Introduction to JAUS for Unmanned Systems Interoperability

– Joint Architecture for Unmanned Systems –

Daniel Serrano
Department of Intelligent Systems, ASCAMM Technology Center
Parc Tecnològic del Vallès, Av. Universitat Autònoma, 23
08290 Cerdanyola del Vallès
SPAIN

dserrano@ascamm.com

ABSTRACT

This paper focuses on the Joint Architecture for Unmanned Systems (JAUS). JAUS is an international standard of the SAE AS-4 Unmanned Systems Steering Committee. This paper will discuss the basic concept of the standard and their applicability to accomplish interoperability among heterogeneous unmanned systems. It will also describe the standard documents and the JAUS service sets. Lastly, the paper highlights the advantages and limitations of the JAUS standard.

1.0 INTRODUCTION

The Joint Architecture for Unmanned Systems (JAUS) is an international standard that establishes a common set of message formats and communication protocols for supporting interoperability within and between unmanned vehicles and ground control stations.

It was originally chartered by the United States Department of Defense (DoD) to provide an open architecture for the domain of Unmanned Ground Robots. JAUS was later converted to an international industry standard - currently it belongs to the Society of Automotives Engineers (SAE), a standards development organization (SDO) with robotics experience, which established the Aerospace Standards Unmanned Systems Steering Committee (AS-4) Committee in August 2004. All of the standard documents that define JAUS can be purchased online directly from SAE.

Its original purpose was to define an open communication standard to support interoperability of robotic systems in the military. Many programs and vendors became involved with JAUS and have successfully demonstrated its use on unmanned systems. Some examples of these systems include several robots from the 2004 and 2005 DARPA Grand Challenges and the 2007 DARPA Urban Challenge.

2.0 JAUS OVERVIEW

The main goal of JAUS is to structure communication and interoperation of unmanned systems within a network.

A JAUS system is made up of subsystems connected to a common data network. A Subsystem typically represents a physical entity in the system network, such as an unmanned vehicle or operator control unit (See Figure 1).

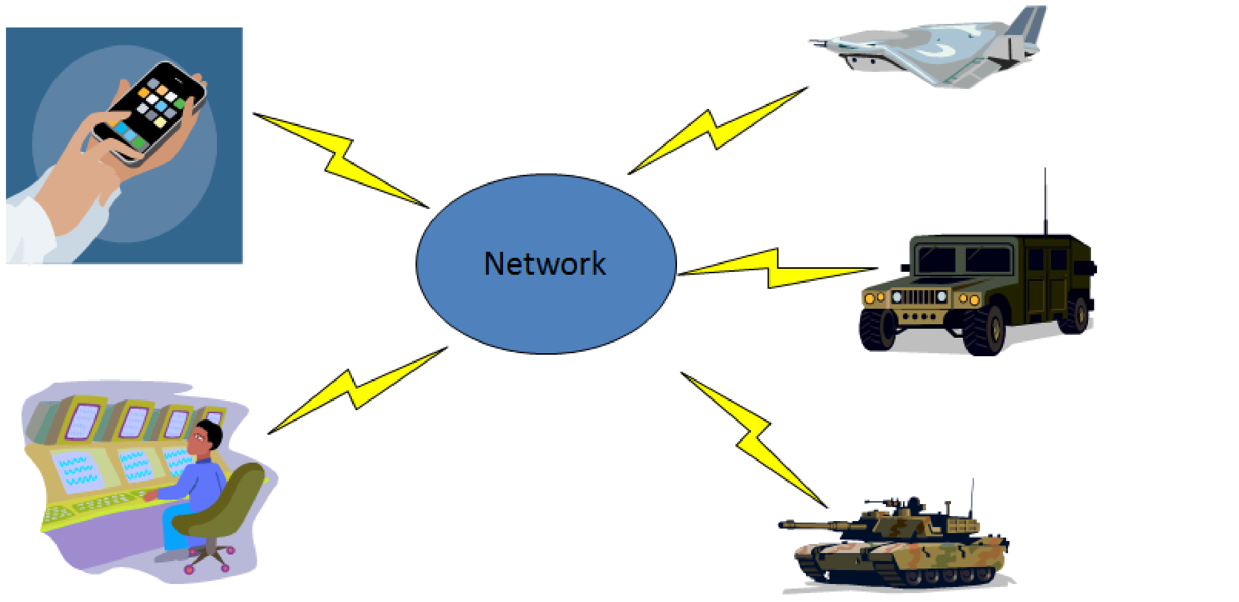

Figure 1: High-Level JAUS system architecture [1].

