Unmanned Maritime Autonomy Architecture (UMAA)
Experimental Services (EXP)
Interface Control Document (ICD)
(UMAA-SPEC-EXPICD)

Version 6.0

6 June 2024

Contents

List of Figures

List of Tables

1 Scope

1.1 Identification

This document defines a set of ***experimental*** services as part of the Unmanned Maritime Autonomy Architecture (UMAA)—experimental services are not required to satisfy UMAA compliance, but are provided to industry for feedback. As such, it provides services that are in an experimental state and are in the process of being developed. This document is generated automatically from data models that define its services and their interfaces as part of the Unmanned Systems (UxS) Control Segment (UCS) Architecture as extended by UMAA to provide autonomy services for unmanned vehicles.

To put each ICD in context of the UMAA Architecture Design Description (ADD), the UMAA functional decomposition mapping to UMAA ICDs is shown in Figure 1.

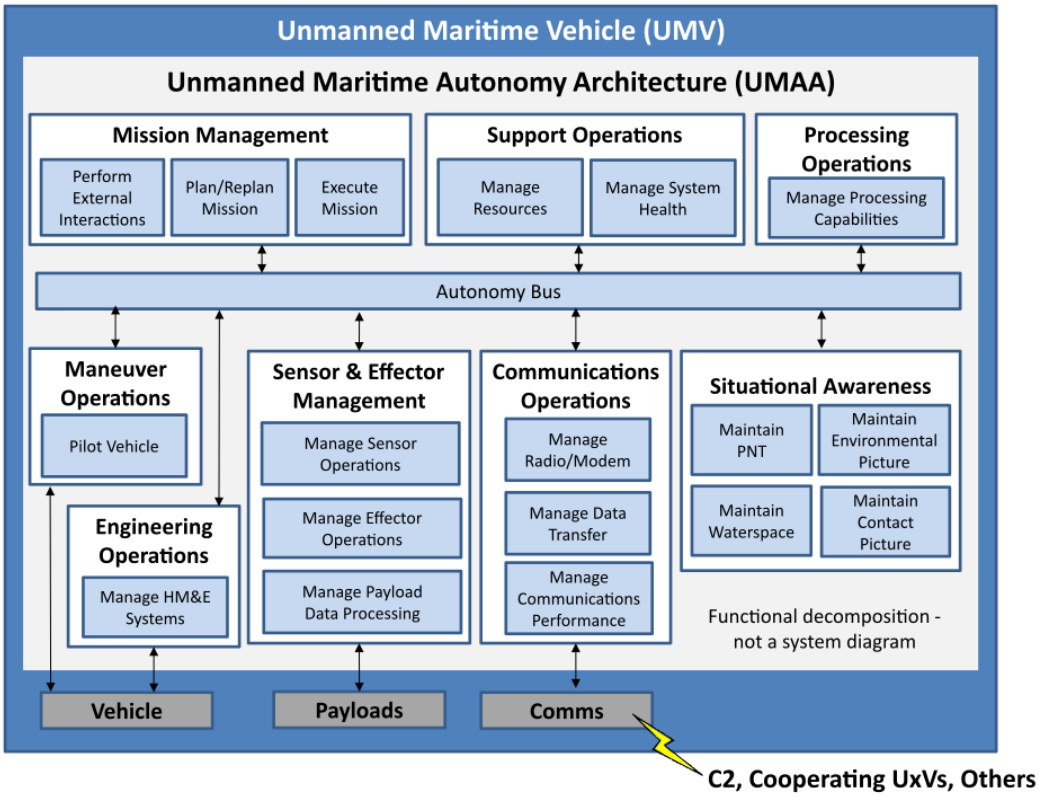

Figure 1: UMAA Functional Organization.

1.2 Overview

The fundamental purpose of UMAA is to promote the development of common, modular, and scalable software for unmanned vehicles that is independent of a particular autonomy implementation. Unmanned Maritime Systems (UMSs) consist of Command and Control (C2), one or more unmanned vehicles, and support equipment and software (e.g. recovery system, Post Mission Analysis applications). The scope of UMAA is focused on the autonomy that resides on-board the unmanned vehicle. This includes the autonomy for all classes of unmanned vehicles and must support varying levels of communication in mission (i.e., constant, intermittent, or none) with external systems. To enable modular development and upgrade of the functional capabilities of the on-board autonomy, UMAA defines eight high-level functions. These core functions include: Communications Operations, Engineering Operations, Maneuver Operations, Mission Management, Processing Operations, Sensor and Effector Operations, Situational Awareness, and Support Operations. In each of these areas, it is anticipated that new capabilities will be required to satisfy evolving Navy missions over time. UMAA seeks to define standard interfaces for these functions so that individual programs can leverage capabilities developed to these standard interfaces across programs that meet the standard interface specifications. Individual programs may group services and interfaces into components in different ways to serve their particular vehicle's needs. However, the entire interface defined by UMAA will be required as defined in the ICDs for all services that are included in a component. This requirement is what enables autonomy software

to be ported between heterogeneous UMAA-compliant vehicles with their disparate vendor-defined vehicle control interfaces without recoding to a vehicle-specific interface.

Experimental Services defines the services that are still under early state development. Figure 2 depicts an example of possible component service groupings (designated by dashed lines).

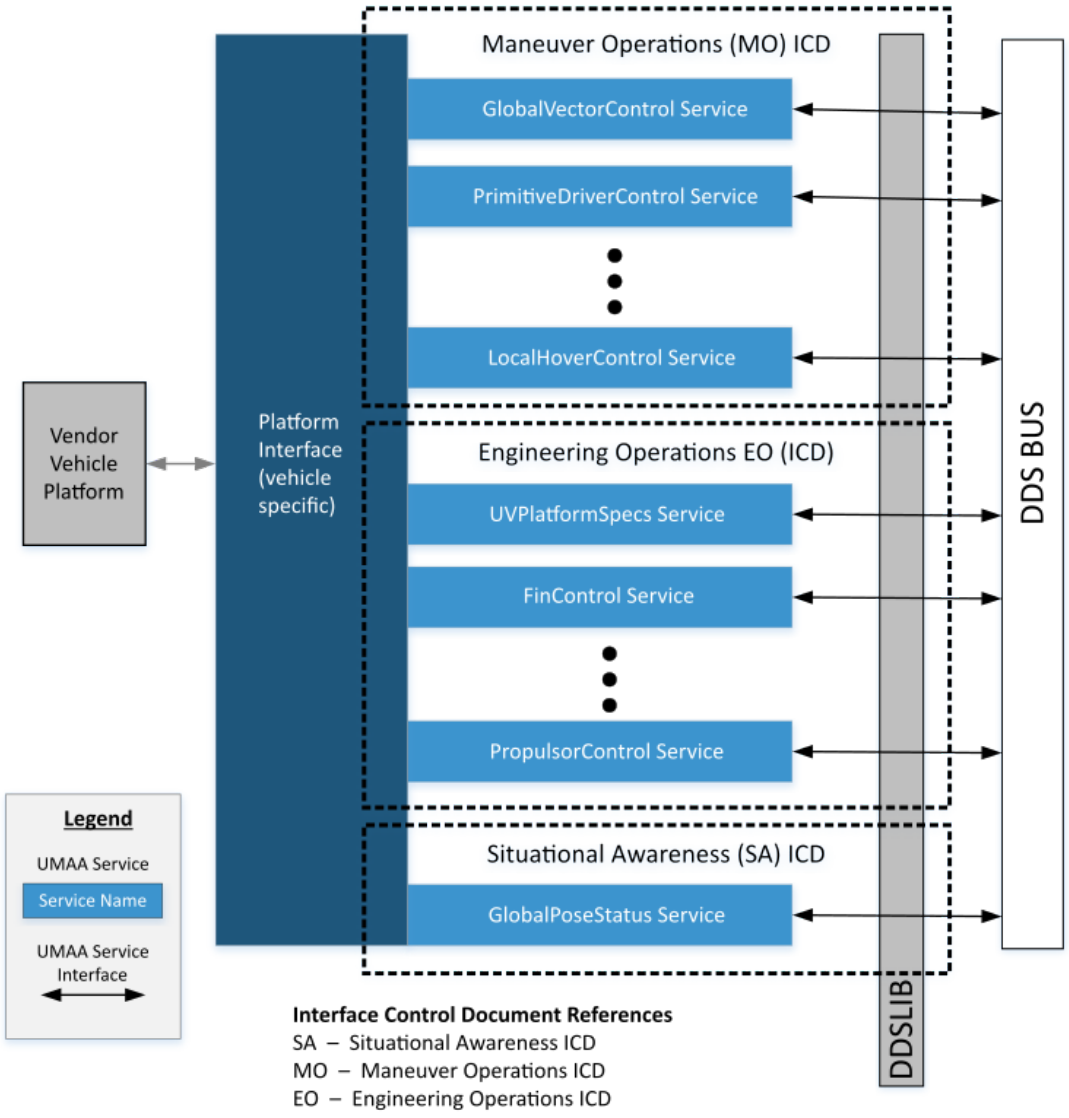

Figure 2: UMAA Services and Interfaces Example.

1.3 Document Organization

This interface control document is organized as follows:

Section 1 – Scope: A brief purview of this document

Section 2 – Referenced Documents: A listing of associated of government and non-government documents and standards

Section 3 – Introduction to Data Model, Services, and Interfaces: A description of the common data model across all services and interfaces

Section 4 – Introduction to Coordinate Reference Frames and Position Model: An overview of the reference frame model used by UMAA

Section 5 – Flow Control: A description of different flow control patterns used throughout UMAA

Section 6 – Experimental Services (EXP) Services and Interfaces: A description of specific services and interfaces for this ICD

2 Referenced Documents

The documents in the following table were used in the creation of the UMAA interface design documents. Not all references may be applicable to this particular document.

Table 1: Standards Documents

Title	Release Date
A Universally Unique IDentifier (UUID) URN Namespace	July 2005
Data Distribution Service for Real-Time Systems Specification, Version 1.4	March 2015
Data Distribution Service Interoperability Wire Protocol (DDSI-RTPS), Version 2.3	April 2019
Object Management Group Interface Definition Language Specification (IDL)	March 2018
Extensible and Dynamic Topic Types for DDS, Version 1.3	February 2020
UAS Control Segment (UCS) Architecture, Architecture Description, Version 2.4	27 March 2015
UCS Architecture, Conformance Specification, Version 2.2	27 September 2014
UCS-SPEC-MODEL v3.4 Enterprise Architect Model	27 March 2015
UCS Architecture, Architecture Technical Governance, Version 2.5	27 March 2015
System Modeling Language Specification, Version 1.5	May 2017
Unified Modeling Language Specification, Version 2.5.1	December 2017
Interface Definition Language (IDL), Version 4.2	March 2018
U.S. Department Of Homeland Security, United States Coast Guard "Navigation Rules International-Inland" COMDTINST M16672.2D	March 1999
IEEE 1003.1-2017 - IEEE Standard for Information Technology–Portable Operating System Interface (POSIX(R)) Base Specifications, Issue 7	December 2017
Guard, U. C. (2018). Navigation Rules and Regulations Handbook: International—Inland. Simon and Schuster.	June 2018
Department of Defense Interface Standard: Joint Military Symbology (MIL-STD-2525D Appendix A)	10 June 2014
DOD Dictionary of Military and Associated Terms	August 2018

Table 2: Government Documents

Title	Release Date
Unmanned Maritime Autonomy Architecture (UMAA) Architecture Design Description (ADD), Version 1.0	January 2019
Manual for the Submission of Oceanographic Data Collected by Unmanned Undersea Vehicles (UUVs)	October 2018

3 Introduction to Data Model, Services, and Interfaces

3.1 Data Model

A common data model is at the heart of UMAA. The common data model describes the entities that represent system state data, the attributes of those entities and relationships between those entities. This is a "data at rest" view of system-level information. It also contains data classes that define types of messages that will be produced by components, or a "data in motion" view of system-level information.

The common data model and coordinated service interfaces are described in a Unified Modeling Language (UML™) modeling tool and are represented as UML™ class diagrams. Interface definition source code for messages/topics and other interface definition products and documentation will be automatically generated from the common data model so that they are consistent with the data model and to ensure that delivered software matches its interface specification.

The data model is maintained as a Multi-Domain Extension (MDE) to the UCS Architecture and will be maintained under configuration control by the UMAA Board as UCSMDE and will be incrementally integrated into the core UCS standard. Section 6 content is automatically generated from this data model, as are other automated products such as IDL that are used for automated code generation.

3.2 Definitions

UMAA ICDs follow the UCS terminology definitions found in the UCS Architecture Description v2.4. The normative (required) implementation to satisfy the requirements of a UMAA ICD is to provide service and interface specification compliance. Components may group services and required interfaces in any manner so long as every service meets its interface specifications. Figure 3 shows a particular grouping of services into components. The interfaces are represented by the blue and green lines and may equate to one or more independent input and output interfaces for each service. The implementation of the service into software components is left up to the individual system development. Given this context, section 6 correspondingly defines services with their interfaces and not components.

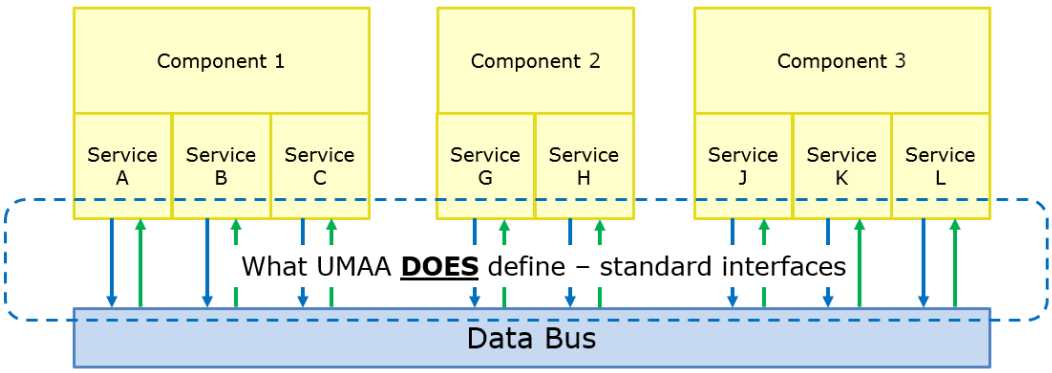

Figure 3: Services and Interfaces Exposed on the UMAA Data Bus.

Services may use other services within this ICD, or in other UMAA defined ICDs, to provide their capability. Additionally, components for acquisition and development may span multiple ICDs. An example of this would be a commercial radar that provides both status and control of the unit via the radar's software Application Programming Interface (API).

3.3 Data Distribution Service (DDS™)

The data bus supporting autonomy messaging (as seen in Figure 3) is implemented via DDS™. DDS is a middleware protocol and API standard for data-centric connectivity from the Object Management Group (OMG). It integrates the components of a system together, providing low-latency data connectivity, extreme reliability, and a scalable architecture. In a distributed system, middleware is the software layer that lies between the operating system and applications. It enables the various system components to more easily communicate and share data. It simplifies the development of distributed systems by letting software developers focus on the specific purpose of their applications rather than the mechanics of passing information between applications and systems. The DDS specification is fully described in free reference material on the OMG website and there are both open source and commercially available implementations.

3.4 Naming Conventions

UMAA services are modeled within the UCS Architecture under the Multi-Domain Extension (MDE). The UCS Architecture uses SoaML concepts of participant, serviceInterface, service port, and request port to describe the interfaces that make up a service and show how the service is used. Each service defines the capability it provides as well as required interfaces. Each interface consists of an operation that accepts a single message (A SoaML MessageType). In SoaML, a MessageType is defined as a unit of information exchanged between participant Request and Service ports via ServiceInterfaces. Instances of a MessageType are passed as parameters in ServiceInterface operations. (Reference: UCS Architecture, Architecture Technical Governance)

To promote commonality across service definitions, a common way of naming services and their sets of operations and messages has been adopted for defining services within UCS-MDE. The convention uses the Service Base Name <SBN> and an optional Function Name [FN] to derive all service names and their associated operations and messages. As this is meant to be a guide, services might not include all of the defined operations and messages and their names might not follow the convention where a more appropriate name adds clarity.

Furthermore, services in UMAA are not required to be defined as indicated in Table 3 when all parts of the service capabilities are required for the service to be meaningful (such as ResourceAllocation).

Additionally, note that for UMAA not all operations defined in UCS-MDE result in a message being published to the DDS bus, e.g., since DDS uses publish/subscribe, most query operations result in a subscription to a topic and do not actually publish the associated request message. In the case of cancel commands, there is no associated implementation of the cancel<SBN>[FN]CommandStatus as it is just the intrinsic response of the DDS dispose function; so, it is essentially a NOOP (no operation) in implementation. The conventions used to define UCS-MDE services are as follows:

Service Name
- <SBN>[FN]Config
- <SBN>[FN]Control
- <SBN>[FN]Specs
- <SBN>[FN]Status OR Report

where the SBN should be descriptive of the task or information provided by the service. Note that the FN is optional and only included if needed to clarify the function of the service. The suffixes Status and Report are interchangeable. If a "Report" is a more appropriate description of the service, it can be used in lieu of "Status".

Table 3: Service Requests and Associated Responses

	Service Requests (Inputs)	**Service Responses (Outputs)**
Config	set<SBN>[FN]Config	report<SBN>[FN]ConfigCommandStatus
	query<SBN>[FN]ConfigAck	report<SBN>[FN]ConfigAck
	query<SBN>[FN]Config	report<SBN>[FN]Config
	cancel<SBN>[FN]Config	report<SBN>[FN]CancelConfigCommandStatus
	query<SBN>[FN]ConfigExecutionStatus	report<SBN>[FN]ConfigExecutionStatus
Control	set<SBN>[FN]	report<SBN>[FN]CommandStatus
	query<SBN>[FN]CommandAck	report<SBN>[FN]CommandAck
	cancel<SBN>[FN]Command	report<SBN>[FN]CancelCommandStatus
	query<SBN>[FN]ExecutionStatus	report<SBN>[FN]ExecutionStatus
Specs	query<SBN>[FN]Specs	report<SBN>[FN]Specs
Status OR Report	query<SBN>[FN]	report<SBN>[FN]

Service Requests (operation:message)

set<SBN>[FN]Config:<SBN>[FN]ConfigCommandType
query<SBN>[FN]Config:<SBN>[FN]ConfigRequestType[1]
set<SBN>[FN]:<SBN>[FN]CommandType
query<SBN>[FN]CommandAck:<SBN>[FN]CommandAckRequestType[1]
cancel<SBN>[FN]Command:<SBN>[FN]CancelCommandType[1]
cancel<SBN>[FN]Config:<SBN>[FN]CancelConfigType[1]
query<SBN>[FN]ExecutionStatus:<SBN>[FN]ExecutionStatusRequestType[1]
query<SBN>[FN]ConfigExecutionStatus:<SBN>[FN]ConfigExecutionStatusRequestType[1]
query<SBN>[FN]ConfigAck:<SBN>[FN]ConfigAckRequestType[1]
query<SBN>[FN]Specs:<SBN>[FN]SpecsRequestType[1]
query<SBN>[FN]:<SBN>[FN]RequestType [1] [2]

Service Responses (operation:message)
report<SBN>[FN]ConfigCommandStatus:<SBN>[FN]ConfigCommandStatusType
report<SBN>[FN]Config:<SBN>[FN]ConfigReportType
report<SBN>[FN]ConfigAck:<SBN>[FN]ConfigAckReportType
report<SBN>[FN]CommandStatus:<SBN>[FN]CommandStatusType
report<SBN>[FN]CommandAck:<SBN>[FN]CommandAckReportType
report<SBN>[FN]CancelCommandStatus:<SBN>[FN]CancelCommandStatusType[1]
report<SBN>[FN]CancelConfigCommandStatus:<SBN>[FN]CancelConfigCommandStatusType[1]
report<SBN>[FN]ExecutionStatus:<SBN>[FN]ExecutionStatusReportType
report<SBN>[FN]ConfigExecutionStatus:<SBN>[FN]ConfigExecutionStatusReportType
report<SBN>[FN]Specs:<SBN>[FN]SpecsReportType
report<SBN>[FN]:<SBN>[FN]ReportType

where,

- Config (Configuration) Command/Report – This is the setup of a resource for operation of a particular task. Attributes may be static or variable. Examples include: maximum RPM allowed, operational sonar frequency range allowed, and maximum allowable radio transmit power.
- Command Status – This is the current state of a particular command (either control or configuration).
- Command – This is the ability to influence or direct the behavior of a resource during operation of a particular task. Attributes are variable. Examples include a vehicle's speed, engine RPM, antenna raising/lowering, and controlling a light or gong.
- Command Ack (Acknowledgement) Report – This is the command currently being executed.
- Cancel – This is the ability to cancel a particular command that has been issued.
- Execution Status Report – This is the status related to executing a particular command. Examples associated with a waypoint command include cross track error, time to achieve, and distance remaining.
- Specs (Specifications) Report – Provides a detailed description of a resource and/or its capabilities and constraints. Attributes are static. Examples include: maximum RPM of a motor, minimum frequency of a passive sonar sensor, length of the unmanned vehicle, and cycle time of a radar.
- Report – This is the current information being provided by a resource. Examples include vehicle speed, rudder angle, current waypoint, and contact bearing.

3.5 Namespace Conventions

Each UMAA service and the messages under the service can be accessed through their appropriate UMAA namespace. The namespace reflects the mapping of a specific service to its parent ICD, and the parent ICD's mapping to the overall UMAA Design Description. For example:

Access the Primitive Driver Control service under Maneuver Operations:

[1] These message types are required for UCS model rules of construction, but are not implemented as messages in the UMAA specification.
[2] At this time, there are no Requests in the specification. This will be the message format when Requests have been added.

UMAA::MO::PrimitiveDriverControl
Access the ContactReport Service under Situational Awareness:
UMAA::SA::ContactReport

The UMAA model uses common data types that are re-used through the model to define service interface topics, interface topics, and other common data topics. These data types are not intended to be directly utilized but, for reference, they can be accessed in the same manner:

Access the common UMAA Status Message Fields:
UMAA::UMAAStatus
Access the common UMAA GeoPosition2D (i.e., latitude and longitude) structure:
UMAA::Common::Measurement::GeoPosition2D

3.6 Cybersecurity

The UMAA standard addressed in this ICD is independent from defining specific measures to achieve Cybersecurity compliance. This UMAA ICD does not preclude the incorporation of security measures, nor does it imply or guarantee any level of Cybersecurity within a system. Cybersecurity compliance will be performed on a program-specific basis and compliance testing is outside the scope of UMAA.

3.7 GUID algorithm

The UMAA standard utilizes the Globally Unique IDentifier (GUID), conforming to the variant defined in RFC 4122 (variant value of 2). Generators of GUIDs may generate GUIDs of any valid, RFC 4122-defined version that is appropriate for their specific use case and requirements. (Reference: A Universally Unique IDentifier (UUID) URN Namespace)

3.8 Large Collections

The UMAA standard defines Large Collections, which are collections of decoupled but related data. Large Collections provide the ability to update one or more elements of the collection without republishing the entire collection to the DDS bus. This avoids two problems related to using an unbounded sequence type in a DDS message: 1) resource consumption growing as the collection is appended to or updated, and 2) DDS implementation-specific limitations on unbounded sequences. There are two implementations of a Large Collection: the Large Set (unordered) and the Large List (ordered).

In both Large Collection implementations, there are two important abstractions: the collection metadata and collection element type. Because Large Collections are specific to the UMAA PSM, the type definitions for the collection metadata and collection element are not part of MDE, and the IDL definitions of these types are generated separately. A particular UMAA message that has a Large Collection attribute will reference the metadata type (LargeSetMetadata or LargeListMetadata). The collection element type is defined under the same namespace as the message that uses it, and follows the naming pattern <parent message name><attribute name><collection type>Element. Each element of the collection is published as a separate message on the DDS bus, and can be tracked back to their related collection using the setID or listID. Users can also trace an element in a set to the source attribute (a NumericGUID) of the Service Provider that generated the report with this set using the collection metadata.

3.8.1 Necessary QoS

To achieve the Large Collection consistency in the update process described below, ordering of samples on the collection element type topic is necessary. Therefore, publishers and subscribers to the collection element type topic must use the `PRESENTATION` QoS policy with an `access_scope` of `DDS_TOPIC_PRESENTATION_QOS` and `ordered_access`.

Note that Large Collection Metadata and Elements are sent on separate DDS topics. DDS QoS does not guarantee ordering across topics. For this reason, implementations must be able to handle cases where elements arrive before or after the associated metadata. Memory must be allocated to await the proper metadata and associated elements.

3.8.2 Creating Large Collections

To create a large collection, a series of element messages and a metadata message must be sent from one DDS participant (the sender) to another (the receiver). The messages should be buffered on the receiving side until a synchronization point is

reached which indicates an atomic update. That is, when both a metadata message and an element message corresponding by list ID, timestamp, and last element ID have been received, yield a complete collection. Figure 4 shows the sequence of exchanges to establish a collection with 3 elements.

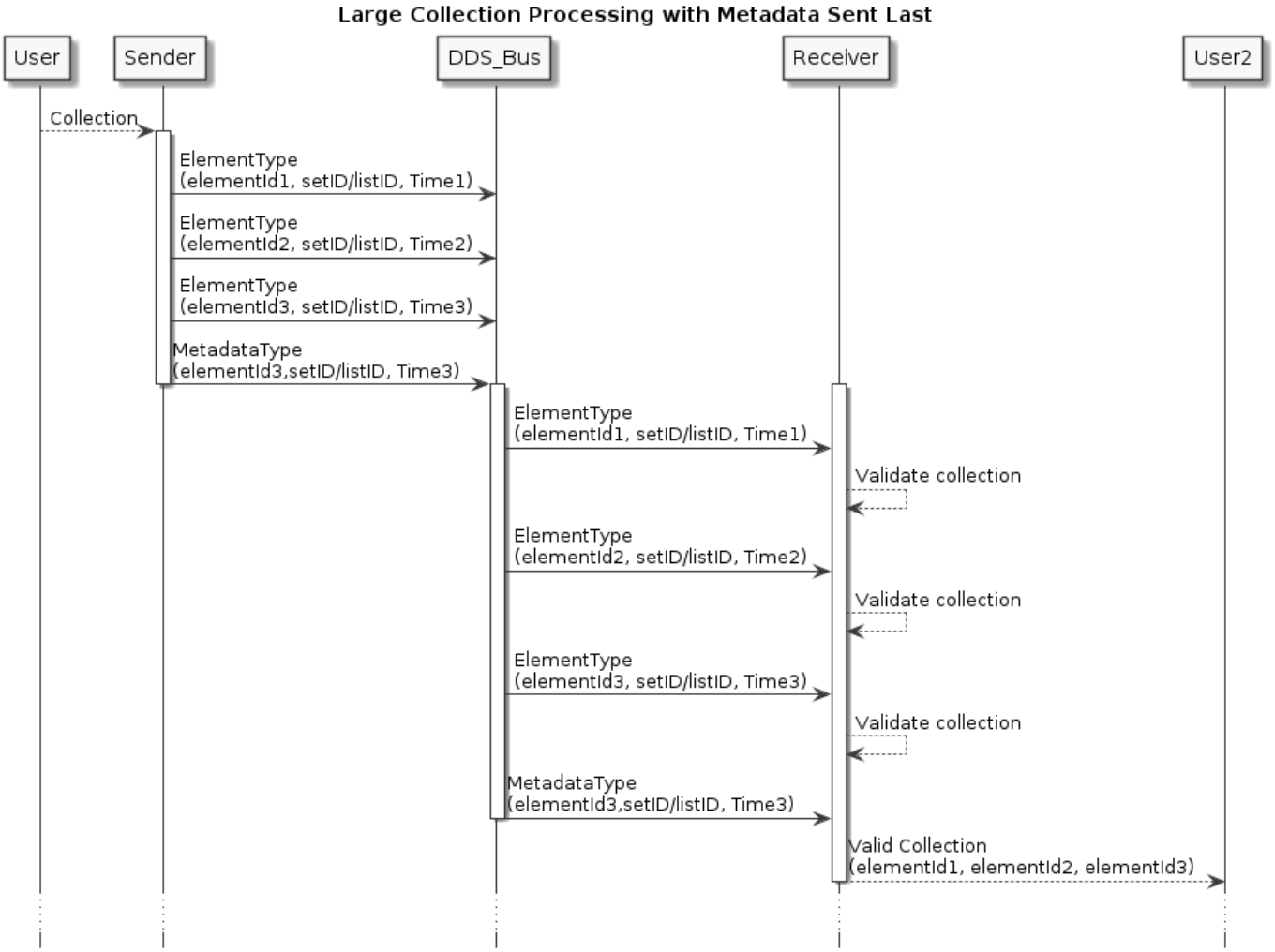

Figure 4: Sequence Diagram for initialization of a Large Collection with 3 elements.

The same collection could be established where the element data arrives after the metadata, creating the same list as depicted in figure 5.

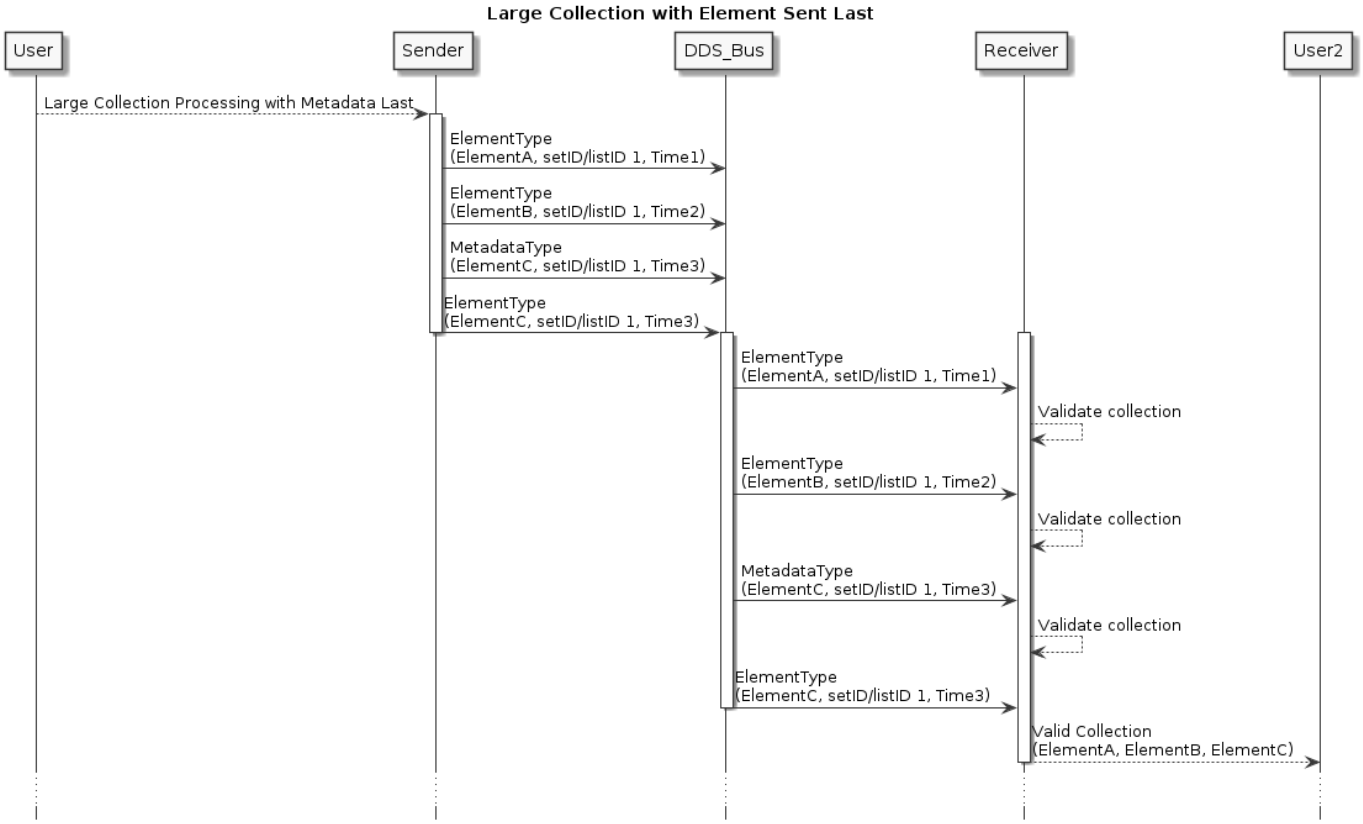

Figure 5: Sequence Diagram for initialization of a Large Collection with 3 elements.

3.8.3 Updating Large Collections

When elements of the collection are updated, the metadata must be updated as well to signal a change in the set. The `updateElementID` is updated to match the elementID of the element whose reception signals the end of the atomic update of the collection. Because of the requirement of an ordered topic described above, this will be the element that is updated last chronologically. The metadata `updateElementTimestamp` must be updated to the timestamp of the same element that signals the end of the update.

The set can be updated as a batch (multiple elements in a single "update cycle," as determined by the provider). This allows for a coarse synchronization: data elements that do not match the metadata `updateElementID` and `updateElementTimestamp` can be assumed to be part of an in-progress update cycle. Consumers can choose to immediately act on those data individually or wait until the matching element is received to signal that the complete update cycle has finished and consider the set as a whole. Note that the coarseness of synchronization is service-dependent: in some cases an intermediate view of a collection update may be logically incorrect to act upon.

Figure 6 shows the sequence of exchanges to update a collection of 3 elements and add a 4th element.

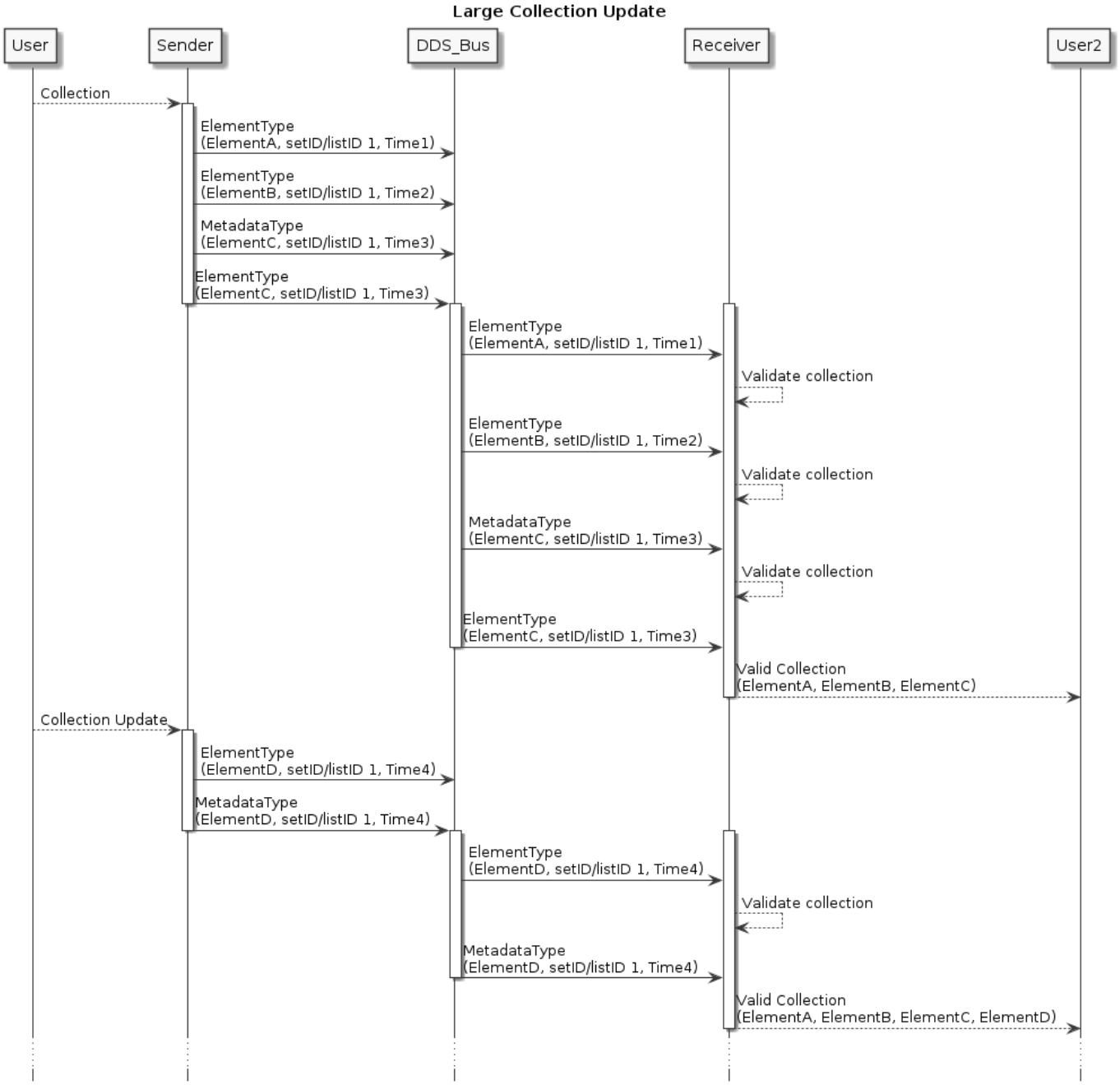

Figure 6: Sequence Diagram for update of Large Collection.

Figure 7 shows the sequence of exchanges to update an element of a collection multiple times.

