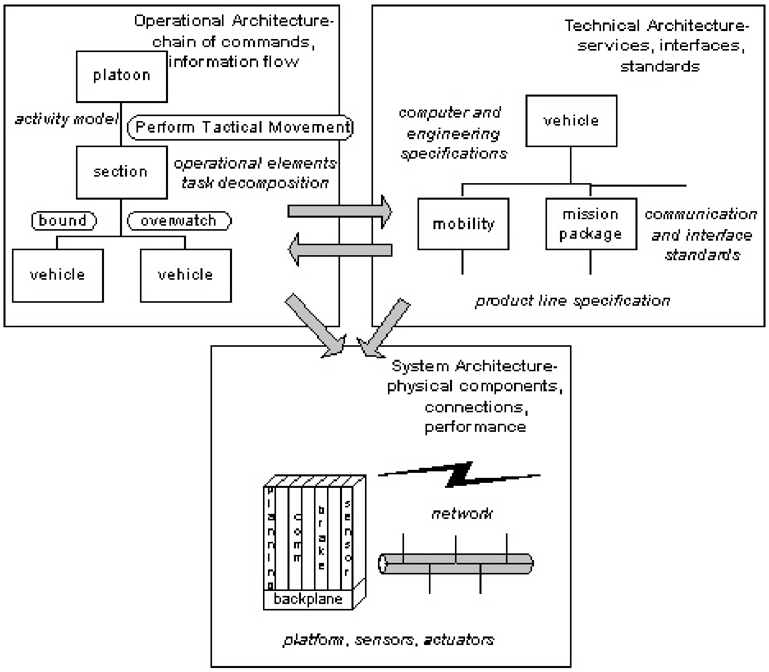

Figure 1: A C4ISR-consistent 4D/RCS unmanned vehicle system

easily integrated to enable joint operations, for the architectures to be consistently presented, and for the architectural components to be reusable.

2.1 Types of architecture and their integration

C4ISR defines the following three types of architectures:

- An Operational Architecture (OA) should describe the tasks, operational units, and information flows required to accomplish a military mission.
- A Technical Architecture (TA) should contain a set of rules governing the organization, interaction, and interdependence of the system components. This is to facilitates interoperability when the system's or system-of-system's components conform to the specification. TA specifies conceptual paradigms of the processing, database, and communications. TA also specifies standards and data dictionary.
- A Systems Architecture (SA) should describe physical system components and interconnections that integrate for particular military missions. The systems architecture is constructed to satisfy operational architecture requirements per standards defined in the technical architecture.

C4ISR prescribes that common terms and approaches for architectural development facilitate integration and interoperation. Different types of integration can occur within and among the three types of architectures.

2.2 Architectural information to be produced

C4ISR specifies that a DoD architecture should contain standard information including the architectural type and its associated list of output, purposes of the architecture, maturity level, context, and findings. These information elements are assigned unique identification numbers in C4ISR.

2.3 An illustrated mapping for a 4D/RCS based unmanned system

Figure 1 shows, in a simplified view, how an unmanned vehicle system can be constructed based on the 4D/RCS architecture and be consistent with C4ISR. The OA defines the control, command, and task structures per mission requirements. The activity shown in this example is for a Platoon control node to command its subordinate Section node to conduct a Perform Tactical Movement task. The Section node will plan for the task and result in a plan consisting of a series of Bound and Overwatch movements for its subordinate Vehicles. The OA may dictate, among others, the interface and communication standards requirements, shown in the TA. New technologies as specified in TA may, in the reverse direction, enhance an OA specification. A system is realized through integrating the software implementation of the OA and the standards and specification as set up in the TA to the physical computing and communication equipment, shown as the SA.

Conformance to these architectures facilitates interoperability. 4D/RCS could provide reference model for system control.