The JAUS network is further subdivided into hierarchical layers. Subsystems are divided into Nodes, which represent a physical computing end-point in the system. For example, a Node might be a computer or microcontroller within a Subsystem.

Nodes can then host one or more Components, which are commonly applications or threads running on the Node. Finally, Components are made up of one or more Services.

Therefore, a system would look like this:

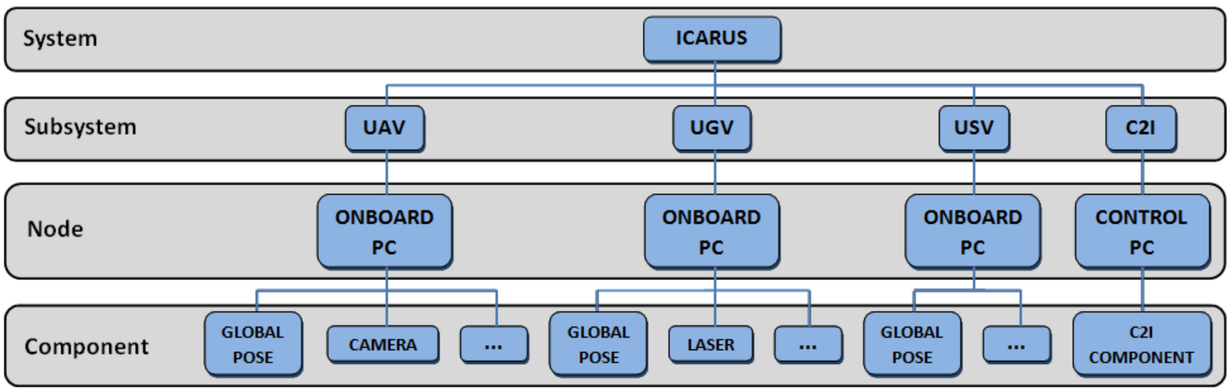

Figure 2: An example of a JAUS topology.

A Service simply provides some useful function for the system. The Service Oriented Architecture (SOA) enables distributed command and control of the unmanned systems. The SOA approach of JAUS attempts to formalize the message format and protocol interaction between system components. This approach is standardized by the JAUS Service Interface Definition Language (JSIDL); an XML-based language that provides the basic structure and syntax for specifying JAUS Services. All of the Services that are standardized by JAUS must be specified in valid JSIDL syntax.

3.0 JAUS STANDARD DOCUMENTS

The specifications of the JAUS standard are published as separate documents. The following two documents provide the foundations for the later specification of the JAUS services:

Table 1: JAUS specification documents.

JAUS Standards	Date	Content
JAUS Transport Specification (SAE AS5669A) [1]	2007.12	Specifications for UDP, TCP, and Serial based data transmission of JAUS messages.
JAUS Service Interface Definition Language (SAE AS5684) [2]	2010.07	Defines the data structures of services, messages and protocol, formalized as an XML schema.

The JAUS standard is built upon JSIDL which defines an XML schema that enables formal specification of JAUS Services, Messages and Message Protocol. This schema aids robust and reliable interoperability by removing some of the ambiguities often found in hand-written standards.

The JAUS services are grouped in Sets and published as related but separate documents. They described generic concepts commonly found in unmanned systems. The following table lists the most relevant ones:

Table 2: JAUS service sets.

