



## **A Practical View and Future Look at JAUS**

Author: Jorgen Pedersen, President & CEO, RE2, Inc.

White Paper  
May 2006

This document is protected by copyright limiting its duplication and distribution. This document may be downloaded and/or distributed in its entirety and without modification. The use of any part of this paper without prior written consent by RE2, Inc. is strictly prohibited.

The purpose of this white paper is to provide an overview of the Joint Architecture for Unmanned Systems (JAUS), a Department of Defense (DoD) mandated messaging protocol for unmanned systems. The paper provides answers to both basic questions, such as what is JAUS and why use it , and more complex issues, such as how to implement JAUS and how to integrate JAUS with STANAG 4586. The author also provides insight into the future of the standard under the Society of Automotive Engineers (SAE).

---

## Contents

WHAT IS JAUS? .....	Page 3
WHY USE JAUS? .....	Page 3
JAUS OVERVIEW .....	Page 4
Terminology .....	Page 4
Messages .....	Page 4
JAUS Topology - Figure 1.....	Page 4
Routing .....	Page 5
Example of Routing – Figure 2.....	Page 6
Message Transmission .....	Page 6
COMPLIANCE .....	Page 7
FUTURE OF JAUS UNDER SAE .....	Page 7
INTEGRATING JAUS AND STANAG 4586 .....	Page 8
INTEGRATING JAUS AND STANAG 4586 – Figure 3.....	Page 8
IMPLEMENTING JAUS AS/4 .....	Page 8
ABOUT THE AUTHOR .....	Page 9
ABOUT RE2, INC. ....	Page 9

## What is JAUS?

The Joint Architecture for Unmanned Systems (JAUS) is a system that has been mandated within the Department of Defense to promote interoperability between Operator Control Units (OCUs), unmanned vehicles, and payloads on unmanned vehicles. Many organizations are reluctant to adopt JAUS because the word “architecture” is part of the acronym. Use of the word architecture is misleading because JAUS does not specify a complete architecture. In its current embodiment, JAUS can be viewed as purely a message set. A true architecture would dictate how to design an unmanned system completely, whereas a message set only defines how higher-level elements of that unmanned system communicate with one another.

JAUS currently promotes interoperability for unmanned vehicles. JAUS does not guarantee interoperability. JAUS should be viewed as a framework for expediting the integration of unmanned systems.

Initially, JAUS was developed primarily for Unmanned Ground Vehicles (UGVs), but the JAUS Working Group ([www.jauswg.com](http://www.jauswg.com)) is now addressing the needs of Unmanned Surface-water Vehicles (USVs), Unmanned Under-water Vehicles (UUVs), and Unmanned Air Vehicle (UAVs).

## Why use JAUS?

From a program perspective, JAUS reduces life-cycle costs, reduces development and integration time, provides a framework for technology insertion, and accommodates expansion of existing systems with new capabilities. From an engineering perspective, JAUS promotes interoperability, provides a standard, robust message set, which has been thoroughly tested, and allows for new messages to be easily inserted.

From a business perspective, JAUS is now a mandated standard within the United States Navy for most unmanned vehicles. At the AUVSI Plenary Session held on June 29<sup>th</sup> 2005, Rear Admiral William Landay, III mandated “If you bring us a UGV, USV, or UUV that’s not JAUS-compliant, we’re not interested”. Congress recently mandated that 30% of the U.S. Military shall be unmanned by 2015 through the Future Combat Systems (FCS) program. JAUS is mandated for the following elements of FCS: UAVs, UGVs, Unattended Ground Sensors (UGS), and Unattended Munition Systems (UMS).

## JAUS Overview

### Terminology

JAUS elements are defined as a hierarchy of elements. At the highest level, a *system* is a collection of subsystems. *Subsystems* are logical stand-alone operational entities such as OCUs or unmanned vehicles. Within a subsystem, computer processors are roughly defined as *nodes*. Software processes executing within a node are defined as *components*. Finally, there can be multiple *instances* of components. In most applications, there is only one instance of any given component. An overview of this hierarchy is depicted in Figure 1.

### Messages

JAUS is based on messages, which are sent between component instances. JAUS messages are organized into several classes. The *command* class defines messages, which cause an action to be performed upon receipt. A component may also ask another component for information using *query* class messages. Upon receiving query messages, a component can respond with *inform* class messages. A component may also send an inform class message to another component without being queried. There are also message classes to handle *event set up and notification* messages. Finally, *user-defined* messages can be employed if the current message set does not provide the functionality required for a specific application.