3. DoD Technical Reference Model (TRM)

TRM describes a common conceptual view for the DoD computing architectures. TRM also defines a set of associated, common vocabulary to facilitate mutual understanding among the involved organizations during the coordinated acquisition, development, and operation of their systems, subsystems, or components. TRM is based on the Society of Automotive Engineers (SAE) Generic Open Architecture (GOA) framework [13] and contains certain extensions. Unmanned/robotic systems certainly can benefit from TRM.

TRM identifies a service view that describes multiple computing layers, namely, application entities, application platform entities, and external environment entities, top down. The service view is further

elaborated to form the four-layered interface view, as illustrated in Figure 2. The application layer is further subdivided into user applications and support

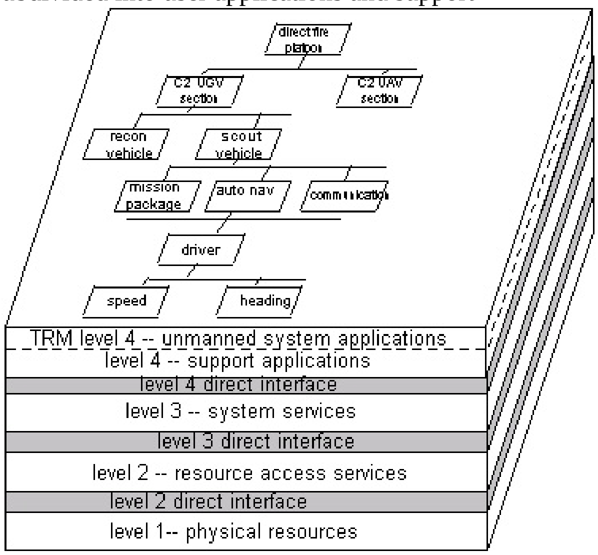

Figure 2: TRM and 4D/RCS based unmanned system

applications. The latter may include functions such as graphic display and communication. These layer and interface names and numbers are parts of the standard vocabulary.

Figure 2 also shows how a 4D/RCS based unmanned system can be designed in the TRM framework. The C4ISR operational architecture resides at layer 4 but extends through level 1, demonstrating how an OA is integrated to its SA.

TRM identifies three types of interfaces. The direct interface goes between the two consecutive computing layers. The 4D/RCS functions of timing handling and sensing and actuation would fall into this category. The internal direct interface goes between the sub computing layer within a computing layer. The 4D/RCS Human-Computer Interface (HCI) displays fall within this category. The logical interface goes among components, subsystems, or systems at the same computing layer. The 4D/RCS command, status, and query messages are of this type.

TRM mapped out a conceptual framework to help understand various DoD robotic and unmanned subsystems and to help integrate individual architectural efforts, and thus, facilitate interoperability. 4D/RCS could serve as a reference architecture operating within the framework.

4. JTA/JTA-A

The Joint Technical Architecture - Army (JTA-Army) is the Army's implementation of the DoD JTA. They, as the names indicate, belong to one of the three architectural types that C4ISR specifies. The architectures state that their first objective is to provide the foundation for seamless flow of information and interoperability among all tactical, strategic, and sustainment/combat support systems that produce, use, or exchange information electronically. The second objective is to mandate standards and guidelines for system development and acquisition that will dramatically reduce cost, development time, and fielding time for improved systems.

JTA/JTA-A can be considered as having two parts. The core specifies standards for general DoD computing requirements, applicable but not specific to unmanned systems. A series of appendices define standards for particular Army system domains, including the weapon systems domain that covers robotic unmanned systems. The latter will be described in Section 5, WSTAWG.

JTA/JTA-A seeks applicable, proven, industrial standards for the areas where standards are needed. Depending on factors including the maturity and applicability level of the standards, they may be designated as either mandates or emerging standards. The latter are potential mandates after the evolution of the standards for an identified period of time. The core of JTA/JTA-A includes standards specifications for the following key areas:

- Programming Languages, User Interface, Document Exchange, Computer Graphics, Operating Systems, and Communication and network. Standards include ISO specified C, C++, and Ada languages, CDE/X windows and Win32, and POSIX and Win32.
- Information and Object Modeling. Standards include IDEF0 for activity modeling, IDEF1X for data modeling, and UML object modeling.
- Data Definitions and Data Exchange. Standards include the Defense Data Dictionary System (DDDS), Joint Variable Message Format (JVMF), and XML.
- Computing and Information Security. Standards include Passwords, network authentication services, Cryptographic Algorithms**,** Secure Sockets Layer (SSL) Protocol for WWW.
- HCI. DoD HCI Style Guide.