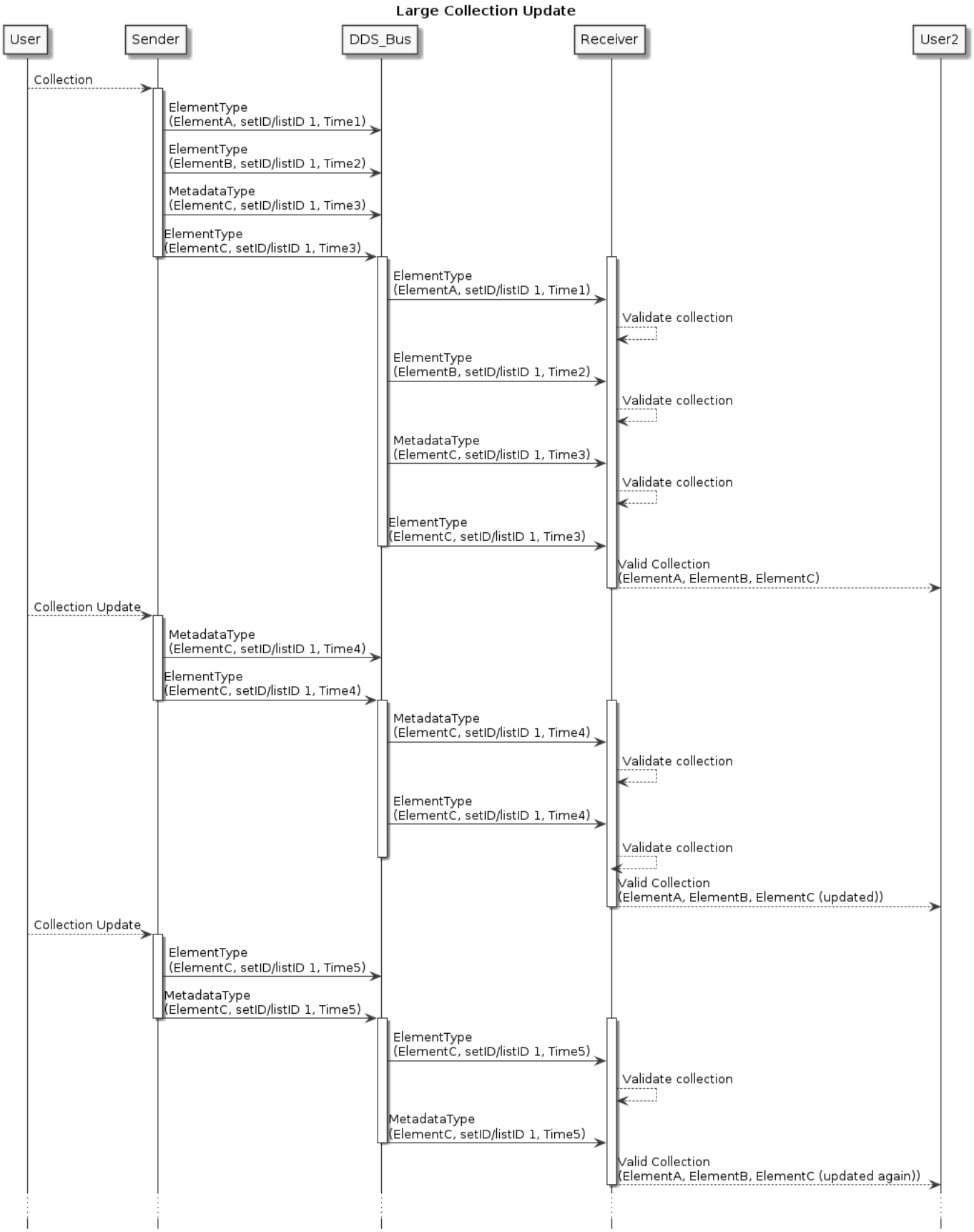

Figure 7: Sequence Diagram for update of an element of a Large Collection multiple times.

3.8.4 Removing an element from Large Collections

To remove an element from a collection, dispose of the element on the element topic and re-publish the metadata. Multiple deletes and inserts can happen for a single metadata update. In the case where the final element of the collection is deleted, the updateElementTimestamp should be unset in the metadata.

Figure 8 shows the sequence of exchanges to delete an element from a Large Collection.

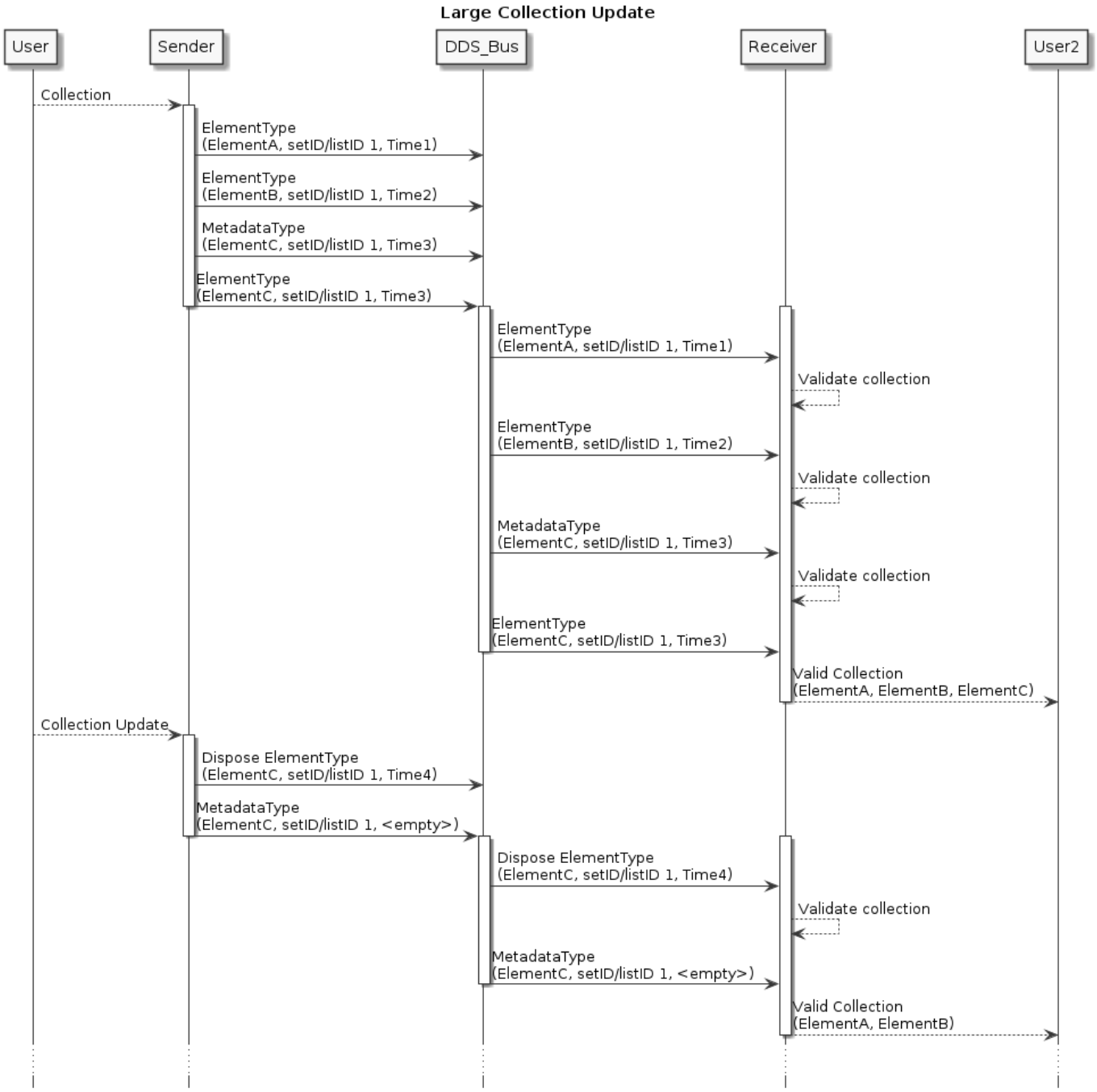

Figure 8: Sequence Diagram for delete of element from Large Collection.

For Large Lists, it may be necessary to update the nextElementID references during delete operations to ensure that the list is still valid. This would cause multiple element messages to be sent along with updated metadata.

3.8.5 Specifying an Empty Large Collection

A particular Large Collection can be empty during initial creation. This is indicated by publishing metadata with a `size` of zero and an `updateElementID` set to the Nil UUID. As specified in section 4.1.7 of the referenced document "A Universally Unique IDentifier (UUID) URN Namespace", this is a "special form of UUID that is specified to have all 128 bits set to zero".

Figure 9 shows the sequence of exchanges to establish an initially empty Large Collection.

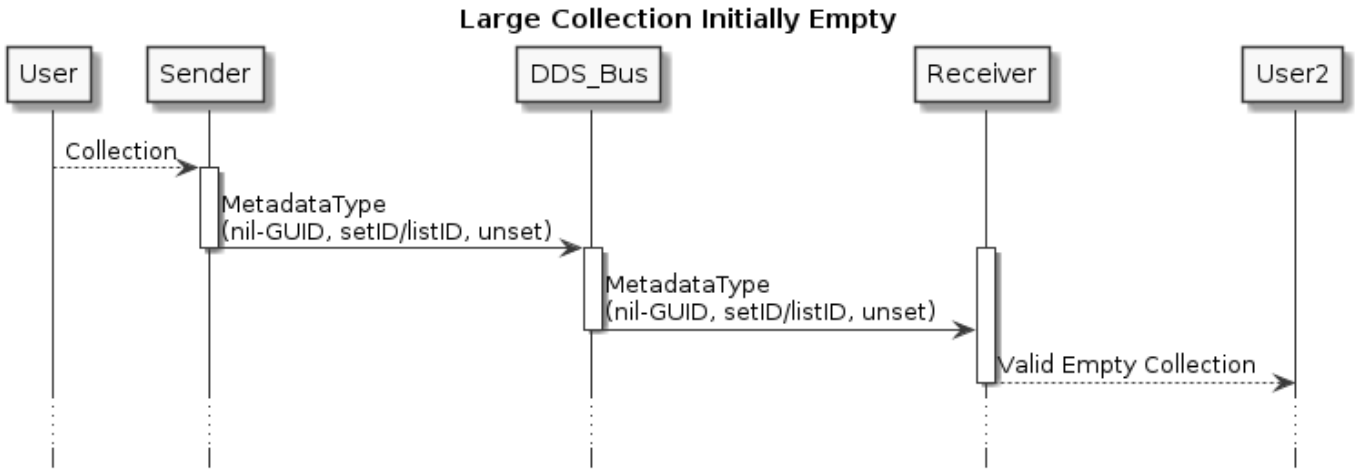

Figure 9: Sequence Diagram for initialization of an empty Large Collection.

3.8.6 Large Set Types

The following details the LargeSetMetadata structure:

Table 4: LargeSetMetadata Structure Definition

Attribute Name	Attribute Type	Attribute Description
setID	NumericGUID	Identifies the Large Set instance this metadata relates to.
updateElementID	NumericGUID	This field references the element ID of the set element whose reception signals the end of an atomic update to this set. This elementID must be used in conjunction with the updateElementTimestamp below to fully identify when the atomic update has completed and the set is stable.
updateElementTimestamp†	DateTime	This field identifies the elementTimestamp of the element, referenced above by updateElementID, that signals the end of an atomic update to this set. This field will be empty in the event that the element update results from a DDS dispose.
size	LargeCollectionSize	Indicates the number of elements associated with this set after the atomic update is complete.

An example element type is shown below, where a `FooReportType` message has a Large Set attribute called "items" whose type is `BarType`

Table 5: Example FooReportTypeItemsSetElement Structure Definition

Attribute Name	Attribute Type	Attribute Description
element	BarType	The value of the set element.
setID*	NumericGUID	Identifies the Large Set instance this element relates to.
elementID*	NumericGUID	Uniquely identifies this element within the set and across all large collection elements that currently exist on the DDS bus.
elementTimestamp	DateTime	The timestamp of this element.

3.8.7 Large List Types

The following details the LargeListMetadata structure:

Table 6: LargeListMetadata Structure Definition

Attribute Name	Attribute Type	Attribute Description
listID	NumericGUID	Identifies the Large List instance this metadata relates to.
updateElementID	NumericGUID	This field references the element ID of the list element whose reception signals the end of an atomic update to this list. This elementID must be used in conjunction with the updateElementTimestamp below to fully identify when the atomic update has completed and the list is stable.
updateElementTimestamp†	DateTime	This field identifies the elementTimestamp of the element, referenced above by updateElementID, that signals the end of an atomic update to this list. This field will be empty in the event that the element update results from a DDS dispose.
startingElementID	NumericGUID	This field identifies the list element, tying to its elementID, that is sequentially first in the list. This is provided for convenience when iterating through the linked list using the nextElementID field.
size	LargeCollectionSize	Indicates the number of elements associated with this set after the atomic update is complete.

An example element type is shown below, where a `FooReportType` message has a Large List attribute called "items" whose type is `BarType`

Table 7: Example FooReportTypeItemsListElement Structure Definition

Attribute Name	Attribute Type	Attribute Description
element	BarType	The value of the list element.
listID*	NumericGUID	Identifies the Large List instance this element relates to.
elementID*	NumericGUID	Uniquely identifies this element within the list and across all large collection elements that currently exist on the DDS bus.
elementTimestamp	DateTime	The timestamp of this element.

Attribute Name	Attribute Type	Attribute Description
nextElementID†	NumericGUID	This field references to the elementID of the element that logically follows this element in the linked list. This is empty if this element is sequentially last.

3.9 Generalizations and Specializations

The UMAA standard makes use of generalization/specialization relationships when defining data types. The generalization/specialization relationship is one where a generalization data structure is defined to contain attributes that are common across some entity and specialization data structures are defined to contain attributes that are specific to a particular type of that entity. This relationship can be modeled as inheritance in UML as shown below.

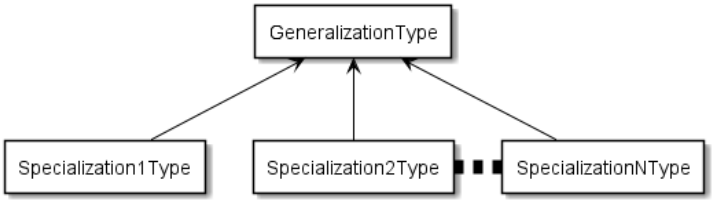

Figure 10: Generalization/Specialization UML diagram.

When the data type of an attribute within a message is a generalization, it is defined to be that generalization plus the data type of one of its specializations. In order to support this relationship, the generalization data structure and its specialization data structure are published to separate topics along with additional metadata linking the two topics. Specifically, the generalization data structure includes: specializationTopic, specializationID, and specializationTimestamp; and the specialization data structure includes: specializationID and specializationTimestamp. The specializationTopic specifies the topic name of the particular specialization, and the specializationID and specializationTimestamp must be equivalent in each topic, respectively, in order to establish the generalization/specialization relationship.

3.9.1 Creating a generalization/specialization

To create a generalization/specialization, both the GeneralizationType and SpecializationType topics must be sent from one DDS participant (the sender) to another (the receiver). The topics should be buffered on the receiving side until a synchronization point is reached that indicates an atomic update.

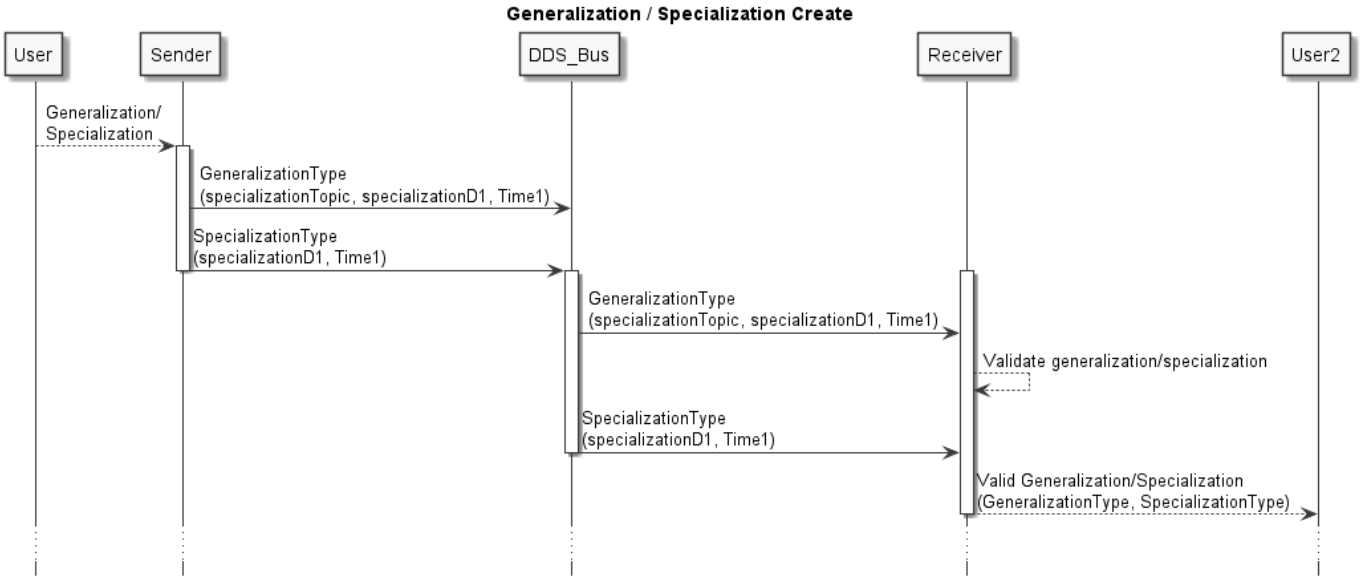

Figure 11: Sequence diagram for creating a generalization/specialization.

3.9.2 Updating a generalization/specialization

An update to a generalization/specialization can occur when there is a change in either data structure. In order for the update to be complete, the specializationTimestamp must be updated in both the GeneralizationType and the SpecializationType, and again they must be equal. Note that if a generalization/specialization exists within a large set or large list that their respective metadata must also be updated as defined in Section 3.8.

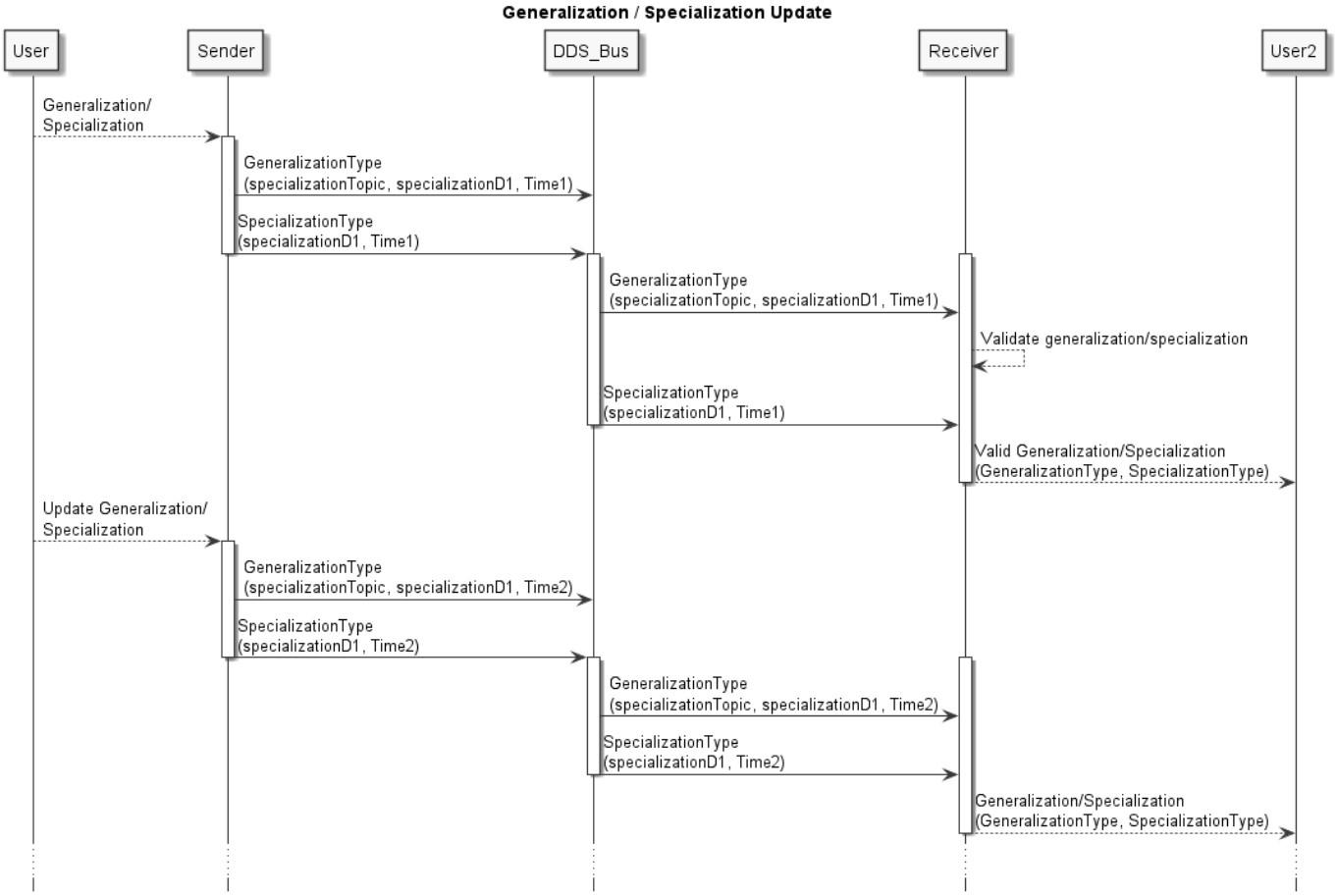

Figure 12: Sequence diagram for updating a generalization/specialization.

3.9.3 Removing a generalization/specialization

To remove a generalization/specialization, both topics must be disposed. Again, note that if a generalization/specialization exists within a large set or large list that their respective metadata must also be updated as defined in Section 3.8.

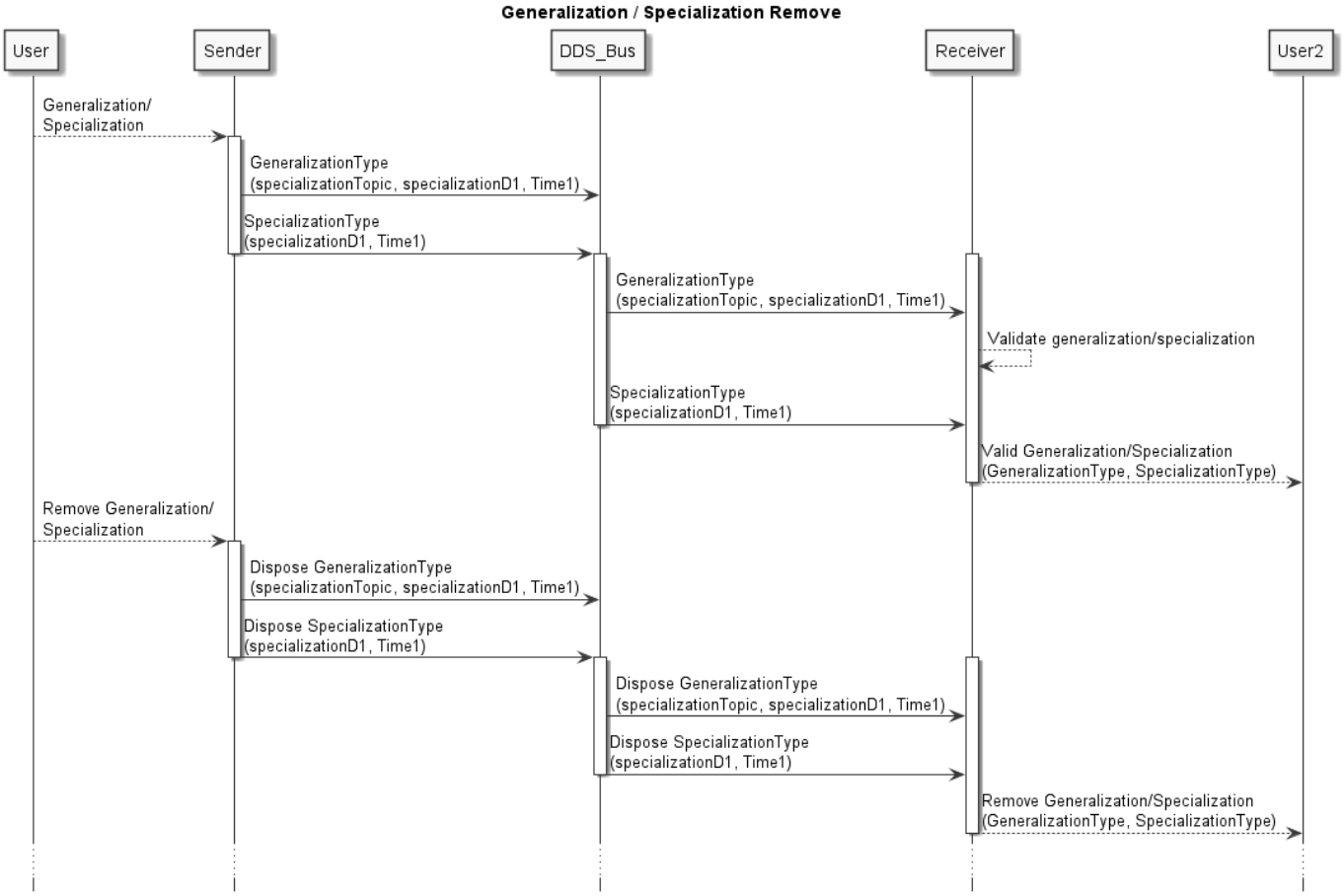

Figure 13: Sequence diagram for removing a generalization/specialization.

4 Introduction to Coordinate Reference Frames and Position Model

4.1 Platform Reference Frame

In the following Service Definitions, we use the parameters yaw, pitch, and roll to define the platform orientation with respect to the specified reference frame. Each parameter is described as a rotation around a given axis: Yaw about the Z axis. Pitch about the Y axis. Roll about the X axis. A UUV is shown in the diagrams because it has more degrees for freedom for its pose and motion, however, the terminology equally applies to both USVs and UUVs.

The axes are defined as:

- X - Positive in the forward direction, negative in the aft.
- Y - Positive in the starboard direction, negative in the port.
- Z - Positive in the down direction, negative in the up.

Additionally, rotations about all axes follow the right-hand rule.

4.2 Earth-Centered Earth-Fixed Frame

The Earth-Centered Earth-Fixed (ECEF) frame is a global reference frame with its origin at the center of the ellipsoid modeling the Earth's surface (Figure 14). The Z-axis points along the Earth's axis of rotation through the North Pole. The X-axis points from the origin to the intersection of the equator with the prime meridian, which defines 0° longitude. The Y-axis completes the right-handed orthogonal system, intersecting the equator at the 90° east meridian.

4.3 North-East-Down Frame

The North-East-Down (NED) frame is defined with an origin at the object described by the navigation solution. The Down axis is defined as normal to the surface of the reference ellipsoid in the direction pointing towards the interior of the Earth. The North axis is the projection of the line from the object to the north pole onto the plane orthogonal to the Down axis. The East axis completes the right-handed orthogonal system and points in the East direction.

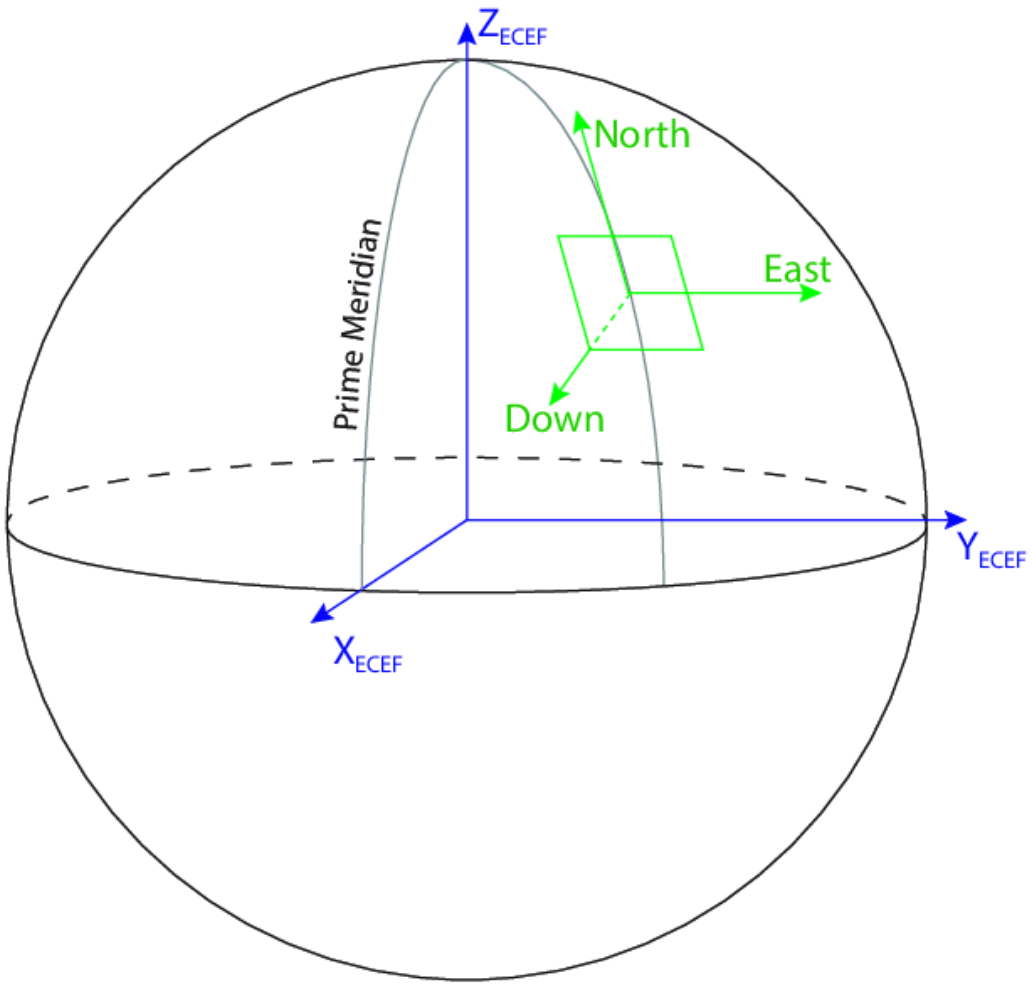

Figure 14: Origins and axes of the Earth-Centered Earth-Fixed (ECEF) and North-East-Down (NED) frames.

4.4 WGS 84

The World Geodetic System (WGS) 1984 defines a standard coordinate system for the Earth. It represents the Earth as an oblate spheroid, and defines the mapping between latitude-longitude-altitude (LLA) coordinates and Cartesian ECEF coordinates. GPS reports positions in WGS 84 LLA coordinates. It has become the standard datum for navigation.

While the UMAA services typically make use of the coordinate systems defined by WGS 84, it also defines an Earth Gravity Model (EGM) and a World Magnetic Model (WMM) which are updated regularly.

4.5 Vehicle Orientation

Determining the orientation of the vehicle (Figure 15) with respect to any reference frame is carried out via the following procedure (Figure 16).

1. Align the vehicle's longitudinal or X axis with the reference frame X axis. In the global reference frame, this is the north direction.
2. Align the vehicle's down or Z axis with the reference frame's Z axis. In the global reference frame, this is the gravity direction.
3. Ensure that the vehicle's transverse or Y axis is aligned with the reference frame's Y axis. In the global reference frame, this is the east direction.
4. Rotate the vehicle about the vehicle's Z axis by the yaw angle (Figure 17).

5. Rotate the vehicle about the vehicle's newly oriented Y axis by the pitch angle (Figure 18).
6. Rotate the vehicle about the vehicle's newly oriented X axis by the roll angle (Figure 19).

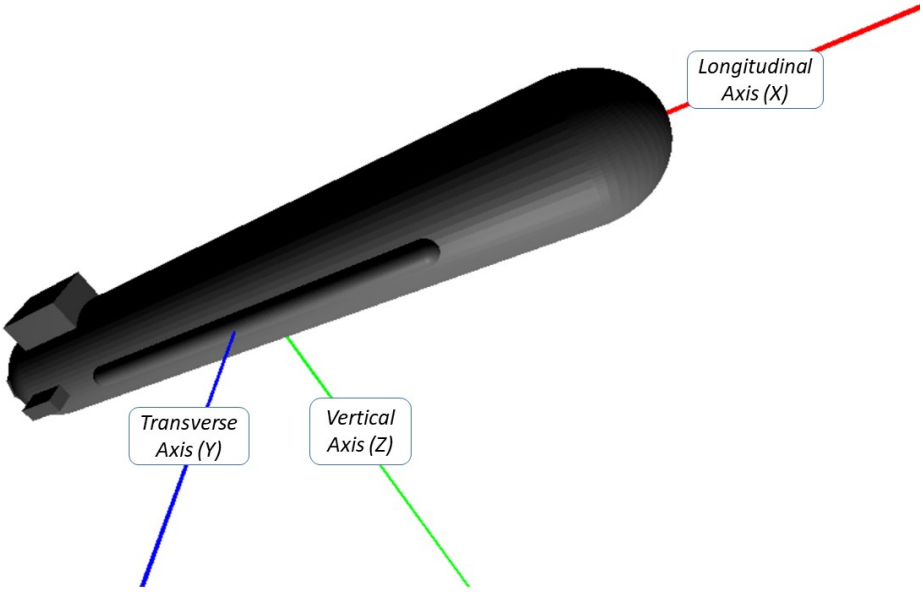

Figure 15: Define the Vehicle Coordinate System

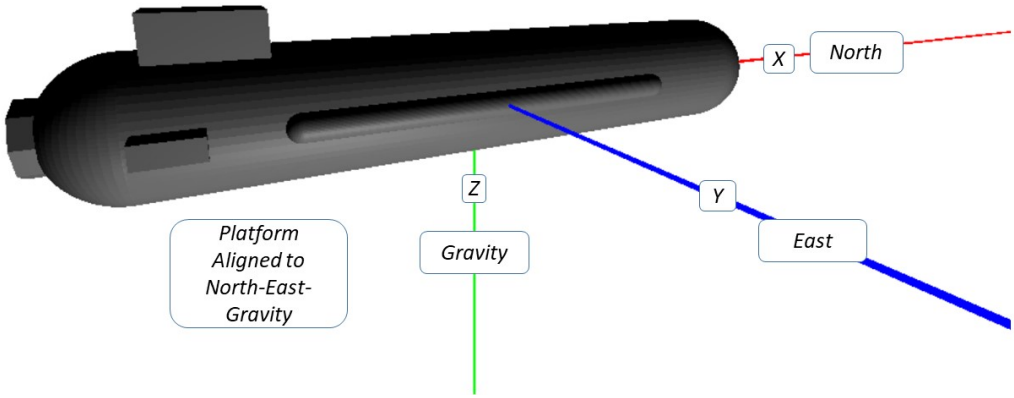

Figure 16: Align the Vehicle with the Reference Frame Axes.

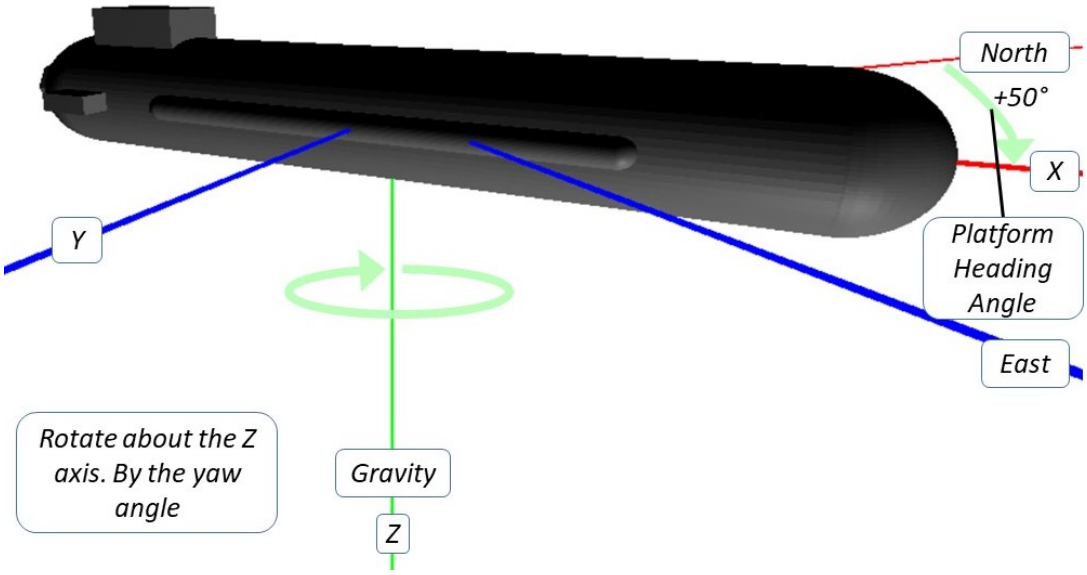

Figure 17: Rotate the Vehicle by the Yaw Angle.

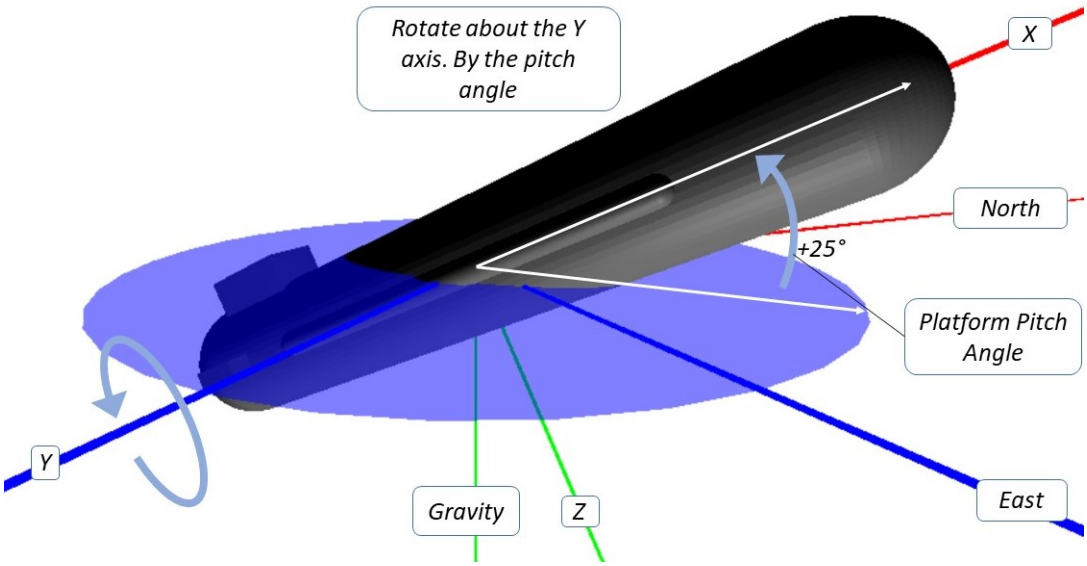

Figure 18: Rotate the Vehicle by the Pitch Angle.