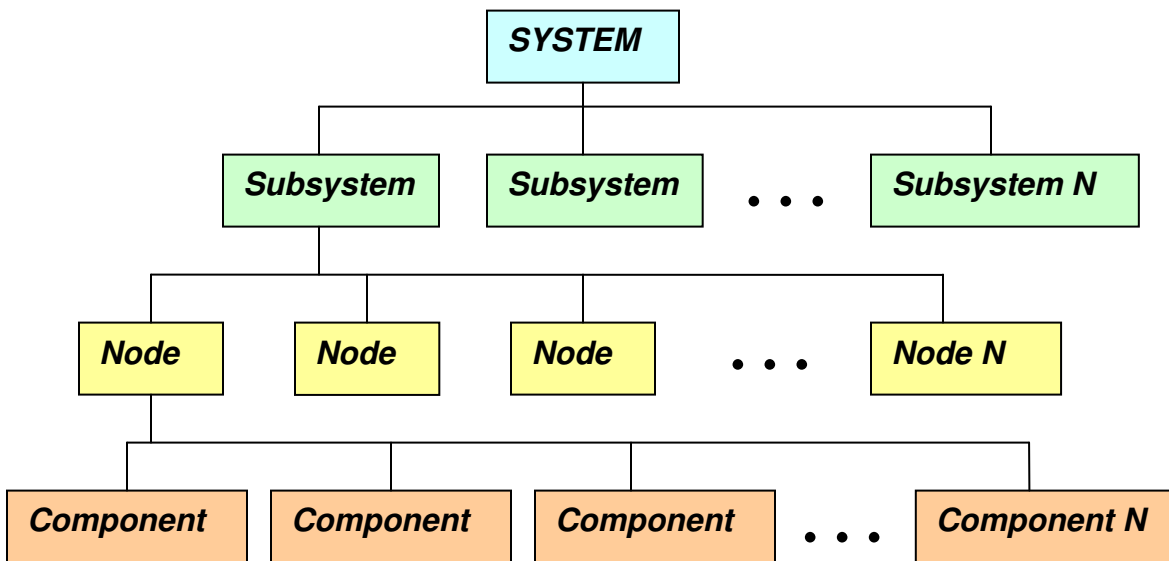


Figure 1: JAUS Topology

As deficiencies are identified within the current JAUS message set, a new subcommittee is formed under the main JAUS Working Group committee. The new subcommittee is tasked with identifying new messages for insertion into the message set. The subcommittee generates user-defined messages, which undergo experimentation before being inserted into the message set.

Although user-defined messages are allowed, they should only be avoided in the spirit of obtaining true interoperability, and they should only be created under guidance from the JAUS committee. For example, suppose a robot system uses UTM (Universal Transverse Mercator) coordinates to represent a robot's position relative to the earth. A solution would be to define a new JAUS message to send the robot's position as UTM coordinates. A better solution, in the spirit of interoperability, would be to convert UTM coordinates to latitude/longitude and use the "Global Pose" message already provided in the JAUS message set.

It is important to realize that there is no perfect standard and that achieving true interoperability is difficult. However, JAUS provides an excellent framework for subsystems (i.e. robots and OCUs) and nodes (i.e. payloads) to communicate. In addition to defining message structure, JAUS helps promote interoperability by providing standard definitions. For example, JAUS defines the reference frames for unmanned vehicles and the linkage definitions for manipulator arms.

## Routing

There are two special JAUS components required for routing messages. At the highest level, a *communicator* is the portal for all messages entering and leaving a subsystem. Similarly, a *node manager* is the portal for all messages entering and leaving a node. Communicators and node managers can be viewed simply as routers. Figure 2 depicts an example of how messages are routed in a JAUS system.

For example, referring to Figure 2, if Component 1 wants to send a message to Component 2, it merely sends the message up to its node manager, Node Manager 1. Node Manager 1 recognizes that the message is intended for one of its components and relays the message down to Component 2. If Component 1 wants to send a message to Component 4, it sends the message up to Node Manager 1. Node Manager 1 realizes that Component 4 is not one of its components and relays the message up to its communicator, Communicator 1. Upon recognizing that the message is not under its purview,

Communicator 1 relays the message to the appropriate Communicator. Communicator 2 then sends the message to Component 4 through Node Manager 3.