JAUS Standards	Date	Content
JAUS Core Service Set (SAE AS5710A) [3]	2010.08	Low level services such as transport and discovery to enable basic interoperation
JAUS Mobility Service Set (SAE AS6009) [4]	2009.04	Common mobility services such as global positioning and vehicle platform control by defining abstract services that are agnostic to specific vehicle mobility types (ground vehicles, aircraft, etc.)
JAUS Environment Sensing Service Set (SAE AS6060) [5]	2010.11	Environment sensing capabilities commonly found across all domains and types of unmanned systems in a platform-independent manner (range, visual, video, etc).
JAUS Manipulator Service Set (SAE AS6057) [6]	2011.03	Service definitions for controlling robotic manipulators. Messages are defined generically so they can be applied to many different types of manipulators (arms, grippers, pan/tilt, etc.).
JAUS HMI Service Set (SAE AS6040) [7]	2010.11	Service definitions for HMI interaction that includes drawing, keyboard input, pointing device input, analog and digital user controls.
JAUS Mission Spooling Service Set (SAE AS6062) [8]	2010.07	Services definition to store mission plans, coordinate mission plans, and parcel out elements of the mission plan for execution

JAUS Unmanned Ground Vehicle Service Set (SAE AS6091) [9]	2014.07	Represent the platform-specific capabilities commonly found in UGVs, and augment the Mobilty Service Set [AS6009] which is platform-agnostic.

Some JAUS services sets are still under development and may become available in the future:

Table 3: JAUS future service sets.

JAUS Standards	Content
JAUS USV Service Set (SAE AS6063)	Unmanned surface vehicle specific capabilities that are not supported by the higher level platform-independent services. This service enables interoperability on common elements specific to unmanned surface vehicles.
JAUS Unmanned Underwater Vehicle Service Set (SAE AS6111)	Unmanned underwater vehicle specific capabilities that are not supported by the higher level platform-independent services.

4.0 JAUS TRANSPORT SPECIFICATIONS

The JAUS Transport specification provides an interface between the application layer and the communication media. Every application-layer payload (packet) is wrapped around by a header that assists in the routing, verification and delivery of the message:

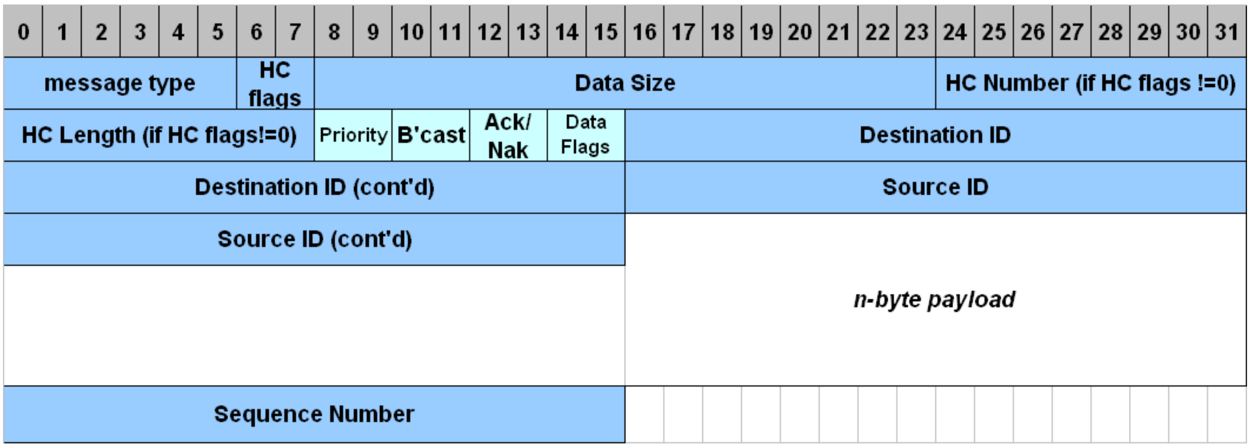

Figure 3: JAUS General Transport Header Format [1].

The contents of the fields required for proper transport in this common transport header follow certain guidelines. Valid ranges, proper combinations for bit mapped fields and other field details are described in the document [1].

A variety of transports may be used to transmit the data – standard Ethernet, wireless IP-based radios (like 802.11b/g), long-range serial radios, etc. Currently, JAUS support UDP (JUDP), TCP (JTCP) and serial (JSerial).

The native JAUS Discovery protocol uses broadcast semantics to discovery JAUS assets. The key in IP-based networks employing port-forwarding is to ensure that this broadcast propagates everywhere it needs to go. The JAUS Discovery protocol multicasts to port 3794. Hence, the process on the host that receives these multicasts must be capable of routing them to all parties that are interested in participating in this protocol.