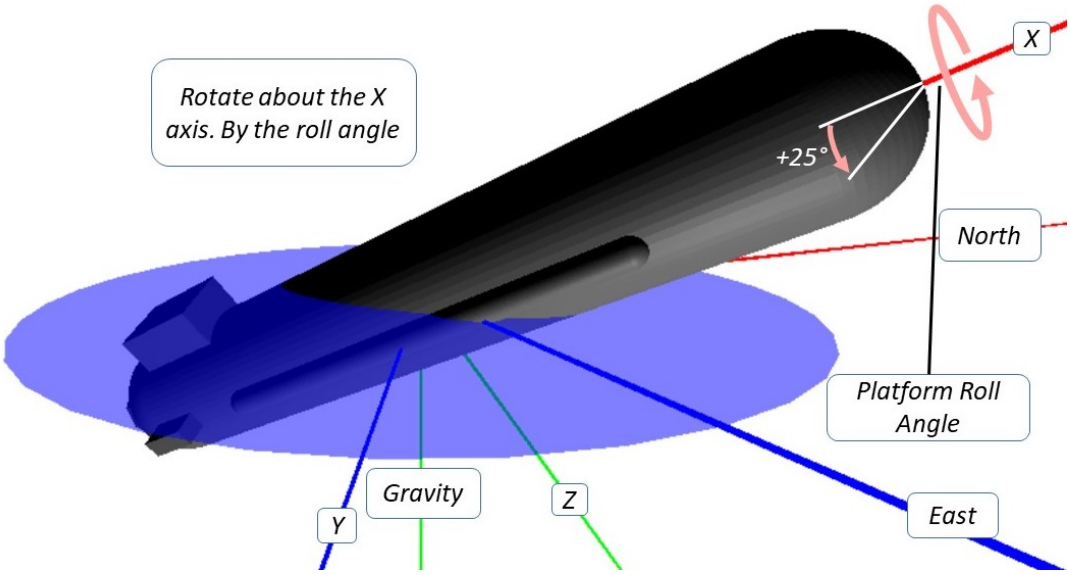

Figure 19: Rotate the Vehicle by the Roll Angle.

4.6 Vehicle Coordinate Reference Frame Origin

UMAA does not specify a required origin for the vehicle coordinate reference frame. However, certain applications may benefit from defining a specific origin such as the registration of multiple sensors with associated offsets for data fusion. Possible origins include the keel/transom intersection, bow/waterline intersection, center of gravity, center of buoyancy and location of INS. A few examples follow.

Definitions

- Keel Transom Intersection
  - Beam at Waterline (BWL) - The maximum distance of the vehicle at the waterline, the distance along the Y axis of the widest point of the hull where it meets the waterline.
  - Design Waterline (DWL) - The line representing the waterline on the vehicle at designed load in summer temperature.
  - Keel - The principal fore-and-aft component of a ship's framing, located along the centerline of the bottom and connected to the stem and stern frames.
  - Length at Waterline (LWL) - The measured distance of the vehicle at the level where it sits in the water, measured along the X axis.
  - Transom - The aftermost transverse flat or shaped plating enclosing the hull.

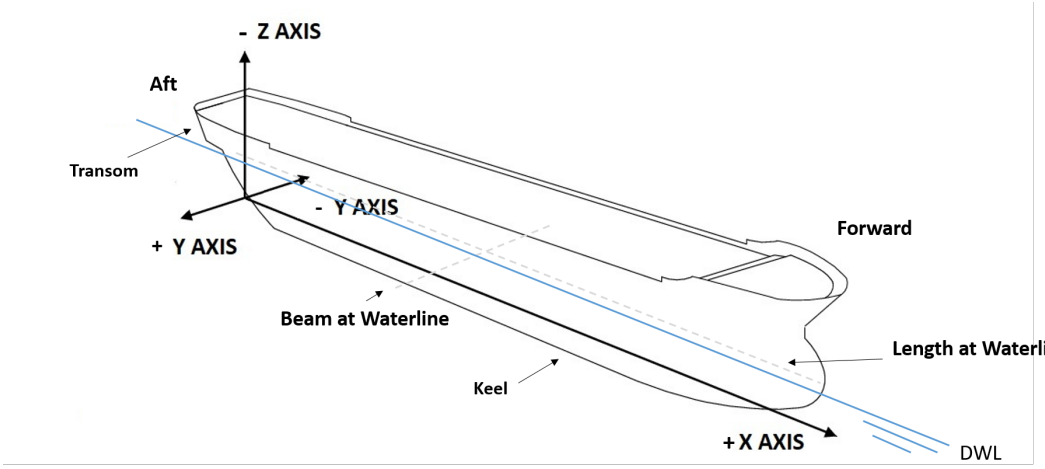

Figure 20: Keel Transom Intersection Origin Location on a USV as Example

- Center of Buoyancy
  - X - The Longitudinal Center of Buoyancy (LCB) when fully submerged.
  - Y - The symmetrical centerline.
  - Z - The Vertical Center of Buoyancy (VCB) when fully submerged.

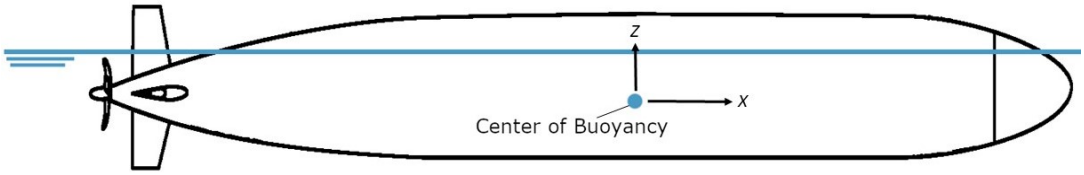

Figure 21: Center of Buoyancy Origin Location on a UUV as Example.

5 Flow Control

5.1 Command / Response

This section defines the flow of control for command/response over the DDS bus. A command/response controls a specific service. While the exact names and processes will depend on the specific service and command being executed, all command/responses in UMAA follow a similar pattern. A notional "Function" command `FunctionCommand` is used in the following examples. As will be described in subsequent paragraphs, DDS publish/subscribe methods are used in implementations to issue commands and responses.

To direct a `FunctionCommand` at a specific Service Provider, UMAA includes a `destination` GUID in all commands. A Service Provider is required to respond to all `FunctionCommand`s where the `destination` is the same as the Service Provider's ID. The Service Consumer will also create a `sessionID` for the command when commanded. The `sessionID` is used to track the command execution as a key into other command-related messages. The `sessionID` must be unique across all `FunctionCommand` instances that are active (i.e. currently on the DDS bus), otherwise the Service Provider will consider the `FunctionCommand` to be a command update (see Section 5.1.4.2). Once a `FunctionCommand` is removed from the DDS bus as part of the Command Cleanup process (see Section 5.1.5), its `sessionID` may be reused for future commands without triggering a command update; therefore it is not necessary for a Service Provider to maintain a complete history of `sessionID`s.

Service Provider and Service Consumer terminology in the following sections is adopted from the OMG Service-oriented architecture Modeling Language (SoaML).

To initialize, a Service Provider (controllable resource) subscribes to the `FunctionCommand` DDS topic. At startup or right before issuing a command, the Service Consumer (controlling resource) subscribes to the `FunctionCommandStatus` DDS topic. Optionally, the Service Consumer may also subscribe to the `FunctionCommandAckReport` to monitor which command is currently being executed, and the `FunctionExecutionStatusReport` (if defined for the Function service) that provides reporting on function-specific data status.

Both Service Providers and Service Consumers are required to recover or clean up any previous persisted commands on the bus during initialization.

To execute a command, the Service Consumer publishes a `FunctionCommandType` to the DDS bus. The Service Provider will be notified and will begin processing the request. During each phase of processing, the Service Provider will provide updates to the Service Consumer via published updates to a related `FunctionCommandStatus` topic. Command responses are correlated to their originating command via the `sessionID`. If a command with a duplicate `sessionID` is received, the Service Provider will regard this as a command update, and follow the flow control detailed in Section 5.1.4.2. Command status updates are provided in the command responses via the `commandStatus` field with additional details included in the `commandStatusReason` field. The Service Provider will also publish the current executing command to the `FunctionCommandAckReport` topic. When defined for the Function service, the Service Provider must also publish the `FunctionExecutionStatusReport` topic and update it as appropriate throughout the execution of the command.

The required state transitions for the `commandStatus` field are shown in Figure 22. Commands may complete normally, or they may terminate early due to failure (Section 5.1.4.4) or cancellation (Section 5.1.4.5). The state machine for a command can also be reset to `ISSUED` via a command update (Section 5.1.4.2). If there is not a self-transition indicated in the diagram, you cannot republish that state in a message. Every command must transition through the states as defined. For example, it is a violation to transition from `ISSUED` to `EXECUTING` without transitioning through `COMMANDED`. Even in the case where there is no logic executing between the `ISSUED` and `EXECUTING` states, the Service Provider is required to transition through `COMMANDED`. This ensures consistent behavior across different Service Providers, including those that do require the `COMMANDED` state.

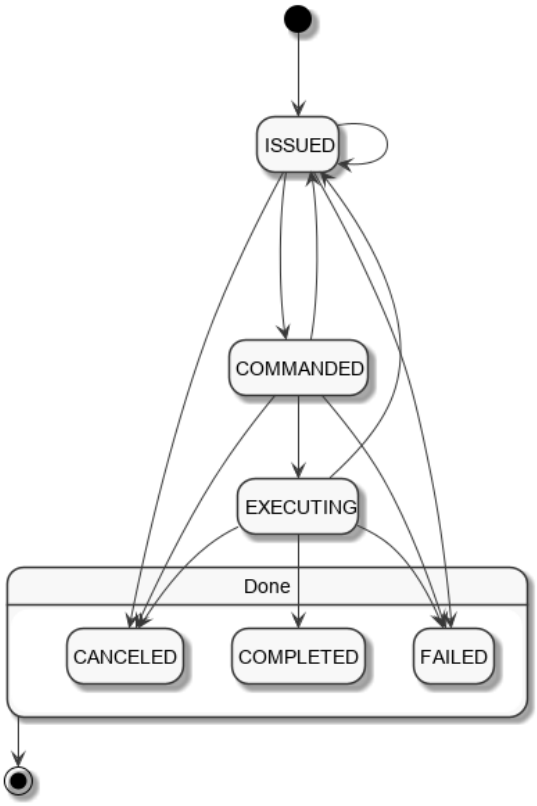

Figure 22: State transitions of the `commandStatus` as commands are processed.

As described above, each time a command transitions to a new state, a `FunctionCommandStatus` message is published containing the updated `commandStatus` and a `commandStatusReason` that indicates why the state transition happened. The table below shows all valid `commandStatusReason` values for each `commandStatus` transition.

	Ending State					
Starting State	ISSUED	COMMANDED	EXECUTING	COMPLETED	FAILED	CANCELED
Initial State	SUCCEEDED	—	—	—	—	—
ISSUED	UPDATED	SUCCEEDED	—	—	VALIDATION_FAILED RESOURCE_FAILED INTERRUPTED TIMEOUT SERVICE_FAILED	CANCELED
COMMANDED	UPDATED	—	SUCCEEDED	—	RESOURCE_REJECTED INTERRUPTED TIMEOUT SERVICE_FAILED	CANCELED
EXECUTING	UPDATED	—	—	SUCCEEDED	OBJECTIVE_FAILED RESOURCE_FAILED INTERRUPTED TIMEOUT SERVICE_FAILED	CANCELED
COMPLETED	—	—	—	—	—	—
FAILED	—	—	—	—	—	—
CANCELED	—	—	—	—	—	—

Figure 23: Valid `commandStatusReason` values for each `commandStatus` state transition. Entries marked with a (—) indicate that the state transition is invalid.

In the following sections, the sequence diagrams demonstrate different exchanges between a Service Consumer and Service Provider. Within the diagrams, the dashed arrows represent implementation-specific communications that are outside of UMAA's scope. These sequence diagrams are just an example of one possible implementation. Other implementations may have different communication patterns between the Service Provider and the Resource or be implemented completely within the Service Provider process itself (no dependency on an external Resource). Likewise, the interactions between the User and Service Consumer may follow similar or different patterns. However, the UMAA-defined exchanges with the DDS bus between the Service Consumer and Service Provider must happen in the order shown within the sequence diagrams.

5.1.1 High-Level Flow

The high-level flow of a command sequence is shown in Figure 24 and can be described as follows:

1. The Command Startup Sequence is performed.
2. For each command to be executed:
   (a) The Command Start Sequence is performed.
   (b) The command is executed (sequence depends on the execution path, i.e., success, failure, or cancel).
   (c) The Command Cleanup Sequence is performed.
3. The Command Shutdown Sequence is performed.

The `ref` blocks will be defined in later sequence diagrams. Note that the duration of the system execution for any particular `FunctionCommandType` is defined by the combination of the Service Provider(s) and Service Consumer(s) in the system and may not be identical to the overall system execution duration. For example, providers may only be available to execute certain commands during specific mission phases or when certain hardware is in specific configurations. This Command Startup Sequence is not required to happen during a system startup phase. The only requirement is that it must be completed by at least one Service Provider and one Service Consumer before any `FunctionCommandType` commands can be fully executed. Likewise, the Command Shutdown sequence may occur at any time the `FunctionCommandType` will no longer be supported. There is no requirement stating that the Command Shutdown Sequence only be performed during a system shutdown phase.

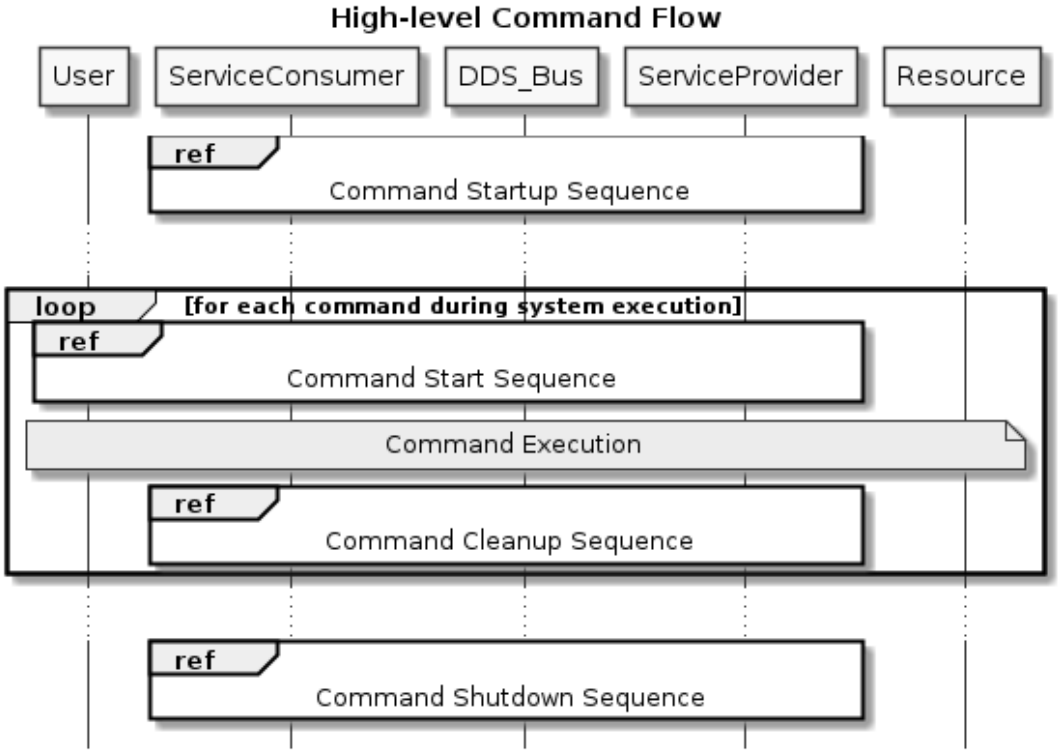

Figure 24: Sequence Diagram for the High-Level Description of a Command Execution.

5.1.2 Command Startup Sequence

As part of initialization both the Service Provider and Service Consumer are required to perform a startup sequence. This startup prepares the Service Provider to execute commands and the Service Consumer to request commands and monitor the progress of those requested commands.

The Service Provider and Service Consumer can initialize in any order. Commands will not be completely executed until both have completed their initialization. The sequence diagram is shown in Figure 25.

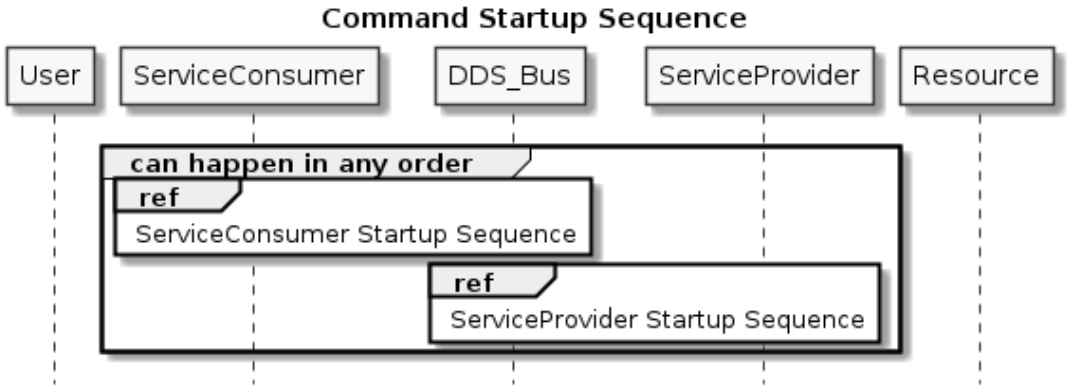

Figure 25: Sequence Diagram for Command Startup.

5.1.2.1 Service Provider Startup Sequence During startup, the Service Provider is required to register as a publisher to the `FunctionCommandStatus`, `FunctionCommandAckReport`, and (if defined for the Function service) the `FunctionExecutionStatusReport` topics.

The Service Provider is also required to subscribe to the `FunctionCommand` topic to be notified when new commands are published.

Finally, the Service Provider is required to handle any existing `FunctionCommandType` commands persisted on the DDS bus with the Service Provider's ID. For each command, if the Service Provider can and wishes to recover, it can continue to execute the command. To obtain the last published state of the command, the Service Provider must subscribe to the `FunctionCommandStatusType`. The Service Provider will continue following the normal status update sequence, picking up from the last status on the bus. If the Service Provider cannot or chooses not to continue processing the command, it must fail the command by publishing a `FunctionCommandStatus` with a `commandStatus` of `FAILED` and a `reason` of `SERVICE_FAILED`.

The Service Provider Startup sequence is shown in Figure 26.

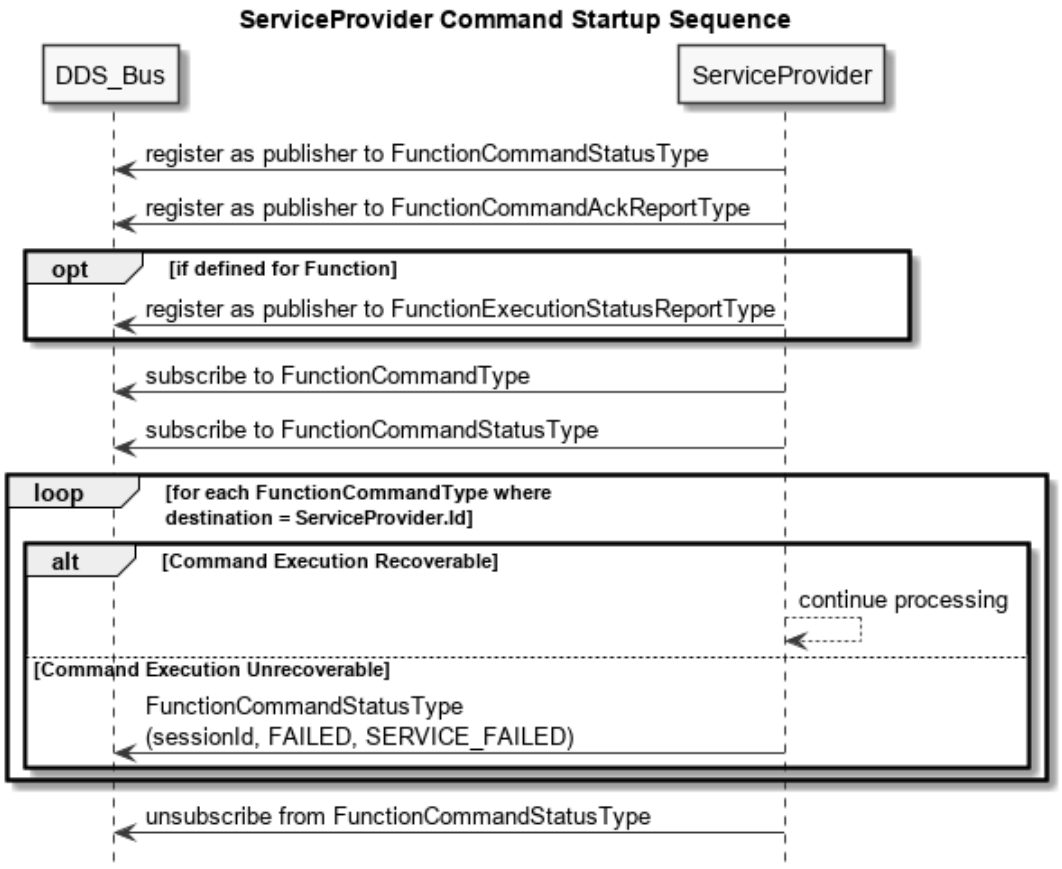

Figure 26: Sequence Diagram for Command Startup for Service Providers.

5.1.2.2 Service Consumer Startup Sequence During startup, the Service Consumer is required to register as a publisher of the `FunctionCommandType`.

The Service Consumer is also required to subscribe to the `FunctionCommandStatusType` to monitor the execution of any published commands. The Service Consumer can optionally register for the `FunctionCommandAckReportType` and, if defined for the Function service, the `FunctionExecutionStatusReportType` if it desires to track additional status of the execution of commands.

Finally, the Service Consumer is required to handle any existing `FunctionCommandType` commands persisted on the DDS bus with this Service Consumer's ID. To find existing `FunctionCommandType`s on the bus, it must first subscribe to the topic. If the Service Consumer can and wishes to recover, it can continue to monitor the execution of the command. If the Service Consumer cannot or chooses not to continue the execution of the command, it must cancel the command via the normal command cancel method.

The Service Consumer Startup sequence is shown in Figure 27.

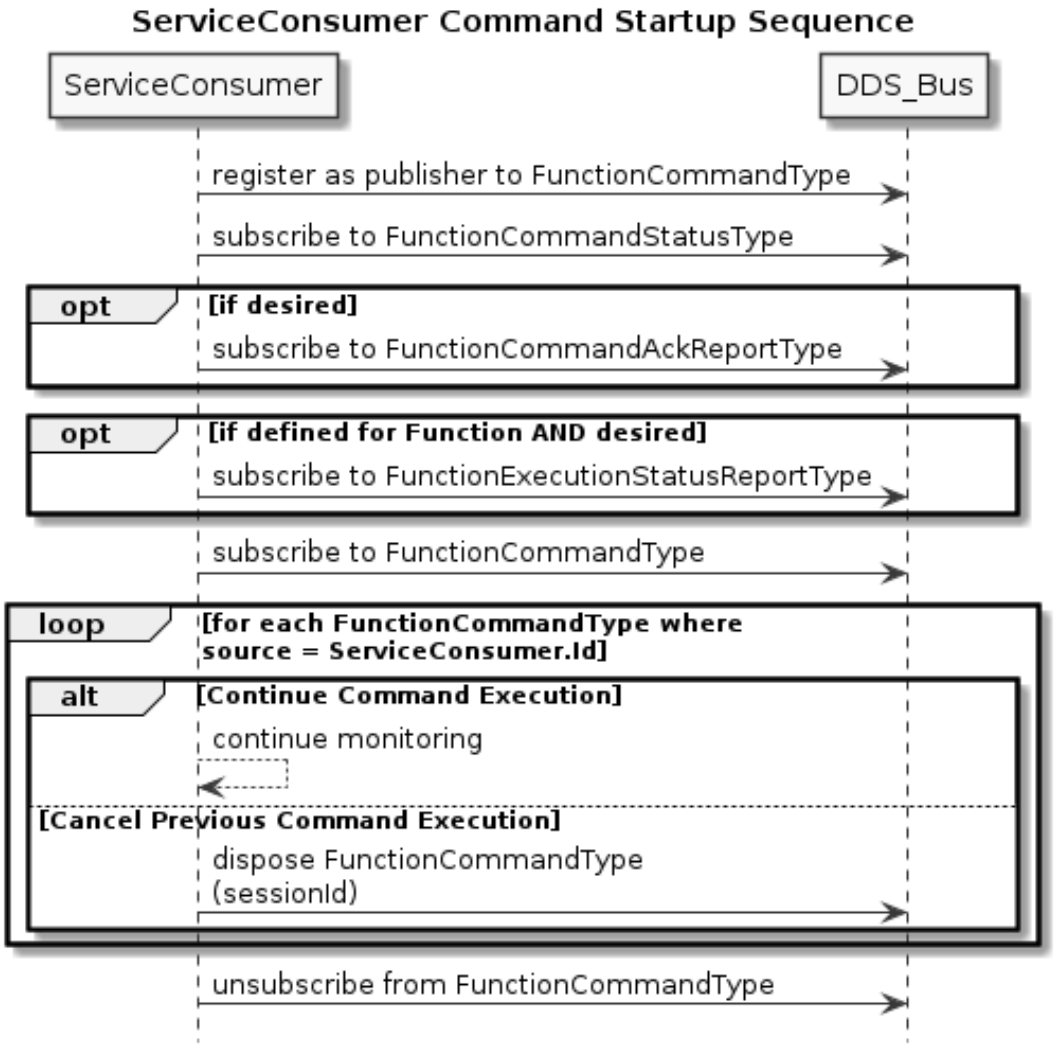

Figure 27: Sequence Diagram for Command Startup for Service Consumers.

5.1.3 Command Execution Sequences

Once both the Service Provider and Service Consumer have performed the startup sequence, the system is ready to begin issuing and executing commands.

5.1.4 Command Start Sequence

The initial start sequence to execute a single new command follows this pattern:

1. The User of the Service Consumer issues a request for a command to be executed.
2. The Service Consumer publishes the `FunctionCommandType` with a unique session ID, the source ID of the Service Consumer, and the destination ID of the desired Service Provider.
3. The Service Provider, upon notification of the new `FunctionCommandType`, publishes a new `FunctionCommandStatusType` with (1) the same session ID as the new `FunctionCommandType`, (2) the status of `ISSUED` and (3) the reason of `SUCCEEDED` to notify the Service Consumer it has received the new command.

The Command Start Sequence for a new command is shown in Figure 28. This pattern will be repeated each time a new command is requested. Note that the Command Start Sequence differs if the `FunctionCommandType` has a `sessionID` that matches another `FunctionCommandType` that currently exists on the DDS bus. This is considered a command update and detailed in Section 5.1.4.2.

After the Command Start Sequence, the sequence can take different paths depending on the actual execution of the command, detailed from Section 5.1.4.1 to Section 5.1.4.5, but they do not enumerate all of the possible execution paths. Other paths (e.g., an objective failing) will follow a similar pattern to other failures; all are required to follow the state diagram shown in Figure 22 and eventually end with the Command Cleanup Sequence (shown in Figure 35).

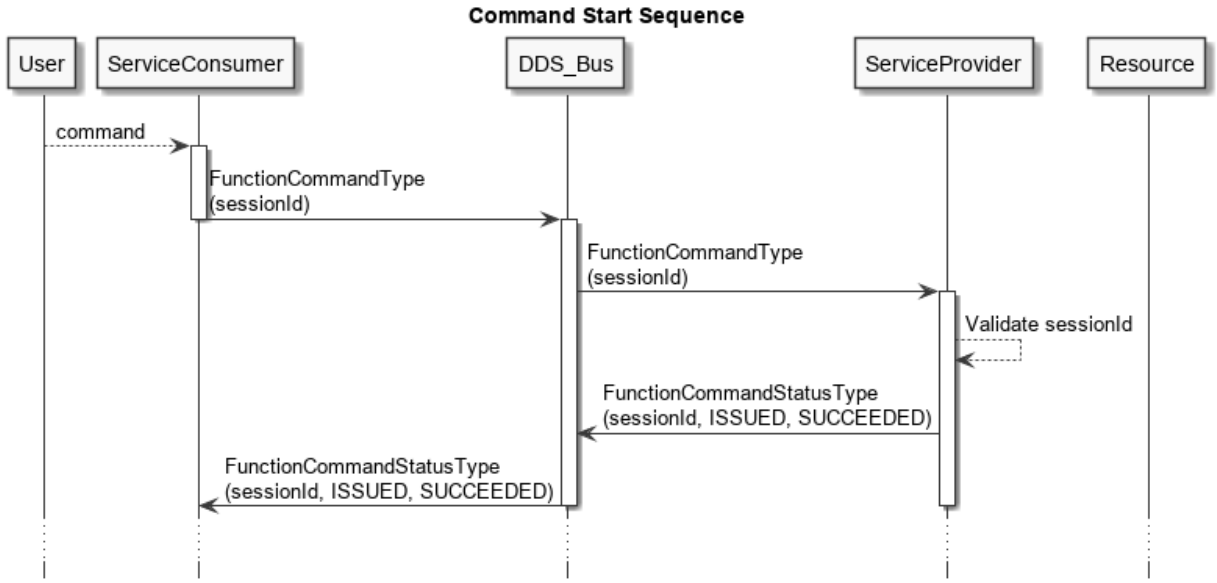

Figure 28: Sequence Diagram for the Start of a Command Execution.

5.1.4.1 Command Execution Once a Service Provider starts to process a command, the Command Execution sequence is:

1. The Service Provider publishes a `FunctionCommandAckReportType` with matching session ID and parameters as the `FunctionCommandType` it is starting to process.
2. The Service Provider performs any validation and negotiation with backing resources as necessary. Once the command is ready to be executed, the Service Provider publishes a `FunctionCommandStatusType` with a status `COMMANDED` and reason `SUCCEEDED` to notify the Service Consumer that the command has been validated and commanded to start execution.
3. Once the command has begun executing, the Service Provider publishes a `FunctionCommandStatusType` with a status `EXECUTING` and reason `SUCCEEDED` to notify the Service Consumer that the command has been validated and commanded to start.
4. If the Function has a defined `FunctionExecutionStatusReportType`, the Service Provider must publish a new instance with matching session ID as the associated `FunctionCommandType`. The `FunctionExecutionStatusReportType` must be updated by the Service Provider throughout the execution as dictated by the definitions of the command-specific attributes in the execution status report.

The command execution sequence is shown in Figure 29. This sequence holds until the command completes execution.

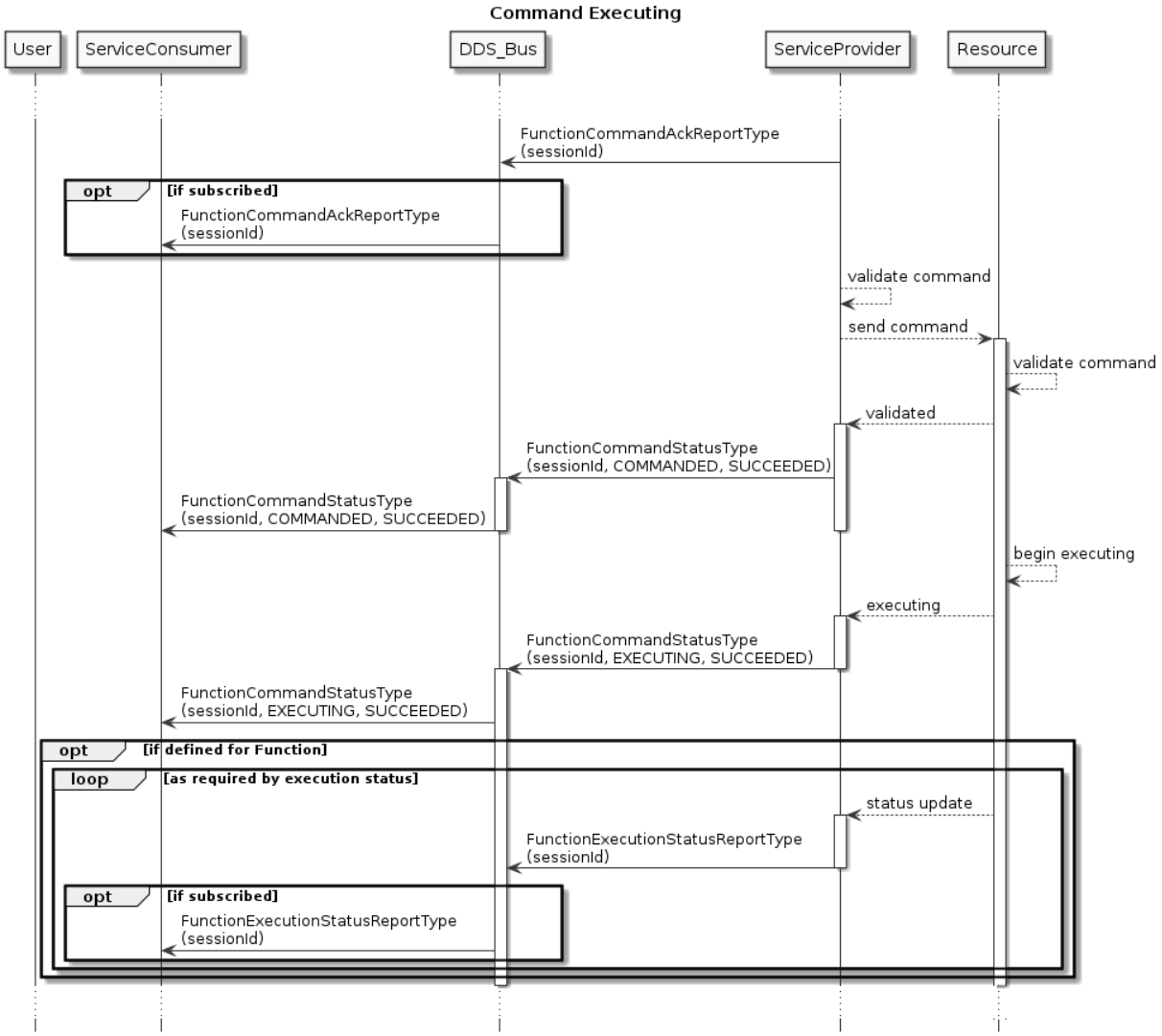

Figure 29: Beginning Sequence Diagram for a Command Execution.

The normal successful conclusion of a command being executed in some cases is initiated by the Service Consumer (an endless GlobalVector command concluded by canceling it) and in other cases is initiated by the Service Provider (a GlobalWaypoint commanded concluded by reaching the last waypoint). Unless otherwise explicitly stated, it is assumed the Service Provider will be able to identify the successful conclusion of a command. In the cases where commands are defined to be indeterminate the Service Consumer must cancel the command when the Service Consumer no longer desires the command to be executed.

5.1.4.2 Updating a Command An updated command is defined as a command with a source ID and session ID identical to the current command being processed by the Service Provider, but whose timestamp is newer than the current command. Only a command that is in a non-terminal state may be updated - otherwise, the Service Consumer must follow the normal command cleanup process and issue a new command with an updated unique session ID. If a command is in a terminal state, the Service Provider must ignore an update to that command.

When the Service Provider receives an updated command, it is required to take one of two possible actions:

1. If the current command is in a non-terminal state (`ISSUED`, `COMMANDED`, or `EXECUTING`), then the Service Provider publishes a `FunctionCommandStatusType` with a status `ISSUED` and reason `UPDATED`. The state machine then restarts and proceeds through the normal command flow detailed in 5.1.4. The Service Provider must consider the updated command as an entirely new command, resetting any internal state related to the command (e.g. a timer that keeps track of command timeout).

2. If the current command is in a terminal state (`COMPLETED`, `CANCELED`, or `FAILED`), then the updated command cannot be processed, and the Service Provider must publish a `FunctionCommandStatusType` with a status `FAILED` and follow the normal command cleanup process.