5. WSTAWG

WSTAWG operates by identifying standards needs in particular embedded systems technical areas before forming Integrated Product Teams (IPTs) to develop the standards. The main IPTs include the Weapon System Common Operating Environment (WSCOE),

Unmanned Vehicle Architecture (UVA), Operating Environment (OE), and mapping service.

WSCOE is identified as a reuse component architecture, seeking to leverage and incorporate various Army weapon and robotic systems and engineering technology. Investigations of the current major architectures and technologies, including 4D/RCS, are being conducted to extract best practices and form the basis for WSCOE. Its scope covers all the levels in TRM and all the architectural types in C4ISR.

The UVA IPT uses an approach of scoping the unmanned vehicle system types, their interface types, and the mission types to set up a foundation for developing required technical standards. It has so far identified over 30 mission types that unmanned vehicles may carry out, including Terrain Control/Denial, Forward Observation, Targeting, and Reconnaissance. 4D/RCS has been identified as a standards candidate.

The OE IPT covers the TRM levels 3 and 2, the System Services and Resource Access Services layers. The IPT has produced an API specification for robotic application systems to interface with real-time operating systems, thus making the application platform independent.

The mapping service IPT has developed an API specification for interfacing various application software to the map display subsystems employed by unmanned robotic systems.

6. JAUS

The Joint Architecture for Unmanned Systems (JAUS) Working Group was chartered by the Office of the Undersecretary of Defense. JAUS specifies an upper level, component based, message passing architecture. NIST is one of the charter members of the working group and has been participating in its development efforts since.

JAUS aims at a set of generic specifications that will be common to all the unique implementation technology, computer hardware, vehicle platforms, and missions.

JAUS contains two technical specifications, namely, domain model (DM) and reference architecture (RA) and three managing documents, namely, strategic plan, document control plan, and standard operating procedure. This paper focuses on the two technical specifications.

6.1 Domain Model

The JAUS DM describes the operational requirements, both known and potential, from a final user's viewpoint. DM, therefore, serves as a repository to capture users' needs. It is written in a user's language. DM also specifies the requirements for developing the RA. DM provides a framework for system acquisition and R&D.

DM identifies and describes, in details, the following architectural elements:

- Functional agents, including commander, communicator, driver, planner, and technician.
- Knowledge stores, including gauge panel, map, library, and log.

The JAUS DM fits into the OA type of C4ISR.

6.2 Reference Architecture

The JAUS RA is the performance specification for implementation of the user requirements stated in the DM. The RA is written in scientist's or engineer's languages. Each message and service defined in the RA can be traced to a DM requirement. The RA will only implement those requirements in the DM that have gone through a technical evaluation process.

6.2.1 JAUS components

6.2.1.1 Groups

The RA specifies the following groups of components: command and control, communications, platform, manipulator, and environmental sensor. However, due to resource constraints, the main focus has been on the platform group up to the current version of JAUS RA, version 3.0. The platform group provides mobility functions, including sensing. In addition, essential components in the command and control and communication groups have been specified to meet users' implementation needs.

Topologically, components form nodes. A node can be regarded as a blackbox that contains all the necessary hardware and software to provide a complete service. Nodes form subsystems. Subsystems are independent and distinct units. Subsystems form systems.