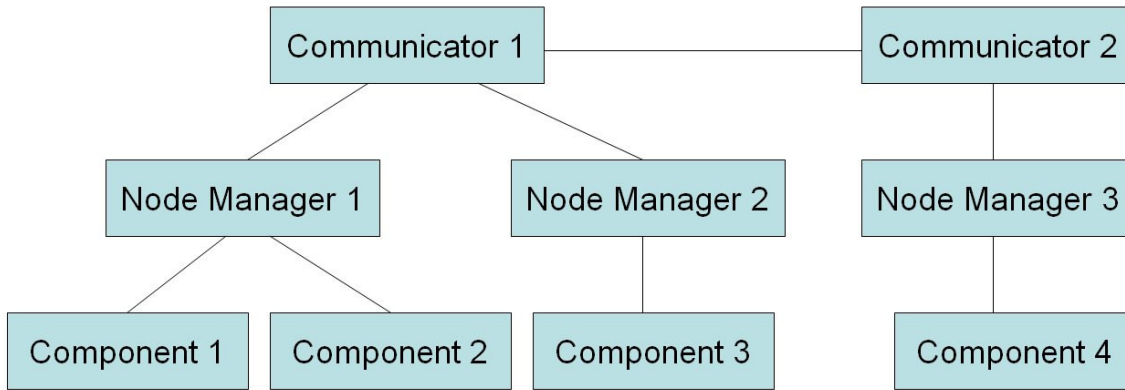


Figure 2: Example of JAUS Routing

**Message Transmission**

JAUS currently *promotes* interoperability. JAUS does not *provide* interoperability. For example, vendor A may be sending JAUS messages as TCP/IP packets and vendor B may be sending JAUS messages as UDP/IP packets. Although both vendors may be JAUS-compliant, since they are using different transport mechanisms, they will never communicate. However, the JAUS committee will soon be releasing best practice guidelines that state that UDP/IP and RS232 shall be transport protocols used by all JAUS messages. UDP/IP is suggested for high bandwidth applications and RS232 is suggested for low bandwidth and low latency applications such as UUVs.

## Compliance

JAUS Compliance currently maps into the JAUS hierarchy. There are three levels of JAUS Compliance. Referring to Figure 1, Level 1 Compliance states that all messages between subsystems are JAUS. If all messages between subsystems and nodes are JAUS, then the system is in Level 2 Compliance. Finally, Level 3 Compliance states that all messages between subsystems, nodes, and components are JAUS. Level 1 Compliance is realistic with the current version (3.2) of the JAUS message set. Level 2 Compliance is realistic with the next version of the message set which will include dynamic registration of payloads. Level 3 Compliance may never be realistic because the message set will most likely not have the granularity required for communications at the component level.

## Future of JAUS under SAE

JAUS has been adopted by the international standards organization, the Society of Automotive Engineers (SAE). JAUS is known as “AS-4” under the SAE Aerospace Division. There are three main committees under AS-4. AS-4A, “Architecture Framework”, is responsible for identifying deficiencies in the current message set and generating a comprehensive list of capabilities to overcome those deficiencies. AS-4B, “Network Environment”, is responsible for the JAUS message structure and the transport layer over which JAUS messages are transmitted. AS-4C, “Information Modeling” is responsible for transforming capabilities into new messages for insertion into the current message set. AS-4C also conducts the experimentation of all new messages.

AS-4 will be releasing new functionality with the next version of JAUS. This functionality includes better plug-n-play capability with dynamic registration messages. This dynamic registration message set will serve as the foundation for interoperability at the node level, making Level 2 Compliance realistic. The new message set will also contain messages for world modeling which will enable autonomous unmanned vehicle applications, mission planning messages, and messages to support the use of weapons on unmanned systems. AS-4 will be providing new, complete documentation, including the best practice guidelines for the transport layer.

### Integrating JAUS and STANAG 4586

The NATO standard “STANAG 4586” is widely used in the UAV community. Integrating JAUS with STANAG 4586 to Level 1 Compliance is easily realized by replacing the data link element, STANAG 7085 with JAUS. JAUS messages map well to 4586 messages, allowing for easy translation of JAUS messages into 4586 messages, and vice versa. Figure 3 shows how JAUS can be integrated within the STANAG 4586 framework.

### Implementing JAUS/AS-4

There are two core elements to JAUS: 1) message packing/unpacking, and 2) message delivery. An organization can either develop this code base in-house, subcontract parts or the entire development effort, or purchase a Software Development Kit (SDK). Purchasing a SDK will save an organization both time and money.

Information about the RE2 JAUS SDK can be found at [http://www.resquared.com/JAUS\\_SDK.html](http://www.resquared.com/JAUS_SDK.html). Details about the JAUS standard can be found on the JAUS Working Group website – [www.jauswg.org](http://www.jauswg.org).