The flow control for command update is detailed below:

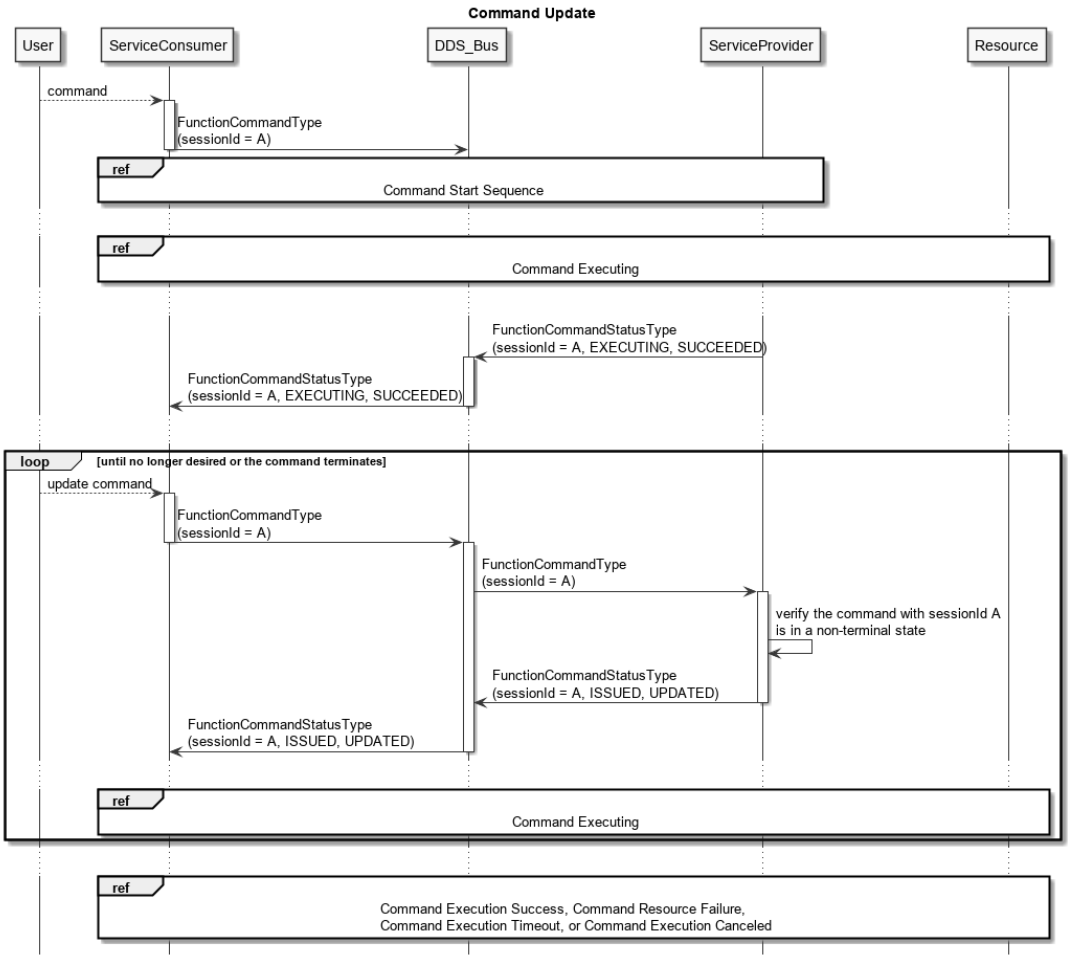

Figure 30: Sequence Diagram for Command Update.

5.1.4.3 Command Execution Success When the Service Provider determines a command has successfully completed, it must update the associated `FunctionCommandStatusType` with as status of `COMPLETED` and reason of `SUCCEEDED`. This signals to the Service Consumer that the command has completed successfully.

The Command Execution Success sequence is shown in Figure 31.

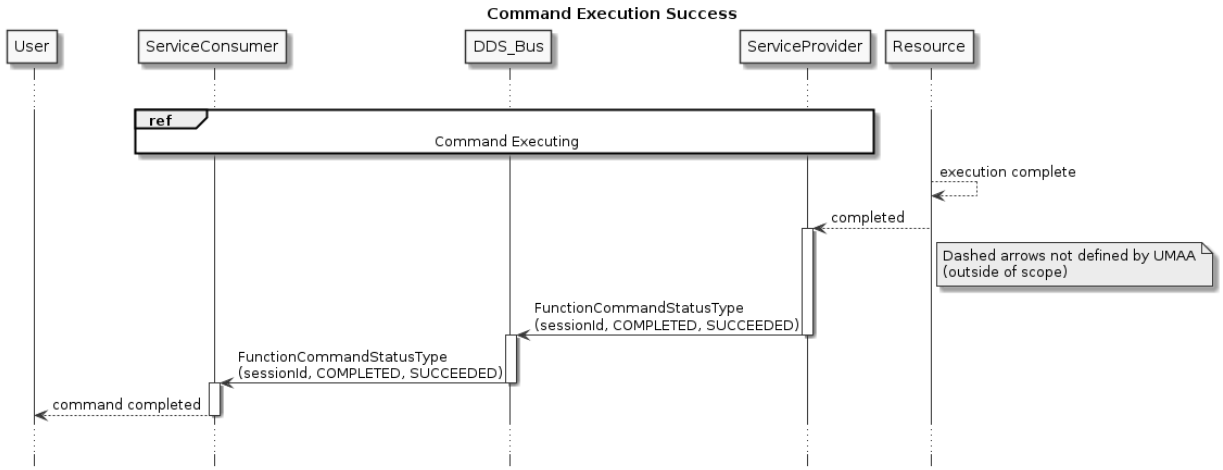

Figure 31: Sequence Diagram for a Command That Completes Successfully.

5.1.4.4 Command Execution Failure The command may fail to complete for any number of reasons including software errors, hardware failures, or unfavorable environmental conditions. The Service Provider may also reject a command for a number of reasons including inability to perform the task, malformed or out of range requests, or a command being interrupted by a higher priority process. In all cases, the Service Provider must publish a `FunctionCommandStatusType` with an identical `sessionID` as the originating `FunctionCommandType` with a status of `FAILED` and the reason that reflects the cause of the failure (`VALIDATION_FAILED, SERVICE_FAILED, OBJECTIVE_FAILED,` etc).

Figure 32 and Figure 33 provide examples where a command has failed.

In the first example, the backing Resource failed and the Service Provider is unable to communicate with it. In this case, the Service Provider will report a `FunctionCommandStatusType` with a status of `FAILED` and a reason of `RESOURCE_FAILED`. This is shown in Figure 32.

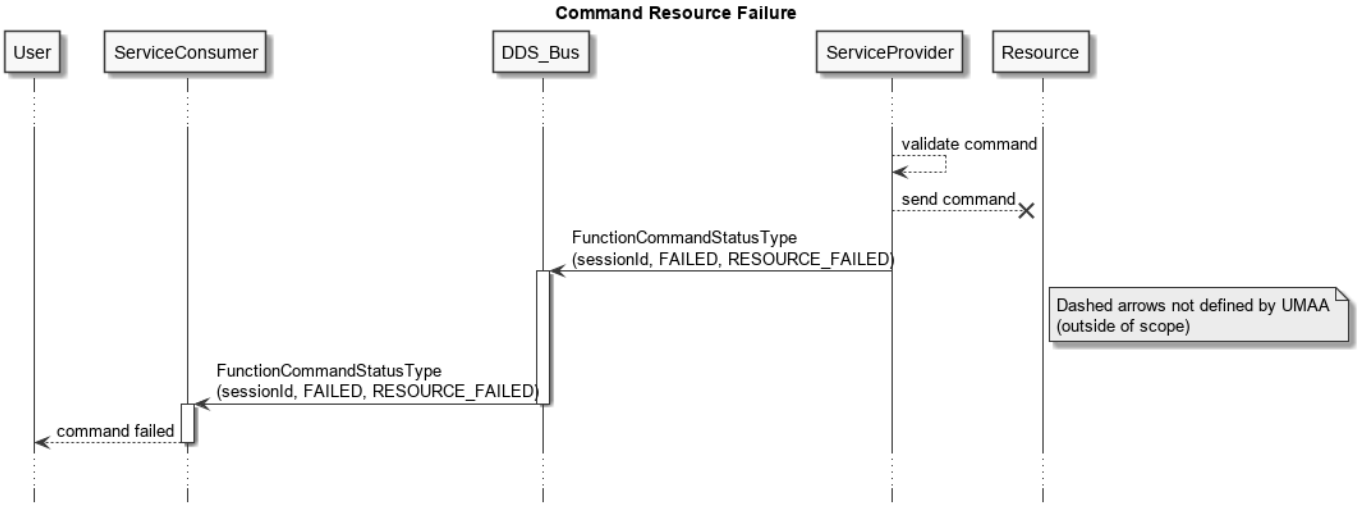

Figure 32: Sequence Diagram for a Command That Fails due to Resource Failure.

In the second example, the Resource takes too long to respond, so the Service Provider cancels the request and reports a `FunctionCommandStatusType` with a status of `FAILED` and a reason of `TIMEOUT`. This is shown in Figure 33.

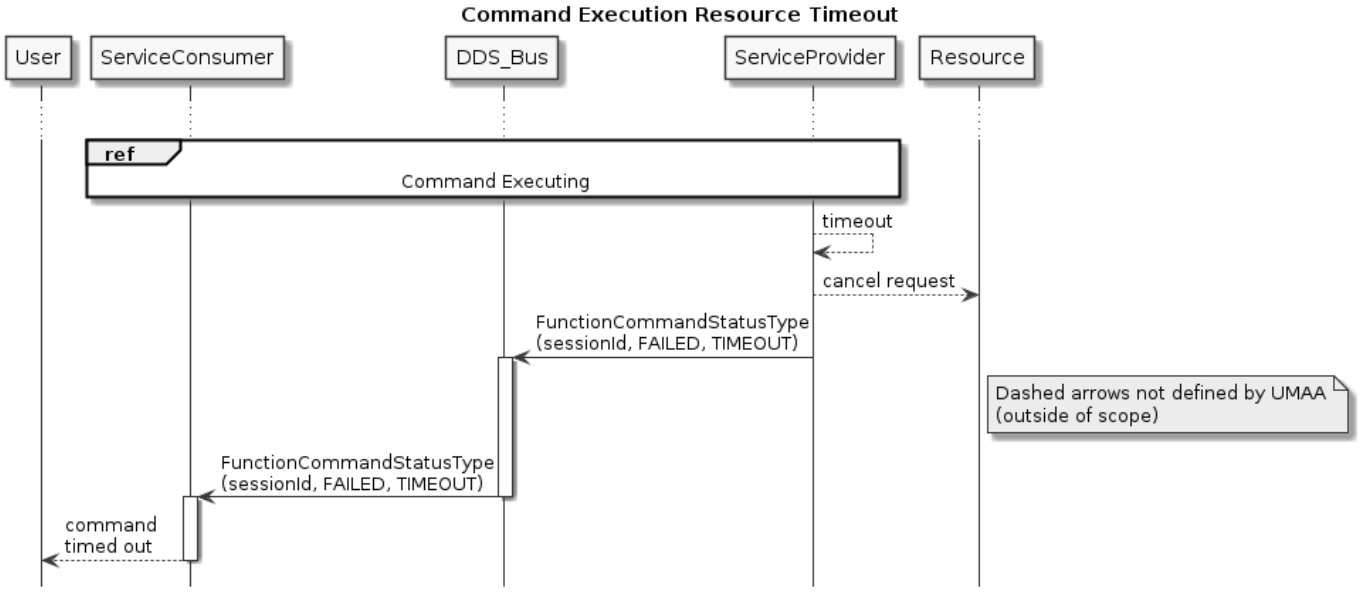

Figure 33: Sequence Diagram for a Command That Times Out Before Completing.

Other failure conditions will follow a similar pattern: when the failure is recognized, the Service Provider will publish a `FunctionCommandStatusType` with a status of `FAILED` and a reason that reflect the cause of the failure.

5.1.4.5 Command Canceled The Service Consumer may decide to cancel the command before processing is finished. To signal a desire to cancel a command, the Service Consumer disposes of the existing `FunctionCommandType` from the DDS bus before the execution is complete. When notified of the command disposal, and if the Service Provider is able to cancel the command, it should respond to the Service Consumer with a `FunctionCommandStatusType` with both the status and reason as `CANCELED`. At this point, the DDS bus should dispose of the `FunctionCommandStatusType`, the `FunctionCommandAckReportType` and, (if defined for the Function service) the `FunctionExecutionStatusReportType`. This is shown in Figure 34. If the command cannot be canceled, then the Service Provider can continue to update the command status until the execution is completed. Reporting will include `FunctionCommandStatusType` with a status of `COMPLETED` and a reason of `SUCCEEDED`. Then, the DDS bus should dispose of the `FunctionCommandStatusType`, the `FunctionCommandAckReportType`, and (if defined for the Function service) the `FunctionExecutionStatusReportType`.

There is no new, unique, or specific status message response to a cancel command from the Service Provider. The cancel command status can be inferred through the corresponding `FunctionCommandStatusType` status and reason updates.

On loss of liveliness of a Service Provider while executing a command, all Service Consumers must cancel (dispose) all in-process commands with that Service Provider.

On loss of liveliness of a Service Consumer while executing a command, all Service Providers must treat the command as canceled. This means the service should report the `CANCELED` status for the command, and then dispose the command status, ack, and execution status (if one exists).

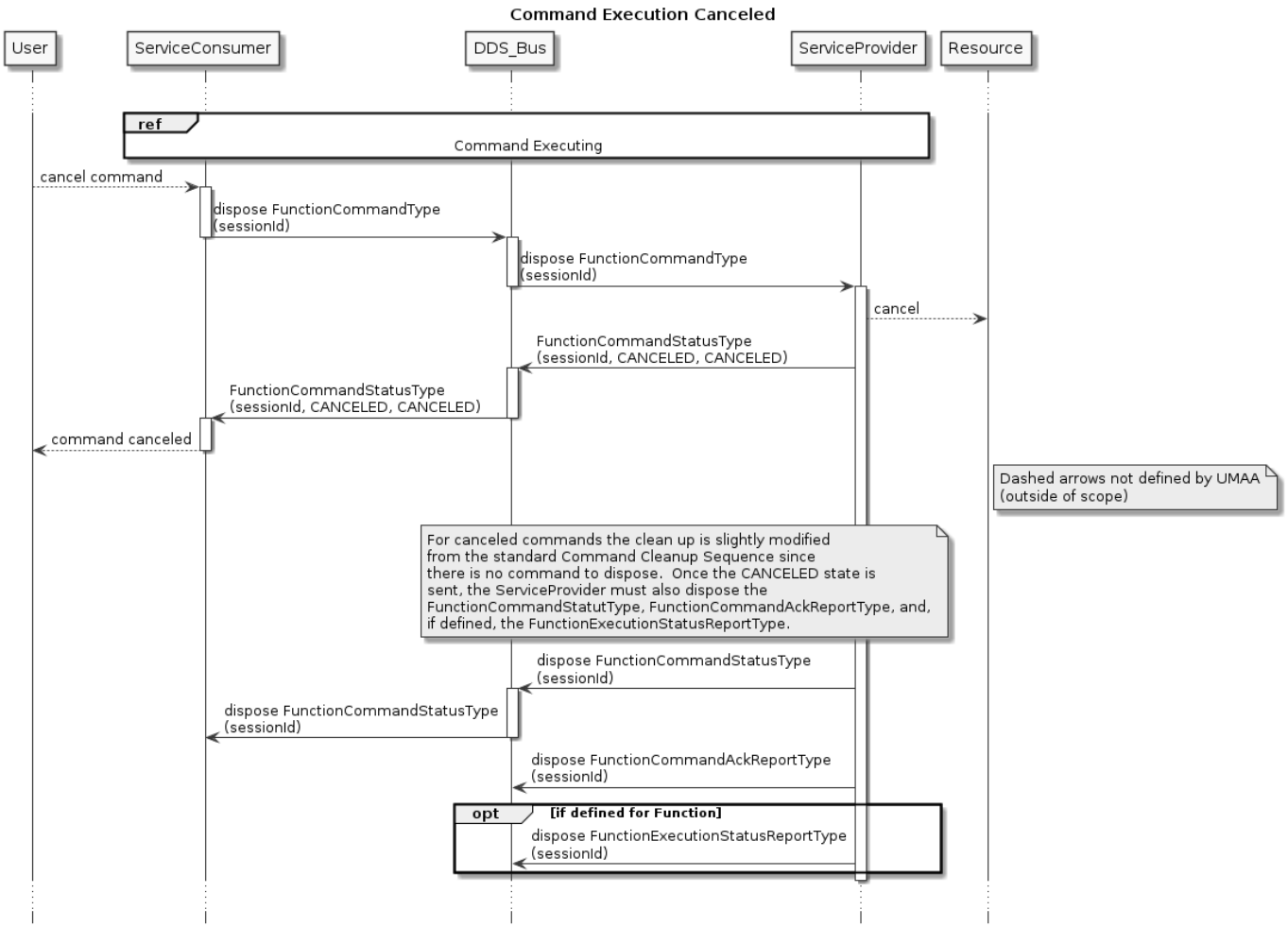

Figure 34: Sequence Diagram for a Command That is Canceled by the Service Consumer Before the Service Provider can Complete It.

5.1.5 Command Cleanup

The Service Consumer and Service Provider are responsible for disposing of corresponding data that is published to the DDS bus when the command is no longer active. With the exception of a canceled command, the signal that a `FunctionCommandType` can be disposed is when the `FunctionCommandStatusType` reports a terminal state (`COMPLETED` or `FAILED`)[3]. In turn, the signal that a `FunctionCommandStatusType`, `FunctionCommandAckReportType`, and (if defined for the Function service) the `FunctionExecutionStatusReportType` can be disposed is when the corresponding `FunctionCommandType` has been disposed. This is shown in Figure 35.

[3]While `CANCELED` is also a terminal state, the `CANCELED` command cleanup is handled specially as part of the cancelling sequence and, as such, does not need to be handled here.

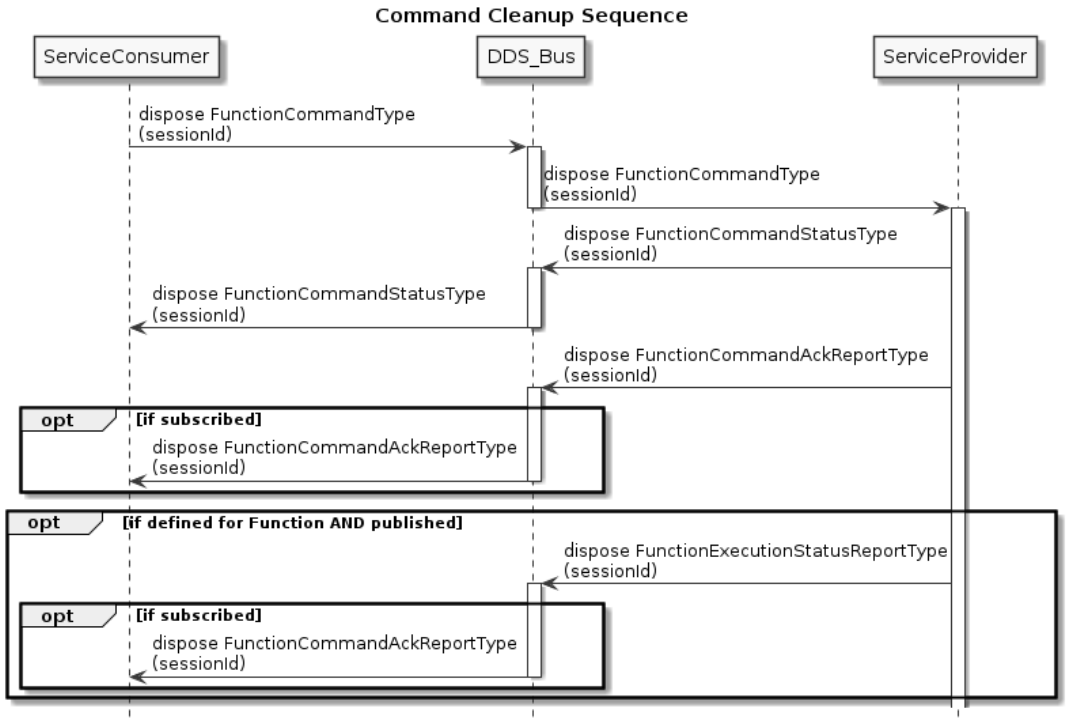

Figure 35: Sequence Diagram Showing Cleanup of the Bus When a Command Has Been Completed and the Service Consumer No Longer Wishes to Maintain the Commanded State.

5.1.6 Command Shutdown Sequence

As part of shutdown, both the Service Provider and Service Consumer are required to perform a shutdown sequence. This shutdown cleans up resources on the DDS bus and informs the system that the Service Provider and Service Consumer are no longer available.

The Service Provider and Service Consumer can shut down in any order. The sequence diagram is shown in Figure 36.

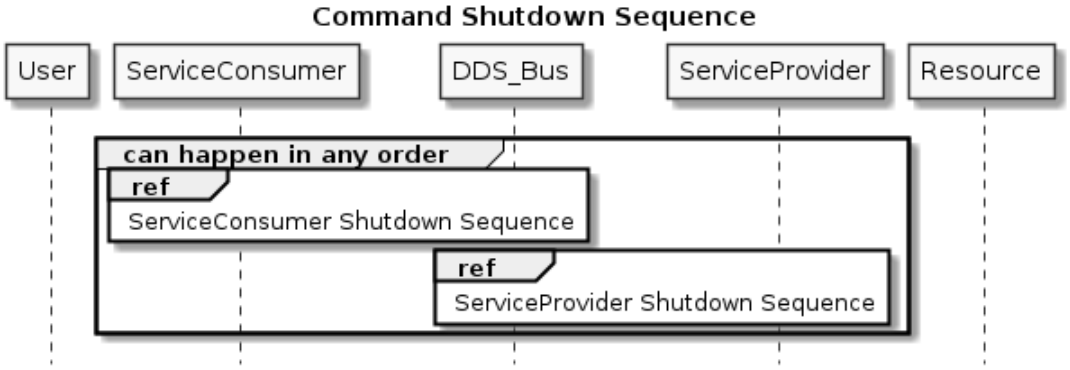

Figure 36: Sequence Diagram for Command Shutdown.

5.1.6.1 Service Provider Shutdown Sequence During shutdown, the Service Provider is required to fail any incomplete requests and then unregisters as a publisher of the `FunctionCommandStatusType`, `FunctionCommandAckReportType`, and (if defined for the Function service) the `FunctionExecutionStatusReportType`.

The Service Provider is also required to unsubscribe from the `FunctionCommandType`.

The Service Provider Shutdown sequence is shown in Figure 37.

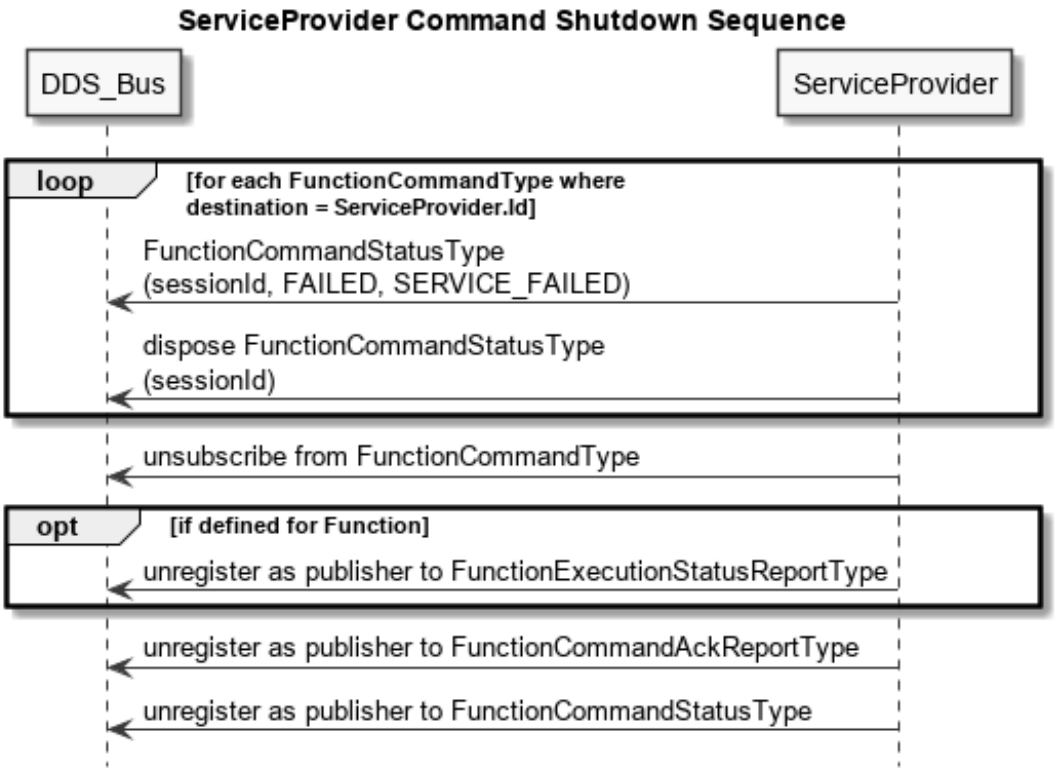

Figure 37: Sequence Diagram for Command Shutdown for Service Providers.

5.1.6.2 Service Consumer Shutdown Sequence During shutdown, the Service Consumer is required to cancel any incomplete requests and then unregister as a publisher of the `FunctionCommandType`.

The Service Consumer is also required to unsubscribe from the `FunctionCommandStatusType`, the `FunctionCommandAckReportType` if subscribed, and the `FunctionExecutionStatusReportType` if defined for the Function service and subscribed.

The Service Consumer Shutdown sequence is shown in Figure 38.

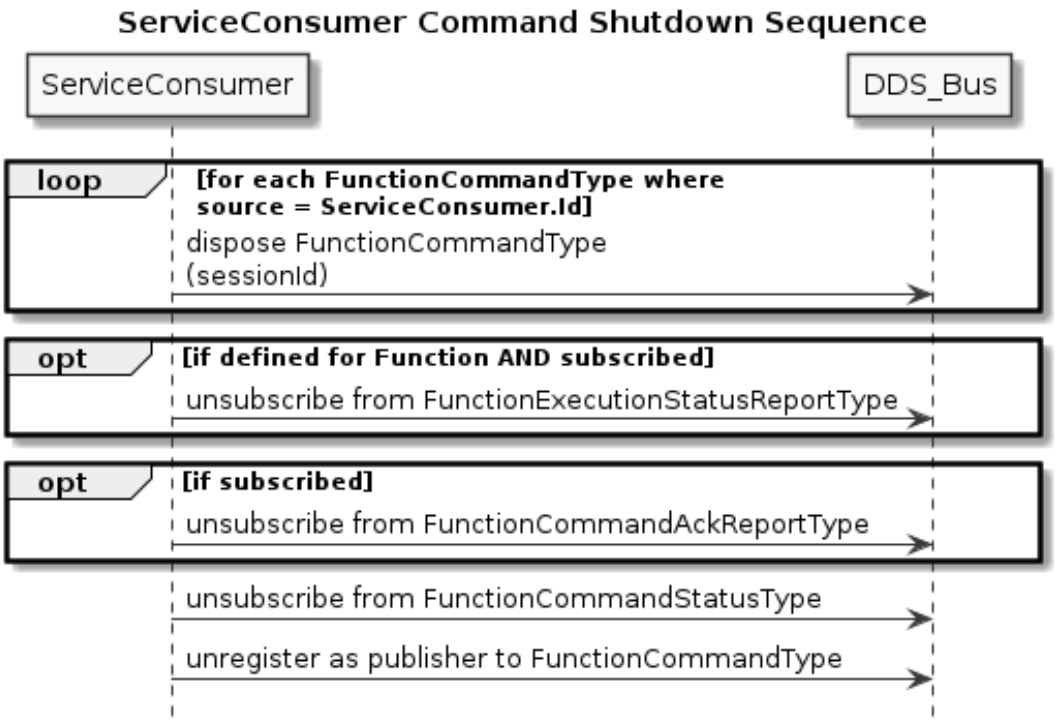

Figure 38: Sequence Diagram for Command Shutdown for Service Consumers.

5.2 Request / Reply

This section defines the flow of control for request/reply over the DDS bus. A request/reply is used to obtain data or status from a specific Service Provider.

A Service Provider is required to reply to all requests it receives. In the case of requests with no query data, this is accomplished via a DDS subscribe. In the case of a request with associated query data, a message with the query data must be published by the requester. To direct a request at a specific Service Provider or set of services, UMAA defines a `destination` GUID as part of requests.

The sequence diagrams in Sections 39 through 43 demonstrate different exchanges between a Service Consumer and Service Provider. Within the diagrams, the dashed arrows represent implementation-specific communications that are outside of UMAA's scope. Additionally, these sequence diagrams are examples of one possible implementation. Other implementations may have different communication patterns between the Service Provider and the Resource, or be implemented completely within the Service Provider process itself (no external Resource). However, in all implementations, UMAA-defined exchanges with the DDS bus between the Service Consumer and Service Provider must happen in the order shown within the sequence diagrams.

5.2.1 Request/Reply without Query Data

Figure 39 shows the sequence of exchanges in the case where there is no specific query data (i.e., the service is always just providing the current data to the bus).

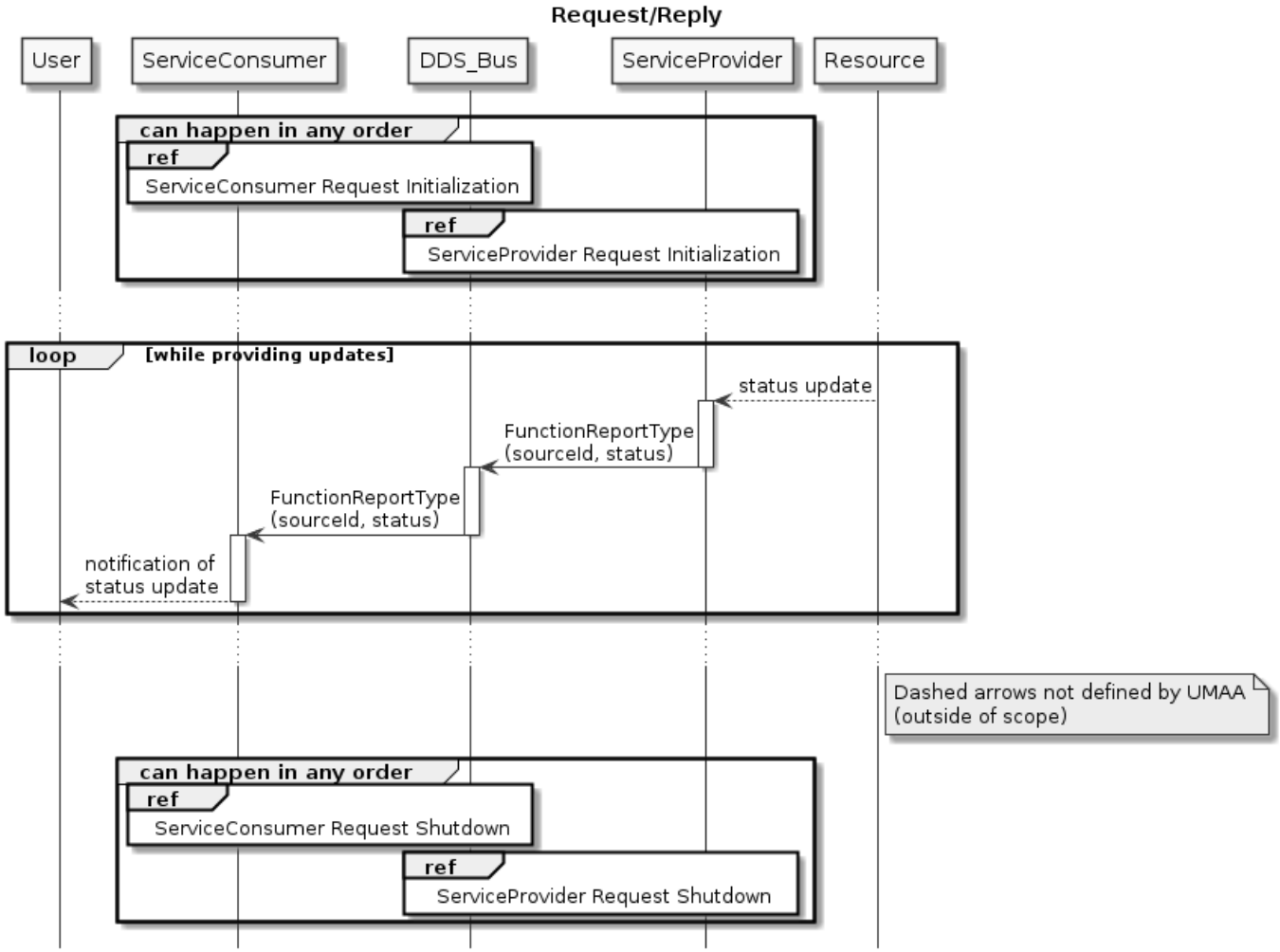

Figure 39: Sequence Diagram for a Request/Reply for Report Data That Does Not Require any Specific Query Data.

5.2.1.1 Service Provider Startup Sequence The Service Provider registers as a publisher of `FunctionReportTypes` to be able to respond to requests. The Service Provider must also handle reports that exist on the bus from a previous instantiation, either by providing an immediate update or, if the status is unrecoverable, disposing of the old `FunctionReportType`. This is shown in Figure 40.

As `FunctionReportType` updates are required (either through event-driven changes or periodic updates), the Service Provider publishes the updated data. The DDS bus will deliver the updates to the Service Consumer.

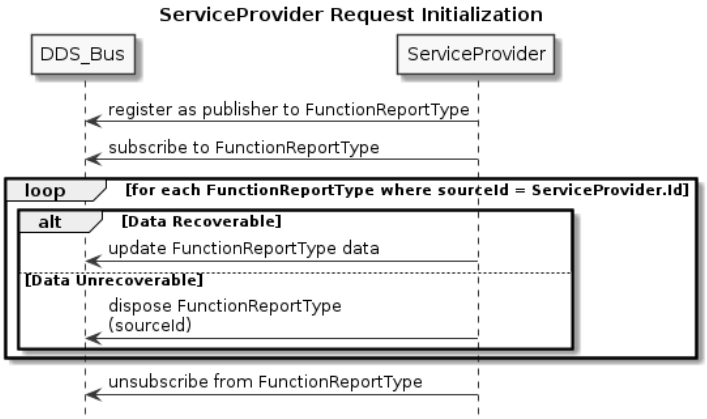

Figure 40: Sequence Diagram for Initialization of a Service Provider to Provide `FunctionReportType`s.

5.2.1.2 Service Consumer Startup Sequence The Service Consumer subscribes to the `FunctionReportType` to signal an outstanding request for updates. This is shown in Figure 41.

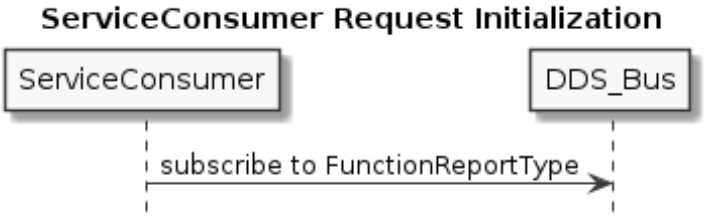

Figure 41: Sequence Diagram for Initialization of a Service Consumer to Request `FunctionReportType`s.

5.2.1.3 Service Provider Shutdown To no longer provide `FunctionReportType`s, the Service Provider disposes of the `FunctionReportType` and unregisters as a publisher of the data (shown in Figure 42).

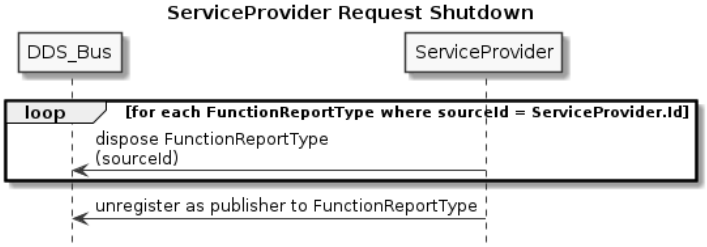

Figure 42: Sequence Diagram for Shutdown of a Service Provider.

5.2.1.4 Service Consumer Shutdown To no longer request `FunctionReportType`s, the Service Consumer unsubscribes from `FunctionReportType` (shown in Figure 43).

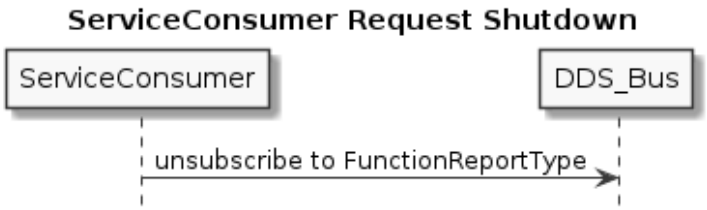

Figure 43: Sequence Diagram for Shutdown of a Service Consumer.

5.2.2 Request/Reply with Query Data

Currently, UMAA does not define any request/reply interactions with query data, but it is expected that some will be defined. When defined, this section will be expanded to describe how they must be used.

6 Experimental Services (EXP) Services and Interfaces

6.1 Services and Interfaces

The interfaces in the following subsections describe how each UCS-UMAA topic is defined by listing the name, namespace, and member attributes. The "name" corresponds with the message name of a given service interface. The "namespace" defines the scope of the "name" where similar commands are grouped together. The "member attributes" are fields that can be populated with differing data types, e.g. a generic "depth" attribute could be populated with a double data value. Note that using a UCS-UMAA "Topic Name" requires using the fully-qualified namespace plus the topic name.

Each interface topic is referenced by a UMAA service and is defined as either an input or output interface.

Attributes ending in one or more asterisk(s) denote the following:
* = Key (annotated with @key in IDL file; vendors may use different notation to indicate a key field)
† = Optional (annotated with @optional in IDL file; vendors may use different notation to indicate an optional field)

Optional fields should be handled as described in the UMAA Compliance Specification.

Commands issued on the DDS bus must be treated as if they are immutable in UMAA and, therefore, if updated (treated incorrectly as mutable), the resulting service actions are indeterminate and flow control protocols are no longer guaranteed.

Operations without DDS Topics

⊕ = Operations that are handled directly in DDS

query<...> - All query operations are used to retrieve the correlated report message. For UMAA, this operation is accomplished through subscribing to the appropriate DDS topic.

cancel<...> - All cancel operations are used to nullify the current command. For UMAA, this operation is accomplished through the DDS dispose action on the publisher.

report<...>CancelCommandStatus - All cancel reports are included here to show completeness of the MDE model mapping to UMAA. For UMAA, this operation is not used. Instead, the cancel status is inferred from the associated command status. If the cancel command is successful, the corresponding command will fail with a command status and reason of CANCELED. If the corresponding command status reports COMPLETED, then this cancel command has failed.

6.1.1 AccelerationStatus

The purpose of this service is to report the current rate of change in linear velocity and rotational rate of the vehicle.

Table 8: AccelerationStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryAcceleration⊕	reportAcceleration

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.1.1 reportAcceleration

Description: This operation is used to report the current status of the Acceleration service.

Namespace: UMAA::SA::AccelerationStatus

Topic: AccelerationReportType

Data Type: AccelerationReportType

Table 9: AccelerationReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
acceleration	Acceleration3DPlatformXYZ	The current rate of change in linear velocity of the vehicle.
accelerationCovariance†	CovarianceAccelerationPlatformXYZType	Current error covariance value of the vehicle acceleration.
rotationalAcceleration†	OrientationAcceleration3DPlatformXYZ	The current rate of change in rotational rate of the vehicle.
rotationalAccelerationCovariance†	CovarianceOrientationAccelerationPlatformXYZType	Current error covariance value of the vehicle rotational acceleration.