Functionally, we observe that the components are grouped in terms of their commanding authority. Figure 3 depicts such a hierarchical notion for the platform component group. The main body of the diagram contains the platform group of components. At the bottom is the primitive driver component that outputs the commanded wrench, i.e., mobility actuator command at six degrees of freedom in a percentage scale, to the actuators. The input to the primitive driver can come from either the reflexive driver or any other drivers as listed. The rationale for the reflexive driver is to provide robotic control system developers with an additional but optional low level reflexive capability to

prevent the vehicle from unstable or unsafe states. The other listed drivers allow different types of robotic control input, ranging from local waypoints to global vectors, to either of the lowest level drivers.

Note that only the primitive driver is open-loop. All the other drivers can receive and respond to sensing and world model information. As shown in the figure,

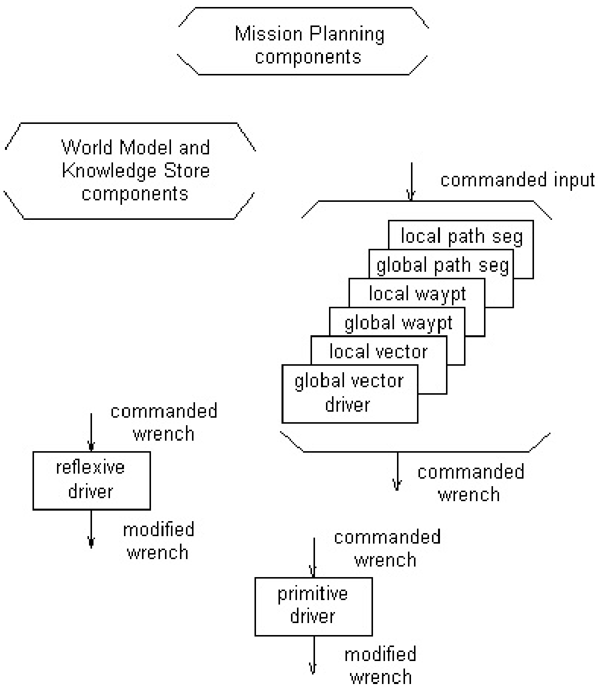

Figure 3: platform component group

the world modeling and knowledge store and the mission planning subgroups are being specified within the JAUS working group. The mission planning components provide input, the mission plans, for the drivers.

6.2.1.2 Structure

A component is specified through its name, ID, function, common behavior, and core and specific messages. The components have a common behavior that essentially models the component's initialization, ready, standby, emergency, failure, and shutdown states and the associated transitions. These core states correspond to a set of core messages for the components.

6.2.2 JAUS Messages

The messages are critical in facilitating interoperability. The following characterizes the JAUS RA messages:

- There are six categories of messages, namely, command, query, inform, event setup, event notification, and node management. Each message is assigned a code, i.e., an ID. Each message class is assigned a code range, or message space.
- Additional, blank message space is reserved for applications that require messages not yet specified in RA. Users are encouraged to submit the proven messages to the Working Group for consideration.
- Messages begin with a standard, 16-byte header. The header specifies 11 fields, including message properties, source and destination IDs for the involved components, nodes, and subsystems. The message property field specifies such properties as message priority, ACK/NAK, whether user-defined or standard message, and the version of RA that the message conforms to.
- For messages carrying large data sets, there is a data control field in the message header to indicate that the message contains a part of a large data set.
- There is a service connection feature in the messaging that support regular, periodic data transmission. JAUS RA also supports the broadcast type of messaging.

6.2.3 Integrating components and messages

The following characterize the relationship between components and messages:

- Each component must be able to receive a set of core command messages that command the component to any of its core state, including shutdown, standby, and clear emergency.
- The core messages also contain the means for creating, confirming, activating, suspending, and terminating a service connection. Core messages also include request, release, confirm, and reject component control.
- The driver components take a command message corresponding to their names. For example, global vector driver receives a message called Set Global Vector.
- The query messages are paired with inform messages. The query message class may query a component's characteristics, including its authority and status. The platform components may receive messages querying the components' driving status.