5.0 JAUS SERVICE INTERFACE DEFINITION LANGUAGE (JSIDL)

The JSIDL is a precise XML schema that can be used to design JAUS services. This approach allows to formalize JAUS as a Service Oriented Architecture and to specify unambiguously how to operate a JAUS service.

A service is defined as a set of input and output JAUS messages, together with a protocol state machine that defines how the messages are to be received, processed and sent by that service.

The XML syntax of JSIDL is only used for the specification of services when they are published in standard. It is not intended to be transferred by JAUS systems at runtime.

JSIDL specifies how messages are to be structured in a JAUS service. It does not specify the format of specific messages itself. These are defined in the JAUS Service Sets document for each specific group of services.

```
<service_def
   xmlns="urn:jaus:jsidl:0.11"
   name="service_name"
   id="service_uri"
   version="1.1">

  <description> </description>

  <assumptions> </assumptions>

  <references> </references>

  <declared_const_set> </declared_const_set>
  <declared_type_set> </declared_type_set>

  <message_set>
    <input_set> </input_set>
    <output_set> </output_set>
  </message_set>

  <internal_events_set> </internal_events_set>

  <protocol_behavior> </protocol_behavior>
</service_def>
```

Figure 4: JSIDL Top-Level structure of a service definition [2].

Some implementations of the JAUS standards, for instance OpenJAUS [10], use the JSIDL specification in of a given service to automatically generate a stub implementation. This approach aids robust and reliable interoperability by removing some of the ambiguities often found in hand-written standards and helps to reduce or eliminate any human error that can be introduced when interpreting JAUS standard services.

6.0 JAUS MESSAGE DEFINITIONS

The format of a JAUS message is defined using a message_def element. :

```
<message_def name="Request_Control" message_id="000d" is_command="true">
    <description xml:space=”preserve”>
        This message is used to request exclusive control of...
    </description>
    <header>
        ...
    <header>
    <body>
        ...
    </body>
    <footer/>
</message_def>
```

Figure 5: Template for message definition [2].

JISDL defines the data field elements available to define a message:

- All Fields
  - Composite Fields
    - array
    - record
    - list
    - variant
    - sequence
  - Simple Fields
    - Primitive Fields
      - fixed_field
      - variable_field
      - bit_field (and sub_field)
    - String Fields
      - fixed_length_string
      - variable_length_string
    - BLOB Fields
      - variable_length_field
      - variable_format_field
  - Meta Fields
    - count_field
    - vtag_field
    - format_field
    - presence_vector
    - type_and_units_field

Figure 6: Data field elements.

Although the message structure and format is specified in XML, the format of the actual JAUS message is a binary format containing only the serialized data.

A JAUS message is defined by a globally unique Message ID composed by 2 bytes. This identifier is always at the beginning of the application-layer payload. There are generally three types of messages to interface to a service: Query, Report and Command.

7.0 JAUS CORE SERVICE SETS

JAUS standard defines a set of common services to enable basic interoperability between and within unmanned systems. These core services allows component to transport data to each other, configure events for messaging, dynamically discover other systems and their services, exclusive access control, etc.

Within the Core services, JAUS defines the following:

- Transport Service: Abstracts the functionality of the underlying communication transport layer
- Events Service: Establishes a publish/subscribe mechanism for automatic messaging
- Access Control: Manages preemptable exclusive control for safety critical operations
- Management: Defines component life-cycle management
- Time: Allows clients to query and set the system time for the component
- Liveness: Provides a means to maintain connection liveness between communicating components
- Discovery: Governs automatic discovery of remote entities and their capabilities

The Transport service provides abstraction from the underlying communication protocol layers. This service establishes a communication endpoint whose address is defined by a triple {SubsystemID, NodeID, ComponentID}. Other services that need to utilize the communication channel provided by the transport service must inherit from the transport service. It provides a bi-directional communication channel (input queue and output queue) with the capability of sending messages to a single destination endpoint or broadcasting messages to all endpoints in the system, and to receive a message from any source endpoint. It also provides the capability to prioritize the delivery of sent messages.