6.1.2 AcousticInterferenceStatus

The purpose of this service is to provide access to the status of acoustic interference.

Table 10: AcousticInterferenceStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryAcousticInterference⊕	reportAcousticInterference

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.2.1 reportAcousticInterference

Description: This operation provides the current acoustic interference status.

Namespace: UMAA::SEM::AcousticInterferenceStatus

Topic: AcousticSelfNoiseInterferenceStatusType

Data Type: AcousticSelfNoiseInterferenceStatusType

Table 11: AcousticSelfNoiseInterferenceStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
bandwidth†	FrequencyHertz	The difference between the upper and lower frequencies for the interference signal.
centerFrequency†	FrequencyHertz	Center of the upper and lower frequencies of the interference signal.
duration	DurationSeconds	Expected duration of the interference.
time	DateTime	Start time of interference.
type†	InterferenceEnumType	Type of interference. Used to help determine if the sonar needs to eliminate the time window of the interference.

6.1.3 BITConfig

The purpose of this service is to configure a Built-In-Test (BIT).

Table 12: BITConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBITConfig	reportBITConfigCommandStatus
cancelBITConfig⊕	reportBITCancelConfigCommandStatus⊕
queryBITConfig⊕	reportBITConfig
queryBITConfigAck⊕	reportBITConfigAck

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.3.1 reportBITConfig

Description: This operation is used to report the current status of the BIT configuration service.

Namespace: UMAA::SO::BITConfig

Topic: BITConfigReportType

Data Type: BITConfigReportType

Table 13: BITConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
continuousTestConfiguration	ContinuousTestEnumType	Specifies the continuous test configuration.
minTimeBetweenTests†	DurationMilliseconds	Specifies the minimum amount of time allowed between continuous tests, with 0 being as fast as possible. If not specified, this represents automatic.
powerOnTestConfiguration	PowerOnTestEnumType	Specifies the power on test configuration.
resourceID*	IdentifierType	Unique identifier of the resource.

6.1.3.2 reportBITConfigAck

Description: This operation is used to report the current BIT configuration.

Namespace: UMAA::SO::BITConfig

Topic: BITConfigAckReportType

Data Type: BITConfigAckReportType

Table 14: BITConfigAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
config	BITConfigCommandType	The source configuration.

6.1.3.3 reportBITConfigCommandStatus

Description: This operation is used to report the status of the current BIT configuration command.

Namespace: UMAA::SO::BITConfig

Topic: BITConfigCommandStatusType

Data Type: BITConfigCommandStatusType

Table 15: BITConfigCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.3.4 setBITConfig

Description: This operation is used to set the BIT configuration command.

Namespace: UMAA::SO::BITConfig

Topic: BITConfigCommandType

Data Type: BITConfigCommandType

Table 16: BITConfigCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
continuousTestConfiguration	ContinuousTestEnumType	Specifies the continuous test configuration.
minTimeBetweenTests†	DurationMilliseconds	Specifies the minimum amount of time allowed between continuous tests, with 0 being as fast as possible. If not specified, this represents automatic.
powerOnTestConfiguration	PowerOnTestEnumType	Specifies the power on test configuration.
resourceID	IdentifierType	Unique identifier of the resource.

6.1.4 BITControl

The purpose of this service is to provide the capability to command a Built-In-Test (BIT).

Table 17: BITControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBIT	reportBITCommandStatus
queryBITCommandAck⊕	reportBITCommandAck
queryBITExecutionStatus⊕	reportBITExecutionStatus
cancelBITCommand⊕	reportBITCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.4.1 reportBITCommandAck

Description: This operation is used to provide the BIT commanded values.

Namespace: UMAA::SO::BITControl

Topic: BITCommandAckReportType

Data Type: BITCommandAckReportType

Table 18: BITCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	BITCommandType	The source command.

6.1.4.2 reportBITCommandStatus

Description: This operation is used to report the status of the current BIT command.

Namespace: UMAA::SO::BITControl

Topic: BITCommandStatusType

Data Type: BITCommandStatusType

Table 19: BITCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.4.3 reportBITExecutionStatus

Description: This operation is used to report the current BIT execution status.

Namespace: UMAA::SO::BITControl

Topic: BITExecutionStatusReportType

Data Type: BITExecutionStatusReportType

Table 20: BITExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
estimatedTestCompletion†	DateTime	Provides the estimated time the BIT will be completed.
resourceID	IdentifierType	Unique identifier of the health detail of the resource.
testCancelable	boolean	Indicates that the BIT can be canceled after it has been started.

6.1.4.4 setBIT

Description: This operation is used to set the BIT command.

Namespace: UMAA::SO::BITControl

Topic: BITCommandType

Data Type: BITCommandType

Table 21: BITCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
initiatedTestType	InitiatedTestEnumType	Specifies the initiated test configuration.
resourceID	IdentifierType	Unique identifier of the health detail of the resource.

6.1.5 BITReport

The purpose of this service is to provide the status of conducting a Built-In-Test (BIT).

Table 22: BITReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBIT⊕	reportBIT

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.5.1 reportBIT

Description: This operation is used to report the current status of the BIT service.

Namespace: UMAA::SO::BITReport

Topic: BITReportType

Data Type: BITReportType

Table 23: BITReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
commandBITAvailable	boolean	Specifies whether or not an initiated Built-In-Test (IBIT) can be performed.
timeOfLastBIT†	DateTime	Provides the time of the last BIT.
resourceID*	IdentifierType	Unique identifier of the resource.

6.1.6 BITSpecs

The purpose of this service is to report the capabilities of a Built-In-Test (BIT).

Table 24: BITSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBITSpecs⊕	reportBITSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.6.1 reportBITSpecs

Description: This operation is used to report the current status of the BIT specifications service.

Namespace: UMAA::SO::BITSpecs

Topic: BITSpecsReportType

Data Type: BITSpecsReportType

Table 25: BITSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
avgMinTimeBetweenContinuousTests†	DurationMilliseconds	Specifies the average minimum time between continuous tests.
fullContinuousTestSupported	boolean	Specifies whether or not full continuous tests are supported.
fullPowerOnTestSupported	boolean	Specifies whether or not full power on tests are supported.
initiatedDestructiveTestSupported	boolean	Specifies whether or not initiated destructive tests are supported.
initiatedNonDestructiveTestSupported	boolean	Specifies whether or not initiated non-destructive tests are supported.
minTimeBetweenTestsSupported	boolean	Specifies whether or not min time between tests are supported.
nonIntrusiveContinuousTestSupported	boolean	Specifies whether or not non-intrusive continuous tests are supported.
quickPowerOnTestSupported	boolean	Specifies whether or not quick power on tests are supported.
resourceID*	IdentifierType	Unique identifier of the resource.

6.1.7 BallastTank

The purpose of this service is to provide the operations and interfaces to control and monitor the ballast tanks and their supporting pumps on the vehicle.

Table 26: BallastTank Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBallastPump	reportBallastPumpCommandStatus
queryBallastPumpCommandAck⊕	reportBallastPumpCommandAck
cancelBallastPumpCommand⊕	reportBallastPumpCancelCommandStatus⊕
queryBallastPump⊕	reportBallastPump
setBallastTank	reportBallastTankCommandStatus
queryBallastTankCommandAck⊕	reportBallastTankCommandAck
cancelBallastTankCommand⊕	reportBallastTankCancelCommandStatus⊕
queryBallastTank⊕	reportBallastTank
queryBallastPumpSpecs⊕	reportBallastPumpSpecs
queryBallastTankSpecs⊕	reportBallastTankSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.7.1 reportBallastPump

Description: This operation is used to report the current status of the ballast pumps on the vehicle.

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpReportType

Data Type: BallastPumpReportType

Table 27: BallastPumpReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
massFillRate†	MassFlowRate	The flow rate to fill or empty the mass.
state	PumpStateEnumType	The current state of the ballast pump.
volumeFlowRate†	VolumetricFlowRate	The flow rate to fill or empty the volume.

6.1.7.2 reportBallastPumpCommandAck

Description: This operation is used to provide the BallastPump commanded values.

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpCommandAckReportType

Data Type: BallastPumpCommandAckReportType

Table 28: BallastPumpCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	BallastPumpCommandType	The source command.

6.1.7.3 reportBallastPumpCommandStatus

Description: This operation is used to report the status of the current BallastPump command.

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpCommandStatusType

Data Type: BallastPumpCommandStatusType

Table 29: BallastPumpCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.7.4 reportBallastPumpSpecs

Description: This operation is used to report the specifications of the ballast pumps on the vehicle.

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpSpecsReportType

Data Type: BallastPumpSpecsReportType

Table 30: BallastPumpSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
maxMassEmptyRate	MassFlowRate	The maximum flow rate to empty the mass.
maxMassFillRate	MassFlowRate	The maximum flow rate to fill the mass.
maxVolumeEmptyRate	VolumetricFlowRate	The maximum flow rate to empty the volume.
maxVolumeFillRate	VolumetricFlowRate	The maximum flow rate to fill the volume.
minMassEmptyRate	MassFlowRate	The minimum flow rate to empty the mass.
minMassFillRate	MassFlowRate	The minimum flow rate to fill the mass.
minVolumeEmptyRate	VolumetricFlowRate	The minimum flow rate to empty the volume.
minVolumeFillRate	VolumetricFlowRate	The minimum flow rate to fill the volume.

6.1.7.5 reportBallastTank

Description: This operation is used to report the current status of the ballast tanks on the vehicle.

Namespace: UMAA::EO::BallastTank

Topic: BallastTankReportType

Data Type: BallastTankReportType

Table 31: BallastTankReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
level†	VolumePercent	The current volume level.
lowPressureLimit	PressureKiloPascals	The minimum allowable pressure of the ballast tank.
mass†	Mass	The current mass level.
pressure	PressureKiloPascals	The current pressure of the ballast tank.
pressureLimit	PressureKiloPascals	The maximum allowable pressure of the ballast tank.
trimActive	boolean	The status of ballast tank trim.

6.1.7.6 reportBallastTankCommandAck

Description: This operation is used to provide the BallastTank commanded values.

Namespace: UMAA::EO::BallastTank

Topic: BallastTankCommandAckReportType

Data Type: BallastTankCommandAckReportType

Table 32: BallastTankCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	BallastTankCommandType	The source command.

6.1.7.7 reportBallastTankCommandStatus

Description: This operation is used to report the status of the current BallastTank command.

Namespace: UMAA::EO::BallastTank

Topic: BallastTankCommandStatusType

Data Type: BallastTankCommandStatusType

Table 33: BallastTankCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.7.8 reportBallastTankSpecs

Description: This operation is used to report the specifications of the ballast tanks on the vehicle.

Namespace: UMAA::EO::BallastTank

Topic: BallastTankSpecsReportType

Data Type: BallastTankSpecsReportType

Table 34: BallastTankSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
massCapacity	Mass	The mass capacity of the ballast tank.
name	StringShortDescription	The name of the ballast tank.
trimTank	boolean	True if this is a trim tank.
volumeCapacity	VolumeCubicMeter	The volume capacity of the ballast tank.

6.1.7.9 setBallastPump

Description: This operation is used to set the BallastPump command.

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpCommandType

Data Type: BallastPumpCommandType

Table 35: BallastPumpCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
ballastPumpFlowRate	BallastPumpFlowRateType	The desired flow rate to fill or empty the ballast pump.

6.1.7.10 setBallastTank

Description: This operation is used to set the BallastTank command.

Namespace: UMAA::EO::BallastTank

Topic: BallastTankCommandType

Data Type: BallastTankCommandType

Table 36: BallastTankCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
ballastFill	BallastFillType	The desired fill amount of the ballast tank.

6.1.8 BilgePumpStatus

The purpose of this service is to provide the operations and interfaces to monitor the bilge pumps on the vehicle. Three modes of operation, Off, On, and Auto are supported per bilge pump. The auto mode means the pump will automatically be turned on by the service when flood is detected in its responsible area. Note: Flood and leak reporting is provided by the Compartment Sensor Service.

Table 37: BilgePumpStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBilgePump⊕	reportBilgePump

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.8.1 reportBilgePump

Description: This operation is used to report the current state of the bilge pumps on the vehicle.

Namespace: UMAA::EO::BilgePumpStatus

Topic: BilgePumpReportType

Data Type: BilgePumpReportType

Table 38: BilgePumpReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	BilgeStateEnumType	The state of the bilge pump.

6.1.9 ClearDataControl

The purpose of this service is to define the parameters needed for an unmanned vehicle to control the clearing of data.

Table 39: ClearDataControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setClearData	reportClearDataCommandStatus
queryClearDataCommandAck⊕	reportClearDataCommandAck
cancelClearDataCommand⊕	reportClearDataCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.9.1 reportClearDataCommandAck

Description: This operation is used to provide the ClearData commanded values.

Namespace: UMAA::SO::ClearDataControl

Topic: ClearDataCommandAckReportType

Data Type: ClearDataCommandAckReportType

Table 40: ClearDataCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	ClearDataCommandType	The source command.

6.1.9.2 reportClearDataCommandStatus

Description: This operation is used to report the status of the current ClearData command.

Namespace: UMAA::SO::ClearDataControl

Topic: ClearDataCommandStatusType

Data Type: ClearDataCommandStatusType

Table 41: ClearDataCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.9.3 setClearData

Description: This operation is used to set the ClearData command.

Namespace: UMAA::SO::ClearDataControl

Topic: ClearDataCommandType

Data Type: ClearDataCommandType

Table 42: ClearDataCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
clearData	boolean	Clear volatile and non-volatile memory containing operational data.
clearEncryption	boolean	Clear encryption keys or other encryption sources.
clearOSMemory	boolean	Clear volatile and non-volatile memory containing operating system.

6.1.10 CompartmentConfig

The purpose of this service is to provide the identification of the compartment sensors.

Table 43: CompartmentConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCompartmentConfig⊕	reportCompartmentConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.10.1 reportCompartmentConfig

Description: This operation is used to report the identification of the compartment sensors on the vehicle.

Namespace: UMAA::SA::CompartmentConfig

Topic: CompartmentConfigReportType

Data Type: CompartmentConfigReportType

Table 44: CompartmentConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	The name of the compartment.

6.1.11 CompartmentStatus

The purpose of this service is to report the status of the compartment sensors (temperature, humidity, flood and leak detection) on the vehicle.

Table 45: CompartmentStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCompartment⊕	reportCompartment

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.11.1 reportCompartment

Description: This operation is used to report the current status of the compartment sensors on the vehicle.

Namespace: UMAA::SA::CompartmentStatus

Topic: CompartmentReportType

Data Type: CompartmentReportType

Table 46: CompartmentReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
floodDetected†	boolean	The response from the flood detection system.
humidity†	RelativeHumidity	The humidity of the compartment.
leakDetected†	boolean	The response from the leak detection system.
pressure†	PressureKiloPascals	The pressure of the compartment.
temperature†	Temperature	The temperature of the compartment.

6.1.12 ConditionalStateReport

The purpose of this service is to report the state of a conditional.

Table 47: ConditionalStateReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryConditionalState⊕	reportConditionalState

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.12.1 reportConditionalState

Description: This operation is used to provide the current state of a conditional for mission plan execution.

Namespace: UMAA::MM::ConditionalStateReport

Topic: ConditionalStateReportType

Data Type: ConditionalStateReportType

Table 48: ConditionalStateReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	boolean	Specifies the state of the conditional.
conditionalID*	NumericGUID	Specifies the identifier of the conditional.

6.1.13 ContactCategoryReport

This service identifies the category (and associated confidence in that category) of a track for a reported contact.

Table 49: ContactCategoryReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryContactCategory⊕	reportContactCategory

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.13.1 reportContactCategory

Description: This operation provides the category (and associated confidence in that category) of a track for a reported contact.

Namespace: UMAA::SA::ContactCategoryReport

Topic: ContactCategoryReportType

Data Type: ContactCategoryReportType

Table 50: ContactCategoryReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
confidence	Percent	The confidence in the visual classification of the contact.
category*	TrackCategoryEnumType	Indicates the type of track, by category.

Attribute Name	Attribute Type	Attribute Description
contactID*	NumericGUID	An identifier of the contact.

6.1.14 ContactFilterConfig

The purpose of this service is to provide a specialized filter that can be used to manage volume for contact reports for external transfer. Enables publishing per configuration information in order to be able to manage comms link bandwidth.

Table 51: ContactFilterConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
setContactFilterConfig	reportContactFilterConfigCommandStatus
cancelContactFilterConfig⊕	reportContactFilterCancelConfigCommandStatus⊕
queryContactFilterConfigAck⊕	reportContactFilterConfigAck

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.14.1 reportContactFilterConfigAck

Description: This operation is used to report the current ContactFilter configuration.

Namespace: UMAA::SA::ContactFilterConfig

Topic: ContactFilterConfigAckReportType

Data Type: ContactFilterConfigAckReportType

Table 52: ContactFilterConfigAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
config	ContactFilterConfigCommandType	The source configuration.

6.1.14.2 reportContactFilterConfigCommandStatus

Description: This operation is used to provide the status of the current ContactFilterConfig command.

Namespace: UMAA::SA::ContactFilterConfig

Topic: ContactFilterConfigCommandStatusType

Data Type: ContactFilterConfigCommandStatusType

Table 53: ContactFilterConfigCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.14.3 setContactFilterConfig

Description: This operation is used to add a new contact filter configuration.

Namespace: UMAA::SA::ContactFilterConfig

Topic: ContactFilterConfigCommandType

Data Type: ContactFilterConfigCommandType

Table 54: ContactFilterConfigCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
bearingChangeLimit	Angle	Specifies to only report if contact bearing change estimate change exceeds this value.
headingChangeLimit	Angle	Specifies to only report if contact heading change estimate change exceeds this value.
messageFilterID	NumericGUID	The identifier of the message filter.
noChangeTimerUpdate	DurationSeconds	Specifies to only report if no change for this amount of time, report timeStamp update so contact is still considered active.
positionChangeLimit	Distance	Specifies to only report if contact distance change estimate exceeds this value.
rangeChangeLimit	Distance	Specifies to only report if contact range change estimate change exceeds this value.
speedChangeLimit	GroundSpeed	Specifies to only report if contact speed change estimate exceeds this value.
withinRangeofOwnship	Distance	Specifies to only report if contact distance from ownship estimate is less that value.

6.1.15 ContactIdentityReport

This service provides the identity (and associated confidence in that identity) of a track for a reported contact.

Table 55: ContactIdentityReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryContactIdentity⊕	reportContactIdentity

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.15.1 reportContactIdentity

Description: This operation provides the identity (and associated confidence in that identity) of a track for a reported contact.

Namespace: UMAA::SA::ContactIdentityReport

Topic: ContactIdentityReportType

Data Type: ContactIdentityReportType

Table 56: ContactIdentityReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
confidence	Percent	The confidence in the visual classification of the contact.
contactID*	NumericGUID	An identifier of the contact.
identity*	TrackIdentityEnumType	The current identity of the contact track.

6.1.16 ControlSystemID

The purpose of this service is to report the information of a control system and its client(s).

Table 57: ControlSystemID Operations

Service Requests (Inputs)	Service Responses (Outputs)
setControlSystemID	reportControlSystemIDCommandStatus
queryControlSystemIDCommandAck⊕	reportControlSystemIDCommandAck
cancelControlSystemIDCommand⊕	reportControlSystemIDCancelCommandStatus⊕
queryControlSystemID⊕	reportControlSystemID
queryClientID⊕	reportClientID

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.16.1 reportClientID

Description: This operation is used to report the information of client(s) within a control system.

Namespace: UMAA::SO::ControlSystemID

Topic: ClientIDReportType

Data Type: ClientIDReportType

Table 58: ClientIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	A name to describe a user or a subset of a control system.

6.1.16.2 reportControlSystemID

Description: This operation is used to report the information of a control system.

Namespace: UMAA::SO::ControlSystemID

Topic: ControlSystemIDReportType

Data Type: ControlSystemIDReportType

Table 59: ControlSystemIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	A name to describe a control station or a control system.

6.1.16.3 reportControlSystemIDCommandAck

Description: This operation is used to provide the ControlSystemID commanded values.

Namespace: UMAA::SO::ControlSystemID

Topic: ControlSystemIDCommandAckReportType

Data Type: ControlSystemIDCommandAckReportType

Table 60: ControlSystemIDCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	ControlSystemIDCommand Type	The source command.

6.1.16.4 reportControlSystemIDCommandStatus

Description: This operation is used to report the status of the set control system ID command.

Namespace: UMAA::SO::ControlSystemID

Topic: ControlSystemIDCommandStatusType

Data Type: ControlSystemIDCommandStatusType

Table 61: ControlSystemIDCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.16.5 setControlSystemID

Description: This operation is used to set the control system ID for an service that might not know it.

Namespace: UMAA::SO::ControlSystemID

Topic: ControlSystemIDCommandType

Data Type: ControlSystemIDCommandType

Table 62: ControlSystemIDCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
name	StringShortDescription	The name of the control system.

6.1.17 ControlTransfer

The purpose of this service is to control and manage ownership of a vehicle, a system, or a payload.

Table 63: ControlTransfer Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryClientControl⊕	reportClientControl
queryControlSystemControl⊕	reportControlSystemControl
queryControlSystemTransfer⊕	reportControlSystemTransfer
queryControlTransfer⊕	reportControlTransfer

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.17.1 reportClientControl

Description: This operation is used to report which client within the control system is in control of a vehicle, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ClientControlReportType

Data Type: ClientControlReportType

Table 64: ClientControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.
clientID	IdentifierType	A unique identification of an operator or a subsystem that controls the vehicle or a payload. However, if status is AVAILABLE, clientID should not be set.
status	ResourceAllocationStatusEnumType	A control status of the vehicle, a system or a payload.

6.1.17.2 reportControlSystemControl

Description: This operation is used to report which control system is in control of a vehicle, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ControlSystemControlReportType

Data Type: ControlSystemControlReportType

Table 65: ControlSystemControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.

Attribute Name	Attribute Type	Attribute Description
controlSystemID	IdentifierType	A unique identification of a system that controls the vehicle or a payload. However, if status is AVAILABLE, controlSystemID should not be set.
status	ResourceAllocationStatusEnumType	A control status of the vehicle, a system or a payload.

6.1.17.3 reportControlSystemTransfer

Description: This operation is used to report a control request by a control system to control of a vehicle, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ControlSystemTransferReportType

Data Type: ControlSystemTransferReportType

Table 66: ControlSystemTransferReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.
result	HandoverResultEnumType	The result of the handover.

6.1.17.4 reportControlTransfer

Description: This operation is used to report a control request by a client to control of a vehicle, a system or a payload.

Namespace: UMAA::MM::ControlTransfer

Topic: ClientControlTransferReportType

Data Type: ClientControlTransferReportType

Table 67: ClientControlTransferReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		

Attribute Name	Attribute Type	Attribute Description
authorityLevel	Count	Value between 0 and 255 with 0 indicating no control and 255 being root access. The control arbiter may use this value to automatically force a control release or this value may be displayed to the controlling client user as an indication of control urgency.
result	HandoverResultEnumType	The result of the handover.

6.1.18 EmitterControl

The purpose of this service is to define the parameters needed for an unmanned vehicle to control the emission state of an emitter.

Table 68: EmitterControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setEmitter	reportEmitterCommandStatus
queryEmitterCommandAck⊕	reportEmitterCommandAck
cancelEmitterCommand⊕	reportEmitterCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.18.1 reportEmitterCommandAck

Description: This operation is used to provide the Emitter commanded values.

Namespace: UMAA::SO::EmitterControl

Topic: EmitterCommandAckReportType

Data Type: EmitterCommandAckReportType

Table 69: EmitterCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	EmitterCommandType	The source command.

6.1.18.2 reportEmitterCommandStatus

Description: This operation is used to report the status of the current Emitter command.

Namespace: UMAA::SO::EmitterControl

Topic: EmitterCommandStatusType

Data Type: EmitterCommandStatusType

Table 70: EmitterCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.18.3 setEmitter

Description: This operation is used to set the Emitter command.

Namespace: UMAA::SO::EmitterControl

Topic: EmitterCommandType

Data Type: EmitterCommandType

Table 71: EmitterCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
emitterID	NumericGUID	The ID of the emitter to control.
endTime†	DateTime	Defines the end time for the desired emitter level.
state	EmitterStateEnumType	Defines the desired emitter state.

6.1.19 EmitterPresetConfig

The purpose of this service is to define the parameters needed for an unmanned vehicle to define the configuration of one or more emission levels. Each level defines a list of emitters that must be secured from generating any emissions and a list of emitters that are allowed to generate emissions as needed. If an emitter is not in either list, then it is assumed that there is no required change in its emission state for that particular emission level.

Table 72: EmitterPresetConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
setEmitterPresetConfig	reportEmitterPresetConfigCommandStatus
cancelEmitterPresetConfig⊕	reportEmitterPresetCancelConfigCommandStatus⊕
queryEmitterPresetConfig⊕	reportEmitterPresetConfig
queryEmitterPresetConfigAck⊕	reportEmitterPresetConfigAck

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.19.1 reportEmitterPresetConfig

Description: This operation is used to report the current status of the EmitterPresetConfig service.

Namespace: UMAA::SO::EmitterPresetConfig

Topic: EmitterPresetConfigReportType

Data Type: EmitterPresetConfigReportType

Table 73: EmitterPresetConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
allowedEmitterID	sequence<NumericGUID> max size = 128	Defines the list of emitters set to the allowed state for this level.
levelName	StringShortDescription	Defines the EmitterPreset name for this level.
securedEmitterID	sequence<NumericGUID> max size = 128	Defines the list of emitters set to the secured state for this level.
levelID*	NumericGUID	Defines the desired EmitterPreset level.

6.1.19.2 reportEmitterPresetConfigAck

Description: This operation is used to report the current EmitterPreset configuration.

Namespace: UMAA::SO::EmitterPresetConfig

Topic: EmitterPresetConfigAckReportType

Data Type: EmitterPresetConfigAckReportType

Table 74: EmitterPresetConfigAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
config	EmitterPresetConfigCommandType	The source configuration.

6.1.19.3 reportEmitterPresetConfigCommandStatus

Description: This operation is used to report the status of the current EmitterPresetConfig command.

Namespace: UMAA::SO::EmitterPresetConfig

Topic: EmitterPresetConfigCommandStatusType

Data Type: EmitterPresetConfigCommandStatusType

Table 75: EmitterPresetConfigCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.19.4 setEmitterPresetConfig

Description: This operation is used to set the EmitterPresetConfig command.

Namespace: UMAA::SO::EmitterPresetConfig

Topic: EmitterPresetConfigCommandType

Data Type: EmitterPresetConfigCommandType

Table 76: EmitterPresetConfigCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
allowedEmitterID	sequence<NumericGUID> max size = 128	Defines the list of emitters that are allowed to generate emissions as needed.
levelID	NumericGUID	Defines the desired EmitterPreset level.
levelName	StringShortDescription	Defines the EmitterPreset name for this level.
securedEmitterID	sequence<NumericGUID> max size = 128	Defines the list of emitters that must be secured from generating any emissions.

6.1.20 EmitterPresetControl

The purpose of this service is to define the parameters needed for an unmanned vehicle to control the emission level.

Table 77: EmitterPresetControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setEmitterPreset	reportEmitterPresetCommandStatus
queryEmitterPresetCommandAck⊕	reportEmitterPresetCommandAck
cancelEmitterPresetCommand⊕	reportEmitterPresetCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.20.1 reportEmitterPresetCommandAck

Description: This operation is used to provide the EmitterPreset commanded values.

Namespace: UMAA::SO::EmitterPresetControl

Topic: EmitterPresetCommandAckReportType

Data Type: EmitterPresetCommandAckReportType

Table 78: EmitterPresetCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	EmitterPresetCommandType	The source command.

6.1.20.2 reportEmitterPresetCommandStatus

Description: This operation is used to report the status of the current EmitterPreset command.

Namespace: UMAA::SO::EmitterPresetControl

Topic: EmitterPresetCommandStatusType

Data Type: EmitterPresetCommandStatusType

Table 79: EmitterPresetCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.20.3 setEmitterPreset

Description: This operation is used to set the EmitterPreset command.

Namespace: UMAA::SO::EmitterPresetControl

Topic: EmitterPresetCommandType

Data Type: EmitterPresetCommandType

Table 80: EmitterPresetCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
endLevelID†	NumericGUID	Defines the level to transition to when endTime expires.
endTime†	DateTime	Defines the end time for the desired EmitterPreset level.
levelID	NumericGUID	Defines the desired EmitterPreset level.

6.1.21 EmitterPresetReport

The purpose of this service is to define the parameters needed for an unmanned vehicle to provide the emission level status.

Table 81: EmitterPresetReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryEmitterPreset⊕	reportEmitterPreset

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.21.1 reportEmitterPreset

Description: This operation is used to report the current status of the EmitterPreset service.

Namespace: UMAA::SO::EmitterPresetReport

Topic: EmitterPresetReportType

Data Type: EmitterPresetReportType

Table 82: EmitterPresetReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
endLevelID†	NumericGUID	Defines the current desired EmitterPreset level ID when the time expires.
endTime†	DateTime	Defines the end time for the desired EmitterPreset level.
isModified	boolean	Whether the level has been modified from its defined configuration.
levelID	NumericGUID	Defines the current EmitterPreset level ID.

6.1.22 EmitterReport

The purpose of this service is to define the parameters needed for an unmanned vehicle to provide the current emission state of an emitter.

Table 83: EmitterReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryEmitter⊕	reportEmitter

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.22.1 reportEmitter

Description: This operation is used to report the current status of the Emitter service.

Namespace: UMAA::SO::EmitterReport

Topic: EmitterReportType

Data Type: EmitterReportType

Table 84: EmitterReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
endTime†	DateTime	Defines the end time for the commanded emitter level.
state	EmitterStateEnumType	Defines the current emitter state.
emitterID*	NumericGUID	The ID of the emitter being reported on.

6.1.23 EmitterSpecs

The purpose of this service is to define the parameters needed for an unmanned vehicle to provide the specifications of an emitter.

Table 85: EmitterSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryEmitterSpecs⊕	reportEmitterSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.23.1 reportEmitterSpecs

Description: This operation is used to report the current status of the EmitterSpecs service.

Namespace: UMAA::SO::EmitterSpecs

Topic: EmitterSpecsReportType

Data Type: EmitterSpecsReportType

Table 86: EmitterSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
frequencyBand	sequence<ElectroMagneticFrequencyHertz> max size = 16	Defines the frequency bands used by the emitter.
name	StringShortDescription	The name of the emitter.
emitterID*	NumericGUID	The ID of the emitter being reported on.

6.1.24 EngineSpecs

The purpose of this service is to report the specifications of the engine on the vehicle.

Table 87: EngineSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryEngineSpecs⊕	reportEngineSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.24.1 reportEngineSpecs

Description: This operation is used to report the system specifications of the engines of the vehicle.

Namespace: UMAA::EO::EngineSpecs

Topic: EngineSpecsReportType

Data Type: EngineSpecsReportType

Table 88: EngineSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
engineKind	EngineKindEnumType	The type of engine.
glowPlugTime†	DurationSeconds	The glow plug preset heating time.
maxCoolantLevel	VolumeCubicMeter	The maximum coolant level limit.
maxCoolantPressure	PressureKiloPascals	The maximum coolant pressure limit.
maxCoolantTemp	Temperature	The maximum coolant temperature limit.
maxEngineTemp	Temperature	The maximum engine temperature limit.

Attribute Name	Attribute Type	Attribute Description
maxGlowPlugTemp†	Temperature	The maximum glow plug temperature limit.
maxManifoldAirTemp	Temperature	The maximum engine manifold air temperature limit.
maxManifoldPressure	PressureKiloPascals	The maximum engine manifold pressure limit.
maxOilPressure	MaxEngineOilPressure	The maximum engine oil pressure limit.
maxOilTemp	Temperature	The maximum engine oil temperature limit.
minCoolantLevel	VolumeCubicMeter	The minimum coolant level limit.
minOilLevel	VolumeCubicMeter	The minimum engine oil level limit.
name	StringShortDescription	The name of the engine.
oilCapacity	VolumeCubicMeter	The oil capacity of the engine.
reverseRPMLowerLimit	EngineSpeed	Describes the lower limit of reverse RPM.
reverseRPMMaxLimit	EngineSpeed	Describes the maximum limit of reverse RPM.
reverseRPMUpperLimit	EngineSpeed	Describes the upper limit of reverse RPM.
reversible	boolean	The reversibility of the engine rotation.
RPMLowerLimit	EngineSpeed	The lower RPM limit to operate the engine.
RPMMaxLimit	EngineSpeed	The physical maximum RPM limit to operate the engine.
RPMUpperLimit	EngineSpeed	The upper RPM limit to operate the engine.

6.1.25 FLSControl

The purpose of this service is to configure a Forward Looking Sonar (FLS).

Table 89: FLSControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setFLS	reportFLSCommandStatus
queryFLSCommandAck⊕	reportFLSCommandAck
cancelFLSCommand⊕	reportFLSCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.25.1 reportFLSCommandAck

Description: This operation is used to provide the FLS commanded values.

Namespace: UMAA::SEM::FLSControl

Topic: FLSCommandAckReportType

Data Type: FLSCommandAckReportType

Table 90: FLSCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	FLSCommandType	The source command.

6.1.25.2 reportFLSCommandStatus

Description: This operation is used to report the status of the current FLS command.

Namespace: UMAA::SEM::FLSControl

Topic: FLSCommandStatusType

Data Type: FLSCommandStatusType

Table 91: FLSCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.25.3 setFLS

Description: This operation is used to set the FLS command.

Namespace: UMAA::SEM::FLSControl

Topic: FLSCommandType

Data Type: FLSCommandType

Table 92: FLSCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
additionalConfig†	FLSAdditionalConfigType	Additional configuration parameters for the FLS. If configMode correlates to one of the types in this union, this field is required. Otherwise, this field is ignored.
configMode	FLSConfigModeEnumType	The desired configuration mode for the subsystem.
operationalState	ActivationStateTargetEnumType	The desired operational state of the subsystem. Determines if the subsystem is active.

6.1.26 FLSPreCalcControl

The purpose of this service is to configure a Forward Looking Sonar (FLS).

Table 93: FLSPreCalcControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setFLSPreCalc	reportFLSPreCalcCommandStatus
queryFLSPreCalcCommandAck⊕	reportFLSPreCalcCommandAck
cancelFLSPreCalcCommand⊕	reportFLSPreCalcCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.26.1 reportFLSPreCalcCommandAck

Description: This operation is used to provide the FLSPreCalc commanded values.

Namespace: UMAA::SEM::FLSPreCalcControl

Topic: FLSPreCalcCommandAckReportType

Data Type: FLSPreCalcCommandAckReportType

Table 94: FLSPreCalcCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	FLSPreCalcCommandType	The source command.

6.1.26.2 reportFLSPreCalcCommandStatus

Description: This operation is used to report the status of the current FLSPreCalc command.

Namespace: UMAA::SEM::FLSPreCalcControl

Topic: FLSPreCalcCommandStatusType

Data Type: FLSPreCalcCommandStatusType

Table 95: FLSPreCalcCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.26.3 setFLSPreCalc

Description: This operation is used to set the FLSPreCalc command.

Namespace: UMAA::SEM::FLSPreCalcControl

Topic: FLSPreCalcCommandType

Data Type: FLSPreCalcCommandType

Table 96: FLSPreCalcCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
additionalConfig†	FLSAdditionalConfigType	Additional configuration parameters for the FLS. If configMode correlates to one of the types in this union, this field is required. Otherwise, this field is ignored.
configMode	FLSConfigModeEnumType	The desired configuration mode of the FLS.

6.1.27 FileSystemStatus

The purpose of this service is to provide a mechanism to report the file system status.

Table 97: FileSystemStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFileSystem⊕	reportFileSystem

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.27.1 reportFileSystem

Description: This operation is used to report the file system status.

Namespace: UMAA::SO::FileSystemStatus

Topic: FileSystemReportType

Data Type: FileSystemReportType

Table 98: FileSystemReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
availableInodes†	LargeCount	Number of free inodes for unprivileged users.

Attribute Name	Attribute Type	Attribute Description
availableSpace†	SizeLargeBytes	Available space for unprivileged users.
freeInodes†	LargeCount	Number of free inodes on the file system.
freeSpace†	SizeLargeBytes	Free space on the file system.
inodes†	LargeCount	Number of inodes on the file system.
maxFilenameLength†	Count	The maximum file name length.
path	StringShortDescription	The pathname of any file within the mounted file system
reachable	boolean	True if file system is working properly, False otherwise
readOnly†	boolean	True if the path is read only, False otherwise.
totalSpace†	SizeLargeBytes	Total space on the file system.

6.1.28 FinsControl

The purpose of this service is to provide the control of the deflection of a set of 1 or more fins (a fin system) for stabilization and mobility of the vehicle.

Table 99: FinsControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setFins	reportFinsCommandStatus
queryFinsCommandAck⊕	reportFinsCommandAck
cancelFinsCommand⊕	reportFinsCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.28.1 reportFinsCommandAck

Description: This operation is used to report the commanded deflection of a set of 1 or more fins (a fin system) on the vehicle.

Namespace: UMAA::EO::FinsControl

Topic: FinsCommandAckReportType

Data Type: FinsCommandAckReportType

Table 100: FinsCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	FinsCommandType	The source command.

6.1.28.2 reportFinsCommandStatus

Description: This operation is used to report the status of the fins deflection command.

Namespace: UMAA::EO::FinsControl

Topic: FinsCommandStatusType

Data Type: FinsCommandStatusType

Table 101: FinsCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.28.3 setFins

Description: This operation is used to control the fin deflection of a set of 1 or more fins (a fin system) on the vehicle. The consumer must perform a cancel of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::EO::FinsControl

Topic: FinsCommandType

Data Type: FinsCommandType

Table 102: FinsCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
fins	sequence<FinCommandType> max size = 16	Contains set of simultaneous commands for the fin system.