6.3 C4ISR, TRM, and 4D/RCS concerns

Comparing to C4ISR's architecture types, RA fits the TA type in that it specifies component and message standards. JAUS specifies itself to be mission isolated and computer hardware independent. RA is, therefore, transparent to both the operational and systems architectures. DM, on the other hand, provides a

generic model for the Operational Architectural specification. RA components fit into TRM level 4.

4D/RCS's universal controller aims to be a comprehensive component model for hierarchical control at any level. The controller complements the JAUS components in providing a mechanism to integrate the JAUS components. Further investigation is being planned to discover whether the two architectures could be further related.

6.4 Interoperability

Figure 4 demonstrates interoperability involving architectures like 4D/RCS using JAUS messages. Within the 4D/RCS vehicle control architecture, each node can receive command messages corresponding to the node's capability. A communicator subsystem is included to convert and route the JAUS messages sent from the Operator Control Unit (OCU). Status reports from 4D/RCS are also converted to the JAUS format before being sent to OCU.

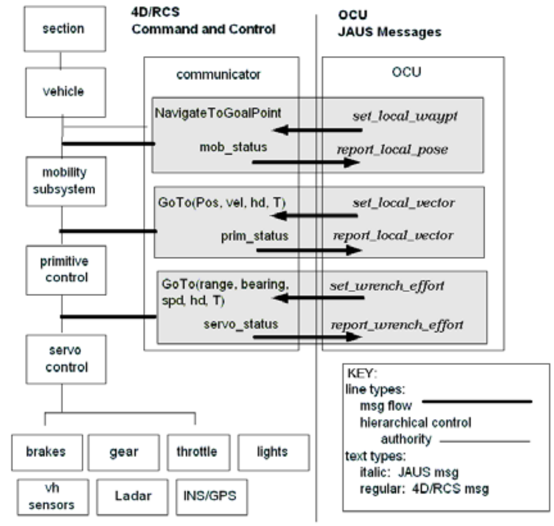

Figure 4: interoperability enabled with JAUS messages

7. Summary

We have described several major DoD architectural standards. We focus on how they are applicable to intelligent robotic systems. We have illustrated, by using the 4D/RCS reference model architecture, how all these architectural standards can facilitate system interoperability.

4D/RCS provides an architectural framework to facilitate component and interface standards development in the areas including command and control, decision aid, sensors, communication, mapping, operating environments, safety, security, software engineering, user interface, and data interchange. Therefore, the 4D/RCS reference model architecture is naturally adaptable and could provide value-added to the DoD and Army architectural standards framework.

Our analysis shows that all the architectural standards covered in this paper are conceptually consistent to each other. They are all extremely positive steps toward DoD's objectives of interoperability, information superiority, and joint and integrated operation.

Product/Company Disclaimer

Certain commercial products or company names are identified in this paper to describe our study adequately. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products or names identified are necessarily the best available for the purpose.

References

1 http://www-jta.itsi.disa.mil/
2 http://www.boeing.com/fcs
3 http://www.fas.org/irp/doddir/dod/c4isr/es.htm
4 http://www.c3i.osd.mil/org/cio/doc/GPM11-8450.pdf
5 http://www-trm.itsi.disa.mil/
6 http://www.jauswg.org/
7 Joint Technical Architecture-Army, Department of Army, Version 6.5, 10 May 2002.
8 http://wstawg.army.mil/
9 Vehicle Electronics (Vetronics) Reference Architecture, Draft version 1.0, U.S. Army Tank-Automotive Research, Development & Engineering Center (TARDEC) Electronic Architecture Team, Warren, MI, October 1, 2001.
10 http://nrac.onr.navy.mil/webspace/list/reportlistalpha.html
11 James Albus, et al., 4D/RCS: A Reference Model Architecture For Unmanned Vehicle Systems, Version 2.0, NISTIR 6910, Gaithersburg, MD, 2002.
12 Chatfield, J. et al., "New Architecture Directions", The Edge, The Mitre Advanced Technology Newsletter, Vol. 01, No. 03, January 1998.
13 SAE 4893, Generic Open Architecture (GOA) Framework, SAE AS-5 GOA Task Group, January 1996