The Access Control service is used for acquiring preemptable exclusive control to one service. Once the exclusive control is established, the related services shall only execute commands originating from the controlling component. The authority code parameter of this service is used for preemption and is to be set equal to that of its controlling client. This service always grants control to the highest authority client that is requesting exclusive control. For example, some of the JAUS mobility services uses the core access control to prevent a

The Discovery service enables the Components to discover one another online. This dynamic discovery mechanism greatly aids the integration process because it allows JAUS developers to quickly network their systems together.

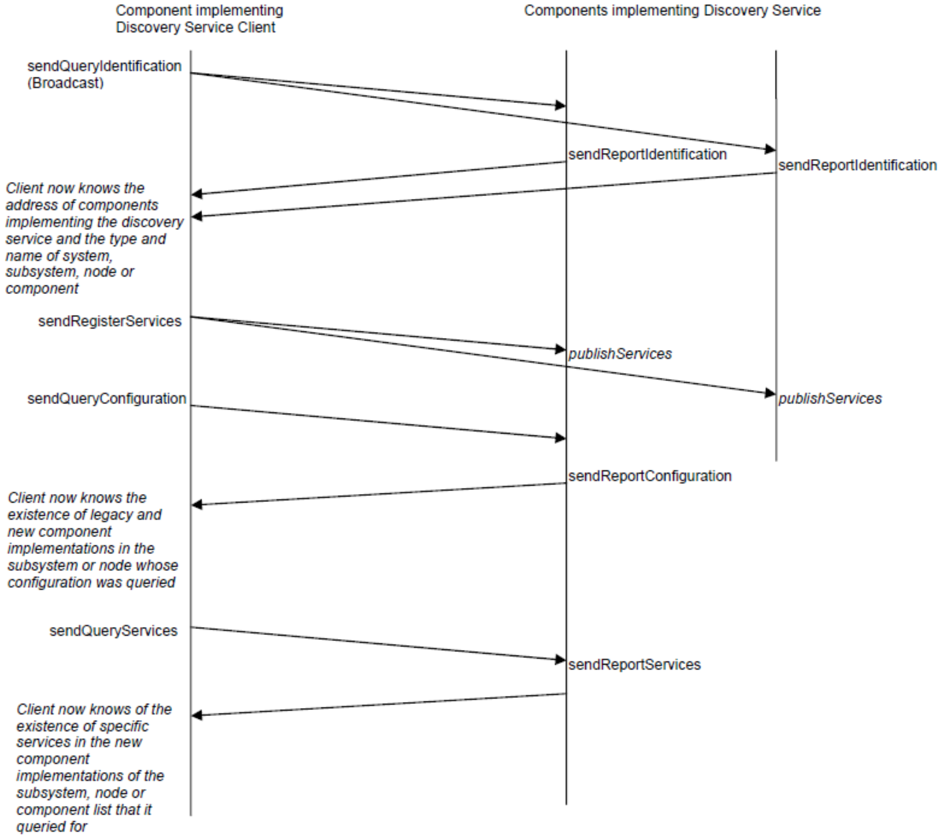

Figure 7: Sequence diagram of service discovery [3].

8.0 JAUS MOBILITY SERVICE SETS

The Mobility Services represent the platform-independent capabilities commonly found across all domains and types of unmanned systems. At present, 15 services are defined in this document. The following image illustrates:

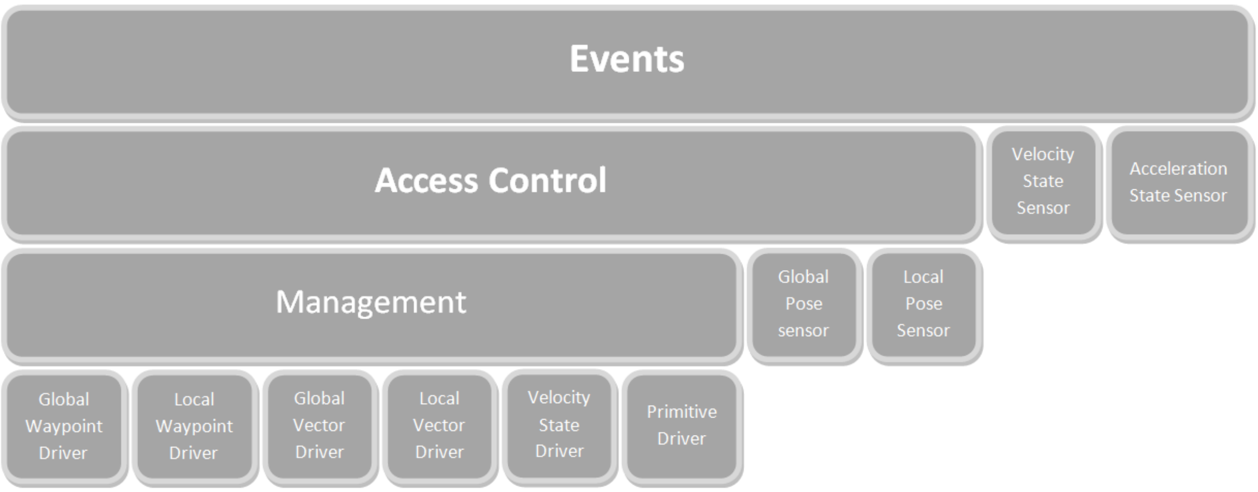

Figure 8: Mobility service set.

Some examples include:

- Pose Sensors: Determine the instantaneous position and orientation of a platform in global or local coordinates.
- Velocity State Sensor: Determines the instantaneous velocity of a platform.
- Acceleration State Sensor: Determines the instantaneous acceleration of a platform.
- Waypoint Drivers: Perform closed loop mobility to a location specified or using a series of locations.
- Vector Drivers: Perform closed loop mobility for straight line travel.
- Velocity State Driver: Similar to vector drivers, but with additional degrees of freedom.
- Primitive Driver: Performs basic mobility for a platform based on force/torque efforts.

The mobility service set defines unambiguously concepts such as platform pose, waypoint and path tolerance. For example, the following image illustrates the definition of the parameters Ψ (yaw1, or heading), θ (pitch), and φ (roll) define the orientation of the mobile platform.

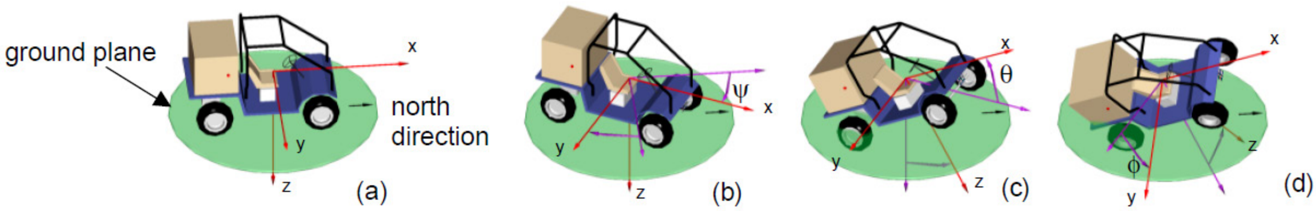

Figure 9: JAUS platform orientation.

9.0 JAUS ENVIRONMENT SERVICE SETS

The Environment Sensing Services represent typical environment sensing capabilities commonly found across all domains and types of unmanned systems in a platform-independent manner. At present, five services are defined in this document:

- Range Sensor: Determine the proximity of objects in the platform's environment
- Visual Sensor: Provides common configuration and setup for different types of imaging systems
- Digital Video: A type of Visual Sensor that manages digital video
- Analog Video: A type of Visual Sensor that manages analog video
- Still Image: A type of Visual Sensor that manages and encodes individual digital images

The function of the Range Sensor Service is to provide information from proximity sensors. This service will output the location of various Data Points with a certain measure of accuracy. A given Range Sensor service may be comprised of one to many actual physical sensors or technologies. Each sub-sensor can be assigned (by the developer) a unique Sensor ID.

The visual sensor service provides access to the basic capabilities and configuration of a visual sensor. There are three specific types: Digital and Analog Video and Still Images.