6.1.29 FinsSpecs

The purpose of this service is to provide the specifications of a set of 1 or more fins (a fin system) on the vehicle.

Table 103: FinsSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFinsSpecs⊕	reportFinsSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.29.1 reportFinsSpecs

Description: This operation is used to report the specifications of a set of 1 or more fins (a fin system) on the vehicle.

Namespace: UMAA::EO::FinsSpecs

Topic: FinsSpecsReportType

Data Type: FinsSpecsReportType

Table 104: FinsSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
finSpecs	sequence<FinSpecsType> max size = 16	Contains specification of each of the fins for the fin system.

6.1.30 FinsStatus

The purpose of this service is to provide the current deflection of a set of 1 or more fins (a fin system) on the vehicle.

Table 105: FinsStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFins⊕	reportFins

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.30.1 reportFins

Description: This operation is used to report the current deflection of a set of 1 or more fins (a fin system) on the vehicle.

Namespace: UMAA::EO::FinsStatus

Topic: FinsStatusType

Data Type: FinsStatusType

Table 106: FinsStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
fins	sequence<FinStatusType> max size = 16	Contains specification of each of the fins for the fin system

6.1.31 FreeFloatControl

The purpose of this service is to command the vehicle to free float.

Table 107: FreeFloatControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setFreeFloat	reportFreeFloatCommandStatus
queryFreeFloatCommandAck⊕	reportFreeFloatCommandAck
queryFreeFloatExecutionStatus⊕	reportFreeFloatExecutionStatus
cancelFreeFloatCommand⊕	reportFreeFloatCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.31.1 reportFreeFloatCommandAck

Description: This operation is used to provide the FreeFloat commanded values.

Namespace: UMAA::MO::FreeFloatControl

Topic: FreeFloatCommandAckReportType

Data Type: FreeFloatCommandAckReportType

Table 108: FreeFloatCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	FreeFloatCommandType	The source command.

6.1.31.2 reportFreeFloatCommandStatus

Description: This operation is used to report the status of the current FreeFloat command.

Namespace: UMAA::MO::FreeFloatControl

Topic: FreeFloatCommandStatusType

Data Type: FreeFloatCommandStatusType

Table 109: FreeFloatCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.31.3 reportFreeFloatExecutionStatus

Description: This operation is used to report the time that the vehicle was free floating.

Namespace: UMAA::MO::FreeFloatControl

Topic: FreeFloatExecutionStatusReportType

Data Type: FreeFloatExecutionStatusReportType

Table 110: FreeFloatExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
timeFreeFloatAchieved	DateTime	The absolute time at which free float is estimated to be achieved or was actually first achieved.
timeFreeFloatCompleted†	DateTime	The absolute time at which free float is estimated to be completed (optional for the case where an end time is not specified).

6.1.31.4 setFreeFloat

Description: This operation is used to set the FreeFloat command.

Namespace: UMAA::MO::FreeFloatControl

Topic: FreeFloatCommandType

Data Type: FreeFloatCommandType

Table 111: FreeFloatCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
endTime†	DateTime	Specifies the end of the command execution time period for the free float; if not specified runs indefinitely until command is changed externally or another terminating condition occurs.

6.1.32 GlobalPoseConfig

The purpose of this service is to command the current position and orientation of the vehicle in the global coordinate system. The service exposes interfaces to set the position and orientation for those vehicles requiring external pose updates. It is designated to provide an initial position for dead-reckoning.

Table 112: GlobalPoseConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalPoseConfig	reportGlobalPoseConfigCommandStatus
cancelGlobalPoseConfig⊕	reportGlobalPoseCancelConfigCommandStatus⊕
queryGlobalPoseConfig⊕	reportGlobalPoseConfig
queryGlobalPoseConfigAck⊕	reportGlobalPoseConfigAck

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.32.1 reportGlobalPoseConfig

Description: This operation is used to report the current configuration command.

Namespace: UMAA::SA::GlobalPoseConfig

Topic: GlobalPoseConfigReportType

Data Type: GlobalPoseConfigReportType

Table 113: GlobalPoseConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		

6.1.32.2 reportGlobalPoseConfigAck

Description: This operation is used to report the current GlobalPose configuration.

Namespace: UMAA::SA::GlobalPoseConfig

Topic: GlobalPoseConfigAckReportType

Data Type: GlobalPoseConfigAckReportType

Table 114: GlobalPoseConfigAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
config	GlobalPoseConfigCommandType	The source configuration.

6.1.32.3 reportGlobalPoseConfigCommandStatus

Description: This operation is used to report the status of the current configuration command.

Namespace: UMAA::SA::GlobalPoseConfig

Topic: GlobalPoseConfigCommandStatusType

Data Type: GlobalPoseConfigCommandStatusType

Table 115: GlobalPoseConfigCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.32.4 setGlobalPoseConfig

Description: This operation is used to set the configuration of the vehicle's global pose.

Namespace: UMAA::SA::GlobalPoseConfig

Topic: GlobalPoseConfigCommandType

Data Type: GlobalPoseConfigCommandType

Table 116: GlobalPoseConfigCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
attitude	Orientation3DNEDRequirement	The desired orientation (roll, pitch, yaw) of the vehicle.
attitudeCovariance†	CovarianceOrientationNEDType	The desired covariance value of the validity of the orientation data.
elevation	ElevationVariantType	Specifies the elevation of the vector.
position	GeoPosition2D	The commanded initial position of the vehicle.
positionCovariance†	CovariancePositionNEDType	The commanded initial covariance value of the validity of the position data.

6.1.33 HazardAvoidanceConfig

The purpose of this service is to report the desired hazard avoidance configuration for individual contacts. The service maps contact IDs to desired hazard avoidance parameters for that particular contact. All service implementations are required to always report a Nil UUID that maps to the default parameters for any contact ID where a configuration is not explicitly defined.

Table 117: HazardAvoidanceConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryHazardAvoidanceConfig⊕	reportHazardAvoidanceConfig

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.33.1 reportHazardAvoidanceConfig

Description: This operation is used to report the hazard avoidance configuration parameters.

Namespace: UMAA::MO::HazardAvoidanceConfig

Topic: HazardAvoidanceConfigReportType

Data Type: HazardAvoidanceConfigReportType

Table 118: HazardAvoidanceConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
hazardAvoidanceConfig†	ContactHazardAvoidanceType	The hazard avoidance configuration for the associated contact ID. If not defined, hazard avoidance is turned off for this contact.
contactID*	NumericGUID	The contact for which this configuration defines the hazard avoidance parameters. If the contact ID is the Nil GUID (as specified in RFC 4122), this defines the default behavior for any contact ID not explicitly reported by the service.

6.1.34 HeartbeatPulseStatus

The purpose of this service is to provide a means to maintain the periodic communication connection with the vehicle.

Table 119: HeartbeatPulseStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryHeartbeatPulse⊕	reportHeartbeatPulse

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.34.1 reportHeartbeatPulse

Description: This operation is used to report the heartbeat pulse status of the vehicle.

Namespace: UMAA::SO::HeartbeatPulseStatus

Topic: HeartbeatPulseReportType

Data Type: HeartbeatPulseReportType

Table 120: HeartbeatPulseReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
heartBeat	Count	The current heartbeat pulse to report the HeartbeatPulse connection of the vehicle.

6.1.35 IlluminatorControl

The purpose of this service is to control Illuminators, e.g., turn on/off and controlling the light characteristics.

Table 121: IlluminatorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setIlluminator	reportIlluminatorCommandStatus
queryIlluminatorCommandAck⊕	reportIlluminatorCommandAck
cancelIlluminatorCommand⊕	reportIlluminatorCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.35.1 reportIlluminatorCommandAck

Description: This operation is used to provide the Illuminator commanded values.

Namespace: UMAA::SEM::IlluminatorControl

Topic: IlluminatorCommandAckReportType

Data Type: IlluminatorCommandAckReportType

Table 122: IlluminatorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	IlluminatorCommandType	The source command.

6.1.35.2 reportIlluminatorCommandStatus

Description: This operation is used to report the status of an illuminator command.

Namespace: UMAA::SEM::IlluminatorControl

Topic: IlluminatorCommandStatusType

Data Type: IlluminatorCommandStatusType

Table 123: IlluminatorCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.35.3 setIlluminator

Description: This operation is used to set the intensity and beam width of an illuminator.

Namespace: UMAA::SEM::IlluminatorControl

Topic: IlluminatorCommandType

Data Type: IlluminatorCommandType

Table 124: IlluminatorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
beamWidth†	IlluminatorBeamWidth	The horizontal field of illumination.
color†	RGBType	The color of illumination.
intensity†	IlluminatorIntensityLevel	The intensity level as a percentage.
state	IlluminatorStateEnumType	Describes the current illuminator state.

6.1.36 IlluminatorSpecs

The purpose of this service is to report the capabilities of an Illuminator.

Table 125: IlluminatorSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryIlluminatorSpecs⊕	reportIlluminatorSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.36.1 reportIlluminatorSpecs

Description: This operation is used to report the specifications of an illuminator.

Namespace: UMAA::SEM::IlluminatorSpecs

Topic: IlluminatorSpecsReportType

Data Type: IlluminatorSpecsReportType

Table 126: IlluminatorSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
beamWidthSpecs†	BeamWidthSpecsType	The range of horizontal field of illumination. If not specified, then the width is fixed.
name	StringShortDescription	The name of the illuminator.
supportedColor	boolean	Indicates that color is controllable.
supportedInfrared	boolean	Indicates that the source is infrared.
supportedIntensityLevel	boolean	Indicates that intensity is controllable.

6.1.37 IlluminatorStatus

The purpose of this service is to report the current state of the illuminators, e.g. on/off, and the light characteristics.

Table 127: IlluminatorStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryIlluminator⊕	reportIlluminator

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.37.1 reportIlluminator

Description: This operation is used to report the illuminator state and light characteristics.

Namespace: UMAA::SEM::IlluminatorStatus

Topic: IlluminatorReportType

Data Type: IlluminatorReportType

Table 128: IlluminatorReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
beamWidth†	IlluminatorBeamWidth	The horizontal field of illumination.
color†	RGBType	The color of illumination.
intensity†	IlluminatorIntensityLevel	The intensity level as a percentage.
state	IlluminatorStateEnumType	Describes the current illuminator state.

6.1.38 LandmarkReport

This service provides landmark related data.

Table 129: LandmarkReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryLandmark⊕	reportLandmark

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.38.1 reportLandmark

Description: This operation is used to report the current status of the Landmark service.

Namespace: UMAA::SA::LandmarkReport

Topic: LandmarkReportType

Data Type: LandmarkReportType

Table 130: LandmarkReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
depth	DistanceBSL	The depth of the landmark.
landmarkType†	LandmarkEnumType	The type of landmark.

Attribute Name	Attribute Type	Attribute Description
location	GeoPosition2D	The two-dimensional location of the landmark.
landmarkID*	NumericGUID	The unique identifier of the landmark.

6.1.39 MapSegmentControl

The purpose of this service is to control survey map segments.

Table 131: MapSegmentControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setMapSegment	reportMapSegmentCommandStatus
queryMapSegmentCommandAck⊕	reportMapSegmentCommandAck
cancelMapSegmentCommand⊕	reportMapSegmentCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.39.1 reportMapSegmentCommandAck

Description: This operation is used to provide the MapSegment commanded values.

Namespace: UMAA::SEM::MapSegmentControl

Topic: MapSegmentCommandAckReportType

Data Type: MapSegmentCommandAckReportType

Table 132: MapSegmentCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	MapSegmentCommandType	The source command.

6.1.39.2 reportMapSegmentCommandStatus

Description: This operation is used to report the status of the current MapSegment command.

Namespace: UMAA::SEM::MapSegmentControl

Topic: MapSegmentCommandStatusType

Data Type: MapSegmentCommandStatusType

Table 133: MapSegmentCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.39.3 setMapSegment

Description: This operation is used to set the MapSegment command.

Namespace: UMAA::SEM::MapSegmentControl

Topic: MapSegmentCommandType

Data Type: MapSegmentCommandType

Table 134: MapSegmentCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
segment	SegmentID	The index number for the segment. If set to a value that has previously been commanded, that map with that segment number will be replaced.
survey	StringShortDescription	A user-readable explanation of the control transfer request.

6.1.40 MemoryStatus

The purpose of this service is to provide a mechanism to report the memory status.

Table 135: MemoryStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryMemory⊕	reportMemory

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.40.1 reportMemory

Description: This operation is used to report the memory status.

Namespace: UMAA::SO::MemoryStatus

Topic: MemoryReportType

Data Type: MemoryReportType

Table 136: MemoryReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
bufferedRam†	SizeLargeBytes	Memory used by buffers.
freeMemory	SizeLargeBytes	Available memory size.
freeSwap†	SizeLargeBytes	Swap space still available.
sharedMemory†	SizeLargeBytes	Amount of shared memory.
totalMemory	SizeLargeBytes	Total usable main memory size.
totalSwap†	SizeLargeBytes	Total swap space size.

6.1.41 MissionPlanAssignmentControl

The purpose of this service is to provide operations and interfaces required to assign tasks and objectives to available resources.

Table 137: MissionPlanAssignmentControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setMissionPlanAssignment	reportMissionPlanAssignmentCommandStatus
queryMissionPlanAssignmentCommandAck⊕	reportMissionPlanAssignmentCommandAck
cancelMissionPlanAssignmentCommand⊕	reportMissionPlanAssignmentCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.41.1 reportMissionPlanAssignmentCommandAck

Description: This operation is used to provide the MissionPlanAssignment commanded values.

Namespace: UMAA::MM::MissionPlanAssignmentControl

Topic: MissionPlanAssignmentCommandAckReportType

Data Type: MissionPlanAssignmentCommandAckReportType

Table 138: MissionPlanAssignmentCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	MissionPlanAssignmentCommandType	The source command.

6.1.41.2 reportMissionPlanAssignmentCommandStatus

Description: This operation is used to provide the current status of assigning a mission to a resource.

Namespace: UMAA::MM::MissionPlanAssignmentControl

Topic: MissionPlanAssignmentCommandStatusType

Data Type: MissionPlanAssignmentCommandStatusType

Table 139: MissionPlanAssignmentCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.41.3 setMissionPlanAssignment

Description: This operation is used to assign a mission to an available resource.

Namespace: UMAA::MM::MissionPlanAssignmentControl

Topic: MissionPlanAssignmentCommandType

Data Type: MissionPlanAssignmentCommandType

Table 140: MissionPlanAssignmentCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
missionID	NumericGUID	Identifies the mission plan
resourceIDs	sequence<IdentifierType> max size = 256	Identifies the resources that are assigned to a specific task.

6.1.42 MissionPlanAssignmentReport

The purpose of this service is to assign a mission plan to a particular resource or set of resources required to accomplish the mission plan.

Table 141: MissionPlanAssignmentReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryMissionPlanAssignment⊕	reportMissionPlanAssignment

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.42.1 reportMissionPlanAssignment

Description: This operation is used to report the current mission plan assignment.

Namespace: UMAA::MM::MissionPlanAssignmentReport

Topic: MissionPlanAssignmentReportType

Data Type: MissionPlanAssignmentReportType

Table 142: MissionPlanAssignmentReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
resourceIDs	sequence<IdentifierType> max size = 256	Identifies the resources that are assigned to accomplish the mission plan.
missionID*	NumericGUID	The identifier of the mission plan.

6.1.43 ObjectiveAssignmentControl

The purpose of this service is to provide operations and interfaces required to assign tasks and objectives to available resources.

Table 143: ObjectiveAssignmentControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setObjectiveAssignment	reportObjectiveAssignmentCommandStatus
queryObjectiveAssignmentCommandAck⊕	reportObjectiveAssignmentCommandAck
cancelObjectiveAssignmentCommand⊕	reportObjectiveAssignmentCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.43.1 reportObjectiveAssignmentCommandAck

Description: This operation is used to provide the ObjectiveAssignment commanded values.

Namespace: UMAA::MM::ObjectiveAssignmentControl

Topic: ObjectiveAssignmentCommandAckReportType

Data Type: ObjectiveAssignmentCommandAckReportType

Table 144: ObjectiveAssignmentCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	ObjectiveAssignmentCommandType	The source command.

6.1.43.2 reportObjectiveAssignmentCommandStatus

Description: This operation is used to provide the current status of the command assigning an objective to a resource.

Namespace: UMAA::MM::ObjectiveAssignmentControl

Topic: ObjectiveAssignmentCommandStatusType

Data Type: ObjectiveAssignmentCommandStatusType

Table 145: ObjectiveAssignmentCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.43.3 setObjectiveAssignment

Description: This operation is used to assign an objective to an available resource.

Namespace: UMAA::MM::ObjectiveAssignmentControl

Topic: ObjectiveAssignmentCommandType

Data Type: ObjectiveAssignmentCommandType

Table 146: ObjectiveAssignmentCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
missionID	NumericGUID	Identifies the mission plan.
objectiveID	NumericGUID	Identifies the associated objective within the mission plan.
resourceIDs	sequence<IdentifierType> max size = 256	Identifies the resources that are assigned to a specific task/objective.
taskID	NumericGUID	Identifies the associated task within the mission plan.

6.1.44 ObjectiveAssignmentReport

The purpose of this service is to associate an objective to a particular resource or set of resources required to accomplish the objective.

Table 147: ObjectiveAssignmentReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryObjectiveAssignment⊕	reportObjectiveAssignment

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.44.1 reportObjectiveAssignment

Description: This operation is used to report the current assignment of an objective to a resource.

Namespace: UMAA::MM::ObjectiveAssignmentReport

Topic: ObjectiveAssignmentReportType

Data Type: ObjectiveAssignmentReportType

Table 148: ObjectiveAssignmentReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
resourceIDs	sequence<IdentifierType> max size = 256	Identifies the resources that are assigned to a specific objective.
missionID*	NumericGUID	Identifies the associated mission plan.
objectiveID*	NumericGUID	Identifies the associated objective within a task plan of a mission plan.
taskID*	NumericGUID	Identifies the associated task plan within the mission plan.

6.1.45 ObjectiveExecutorControl

The purpose of this service is to manage the execution of an objective. This includes providing the objective to be executed as well as controlling and monitoring its current execution state.

Table 149: ObjectiveExecutorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setObjectiveExecutor	reportObjectiveExecutorCommandStatus
queryObjectiveExecutorCommandAck⊕	reportObjectiveExecutorCommandAck
queryObjectiveExecutorExecutionStatus⊕	reportObjectiveExecutorExecutionStatus
cancelObjectiveExecutorCommand⊕	reportObjectiveExecutorCancelCommandStatus⊕
setObjectiveExecutorState	reportObjectiveExecutorStateCommandStatus

Service Requests (Inputs)	Service Responses (Outputs)
queryObjectiveExecutorStateCommandAck⊕	reportObjectiveExecutorStateCommandAck
cancelObjectiveExecutorStateCommand⊕	reportObjectiveExecutorStateCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.45.1 reportObjectiveExecutorCommandAck

Description: This operation is used to provide the ObjectiveExecutor commanded values.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorCommandAckReportType

Data Type: ObjectiveExecutorCommandAckReportType

Table 150: ObjectiveExecutorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	ObjectiveExecutorCommandType	The source command.

6.1.45.2 reportObjectiveExecutorCommandStatus

Description: This operation is used to report the status of the current ObjectiveExecutor command.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorCommandStatusType

Data Type: ObjectiveExecutorCommandStatusType

Table 151: ObjectiveExecutorCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.45.3 reportObjectiveExecutorExecutionStatus

Description: This operation is used to report the current ObjectiveExecutor execution status.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorExecutionStatusReportType

Data Type: ObjectiveExecutorExecutionStatusReportType

Table 152: ObjectiveExecutorExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
missionID	NumericGUID	The current mission plan identification.
objectiveDetailedStatus	ObjectiveDetailedStatusType	The objective detailed status.
taskID	NumericGUID	The current task plan identification within the current mission plan.

6.1.45.4 reportObjectiveExecutorStateCommandAck

Description: This operation is used to provide the ObjectiveExecutorState commanded values.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorStateCommandAckReportType

Data Type: ObjectiveExecutorStateCommandAckReportType

Table 153: ObjectiveExecutorStateCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	ObjectiveExecutorStateCommandType	The source command.

6.1.45.5 reportObjectiveExecutorStateCommandStatus

Description: This operation is used to report the status of the current ObjectiveExecutorState command.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorStateCommandStatusType

Data Type: ObjectiveExecutorStateCommandStatusType

Table 154: ObjectiveExecutorStateCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.45.6 setObjectiveExecutor

Description: This operation is used to set the ObjectiveExecutor command.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorCommandType

Data Type: ObjectiveExecutorCommandType

Table 155: ObjectiveExecutorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
missionID	NumericGUID	Specifies the current mission plan identification.
objective	ObjectiveType	Specifies the objective details.
serviceInterruptTransitionToPause	boolean	If a lower level service leveraged by the Objective Executor becomes interrupted by a higher priority entity, should the ObjectiveExecutor transition to PAUSE state or to FAILED state (True = transition to PAUSE state).
taskID	NumericGUID	Specifies the current task plan identification within the current mission plan.

6.1.45.7 setObjectiveExecutorState

Description: This operation is used to set the ObjectiveExecutorState command.

Namespace: UMAA::MM::ObjectiveExecutorControl

Topic: ObjectiveExecutorStateCommandType

Data Type: ObjectiveExecutorStateCommandType

Table 156: ObjectiveExecutorStateCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
missionID	NumericGUID	Specifies the current mission plan identification.

Attribute Name	Attribute Type	Attribute Description
objectiveID	NumericGUID	Specifies the current objective identification within the current task plan.
objectiveState	ObjectiveExecutorControlEnumType	The desired state of the current objective.
taskID	NumericGUID	Specifies the current task plan identification within the current mission plan.

6.1.46 OperationalModeControl

The purpose of this service is to define the parameters needed to control the operational mode, where the mode determines if associated operational commands are allowed to execute. Note that the associated operational commands are not specified by the service interface, and is therefore determined by some other mechanism (e.g., a configuration file). The modes can be grouped as either manned or unmanned (see state transition diagram). There is a single unmanned mode, MANUAL, where all commands are physically unable to be executed, and is therefore assumed to be manually operated by persons onboard. There are three unmanned modes: STANDBY, REMOTE and AUTONOMOUS. In STANDBY mode, no commands are allowed to execute, i.e., all associated commands received must fail. In REMOTE mode, only commands received from an off board operator are executed, and all associated commands received from onboard autonomy must fail. In AUTONOMOUS mode, commands received from onboard autonomy are allowed to execute. The state transition diagram shows all allowable transitions. Note that transition to/from MANUAL mode is not determined by the service interface and is therefore handled by some other mechanism (e.g., a physical switch). Also note that transition to REMOTE mode can either occur using the service command or by receiving an associated operational command from an off board operator (i.e., commands from the off board operator always have precedence over onboard autonomy).
See figure for reference.

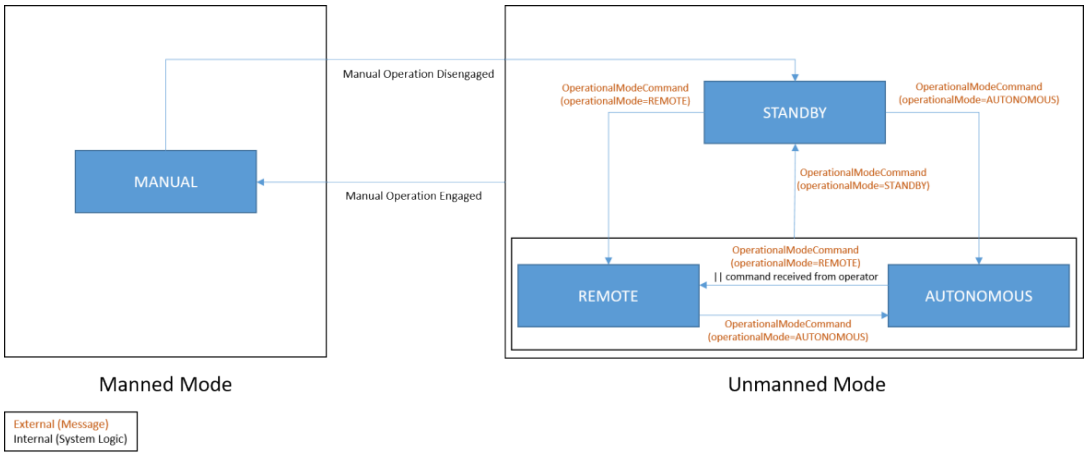

Figure 44: OperationalMode State Diagram

Table 157: OperationalModeControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setOperationalMode	reportOperationalModeCommandStatus
queryOperationalModeCommandAck⊕	reportOperationalModeCommandAck
cancelOperationalModeCommand⊕	reportOperationalModeCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.46.1 reportOperationalModeCommandAck

Description: This operation is used to provide the OperationalMode commanded values.

Namespace: UMAA::MM::OperationalModeControl

Topic: OperationalModeCommandAckReportType

Data Type: OperationalModeCommandAckReportType

Table 158: OperationalModeCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	OperationalModeCommandType	The source command.

6.1.46.2 reportOperationalModeCommandStatus

Description: This operation is used to report the status of the current OperationalMode command.

Namespace: UMAA::MM::OperationalModeControl

Topic: OperationalModeCommandStatusType

Data Type: OperationalModeCommandStatusType

Table 159: OperationalModeCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.46.3 setOperationalMode

Description: This operation is used to set the OperationalMode command.

Namespace: UMAA::MM::OperationalModeControl

Topic: OperationalModeCommandType

Data Type: OperationalModeCommandType

Table 160: OperationalModeCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
operationalMode	OperationalModeControlEnumType	Defines the commanded vehicle operational control mode. Note that MANUAL mode is controlled outside of UMAA (e.g., a physical switch).

6.1.47 OperationalModeStatus

The purpose of this service is to provide the current operational mode.
See figure for reference. (Figure 44)

Table 161: OperationalModeStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryOperationalMode⊕	reportOperationalMode

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.47.1 reportOperationalMode

Description: This operation is used to report the current status of the OperationalMode service.

Namespace: UMAA::MM::OperationalModeStatus

Topic: OperationalModeReportType

Data Type: OperationalModeReportType

Table 162: OperationalModeReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
operationalMode	OperationalModeEnumType	Defines the current vehicle operational mode.

6.1.48 OrientationStatus

The purpose of this service is to report the orientation and rotational rate of the vehicle. The rotational state of a rigid body is defined by six parameters that represent the rotational components of a point in the body that is coincident with the origin of the fixed reference and the instantaneous angular orientation components. The reference system is defined as a fixed coordinate system that at that instant, aligns with the vehicle or system coordinate system.

Table 163: OrientationStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryOrientation⊕	reportOrientation

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.48.1 reportOrientation

Description: This operation is used to report the current orientation and rotational rate of the vehicle.

Namespace: UMAA::SA::OrientationStatus

Topic: OrientationReportType

Data Type: OrientationReportType

Table 164: OrientationReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
attitude	Orientation3DNEDType	The current orientation (roll, pitch, yaw) of the vehicle.
attitudeCovariance†	CovarianceOrientationNEDType	The current error covariance value of the attitude data.
attitudeRate†	OrientationVel3D	The current rotational rate of the vehicle.
attitudeRateCovariance†	CovarianceOrientationVelocityNEDType	The current error covariance value of the attitude rate of change data.

6.1.49 PassiveContactReport

The purpose of this service is to provide passive contact (object) related data to the vehicles and/or the operator for situational awareness in the operational area.

Table 165: PassiveContactReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPassiveContact⊕	reportPassiveContact

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.49.1 reportPassiveContact

Description: This operation is used to report the current status of the PassiveContact service.

Namespace: UMAA::SA::PassiveContactReport

Topic: PassiveContactReportType

Data Type: PassiveContactReportType

Table 166: PassiveContactReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
contacts→setID	LargeSet<PassiveContactType>	A list of passive object structures. This attribute is implemented as a large set, see subsection 3.8 for an explanation. The associated topic is UMAA::SA::PassiveContactReport::PassiveContactReportTypeContactsSetElement.
platformPose	PoseType	The vehicle pose.

6.1.50 PathReporterSpecs

The purpose of this service is to provide the capabilities of the path reporting mechanism of the vehicle.

Table 167: PathReporterSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPathReporterSpecs⊕	reportPathReporterSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.50.1 reportPathReporterSpecs

Description: This operation is used to report the capabilities.

Namespace: UMAA::SA::PathReporterSpecs

Topic: PathReporterSpecsReportType

Data Type: PathReporterSpecsReportType

Table 168: PathReporterSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
pathReporters	sequence<PathReporterType> max size = 4	A list of capabilities of path reporter.

6.1.51 PathReporterStatus

The purpose of this service is to provide a mechanism for reporting the past and/or future expected path of the vehicle. The service is used in cooperation with Global Waypoint Driver, Local Waypoint Driver, Global Waypoint List Driver, Local Waypoint List Driver, Global Pose Sensor and/or Local Pose Sensor. The amount of data reported may be limited by specifying a maximum number of data points, maximum time, maximum distance, and/or path resolution, within the limits of the implementation's reported capabilities. Note that the historical path is assumed to be represented by a FIFO queue. As a result, the Report Path message may be limited by the storage capabilities of the underlying implementation to reporting only the most recent data, e.g. the points nearest the current position. Older data may be discarded as needed by the implementation. Such limits should be specified by the PathReporterSpecs service. Also, note that the future path may be valid only at that instance in time. It represents the current planned path, at the given resolution. However, some waypoint drivers may frequently update the planned path, based on new information about the environment. As a result, planned path information may quickly become stale. Furthermore, the planned path represents the actual path the vehicle expects to follow. It does not replace the Waypoint Driver's Query Waypoint messages. Rather, it provides additional information about how the vehicle plans to achieve the desired waypoints.

Table 169: PathReporterStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPathReporter⊕	reportPathReporter

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.51.1 reportPathReporter

Description: This operation is used to report the current path.

Namespace: UMAA::SA::PathReporterStatus

Topic: PathReporterReportType

Data Type: PathReporterReportType

Table 170: PathReporterReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
historicalGlobalPaths→listID	LargeList<WaypointType>	Historical Global Path. This attribute is implemented as a large list, see subsection 3.8 for an explanation. The associated topic is UMAA::SA::PathReporterStatus::PathReporterReportTypeHistoricalGlobalPathsListElement.
historicalLocalPaths→listID	LargeList<WaypointType>	Historical Local Path. This attribute is implemented as a large list, see subsection 3.8 for an explanation. The associated topic is UMAA::SA::PathReporterStatus::PathReporterReportTypeHistoricalLocalPathsListElement.
plannedGlobalPaths→listID	LargeList<WaypointType>	Planned Global Path. This attribute is implemented as a large list, see subsection 3.8 for an explanation. The associated topic is UMAA::SA::PathReporterStatus::PathReporterReportTypePlannedGlobalPathsListElement.

Attribute Name	Attribute Type	Attribute Description
plannedLocalPaths→listID	LargeList<WaypointType>	Planned Local Path. This attribute is implemented as a large list, see subsection 3.8 for an explanation. The associated topic is UMAA::SA::PathReporterStatus::PathReporterReportTypePlannedLocalPathsListElement.

6.1.52 PowerControl

The purpose of this service is to provide the power control of a resource.

Table 171: PowerControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPower	reportPowerCommandStatus
queryPowerCommandAck⊕	reportPowerCommandAck
cancelPowerCommand⊕	reportPowerCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.52.1 reportPowerCommandAck

Description: This operation is used to report the commanded power state of the resource on the vehicle.

Namespace: UMAA::EO::PowerControl

Topic: PowerCommandAckReportType

Data Type: PowerCommandAckReportType

Table 172: PowerCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	PowerCommandType	The source command.

6.1.52.2 reportPowerCommandStatus

Description: This operation is used to report the status of the power control command message.

Namespace: UMAA::EO::PowerControl

Topic: PowerCommandStatusType

Data Type: PowerCommandStatusType

Table 173: PowerCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.52.3 setPower

Description: This operation is used to control the power state of the specified resource on the vehicle.

Namespace: UMAA::EO::PowerControl

Topic: PowerCommandType

Data Type: PowerCommandType

Table 174: PowerCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
resourceID	IdentifierType	A unique identifier of the resource.
state	PowerStateEnumType	The desired power state of the resource.

6.1.53 PowerStatus

The purpose of this service is to provide the current power status of a resource.

Table 175: PowerStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPower⊕	reportPower

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.53.1 reportPower

Description: This operation is used to report the current power status of a resource on the vehicle.

Namespace: UMAA::EO::PowerStatus

Topic: PowerReportType

Data Type: PowerReportType

Table 176: PowerReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	PowerStateEnumType	The current power status of the resource.
resourceID*	IdentifierType	A unique identifier of the resource.

6.1.54 ProcessingUnitStatus

The purpose of this service is to provide a mechanism to report the processing unit status.

Table 177: ProcessingUnitStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryProcessingUnit⊕	reportProcessingUnit

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.54.1 reportProcessingUnit

Description: This operation is used to report the processing unit status.

Namespace: UMAA::SO::ProcessingUnitStatus

Topic: ProcessingUnitReportType

Data Type: ProcessingUnitReportType

Table 178: ProcessingUnitReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
load15MinuteAverage†	Percent	The processor average utilization during the last fifteen minutes.
load1MinuteAverage†	Percent	The processor average utilization during the last minute.
load5MinuteAverage†	Percent	The processor average utilization during the last five minutes.
numberOfBlockedProcesses†	Count	The number of processes awaiting I/O.
numberOfProcesses†	Count	The number of processes.
numberOfRunningProcesses†	Count	The number of currently running processes.
processorTemperature†	Temperature	The current average temperature of all processor cores.

Attribute Name	Attribute Type	Attribute Description
type	ProcessingUnitEnumType	Processor type.
uptime†	DurationSeconds	The current processor uptime.

6.1.55 PropulsorsControl

The purpose of this service is to provide the operations and interfaces to control the vehicle propulsors. A propulsor is assumed to be any mechanical device that gives propulsion, such as thrusters, propellers, water jets, etc. that is either fixed or has up to two articulations.

Table 179: PropulsorsControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPropulsors	reportPropulsorsCommandStatus
queryPropulsorsCommandAck⊕	reportPropulsorsCommandAck
cancelPropulsorsCommand⊕	reportPropulsorsCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.55.1 reportPropulsorsCommandAck

Description: This operation is used to provide the Propulsors commanded values.

Namespace: UMAA::EO::PropulsorsControl

Topic: PropulsorsCommandAckReportType

Data Type: PropulsorsCommandAckReportType

Table 180: PropulsorsCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	PropulsorsCommandType	The source command.

6.1.55.2 reportPropulsorsCommandStatus

Description: This operation is used to report the status of the propulsor configuration command.

Namespace: UMAA::EO::PropulsorsControl

Topic: PropulsorsCommandStatusType

Data Type: PropulsorsCommandStatusType

Table 181: PropulsorsCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.55.3 setPropulsors

Description: This operation is used to control a set of one or more propulsors on the vehicle. The consumer must perform a cancel of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::EO::PropulsorsControl

Topic: PropulsorsCommandType

Data Type: PropulsorsCommandType

Table 182: PropulsorsCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
propulsors	sequence<PropulsorCommandType> max size = 16	Contains a set of simultaneous commands for the propulsor system.

6.1.56 PropulsorsSpecs

The purpose of this service is to provide the operations and interfaces to report the specifications of the vehicle propulsors.

Table 183: PropulsorsSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPropulsorsSpecs⊕	reportPropulsorsSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.56.1 reportPropulsorsSpecs

Description: This operation is used to report the specifications of the propulsors on the vehicle.

Namespace: UMAA::EO::PropulsorsSpecs

Topic: PropulsorsSpecsReportType

Data Type: PropulsorsSpecsReportType

Table 184: PropulsorsSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
propulsorSpecs	sequence<PropulsorSpecsType> max size = 16	Contains the specification of each of the propulsors for the propulsor system.

6.1.57 PropulsorsStatus

The purpose of this service is to provide the operations and interfaces to monitor the status of the vehicle propulsors.

Table 185: PropulsorsStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPropulsors⊕	reportPropulsors

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.57.1 reportPropulsors

Description: This operation is used to report the current status of the propulsors on the vehicle.

Namespace: UMAA::EO::PropulsorsStatus

Topic: PropulsorsStatusType

Data Type: PropulsorsStatusType

Table 186: PropulsorsStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
propulsorStatus	sequence<PropulsorStatusType> max size = 16	Contains the status for each of the propulsors for the propulsor system.

6.1.58 RecordingSpecs

The purpose of this service is to provide various static information related to the recording.

Table 187: RecordingSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRecordingSpecs⊕	reportRecordingSpecs

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.58.1 reportRecordingSpecs

Description: This operation is used to report the current status of the RecordingSpecs service.

Namespace: UMAA::SO::RecordingSpecs

Topic: RecordingSpecsReportType

Data Type: RecordingSpecsReportType

Table 188: RecordingSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
availableRecordingSpace	SizeBytes	Amount of space available for recording.

6.1.59 RecordingStatus

The purpose of this service is to give status info for recording.

Table 189: RecordingStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRecordingStatus⊕	reportRecordingStatus

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.59.1 reportRecordingStatus

Description: This operation is used to report the current status of the RecordingStatus service.

Namespace: UMAA::SO::RecordingStatus

Topic: RecordingStatusReportType

Data Type: RecordingStatusReportType

Table 190: RecordingStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
isRecording	boolean	True when actively recording, false otherwise.
received	Count	Number of records received by the recorder; should be increasing while actively recording.
receiveErrors	Count	Total number of errors in the records received by the recording system (either missing or invalid).
spaceUsed	Percent	Total current space occupied by recorded data.
writeErrors	Count	Total number of recording write errors.

6.1.60 RelativeContactReport

The purpose of this service is to provide the vehicles and/or the operator with the contact related data relative to ownship for situational awareness in the operational area.