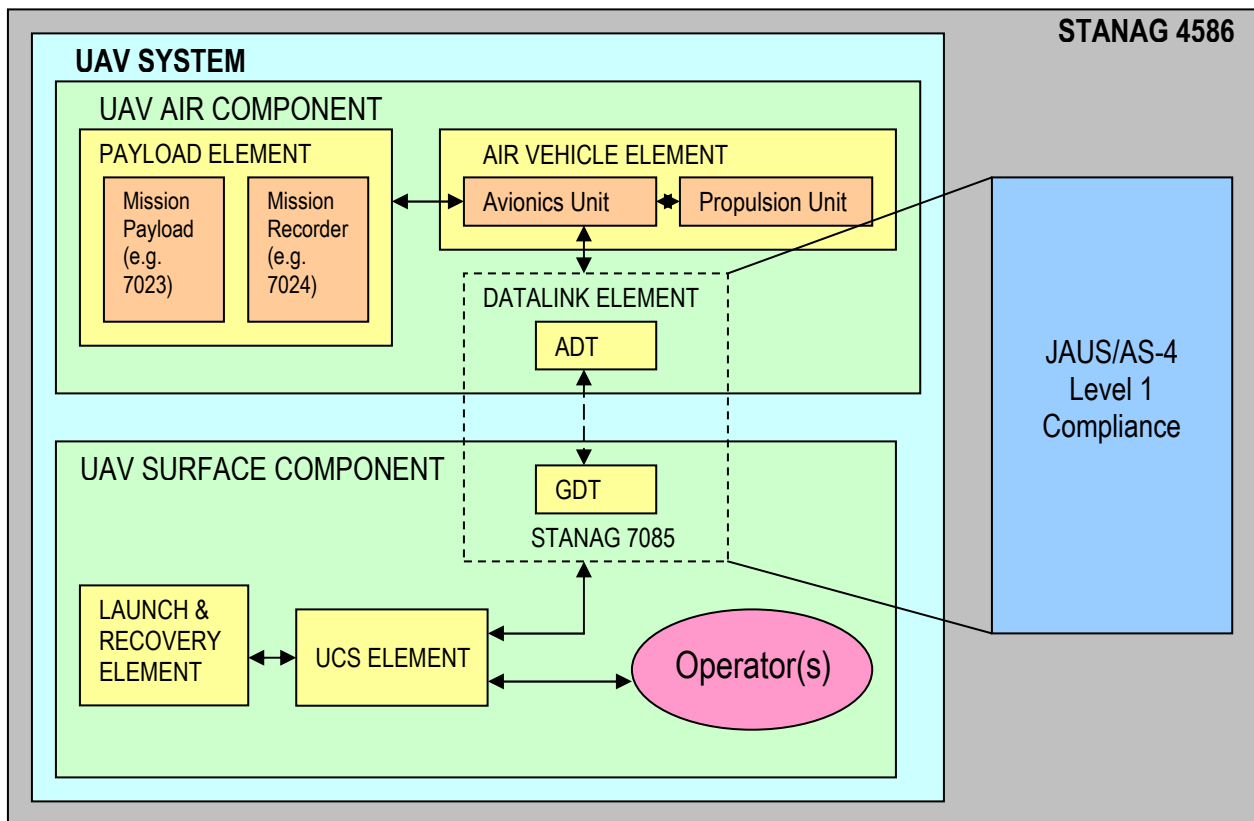


Figure 3: Integrating JAUS with STANAG 4586

## About the Author

Jorgen Pedersen founded RE2, Inc. in July of 2001. Mr. Pedersen, RE2 President & CEO, holds a M.S. degree in Robotics and a B.S. degree in Electrical and Computer Engineering from Carnegie Mellon University. Mr. Pedersen has also taken the executive course in Technology Commercialization at the Carnegie Mellon Graduate School of Industrial Administration. He has been involved in robotics for over 10 years, gaining engineering, managerial, and business experience. Mr. Pedersen is a voting member of the JAUS Working Group ([www.jauswg.org](http://www.jauswg.org)) and the SAE AS-4 committees. He sits on the AS-4A, AS-4B, AS-4C and AS-4 Executive committees. He is also the Chairman of the JAUS/AS-4 Weapons Payload Subcommittee/Task Group. Mr. Pedersen holds two patents related to robotics.

## About RE2, Inc.

RE2, Inc. is a Carnegie Mellon spin-off company specializing in mobile defense robotics with an emphasis on JAUS compliant unmanned systems and components. RE2 provides a comprehensive JAUS SDK ([http://www.resquared.com/JAUS\\_SDK.html](http://www.resquared.com/JAUS_SDK.html)) and integration services to ensure that unmanned systems meet the Department of Defense's requirements for JAUS compliance. RE2 also provides unmanned system platforms and payloads. Through extensive engineering and project experience, RE2 has developed software and robotic solutions to enhance, improve, and test robotic systems. For more information, visit [www.resquared.com](http://www.resquared.com) or call 412.681.6382.

RE2, INC.  
32 39<sup>th</sup> Street  
Pittsburgh, PA 15201  
T: 412.681.6382  
F: 412.681.5904  
[info@resquared.com](mailto:info@resquared.com)  
[www.resquared.com](http://www.resquared.com)

This document is protected by copyright limiting its duplication and distribution. This document may be downloaded and/or distributed in its entirety and without modification. The use of any part of this paper without prior written consent by RE2, Inc. is strictly prohibited.