10.0 OTHER RELEVANT JAUS SERVICE SETS

The Manipulator services represent the platform-independent capabilities of serial manipulator. Some examples of services are:

- Manipulator Joint Position, Velocity and Force/Torque
- Manipulator End Effector Pose and Velocity
- Pan/Tilt Driver

The Mission spooling service set services represent the platform-independent mission execution capabilities common across all domains and types of unmanned systems. This document describes only one service to store and coordinate mission plans.

The Mission Spooler Service is responsible for storing and spooling missions. A mission is a set of SAE JAUS messages (e.g. Set Global Waypoint [AS6009]) to be performed by one or more Services of one or more unmanned systems. The mission structure is an N-ary tree, which allows for parallel, sequential, iterative, and coordinated missions. Each mission has a unique ID allowing for multiple missions. A mission is made up of tasks, which contain JAUS messages, and/or children tasks.

A JAUS message within a mission plan can be blocking (synchronous) or non-blocking (asynchoronous). The Mission Spooler shall not spool messages beyond a blocking SAE JAUS message until the unmanned system has completed the action associated with the blocking SAE JAUS message. Payload commands are a good example of where blocking messages may be used. Some payloads can only perform their functions when the unmanned system is stationary (e.g. soil sampling, video image) while other payloads can perform their functions (e.g. start mine flail) while in motion. The blocking flag ensures that no other messages are spooled until the blocking message is complete.

The fundamental actions of any mission plan are based in SAE JAUS messages supported by the underlying services. For example, the Global Path Segment Driver Service specified by the JAUS Mobility Service Set [AS6009] defines the Set Global Path Segment message for driving along a specified route. It is important to note that the mission actions are generalizations of behaviors offered by SAE JAUS services and are not new messages introduced by the Mission Spooler service.

11.0 ADVANTAGES AND LIMITATIONS

JAUS ensures interoperability across different domains and unmanned system by imposing to operate on the same message set, provided that they share a data network and this has been configured according to the specifications. JAUS does not specify how the addresses should be assigned. Each node must be manually assigned a globally unique identifier by the system builder. This fact reduces the interoperability of the systems since some manual configuration is required to enable several systems to work together.

JAUS imposes a hierarchical architecture partitioning subsystems into software components. This ensure modularity. As a result, software reuse of JAUS components should be viable.

JAUS strictly defines the messages that a component providing a given must implement. This may limited the freedom of a system builder to provide solutions to different scenarios.

JAUS specification does not provide any guidelines on real-time requirements.

Compare with other standards in the domain as for instance, The North Atlantic Treaty Organization (NATO) developed a Standard Agreement (STANAG) 4586 [11] about the standardization of control of unmanned aerial vehicles (UAVs), there is significant overlap between them. JAUS is not restricted to aerial systems and seems more accessible to small robotics platforms. However, STANAG 4586 is well established and only the merging or the adoption of a unified standard, coupled by strong policy, will be able to truly achieve unmanned systems interoperability [12].

REFERENCES

[1] AS5669 JAUS / SDP Transport Specification *http://standards.sae.org/as5669a*

[2] AS5684 JAUS Service Interface Definition Language *http://standards.sae.org/as5684a*

[3] AS5710 JAUS Core Service Set *http://standards.sae.org/as5710*

[4] AS6009 JAUS Mobility Service Set *http://standards.sae.org/as6009*

[5] AS6060 JAUS Environment Sensing Service Set *http://standards.sae.org/as6060*

[6] AS6057 JAUS Manipulator Service Set *http://standards.sae.org/as6057*

[7] AS6040 JAUS HMI Service Set *http://standards.sae.org/as6040*

[8] AS6062 JAUS Mission Spooling Service Set *http://standards.sae.org/as6062*

[9] AS6091 JAUS Unmanned Ground Vehicle Service Set *http://standards.sae.org/as6091*

[10] Galluzo, T. and D. Kent, "The OpenJAUS approach to designing and Implementing the new SAE JAUS standards," AUVSI Unmanned Systems Conference. 2010.

[11] STANAG 4586 Edition No 2 Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability *http://nso.nato.int/nso/zPublic/stanags/current/4586eed02a2.pdf*

[12] J. Pedersen. "Interoperability Standard Analysis (ISA)," Standards Committee of the National Defense Industry Association (NDIA) Robotics Division. 2007.