Table 191: RelativeContactReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRelativeContact⊕	reportRelativeContact

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.60.1 reportRelativeContact

Description: This operation provides the current contact data relative to ownship.

Namespace: UMAA::SA::RelativeContactReport

Topic: RelativeContactReportType

Data Type: RelativeContactReportType

Table 192: RelativeContactReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
bearing†	Angle	The relative bearing of the ownship vessel from the contact.
CPA†	GeoPosition2D	An estimated point in which the distance between the vehicle and contact will reach the minimum value.
CPATime†	DateTime	Time of contact CPA.
contactID*	NumericGUID	An identifier of the contact.

6.1.61 ResourceAllocation

This service provides the interfaces necessary for attempting exclusive control of a resource, setting the priority ordering of resource consumers, and retrieving the current configuration and control information for a resource.

Resource Definition
For UMAA, a resource is defined as a logical grouping of UMAA service providers whose execution is mutually exclusive within the group. For example, when executing a GlobalWaypointControl command, a FinControl command cannot execute at the same time. A resource is identified by its resourceId, and the relationship of service providers to their resource is reported by the resource configuration message. Note that there can be different groupings of resources for different purposes. For example, an autopilot and its associated physical devices can be a viewed as a single resource for control, but each fin could be a resource for Built-In-Test (BIT).

Resource Consumer Definition
Resource consumers are simply consumers of UMAA services whose access is controlled by ResourceAllocation. A resource consumer is identified by the source attribute in the header of the intended command message. This identifier is used by ResourceAllocation when determining the consumer's priority (set using the priority command).

Configuring Resource Consumer Priority
Priority is configured at the resource level, based on resourceId. Priority is modelled using a sequence of resource consumer identifiers (see above), ordered by priority (low to high). Priority may be changed during runtime using the ResourceAllocation priority command, or it may be treated as static configuration. Updating the priority does not change the process in control - rather, it changes the priority state that will be used for the next command. In either case, the priority ordering for all resource consumers in the system must be configured before any resource allocation can take place. Similarly, if a resource consumer attempts a resource allocation and its identifier is not present in the priority configuration, the request must be rejected. Priority is primarily used internally by the ResourceAllocation service, but is published to the DDS bus for persistence. Note that service providers do not rely on this report data to determine whether a command can execute; rather, they use the ResourceAllocation control report to make this determination.

Additions to Flow Control - Resource Consumer
Implementing ResourceAllocation adds additional steps to the flow control for UMAA commands. Before sending a command to a service provider, the consumer must first command ResourceAllocation to attempt to allocate the resource for control. The consumer may only proceed with its service command only if it receives a ResourceAllocation command status of EXECUTING, indicating that it now has control of the resource. This process is detailed below:

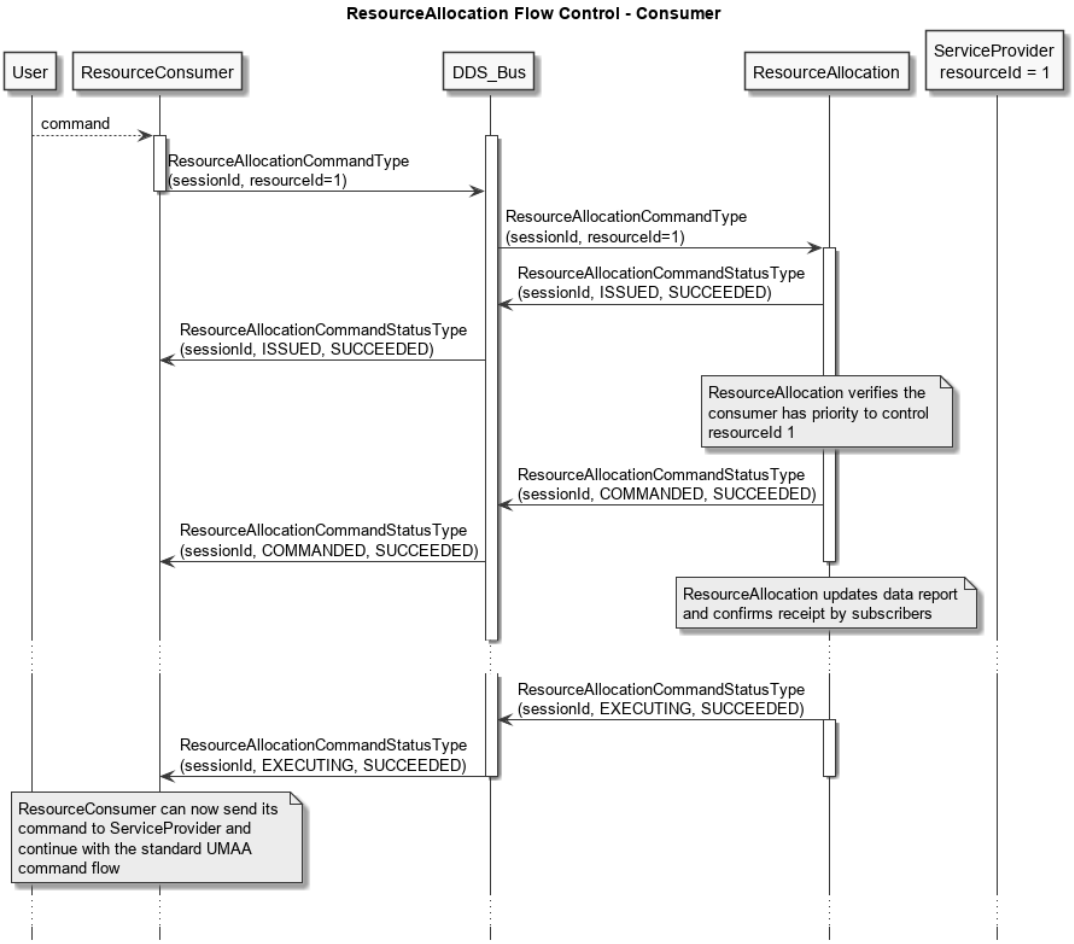

Figure 45: Sequence Diagram of Resource Consumer Requesting Control of a Resource

Additionally, resource consumers can set a duration when attempting control of a resource. If a duration is provided, the consumer must continue to ask for control within the timeout period so control allocation is not lost. This enables robust recovery for software failures and recovery.

Additions to Flow Control - Service Provider

Service providers must subscribe to the ResourceAllocation report. When a service provider receives a command, it uses this report to determine whether the consumer has control of the resource by checking the source of the incoming command message against the consumer currently in control. If the identifiers do not match, or the end time of the control session has elapsed, then the request must be rejected. This process is detailed below:

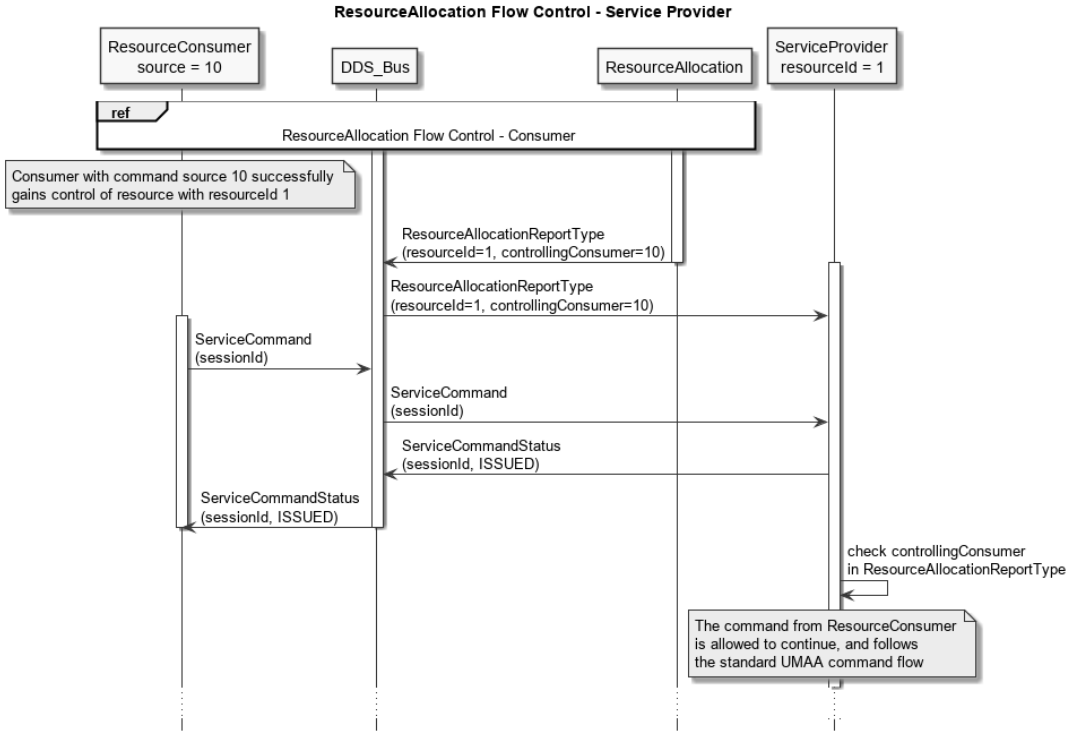

Figure 46: Sequence Diagram of Service Provider Verifying a Command using ResourceAllocation

Additions to Flow Control - Nested Service Provider

The ResourceAllocation command flow control has an additional step for nested services. A nested service provider is commanded by other service providers (and potentially by service consumers as well). For example, in the case where a GlobalWaypoint service implementation can achieve its functionality by sending a series of GlobalVector commands, the GlobalVector service provider is a nested service. When a nested service provider receives a command, it checks the configuration report to determine if the command was sent from another service in the same resource (using the command's source header field). If it was, the command can continue as normal, since the commanding service would have already followed the flow control above. If the command is received directly from the resource consumer originating the command, then the nested service must perform additional verification that the consumer has control of the resource. This process is detailed below:

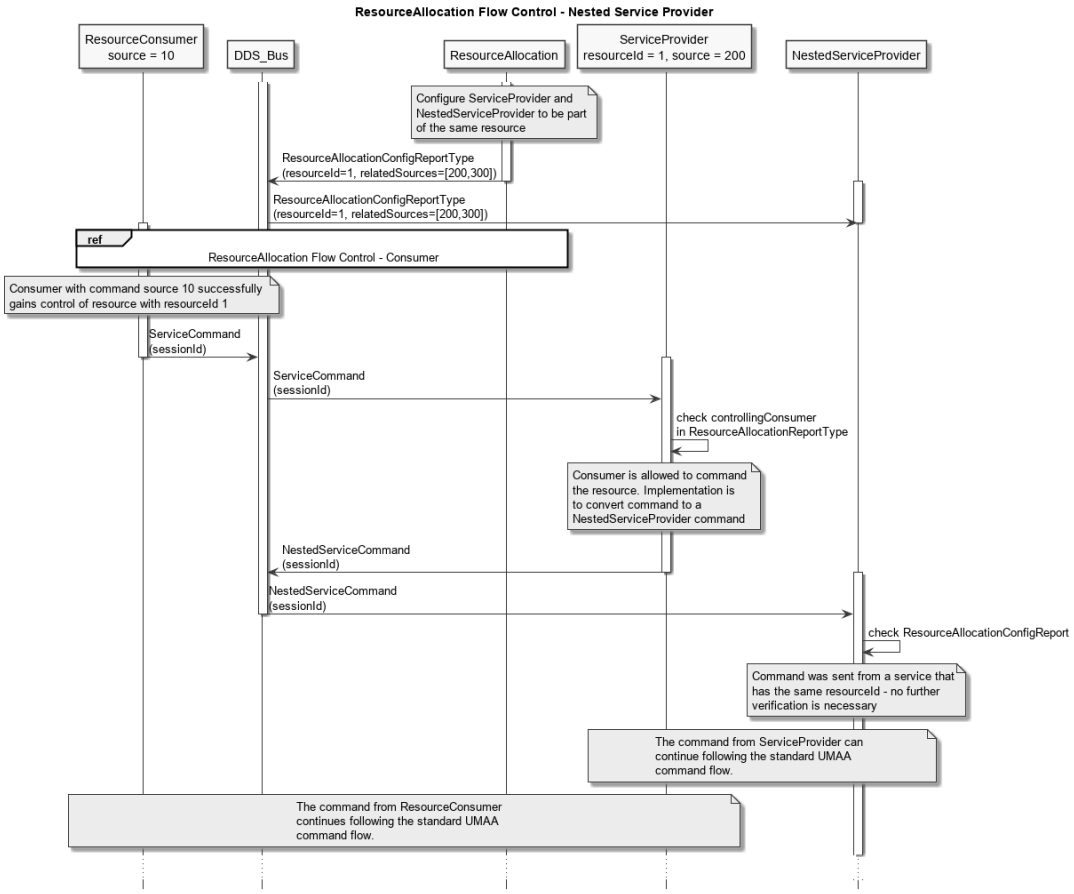

Figure 47: Sequence Diagram of Nested Service Provider Verifying a Command using ResourceAllocation

Ensuring Strong Data Consistency

To avoid report data race conditions, the ResourceAllocation service implementation must take additional steps to ensure strong consistency of ResourceAllocationReport data. Before allowing control of a resource, the ResourceAllocation service must ensure that all subscribers of ResourceAllocationReport have received the most recent sample. This means that service providers will have the data they need to verify the service consumer's command before it is sent. This consistency can be achieved in a number of ways, such as

- Using standard DDS capabilities by configuring the ResourceAllocation report DataWriter to be reliable and using wait_for_acknowledgements()
- Using implementation-specific DDS extensions
- Implementing multiple ResourceAllocation services and using a combination of flow control and other methods above to communicate across a proxy (see examples online)

Avoiding Pitfalls

The ResourceAllocation service exists to deconflict requests coming from multiple service consumers, which would otherwise cause a fight over a particular resource. There is no mechanism in place to prevent a single consumer who has gained control of a resource to issue concurrent commands to multiple service providers within the resource. Doing so is bad engineering practice and should be avoided to ensure command determinism.

Table 193: ResourceAllocation Operations

Service Requests (Inputs)	Service Responses (Outputs)
setResourceAllocation	reportResourceAllocationCommandStatus
queryResourceAllocationCommandAck⊕	reportResourceAllocationCommandAck

Service Requests (Inputs)	Service Responses (Outputs)
cancelResourceAllocationCommand⊕	reportResourceAllocationCancelCommandStatus⊕
queryResourceAllocation⊕	reportResourceAllocation
queryResourceAllocationConfig⊕	reportResourceAllocationConfig
setResourceAllocationPriority	reportResourceAllocationPriorityCommandStatus
queryResourceAllocationPriorityCommandAck⊕	reportResourceAllocationPriorityCommandAck
cancelResourceAllocationPriorityCommand⊕	reportResourceAllocationPriorityCancelCommandStatus⊕
queryResourceAllocationPriority⊕	reportResourceAllocationPriority

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.61.1 reportResourceAllocation

Description: This operation is used to report the current resources and what consumer currently owns each resource.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationReportType

Data Type: ResourceAllocationReportType

Table 194: ResourceAllocationReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
controlInfo	sequence<ResourceAllocationControlInfo> max size = 256	A list of the control information for every resource defined by ResourceAllocation.

6.1.61.2 reportResourceAllocationCommandAck

Description: This operation is used to provide the ResourceAllocation commanded values.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationCommandAckReportType

Data Type: ResourceAllocationCommandAckReportType

Table 195: ResourceAllocationCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		

Attribute Name	Attribute Type	Attribute Description
command	ResourceAllocationCommandType	The source command.

6.1.61.3 reportResourceAllocationCommandStatus

Description: This operation is used to report the status of the current resource allocation command.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationCommandStatusType

Data Type: ResourceAllocationCommandStatusType

Table 196: ResourceAllocationCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.61.4 reportResourceAllocationConfig

Description: This operation is used to report all service provider identifiers that are sharing a single resource.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationConfigReportType

Data Type: ResourceAllocationConfigReportType

Table 197: ResourceAllocationConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
resources→setID	LargeSet<ResourceAllocationDefinitionType>	The configuration of each resource. This attribute is implemented as a large set, see subsection 3.8 for an explanation. The associated topic is UMAA::SO::ResourceAllocation::ResourceAllocationConfigReportTypeResourcesSetElement.

6.1.61.5 reportResourceAllocationPriority

Description: This operation is used to report the current priority ordering of resource consumers.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationPriorityReportType

Data Type: ResourceAllocationPriorityReportType

Table 198: ResourceAllocationPriorityReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
priorities	sequence<ResourceAllocationPriorityInfo> max size = 24	The priority ordering of resource consumers for each resource defined by ResourceAllocation.

6.1.61.6 reportResourceAllocationPriorityCommandAck

Description: This operation is used to provide the ResourceAllocationPriority commanded values.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationPriorityCommandAckReportType

Data Type: ResourceAllocationPriorityCommandAckReportType

Table 199: ResourceAllocationPriorityCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	ResourceAllocationPriorityCommandType	The source command.

6.1.61.7 reportResourceAllocationPriorityCommandStatus

Description: This operation is used to report the status of the current priority command.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationPriorityCommandStatusType

Data Type: ResourceAllocationPriorityCommandStatusType

Table 200: ResourceAllocationPriorityCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.61.8 setResourceAllocation

Description: This operation is used to set the current resource allocation command. The source attribute in the header of this command is used to identify the consumer that is requesting the resource to be allocated.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationCommandType

Data Type: ResourceAllocationCommandType

Table 201: ResourceAllocationCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
duration†	DurationSeconds	Specifies the end of the valid time period for the control session. Once the control session ends, the resource will become available for other requesters to control. If this field is empty, then the duration is assumed to be infinite.
resourceID	IdentifierType	The identifier of the resource to control.

6.1.61.9 setResourceAllocationPriority

Description: This operation is used to set the ordered priority of consumers who will be requesting access. All potential consumers must be on this list to be enabled to request control allocation.

Namespace: UMAA::SO::ResourceAllocation

Topic: ResourceAllocationPriorityCommandType

Data Type: ResourceAllocationPriorityCommandType

Table 202: ResourceAllocationPriorityCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
priorities	sequence<IdentifierType> max size = 100	The priority-ordered (low to high) sequence of resource consumer source identifiers.
resourceID	IdentifierType	The identifier of the resource being set.

6.1.62 ResourceIdentification

The purpose of this service is to report the information of the vehicle and its subsystems.

Table 203: ResourceIdentification Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryResourceAuthorization⊕	reportResourceAuthorization
querySubsystemID⊕	reportSubsystemID
queryVehicleID⊕	reportVehicleID

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.62.1 reportResourceAuthorization

Description: This operation is used to report a list of levels of authorization of the system or subsystem(s).

Namespace: UMAA::SO::ResourceIdentification

Topic: ResourceAuthorizationReportType

Data Type: ResourceAuthorizationReportType

Table 204: ResourceAuthorizationReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
authorizationLevel*	SpecificLOIEnumType	A list of authorized control stations.

6.1.62.2 reportSubsystemID

Description: This operation is used to report the information of the subsystem(s) on-board or off-board a vehicle.

Namespace: UMAA::SO::ResourceIdentification

Topic: SubsystemIDReportType

Data Type: SubsystemIDReportType

Table 205: SubsystemIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		

Attribute Name	Attribute Type	Attribute Description
isControlTransferCapable	boolean	An subsystem or a subset of a system can be transferred control between control systems.
name	StringShortDescription	A name to describe a payload or a subsystem.
type	StringShortDescription	A name to describe the type of payload or subsystem (e.g. cameras, sonar, batteries, GPS, etc.).

6.1.62.3 reportVehicleID

Description: This operation is used to report the information of a vehicle or a system.

Namespace: UMAA::SO::ResourceIdentification

Topic: VehicleIDReportType

Data Type: VehicleIDReportType

Table 206: VehicleIDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
domain	DomainEnumType	The domain that the vehicle is operated under.
isControlTransferCapable	boolean	a vehicle or a system can be transferred control between control systems.
make	StringShortDescription	The manufacture of the vehicle.
model	StringShortDescription	The model of the vehicle.
name	StringShortDescription	The name of the vehicle or a system.
protocol	StringShortDescription	The protocol used to communicate to the vehicle.
type	StringShortDescription	The type of the vehicle or a system.
vehicleNumber	IdentifierType	A unique identification which specifies the string designation for the vehicle. It's a tail number for air vehicle, hull number for maritime vehicle, or registration number for ground vehicle.

6.1.63 SASConfig

The purpose of this service is to define the parameters needed for an unmanned vehicle to configure the Synthetic Aperture Sonar (SAS).

Table 207: SASConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
setSASConfig	reportSASConfigCommandStatus
cancelSASConfig⊕	reportSASCancelConfigCommandStatus⊕

Service Requests (Inputs)	Service Responses (Outputs)
querySASConfig⊕	reportSASConfig
querySASConfigAck⊕	reportSASConfigAck

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.63.1 reportSASConfig

Description: This operation is used to report the current status of the SASConfig service.

Namespace: UMAA::SEM::SASConfig

Topic: SASConfigReportType

Data Type: SASConfigReportType

Table 208: SASConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
autoOffMode	AutoOffModeEnumType	Indicates whether the sonar is shut off or active elements are turned off when auto-off is triggered by an abnormal condition.
name	UniformResourceIdentifier	The name of the SAS configuration to be loaded when transitioning to the READY state.

6.1.63.2 reportSASConfigAck

Description: This operation is used to report the current SAS configuration.

Namespace: UMAA::SEM::SASConfig

Topic: SASConfigAckReportType

Data Type: SASConfigAckReportType

Table 209: SASConfigAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
config	SASConfigCommandType	The source configuration.

6.1.63.3 reportSASConfigCommandStatus

Description: This operation is used to report the status of the current SASConfig command.

Namespace: UMAA::SEM::SASConfig

Topic: SASConfigCommandStatusType

Data Type: SASConfigCommandStatusType

Table 210: SASConfigCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.63.4 setSASConfig

Description: This operation is used to set the SASConfig command.

Namespace: UMAA::SEM::SASConfig

Topic: SASConfigCommandType

Data Type: SASConfigCommandType

Table 211: SASConfigCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
autoOffMode	AutoOffModeEnumType	Indicates whether the sonar is shut off or active elements are turned off when auto-off is triggered by an abnormal condition.
name	UniformResourceIdentifier	The name of the SAS configuration to be loaded when transitioning to the READY state.

6.1.64 SASControl

The purpose of this service is to define the parameters needed for an unmanned vehicle to control the Synthetic Aperture Sonar (SAS) operational state.

Table 212: SASControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setSAS	reportSASCommandStatus
querySASCommandAck⊕	reportSASCommandAck

Service Requests (Inputs)	Service Responses (Outputs)
cancelSASCommand⊕	reportSASCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.64.1 reportSASCommandAck

Description: This operation is used to provide the SAS commanded values.

Namespace: UMAA::SEM::SASControl

Topic: SASCommandAckReportType

Data Type: SASCommandAckReportType

Table 213: SASCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	SASCommandType	The source command.

6.1.64.2 reportSASCommandStatus

Description: This operation is used to report the status of the current SAS command.

Namespace: UMAA::SEM::SASControl

Topic: SASCommandStatusType

Data Type: SASCommandStatusType

Table 214: SASCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.64.3 setSAS

Description: This operation is used to set the SAS command.

Namespace: UMAA::SEM::SASControl

Topic: SASCommandType

Data Type: SASCommandType

Table 215: SASCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
targetState	ActivationStateTargetEnumType	Defines the target activation state.

6.1.65 SASStatus

The purpose of this service is to report the status of a Synthetic Aperture Sonar (SAS) sensor.

Table 216: SASStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySASStatus⊕	reportSASStatus

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.65.1 reportSASStatus

Description: This operation is used to report the current status of the SASStatus service.

Namespace: UMAA::SEM::SASStatus

Topic: SASStatusReportType

Data Type: SASStatusReportType

Table 217: SASStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
currentState	ActivationStateEnumType	Defines the current activation state.

6.1.66 SoundVelocityProfileReport

The purpose of this service is to provide information about the state of the sound velocity profile.

Table 218: SoundVelocityProfileReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySoundVelocityProfile⊕	reportSoundVelocityProfile

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.66.1 reportSoundVelocityProfile

Description: This operation is used to report the current sound velocity profile.

Namespace: UMAA::SA::SoundVelocityProfileReport

Topic: SoundVelocityProfileReportType

Data Type: SoundVelocityProfileReportType

Table 219: SoundVelocityProfileReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
soundSpeed	sequence<DepthSpeedPairType> max size = 1024	The current sound speed.

6.1.67 StillImageStatus

The purpose of this service is to provide a means to obtain images from the camera. While this service reports each image individually, the Events service can be used to automatically report images at a specified rate thereby simulating video (typically done to create an MJPEG video stream).

Table 220: StillImageStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryStillImage⊕	reportStillImage

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.67.1 reportStillImage

Description: This operation is used to report the current status still image.

Namespace: UMAA::SA::StillImageStatus

Topic: StillImageReportType

Data Type: StillImageReportType

Table 221: StillImageReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
altitudeAGL†	DistanceAGL	The current height above ground level of the image.
altitudeASF†	DistanceASF	The current height above the sea floor of the image.
altitudeGeodetic†	GeodeticAltitude	The current height above the ellipsoid of the image.
altitudeMSL†	MSLAltitude	The current height above the Geoid (Mean Sea Level) of the image.
depth†	DistanceBSL	The current depth of the image.
imageFormat	ImageFormatEnumType	Format of the image file.
imageName†	StringShortDescription	Short descriptive name of the image.
imageURI	UniformResourceIdentifier	The current location of the still image.
position†	GeoPosition3DWGS84	Position of the image.
transform†	WorldTransformType	An optional affine transformation to describe location, scale, and rotation of image (not applicable if image format is GEOTIFF).
contactID*	NumericGUID	Identifier for the contact contained within the image.
imageID*	NumericGUID	Identifier of the image.

6.1.68 SyncDataControl

The purpose of this service is to define the parameters needed for an unmanned vehicle to control synchronization of data.

Table 222: SyncDataControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setSyncData	reportSyncDataCommandStatus
querySyncDataCommandAck⊕	reportSyncDataCommandAck
cancelSyncDataCommand⊕	reportSyncDataCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.68.1 reportSyncDataCommandAck

Description: This operation is used to provide the SyncData commanded values.

Namespace: UMAA::SO::SyncDataControl

Topic: SyncDataCommandAckReportType

Data Type: SyncDataCommandAckReportType

Table 223: SyncDataCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	SyncDataCommandType	The source command.

6.1.68.2 reportSyncDataCommandStatus

Description: This operation is used to report the status of the current SyncData command.

Namespace: UMAA::SO::SyncDataControl

Topic: SyncDataCommandStatusType

Data Type: SyncDataCommandStatusType

Table 224: SyncDataCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.68.3 setSyncData

Description: This operation is used to set the SyncData command.

Namespace: UMAA::SO::SyncDataControl

Topic: SyncDataCommandType

Data Type: SyncDataCommandType

Table 225: SyncDataCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		

6.1.69 TamperDetectionControl

The purpose of this service is to provide a mechanism to enable or disable tamper detection.

Table 226: TamperDetectionControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setTamperDetection	reportTamperDetectionCommandStatus
queryTamperDetectionCommandAck⊕	reportTamperDetectionCommandAck
cancelTamperDetectionCommand⊕	reportTamperDetectionCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.69.1 reportTamperDetectionCommandAck

Description: This operation is used to provide the TamperDetection commanded values.

Namespace: UMAA::SO::TamperDetectionControl

Topic: TamperDetectionCommandAckReportType

Data Type: TamperDetectionCommandAckReportType

Table 227: TamperDetectionCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	TamperDetectionCommandType	The source command.

6.1.69.2 reportTamperDetectionCommandStatus

Description: This operation is used to report the status of the tamper detection command.

Namespace: UMAA::SO::TamperDetectionControl

Topic: TamperDetectionCommandStatusType

Data Type: TamperDetectionCommandStatusType

Table 228: TamperDetectionCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.69.3 setTamperDetection

Description: This operation is used to enable or disable tamper detection and to clear the previously reported activities.

Namespace: UMAA::SO::TamperDetectionControl

Topic: TamperDetectionCommandType

Data Type: TamperDetectionCommandType

Table 229: TamperDetectionCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
state	TamperDetectionStateEnumType	A desired state of tamper detection.

6.1.70 TamperDetectionStatus

The purpose of this service is to provide a mechanism to report when tampering has occurred.

Table 230: TamperDetectionStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryTamperDetection⊕	reportTamperDetection

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.70.1 reportTamperDetection

Description: This operation is used to report the current status of anti-tamper detection. Detected activities are continuously reported until cleared by a setTamperDetectionState message.

Namespace: UMAA::SO::TamperDetectionStatus

Topic: TamperDetectionReportType

Data Type: TamperDetectionReportType

Table 231: TamperDetectionReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
descriptor	StringShortDescription	Description of tamper detection activities.
electricalTamper	boolean	Current result of electrical tamper detection.

Attribute Name	Attribute Type	Attribute Description
hardwareTamper	boolean	Current result of hardware tamper detection.
networkInstrusion	boolean	Current result of network tamper detection.
otherTamper	boolean	Current result of tamper detection.
state	TamperDetectionStateEnumType	The current state of tamper detection.

6.1.71 TaskPlanAssignmentControl

The purpose of this service is to provide operations and interfaces required to assign tasks and objectives to available resources.

Table 232: TaskPlanAssignmentControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setTaskPlanAssignment	reportTaskPlanAssignmentCommandStatus
queryTaskPlanAssignmentCommandAck⊕	reportTaskPlanAssignmentCommandAck
cancelTaskPlanAssignmentCommand⊕	reportTaskPlanAssignmentCancelCommandStatus⊕

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.71.1 reportTaskPlanAssignmentCommandAck

Description: This operation is used to provide the TaskPlanAssignment commanded values.

Namespace: UMAA::MM::TaskPlanAssignmentControl

Topic: TaskPlanAssignmentCommandAckReportType

Data Type: TaskPlanAssignmentCommandAckReportType

Table 233: TaskPlanAssignmentCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
command	TaskPlanAssignmentCommandType	The source command.

6.1.71.2 reportTaskPlanAssignmentCommandStatus

Description: This operation is used to provide the current status of assigning a task to a resource.

Namespace: UMAA::MM::TaskPlanAssignmentControl

Topic: TaskPlanAssignmentCommandStatusType

Data Type: TaskPlanAssignmentCommandStatusType

Table 234: TaskPlanAssignmentCommandStatusType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatus		

6.1.71.3 setTaskPlanAssignment

Description: This operation is used to assign a task to an available resource.

Namespace: UMAA::MM::TaskPlanAssignmentControl

Topic: TaskPlanAssignmentCommandType

Data Type: TaskPlanAssignmentCommandType

Table 235: TaskPlanAssignmentCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
missionID	NumericGUID	Identifies the mission plan.
resourceIDs	sequence<IdentifierType> max size = 256	Identifies the resources that are assigned to a specific task.
taskID	NumericGUID	Identifies the associated task within the mission plan.

6.1.72 TaskPlanAssignmentReport

The purpose of this service is to assign a task plan to a particular resource or set of resources required to accomplish the task plan.

Table 236: TaskPlanAssignmentReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryTaskPlanAssignment⊕	reportTaskPlanAssignment

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.72.1 reportTaskPlanAssignment

Description: This operation is used to provide the current assignment of a task plan to a resource.

Namespace: UMAA::MM::TaskPlanAssignmentReport

Topic: TaskPlanAssignmentReportType

Data Type: TaskPlanAssignmentReportType

Table 237: TaskPlanAssignmentReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
resourceIDs	sequence<IdentifierType> max size = 256	Identifies the resources that are assigned to a specific task.
missionID*	NumericGUID	Identifies the associated mission plan.
taskID*	NumericGUID	Identifies the associated task within the mission plan.

6.1.73 TerrainReport

This service provides terrain related data.

Table 238: TerrainReport Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryTerrain⊕	reportTerrain

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.73.1 reportTerrain

Description: This operation is used to report the current status of the Terrain service.

Namespace: UMAA::SA::TerrainReport

Topic: TerrainReportType

Data Type: TerrainReportType

Table 239: TerrainReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		

Attribute Name	Attribute Type	Attribute Description
terrainDepthURI	UniformResourceIdentifier	The current location of the GeoTIFF, which contains terrain depth values around the vehicle. Terrain depth values are set to a negative number to indicate that the points are either not within the sonar field of view or return no measurable value. Information about resolution, spatial extent, and coordinate reference system are embedded within the GeoTIFF structure. The GeoTIFF projection is typically UTM. The extent is locally bounded around the vehicle. The values for depths are represented as positive 32-bit floating point numbers.

6.1.74 WaterCharacteristicsStatus

The purpose of this service is to report the current characteristics of the water column around the vehicle.

Table 240: WaterCharacteristicsStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryWaterCharacteristics⊕	reportWaterCharacteristics

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.74.1 reportWaterCharacteristics

Description: This operation is used to report the data parameters for the WaterCharacteristics service.

Namespace: UMAA::SA::WaterCharacteristicsStatus

Topic: WaterCharacteristicsReportType

Data Type: WaterCharacteristicsReportType

Table 241: WaterCharacteristicsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
conductivity†	Conductivity	The measured or computed conductivity.
density†	Density	The measured density.
depth	DistanceBSL	The measured depth below sea level.
pressure†	PressurePascals	The measured pressure.
salinity†	Salinity	The measured salinity.
soundVelocity†	Speed	The measured velocity of sound.
temperature†	Temperature	The measured temperature.
turbidity†	Turbidity	The measured cloudiness of the water.

6.1.75 WeatherStatus

The purpose of this service is to provide the current state of the weather.

Table 242: WeatherStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryWeather⊕	reportWeather

See Section 6.1 for an explanation of the inputs and outputs marked with a ⊕.

6.1.75.1 reportWeather

Description: This operation is used to report the current status of the weather.

Namespace: UMAA::SA::WeatherStatus

Topic: WeatherReportType

Data Type: WeatherReportType

Table 243: WeatherReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
airTemperature†	AirTemperature	The ambient air temperature.
barometricPressure†	WeatherBarometricPressure	A measure of air pressure that correlates with weather and altitude.
cloudiness†	CloudCoverEnumType	The current state of cloud cover.
dewPoint†	DewPointTemperature	The temperature at which water vapor condenses into water.
icingSeverity†	WeatherSeverityEnumType	The extent of icing present.
precipitation†	PrecipitationEnumType	The type of precipitation.
relativeHumidity†	Percent	The amount of water vapor in the air.
thunderstormPotential†	Percent	The current probability that there is a thunderstorm.
visibility†	Distance	The distance at which an object or light can be clearly discerned.
waterTemperature†	WaterTemperature	The water temperature at surface.

6.2 Common Data Types

Common data types define DDS types that are referenced throughout the UMAA model. These DDS types are considered common because they can be re-used as the data type for many attributes defined in service interface topics, interface topics, and other common data types. These data types are not intended to be directly published to/subscribed as DDS topics.

6.2.1 UCSMDEInterfaceSet

Namespace: UMAA::UCSMDEInterfaceSet

Description: Defines the common UCSMDE Interface Set Message Fields.

Table 244: UCSMDEInterfaceSet Structure Definition

Attribute Name	Attribute Type	Attribute Description
timeStamp	DateTime	The origination time of the data being conveyed in the message, or as close to the data or command generation time as is reasonably possible.

6.2.2 UMAACommand

Namespace: UMAA::UMAACommand

Description: Defines the common UMAA Command Message Fields.

Table 245: UMAACommand Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	IdentifierType	The unique identifier of the originating source of the command interface.
destination*	IdentifierType	The unique identifier of the destination of the command interface.
sessionID*	NumericGUID	The unique identifier for the session.

6.2.3 UMAAStatus

Namespace: UMAA::UMAAStatus

Description: Defines the common UMAA Status Message Fields.

Table 246: UMAAStatus Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	IdentifierType	The unique identifier of the originating source of the status interface.

6.2.4 UMAACommandStatusBase

Namespace: UMAA::UMAACommandStatusBase

Description: Defines the common UMAA Command Status Base Message Fields.

Table 247: UMAACommandStatusBase Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	IdentifierType	The unique identifier of the originating source of the command status interface.
sessionID*	NumericGUID	The unique identifier for the session.

6.2.5 UMAACommandStatus

Namespace: UMAA::UMAACommandStatus

Description: Defines the common UMAA Command Status Message Fields.

Table 248: UMAACommandStatus Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
commandStatus	CommandStatusEnumType	The status of the command.
commandStatusReason	CommandStatusReasonEnumType	The reason for the status of the command.
logMessage	StringLongDescription	Human-readable description related to response. Systems should not parse or use any information from this for processing purposes.

6.2.6 DateTime

Namespace: UMAA::Common::Measurement::DateTime

Description: Describes an absolute time. Conforms with POSIX time standard (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.

Table 249: DateTime Structure Definition

Attribute Name	Attribute Type	Attribute Description
seconds	DateTimeSeconds	The number of seconds offset from the standard POSIX (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.
nanoseconds	DateTimeNanoSeconds	The number of nanoseconds elapsed within the current DateTimeSecond.

6.2.7 Acceleration3DPlatformXYZ

Namespace: UMAA::Common::Measurement::Acceleration3DPlatformXYZ

Description: Specifies the platform's rate of change of velocity with respect to time in the x, y, and z axes.

Table 250: Acceleration3DPlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
xAccel	AccelerationScalar	Specifies the vehicle's rate of change of velocity with respect to time in the x-axis.
yAccel	AccelerationScalar	Specifies the vehicle's rate of change of velocity with respect to time in the y-axis.
zAccel	AccelerationScalar	Specifies the vehicle's rate of change of velocity with respect to time in the z-axis.

6.2.8 AirSpeedRequirement

Namespace: UMAA::Common::Speed::AirSpeedRequirement

Description: Defines the speed through air.

Table 251: AirSpeedRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	IndicatedAirspeed	Specifies speed through air.
speedTolerance†	AirSpeedTolerance	Specifies the tolerance for a speed through air.

6.2.9 AirSpeedRequirementVariantType

Namespace: UMAA::Common::Speed::AirSpeedRequirementVariantType

Description: Defines the speed through air.

Table 252: AirSpeedRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	AirSpeedRequirement	Specifies speed through air.

6.2.10 AirSpeedTolerance

Namespace: UMAA::Common::Speed::AirSpeedTolerance

Description: Defines the speed through air tolerance.

Table 253: AirSpeedTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	IndicatedAirspeed	Specifies the lower limit of allowable values for the air speed.
upperlimit	IndicatedAirspeed	Specifies the upper limit of allowable values for the air speed.

6.2.11 AirSpeedVariantType

Namespace: UMAA::Common::Speed::AirSpeedVariantType

Description: Defines the speed through air.

Table 254: AirSpeedVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	IndicatedAirspeed	Specifies speed through air.

6.2.12 AlphaXPlatformType

Namespace: UMAA::Common::Orientation::AlphaXPlatformType

Description: A requirement that specifies an alpha angle relative to the Platform coordinate system.

Table 255: AlphaXPlatformType Structure Definition

Attribute Name	Attribute Type	Attribute Description
alpha	RollAngle	Defines an alpha angle relative to the Platform coordinate system.

6.2.13 AltitudeAGLRequirementType

Namespace: UMAA::Common::Measurement::AltitudeAGLRequirementType

Description: Defines the distance above ground level.

Table 256: AltitudeAGLRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	DistanceAGL	Specifies the distance above ground level.
altitudeTolerance†	AltitudeAGLToleranceType	Specifies the tolerance for the distance above ground level.

6.2.14 AltitudeAGLRequirementVariantType

Namespace: UMAA::Common::Measurement::AltitudeAGLRequirementVariantType

Description: The height above ground level.

Table 257: AltitudeAGLRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	AltitudeAGLRequirementType	Specifies the distance above ground level.

6.2.15 AltitudeAGLToleranceType

Namespace: UMAA::Common::Measurement::AltitudeAGLToleranceType

Description: Defines the distance above ground level tolerance.

Table 258: AltitudeAGLToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	DistanceAGL	Specifies the lower limit of allowable values for the distance above ground level.

Attribute Name	Attribute Type	Attribute Description
upperlimit	DistanceAGL	Specifies the upper limit of allowable values for the distance above ground level.

6.2.16 AltitudeAGLVariantType

Namespace: UMAA::Common::Measurement::AltitudeAGLVariantType

Description: The height above ground level.

Table 259: AltitudeAGLVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	DistanceAGL	Specifies the distance above ground level.

6.2.17 AltitudeASFRequirementType

Namespace: UMAA::Common::Measurement::AltitudeASFRequirementType

Description: Defines the height above sea floor.

Table 260: AltitudeASFRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	DistanceASF	Specifies the height above sea floor.
altitudeTolerance†	AltitudeASFToleranceType	Specifies the tolerance for the height above sea floor.

6.2.18 AltitudeASFRequirementVariantType

Namespace: UMAA::Common::Measurement::AltitudeASFRequirementVariantType

Description: The height above sea floor.

Table 261: AltitudeASFRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	AltitudeASFRequirementType	The height above the sea floor.

6.2.19 AltitudeASFToleranceType

Namespace: UMAA::Common::Measurement::AltitudeASFToleranceType

Description: Defines the height above sea floor tolerance.

Table 262: AltitudeASFToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	DistanceASF	Specifies the lower limit of allowable values for the height above sea floor.
upperlimit	DistanceASF	Specifies the upper limit of allowable values for the height above sea floor.

6.2.20 AltitudeASFVariantType

Namespace: UMAA::Common::Measurement::AltitudeASFVariantType

Description: The height above sea floor.

Table 263: AltitudeASFVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	DistanceASF	The height above the sea floor.

6.2.21 AltitudeGeodeticRequirementType

Namespace: UMAA::Common::Measurement::AltitudeGeodeticRequirementType

Description: Defines the geodetic height above the ellipsoid.

Table 264: AltitudeGeodeticRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	GeodeticAltitude	Specifies the geodetic height above the ellipsoid.
altitudeTolerance†	AltitudeGeodeticToleranceType	Specifies the tolerance for the geodetic height above the ellipsoid.

6.2.22 AltitudeGeodeticRequirementVariantType

Namespace: UMAA::Common::Measurement::AltitudeGeodeticRequirementVariantType

Description: The geodetic height above the ellipsoid.

Table 265: AltitudeGeodeticRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	AltitudeGeodeticRequirementType	The altitude above the reference ellipsoid.

6.2.23 AltitudeGeodeticToleranceType

Namespace: UMAA::Common::Measurement::AltitudeGeodeticToleranceType

Description: Defines the geodetic height above the ellipsoid tolerance.

Table 266: AltitudeGeodeticToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	GeodeticAltitude	Specifies the lower limit of allowable values for the geodetic height above the ellipsoid.
upperlimit	GeodeticAltitude	Specifies the upper limit of allowable values for the geodetic height above the ellipsoid.

6.2.24 AltitudeGeodeticVariantType

Namespace: UMAA::Common::Measurement::AltitudeGeodeticVariantType

Description: The geodetic height above the ellipsoid.

Table 267: AltitudeGeodeticVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	GeodeticAltitude	The altitude above the reference ellipsoid.

6.2.25 AltitudeMSLRequirementType

Namespace: UMAA::Common::Measurement::AltitudeMSLRequirementType

Description: Defines the orthometric height above the Geoid (Mean Sea Level).

Table 268: AltitudeMSLRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	MSLAltitude	Specifies the orthometric height above the Geoid (Mean Sea Level).
altitudeTolerance†	AltitudeMSLToleranceType	Specifies the tolerance for the orthometric height above the Geoid (Mean Sea Level).

6.2.26 AltitudeMSLRequirementVariantType

Namespace: UMAA::Common::Measurement::AltitudeMSLRequirementVariantType

Description: The orthometric height above the Geoid (Mean Sea Level).

Table 269: AltitudeMSLRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	AltitudeMSLRequirementType	The orthometric height above the Geoid (Mean Sea Level).

6.2.27 AltitudeMSLToleranceType

Namespace: UMAA::Common::Measurement::AltitudeMSLToleranceType

Description: Defines the orthometric height above the Geoid (Mean Sea Level) tolerance.

Table 270: AltitudeMSLToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	MSLAltitude	Specifies the lower limit of allowable values for the orthometric height above the Geoid (Mean Sea Level).
upperlimit	MSLAltitude	Specifies the upper limit of allowable values for the orthometric height above the Geoid (Mean Sea Level).

6.2.28 AltitudeMSLVariantType

Namespace: UMAA::Common::Measurement::AltitudeMSLVariantType

Description: The orthometric height above the Geoid (Mean Sea Level).

Table 271: AltitudeMSLVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	MSLAltitude	The orthometric height above the Geoid (Mean Sea Level).

6.2.29 AltitudeRateASFRequirementType

Namespace: UMAA::Common::Measurement::AltitudeRateASFRequirementType

Description: Defines the change in altitude as a function of time.

Table 272: AltitudeRateASFRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitudeRate	SpeedASF	Specifies the change in altitude as a function of time.
altitudeRateTolerance†	AltitudeRateASFTolerance Type	Specifies the altitude rate tolerance.

6.2.30 AltitudeRateASFRequirementVariantType

Namespace: UMAA::Common::Measurement::AltitudeRateASFRequirementVariantType

Description: The change in altitude as a function of time.

Table 273: AltitudeRateASFRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitudeRate	AltitudeRateASFRequirementType	Specifies the change in altitude as a function of time.

6.2.31 AltitudeRateASFToleranceType

Namespace: UMAA::Common::Measurement::AltitudeRateASFToleranceType

Description: Defines the altitude rate tolerance.

Table 274: AltitudeRateASFToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	SpeedASF	Specifies the lower limit of allowable values for the change in altitude as a function of time.
upperlimit	SpeedASF	Specifies the upper limit of allowable values for the change in altitude as a function of time.

6.2.32 AnnulusSectorRequirementType

Namespace: UMAA::MM::BaseType::AnnulusSectorRequirementType

Description: A requirement that specifies the area of the annulus sector.

Table 275: AnnulusSectorRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
annulusSectorTolerance†	AnnulusSectorToleranceType	Specifies the tolerance for the annulus sector.
maxRange	Distance	Maximum range of the annulus sector.
minRange	Distance	Minimum range of the annulus sector.
sector	BearingSectorVariantType	Specifies the bearing sector.

6.2.33 AnnulusSectorToleranceType

Namespace: UMAA::MM::BaseType::AnnulusSectorToleranceType

Description: Defines the annulus sector tolerance.

Table 276: AnnulusSectorToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
limit	Distance	Specifies the amount of error in position allowed from the annulus sector.

6.2.34 AreaRandomWalkObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::AreaRandomWalkObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the area random walk objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::AreaRandomWalkObjectiveDetailedStatusType

Table 277: AreaRandomWalkObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isAreaAchieved	boolean	Indicates that the area requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing area random walk (True = system is executing area random walk, False = system is in transit).
timePatternAchieved	DateTime	Defines the absolute time at which the area random walk is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the area random walk is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.

Attribute Name	Attribute Type	Attribute Description
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.35 AreaRandomWalkObjectiveType

Namespace: UMAA::MM::BaseType::AreaRandomWalkObjectiveType

Description: The goal of the area random walk objective is to execute a random walk maneuver within a given area. This structure is used to specify the area where the random walk must be conducted. The area random walk objective is achieved by having the vehicle execute random vectors at a specified elevation (or current elevation if not specified) while maintaining the vehicle location within a defined area. The area is defined by specifying the vertices of a polygon, and the random vectors within this area can be configured by specifying min/max speeds and min/max time on course. Area and elevation include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::AreaRandomWalkObjectiveType

Figure 48: An Area Random Walk

Table 278: AreaRandomWalkObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
area	PolygonAreaRequirementType	Defines the area the vehicle must stay in while executing the random walk maneuver.
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the random walk portion of the objective. If not specified, duration is not used to determine when the random walk maneuver is complete.

Attribute Name	Attribute Type	Attribute Description
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the random walk. If not specified, the maneuver is performed at the current elevation.
maxSpeed	SpeedVariantType	Defines the maximum vehicle speed on a given vector.
maxTimeOnCourse	DurationSeconds	Defines the maximum time spent on a given vector.
minSpeed	SpeedVariantType	Defines the minimum vehicle speed on a given vector.
minTimeOnCourse	DurationSeconds	Defines the minimum time spent on a given vector.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the area before transitioning to the random walk maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the random walk area location before transitioning to the random walk maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.36 BallastFillType

Namespace: UMAA::EO::BallastTank::BallastFillType

Description: Union Type. The desired fill amount of the ballast tank.

Table 279: BallastFillType Union(s)

Type Name	Type Description
BallastMassType	The desired filled or empty level of the ballast tank measured by mass.
LevelType	Defines the desired percentage filled or empty level of the ballast tank.

6.2.37 BallastMassType

Namespace: UMAA::Common::Measurement::BallastMassType

Description: The desired filled or empty level of the ballast tank measured by mass.

Table 280: BallastMassType Structure Definition

Attribute Name	Attribute Type	Attribute Description
mass	Mass	Specifies the desired filled or empty level of the ballast tank measured by mass.

6.2.38 BallastPumpFlowRateType

Namespace: UMAA::EO::BallastTank::BallastPumpFlowRateType

Description: Union Type. The desired flow rate to fill or empty the ballast pump.

Table 281: BallastPumpFlowRateType Union(s)

Type Name	Type Description
MassBallastFlowRateType	The desired flow rate to fill or empty the ballast pump measured by mass.
VolumeBallastFlowRateType	The desired flow rate to fill or empty the ballast pump measured by volume.

6.2.39 BeamWidthSpecsType

Namespace: UMAA::SEM::IlluminatorSpecs::BeamWidthSpecsType

Description: The structure is used to describe the specifications of beam width.

Table 282: BeamWidthSpecsType Structure Definition

Attribute Name	Attribute Type	Attribute Description
maxBeamWidth	IlluminatorBeamWidth	The maximum horizontal field of illumination.
minBeamWidth	IlluminatorBeamWidth	The minimum horizontal field of illumination.

6.2.40 BearingSectorGuideCourseVariantType

Namespace: UMAA::Common::Orientation::BearingSectorGuideCourseVariantType

Description: This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing. The bearing sector is defined relative to a guide (e.g., a contact) location with respect to the guide's course.

Table 283: BearingSectorGuideCourseVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
endBearing	HeadingTarget	Provides the end bearing of the bearing sector. The endBearing is defined relative to a guide location with respect to the guide's course.
startBearing	HeadingTarget	Provides the start bearing of the bearing sector. The startBearing is defined relative to a guide location with respect to the guide's course.

6.2.41 BearingSectorMagneticNorthVariantType

Namespace: UMAA::Common::Orientation::BearingSectorMagneticNorthVariantType

Description: This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing. The bearing sector is defined relative to a guide (e.g., contact) location with respect to magnetic north.

Table 284: BearingSectorMagneticNorthVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
endBearing	HeadingMagneticNorth	Provides the end bearing of the bearing sector. The endBearing is defined relative to a guide location with respect to magnetic north.
startBearing	HeadingMagneticNorth	Provides the start bearing of the bearing sector. The startBearing is defined relative to a guide location with respect to magnetic north.

6.2.42 BearingSectorTrueNorthVariantType

Namespace: UMAA::Common::Orientation::BearingSectorTrueNorthVariantType

Description: This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing. The bearing sector is defined relative to a guide (e.g., contact) location with respect to true north.

Table 285: BearingSectorTrueNorthVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
endBearing	HeadingTrueNorthAngle	Provides the end bearing of the bearing sector. The endBearing is defined relative to a guide location with respect to true north.
startBearing	HeadingTrueNorthAngle	Provides the start bearing of the bearing sector. The startBearing is defined relative to a guide location with respect to true north.

6.2.43 BearingSectorVariantType

Namespace: UMAA::Common::Orientation::BearingSectorVariantType

Description: Union Type. This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing.

Table 286: BearingSectorVariantType Union(s)

Type Name	Type Description
BearingSectorGuideCourseVariantType	This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing. The bearing sector is defined relative to a guide (e.g., a contact) location with respect to the guide's course.
BearingSectorMagneticNorthVariantType	This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing. The bearing sector is defined relative to a guide (e.g., contact) location with respect to magnetic north.
BearingSectorTrueNorthVariantType	This structure defines a bearing sector, which is defined to be the sector created by rotating in a positive sense from the startBearing to the endBearing. The bearing sector is defined relative to a guide (e.g., contact) location with respect to true north.

6.2.44 BetaYPlatformType

Namespace: UMAA::Common::Orientation::BetaYPlatformType

Description: A requirement that specifies a beta angle relative to the Platform coordinate system.

Table 287: BetaYPlatformType Structure Definition

Attribute Name	Attribute Type	Attribute Description
beta	PitchHalfAngle	Defines a beta angle relative to the Platform coordinate system.

6.2.45 COLREGSConfigurationType

Namespace: UMAA::MO::HazardAvoidanceConfig::COLREGSConfigurationType

Description: This structure describes COLREGS configuration.

Table 288: COLREGSConfigurationType Structure Definition

Attribute Name	Attribute Type	Attribute Description
dangerRange	Distance	The distance from a contact at which the vehicle may prioritize maneuvering action (reactive) to avoid collision over other mission maneuvering objectives (route, hover, payload, etc.). A contact within this range could create an in extremis situation.
doubtRange	Distance	The distance from a contact at which the vehicle shall evaluate the other contact with respect to COLREGS compliance, and shall take action to remain outside of minimum standoff distance if a ROC exists.

Attribute Name	Attribute Type	Attribute Description
influenceRange	Distance	The distance from a contact at which the vehicle shall evaluate contacts for a risk of collision (ROC). If a ROC exists and the vehicle is the give-way vessel, the vehicle shall take action.

6.2.46 CircleObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::CircleObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the circle objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::CircleObjectiveDetailedStatusType

Table 289: CircleObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isCrossTrackLimitAchieved	boolean	Indicates that the cross track limit requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing the circle maneuver (True = system is executing the circle maneuver, False = system is in transit).
isSpeedAchieved	boolean	Indicates that the speed requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
referencePosition	GeoPosition2D	Defines the reference position for the circle pattern. (Important if reference position was not defined in original objective).
timePatternAchieved	DateTime	Defines the absolute time at which the circle is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the circle is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.47 CircleObjectiveType

Namespace: UMAA::MM::BaseType::CircleObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for following the circle pattern. The circle objective is achieved by having the vehicle execute the circle pattern maneuver at a specified

elevation (or current elevation if not specified) as specified by the center position and radius, with the defined speed, trackTolerance, and turnDirection. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::CircleObjectiveType

Table 290: CircleObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the remaining pattern portion of the objective. If not specified, duration is not used to determine when the circle maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the circle maneuver. If not specified, the maneuver is performed at the current elevation.
loops†	SizeReal	Defines the number of loops around the circle pattern to execute. If not specified, the loops attribute is not used to determine when the circle maneuver is complete.
position†	GeoPosition2D	Defines the reference position for the circle pattern. If not specified, the reference position is the current vehicle position.
radius	Distance	Defines the radius for the circle pattern.
speed	SpeedRequirementVariantType	Defines the vehicle speed to maintain while executing the circle maneuver.
trackTolerance	DistanceRequirementType	Defines the maximum allowable cross track error while executing the circle maneuver.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the circle pattern location before transitioning to the circle maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the circle pattern location before transitioning to the circle maneuver.
turnDirection	WaterTurnDirectionEnumType	Defines the turn direction while executing the circle maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.48 ContactHazardAvoidanceType

Namespace: UMAA::MO::HazardAvoidanceConfig::ContactHazardAvoidanceType

Description: This structure describes hazard avoidance configuration for the associated contact.

Table 291: ContactHazardAvoidanceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
colregsConfig†	COLREGSConfigurationType	The COLREGS configuration for the associated contact ID. If not defined, COLREGS is turned off for this contact.
minimumStandoff	Distance	The distance from a contact, that is equal to the "standing order" for minimum separation, that shall not be violated by the vehicle.

6.2.49 CovarianceAccelerationPlatformXYZType

Namespace: UMAA::Common::Measurement::CovarianceAccelerationPlatformXYZType

Description: An Entity that specifies a range of uncertainty for values of an Observable, usually an Observable associated with kinematics. They are used with a corresponding Observable as attributes of ResourceTypes which provide such state information. An ObservableError may be expressed as a lower and upper limit, delta values, or squares of the Observable, depending on the Frame of Reference and Coordinate Set used for the Observable. The number of values which constitutes the ObservableError is also dependent on the dimensionality of the Observable. Since Frame of Reference, Coordinates, and Dimensionality are determined in refinements specified in the Logical Data Model, the selection of attributes for specializations of this Entity are undefined in the CDM.

Table 292: CovarianceAccelerationPlatformXYZType Structure Definition

Attribute Name	Attribute Type	Attribute Description
axAx	CovarAccelPlatformXYZ	X-X angular acceleration error covariance.
axAy†	CovarAccelPlatformXYZ	X-Y angular acceleration error covariance.
axAz†	CovarAccelPlatformXYZ	X-Z angular acceleration error covariance.
ayAy	CovarAccelPlatformXYZ	Y-Y angular acceleration error covariance.
ayAz†	CovarAccelPlatformXYZ	Y-Z angular acceleration error covariance.
azAz	CovarAccelPlatformXYZ	Z-Z angular acceleration error covariance.

6.2.50 CovarianceOrientationAccelerationPlatformXYZType

Namespace: UMAA::Common::Measurement::CovarianceOrientationAccelerationPlatformXYZType

Description: An Entity that specifies a range of uncertainty for values of an Observable, usually an Observable associated with kinematics. They are used with a corresponding Observable as attributes of ResourceTypes which provide such state information. An ObservableError may be expressed as a lower and upper limit, delta values, or squares of the Observable, depending on the Frame of Reference and Coordinate Set used for the Observable. The number of values which

constitutes the ObservableError is also dependent on the dimensionality of the Observable. Since Frame of Reference, Coordinates, and Dimensionality are determined in refinements specified in the Logical Data Model, the selection of attributes for specializations of this Entity are undefined in the CDM.

Table 293: CovarianceOrientationAccelerationPlatformXYZType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rxRx	CovarOrientationAccelPlatformXYZ	X-X angular acceleration error covariance.
rxRy†	CovarOrientationAccelPlatformXYZ	X-Y angular acceleration error covariance.
rxRz†	CovarOrientationAccelPlatformXYZ	X-Z angular acceleration error covariance.
ryRy	CovarOrientationAccelPlatformXYZ	Y-Y angular acceleration error covariance.
ryRz†	CovarOrientationAccelPlatformXYZ	Y-Z angular acceleration error covariance.
rzRz	CovarOrientationAccelPlatformXYZ	Z-Z acceleration error covariance.

6.2.51 CovarianceOrientationNEDType

Namespace: UMAA::Common::Measurement::CovarianceOrientationNEDType

Description: Contains variances and covariances for random variables representing estimates of Observable values. Given an Observable with true value x the estimated value $\hat{x}$ is taken to be a random variable with sample space and distribution determined by the random processes which contribute to the calculation of the sampled Observable estimation. The ObservableError for the Observable is defined to be the Mean Squared Error, $\text{MSE} = E[(\hat{x} - x)^2]$. The MSE is equal to the sum of the variance of $\hat{x}$ and the square of the estimation bias: $\text{MSE} = \text{Var}[\hat{x}] + \text{Bias}[\hat{x}]^2 = E[(\hat{x} - E[\hat{x}])^2] - E[\hat{x} - x]^2$. The variance of $\hat{x}$ is equal to the MSE, and provides an estimate of the ObservableError, if and only if the underlying estimation process is unbiased.

Table 294: CovarianceOrientationNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rpRp	CovarOrientationNED	Pitch-Pitch orientation-orientation error covariance.
rpRy†	CovarOrientationNED	Pitch-Yaw orientation-orientation error covariance.
rrRp†	CovarOrientationNED	Roll-Pitch orientation-orientation error covariance.
rrRr	CovarOrientationNED	Roll-Roll orientation-orientation error covariance.
rrRy†	CovarOrientationNED	Roll-Yaw orientation-orientation error covariance.
ryRy	CovarOrientationNED	Yaw-Yaw orientation-orientation error covariance.

6.2.52 CovarianceOrientationVelocityNEDType

Namespace: UMAA::Common::Measurement::CovarianceOrientationVelocityNEDType

Description: Contains variances and covariances for random variables representing estimates of Observable values. Given an Observable with true value x the estimated value $\hat{x}$ is taken to be a random variable with sample space and distribution determined by the random processes which contribute to the calculation of the sampled Observable estimation. The ObservableError for the Observable is defined to be the Mean Squared Error, $\text{MSE} = E[(\hat{x} - x)^2]$. The MSE is equal to the sum of the variance of $\hat{x}$ and the square of the estimation bias: $\text{MSE} = \text{Var}[\hat{x}] + \text{Bias}[\hat{x}]^2 = E[(\hat{x} - E[\hat{x}])^2] - E[\hat{x} - x]^2$. The variance of $\hat{x}$ is equal to the MSE, and provides an estimate of the ObservableError, if and only if the underlying estimation process is unbiased.

Table 295: CovarianceOrientationVelocityNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rpRp	CovarOrientationVelNED	Pitch-Pitch orientation velocity-orientation velocity error covariance.
rpRy†	CovarOrientationVelNED	Pitch-Yaw orientation velocity-orientation velocity error covariance.
rrRp†	CovarOrientationVelNED	Roll-Pitch orientation velocity-orientation velocity error covariance.
rrRr	CovarOrientationVelNED	Roll-Roll orientation velocity-orientation velocity error covariance.
rrRy†	CovarOrientationVelNED	Roll-Yaw orientation velocity-orientation velocity error covariance.
ryRy	CovarOrientationVelNED	Yaw-Yaw orientation velocity-orientation velocity error covariance.

6.2.53 CovariancePositionNEDType

Namespace: UMAA::Common::Measurement::CovariancePositionNEDType

Description: Contains variances and covariances for random variables representing estimates of Observable values. Given an Observable with true value x the estimated value $\hat{x}$ is taken to be a random variable with sample space and distribution determined by the random processes which contribute to the calculation of the sampled Observable estimation. The ObservableError for the Observable is defined to be the Mean Squared Error, $\text{MSE} = E[(\hat{x} - x)^2]$. The MSE is equal to the sum of the variance of $\hat{x}$ and the square of the estimation bias: $\text{MSE} = \text{Var}[\hat{x}] + \text{Bias}[\hat{x}]^2 = E[(\hat{x} - E[\hat{x}])^2] - E[\hat{x} - x]^2$. The variance of $\hat{x}$ is equal to the MSE, and provides an estimate of the ObservableError, if and only if the underlying estimation process is unbiased.

Table 296: CovariancePositionNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pdPd†	CovarPosNED	Down-Down position-position error covariance.
pePd†	CovarPosNED	East-Down position-position error covariance.
pePe	CovarPosNED	East-East position-position error covariance.
pnPd†	CovarPosNED	North-Down position-position error covariance.
pnPe†	CovarPosNED	North-East position-position error covariance.

Attribute Name	Attribute Type	Attribute Description
pnPn	CovarPosNED	North-North position-position error covariance.

6.2.54 DateTimeRequirementType

Namespace: UMAA::Common::Time::DateTimeRequirementType

Description: Defines a date time requirement.

Table 297: DateTimeRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
time	DateTime	Specifies the required time.
timeTolerance†	DateTimeToleranceType	Specifies the time tolerance.

6.2.55 DateTimeToleranceType

Namespace: UMAA::Common::Time::DateTimeToleranceType

Description: Defines the date time tolerance.

Table 298: DateTimeToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	DateTime	Specifies the lower limit of allowable values.
upperlimit	DateTime	Specifies the upper limit of allowable values.

6.2.56 DepthRateRequirementType

Namespace: UMAA::Common::Measurement::DepthRateRequirementType

Description: Defines the change in depth as a function of time.

Table 299: DepthRateRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depthRate	SpeedBSL	Specifies the change in depth as a function of time.
depthRateTolerance†	DepthRateToleranceType	Specifies the depth rate tolerance.

6.2.57 DepthRateRequirementVariantType

Namespace: UMAA::Common::Measurement::DepthRateRequirementVariantType

Description: The change in depth as a function of time.

Table 300: DepthRateRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depthRate	DepthRateRequirementType	Specifies the change in depth as a function of time.

6.2.58 DepthRateToleranceType

Namespace: UMAA::Common::Measurement::DepthRateToleranceType

Description: Defines the depth rate tolerance.

Table 301: DepthRateToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	SpeedBSL	Specifies the lower limit of allowable values for the change in depth as a function of time.
upperlimit	SpeedBSL	Specifies the upper limit of allowable values for the change in depth as a function of time.

6.2.59 DepthRequirementType

Namespace: UMAA::Common::Measurement::DepthRequirementType

Description: Defines the depth below sea level.

Table 302: DepthRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depth	DistanceBSL	Specifies the depth below sea level.
depthTolerance†	DepthToleranceType	Specifies the tolerance for the depth below sea level.

6.2.60 DepthRequirementVariantType

Namespace: UMAA::Common::Measurement::DepthRequirementVariantType

Description: The depth below sea level.

Table 303: DepthRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depth	DepthRequirementType	The depth below sea level.

6.2.61 DepthSpeedPairType

Namespace: UMAA::Common::Environment::DepthSpeedPairType

Description: Specifies an element that specifies the speed of sound at a specific depth.

Table 304: DepthSpeedPairType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depth	DistanceBSL	Distance below water surface at which the measurement is made.
soundSpeed	Speed	Speed of sound at the measurement depth.

6.2.62 DepthToleranceType

Namespace: UMAA::Common::Measurement::DepthToleranceType

Description: Defines the depth below sea level tolerance.

Table 305: DepthToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerLimit	DistanceBSL	Specifies the lower limit of allowable values for the depth below sea level.
upperlimit	DistanceBSL	Specifies the upper limit of allowable values for the depth below sea level.

6.2.63 DepthVariantType

Namespace: UMAA::Common::Measurement::DepthVariantType

Description: The depth below sea level.

Table 306: DepthVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depth	DistanceBSL	The depth below sea level.

6.2.64 DirectionCurrentRequirement

Namespace: UMAA::Common::Orientation::DirectionCurrentRequirement

Description: A requirement that specifies the direction with respect to the current, where 0 is defined to be with the direction of the current (i.e. downstream).

Table 307: DirectionCurrentRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingCurrentDirection	Specifies the heading offset angle relative to the current, where 0 is defined to be with the direction of the current (i.e. downstream).
directionTolerance†	DirectionToleranceType	Specifies the heading reference angle tolerance relative to the current, where 0 is defined to be with the direction of the current (i.e. downstream).

6.2.65 DirectionCurrentRequirementVariantType

Namespace: UMAA::Common::Orientation::DirectionCurrentRequirementVariantType

Description: Specifies the direction with respect to the current, where 0 is defined to be with the direction of the current (i.e. downstream).

Table 308: DirectionCurrentRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	DirectionCurrentRequirement	Specifies the heading offset angle relative to the current, where 0 is defined to be with the direction of the current (i.e. downstream).

6.2.66 DirectionCurrentVariantType

Namespace: UMAA::Common::Orientation::DirectionCurrentVariantType

Description: Specifies the direction with respect to the current, where 0 is defined to be with the direction of the current (i.e. downstream).

Table 309: DirectionCurrentVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingCurrentDirection	Specifies the heading offset angle relative to the current, where 0 is defined to be with the direction of the current (i.e. downstream).

6.2.67 DirectionMagneticNorthRequirement

Namespace: UMAA::Common::Orientation::DirectionMagneticNorthRequirement

Description: A requirement that specifies the direction with respect to magnetic north.

Table 310: DirectionMagneticNorthRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingMagneticNorth	Specifies the heading reference angle relative to magnetic north.
directionTolerance†	DirectionToleranceType	Specifies the heading reference angle tolerance relative to magnetic north.

6.2.68 DirectionMagneticNorthRequirementVariantType

Namespace: UMAA::Common::Orientation::DirectionMagneticNorthRequirementVariantType

Description: Specifies the direction with respect to magnetic north.

Table 311: DirectionMagneticNorthRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	DirectionMagneticNorthRequirement	Specifies the heading reference angle relative to magnetic north.

6.2.69 DirectionMagneticNorthVariantType

Namespace: UMAA::Common::Orientation::DirectionMagneticNorthVariantType

Description: Specifies the direction with respect to magnetic north.

Table 312: DirectionMagneticNorthVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingMagneticNorth	Specifies the heading reference angle relative to magnetic north.

6.2.70 DirectionRequirementVariantType

Namespace: UMAA::Common::Orientation::DirectionRequirementVariantType

Description: Union Type. Direction of the vehicle motion or pattern being performed.

Table 313: DirectionRequirementVariantType Union(s)

Type Name	Type Description
DirectionCurrentRequirementVariantType	Specifies the direction with respect to the current, where 0 is defined to be with the direction of the current (i.e. downstream).
DirectionMagneticNorthRequirementVariantType	Specifies the direction with respect to magnetic north.
DirectionTrueNorthRequirementVariantType	Specifies the direction with respect to true north.
DirectionTurnRateRequirementVariantType	Specifies the change in direction as a function of time.
DirectionWindRequirementVariantType	Specifies the direction with respect to the direction of the wind, where 0 is defined to be the direction into the wind.

6.2.71 DirectionToleranceType

Namespace: UMAA::Common::Orientation::DirectionToleranceType

Description: An angle tolerance associated with a direction.

Table 314: DirectionToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	Angle	Describes the direction bound counterclockwise from the specified direction.
upperlimit	Angle	Describes the direction bound clockwise from the specified direction.

6.2.72 DirectionTrueNorthRequirement

Namespace: UMAA::Common::Orientation::DirectionTrueNorthRequirement

Description: A requirement that specifies the direction with respect to true north.

Table 315: DirectionTrueNorthRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingTrueNorthAngle	Specifies the heading reference angle relative to true north.
directionTolerance†	DirectionToleranceType	Specifies the heading reference angle tolerance relative to true north.

6.2.73 DirectionTrueNorthRequirementVariantType

Namespace: UMAA::Common::Orientation::DirectionTrueNorthRequirementVariantType

Description: Specifies the direction with respect to true north.

Table 316: DirectionTrueNorthRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	DirectionTrueNorthRequirement	Specifies the heading reference angle relative to true north.

6.2.74 DirectionTrueNorthVariantType

Namespace: UMAA::Common::Orientation::DirectionTrueNorthVariantType

Description: Specifies the direction with respect to true north.

Table 317: DirectionTrueNorthVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingTrueNorthAngle	Specifies the heading reference angle relative to true north.

6.2.75 DirectionTurnRateRequirementType

Namespace: UMAA::Common::Orientation::DirectionTurnRateRequirementType

Description: A requirement that specifies the change in direction of the vehicle's motion as a function of time.

Table 318: DirectionTurnRateRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
directionRate	TurnRate	Specifies a change in direction as a function of time.
directionRateTolerance†	DirectionTurnRateToleranceType	Specifies the direction turn rate tolerance.

6.2.76 DirectionTurnRateRequirementVariantType

Namespace: UMAA::Common::Orientation::DirectionTurnRateRequirementVariantType

Description: Specifies the change in direction as a function of time.

Table 319: DirectionTurnRateRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
directionRate	DirectionTurnRateRequirementType	Specifies the change in direction of the vehicle's motion as a function of time.

6.2.77 DirectionTurnRateToleranceType

Namespace: UMAA::Common::Orientation::DirectionTurnRateToleranceType

Description: Defines the direction turn rate tolerance.

Table 320: DirectionTurnRateToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	TurnRate	Specifies the lower limit of allowable values for the change in direction as a function of time.
upperlimit	TurnRate	Specifies the upper limit of allowable values for the change in direction as a function of time.

6.2.78 DirectionVariantType

Namespace: UMAA::Common::Orientation::DirectionVariantType

Description: Union Type. Direction of the vehicle motion or pattern being performed.

Table 321: DirectionVariantType Union(s)

Type Name	Type Description
DirectionCurrentVariantType	Specifies the direction with respect to the current, where 0 is defined to be with the direction of the current (i.e. downstream).
DirectionMagneticNorthVariantType	Specifies the direction with respect to magnetic north.
DirectionTrueNorthVariantType	Specifies the direction with respect to true north.
DirectionWindVariantType	Specifies the direction with respect to the direction of the wind, where 0 is defined to be the direction into the wind.

6.2.79 DirectionWindRequirement

Namespace: UMAA::Common::Orientation::DirectionWindRequirement

Description: A requirement that specifies the direction with respect to the direction of the wind, where 0 is defined to be the direction into the wind

Table 322: DirectionWindRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingWindDirection	Specifies the heading offset angle relative to the wind, where 0 is defined to be the direction into the wind.
directionTolerance†	DirectionToleranceType	Specifies the heading reference angle tolerance relative to the wind direction, where 0 is defined to be the direction into the wind.

6.2.80 DirectionWindRequirementVariantType

Namespace: UMAA::Common::Orientation::DirectionWindRequirementVariantType

Description: Specifies the direction with respect to the direction of the wind, where 0 is defined to be the direction into the wind.

Table 323: DirectionWindRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	DirectionWindRequirement	Specifies the heading offset angle relative to the wind, where 0 is defined to be the direction into the wind.

6.2.81 DirectionWindVariantType

Namespace: UMAA::Common::Orientation::DirectionWindVariantType

Description: Specifies the direction with respect to the direction of the wind, where 0 is defined to be the direction into the wind.

Table 324: DirectionWindVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
direction	HeadingWindDirection	Specifies the heading offset angle relative to the wind, where 0 is defined to be the direction into the wind.

6.2.82 DistanceRequirementType

Namespace: UMAA::Common::Distance::DistanceRequirementType

Description: Defines a distance requirement.

Table 325: DistanceRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
distance	Distance	Specifies the required distance.
distanceTolerance†	DistanceToleranceType	Specifies the distance tolerance.

6.2.83 DistanceToleranceType

Namespace: UMAA::Common::Distance::DistanceToleranceType

Description: Defines the distance tolerance.

Table 326: DistanceToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
limit	Distance	Specifies the limit of the tolerance.

6.2.84 DriftObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::DriftObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the drift objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::DriftObjectiveDetailedStatusType

Table 327: DriftObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
distanceFromReference	Distance	Defines the distance from the reference position.

Attribute Name	Attribute Type	Attribute Description
isDriftAchieved	boolean	Indicates that the drift position requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing the drift maneuver (True = system is executing the drift maneuver, False = system is in transit).
isInPoweredDriving	boolean	Is system currently driving (True = system is driving to reference position, False = system is drifting unpowered).
referencePosition	GeoPosition2D	Defines the reference position for the drifting. (Important if reference position was not defined in original objective).
timePatternAchieved	DateTime	Defines the absolute time at which the drift is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the drift is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.85 DriftObjectiveType

Namespace: UMAA::MM::BaseType::DriftObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for drifting. The drift objective is achieved by having the vehicle, under a reduced power mode, maintain its position within the circle at a defined elevation (or current elevation if not defined) as specified by the reference position and driftRadius. If a position is not specified, then the current vehicle position is used as the reference position for drifting. DriftRadius and elevation include optional tolerances. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::DriftObjectiveType

Table 328: DriftObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
driftRadius	DistanceRequirementType	Defines the drift radius that specifies the maximum distance from the reference position the vehicle is allowed to drift.
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the drifting portion of the objective. If not specified, duration is not used to determine when drifting is complete.

Attribute Name	Attribute Type	Attribute Description
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while drifting. If not specified, the maneuver is performed at the current elevation.
position†	GeoPosition2D	Defines the reference position for drifting. If not specified, the reference position is the current vehicle position.
speed	SpeedVariantType	Defines the desired vehicle speed when maneuvering within the area defined by driftRadius.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the drift location before transitioning to drifting. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the drift location before transitioning to drifting.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.86 ElevationRequirementVariantType

Namespace: UMAA::Common::Measurement::ElevationRequirementVariantType

Description: Union Type. The desired elevation used for the vehicle.

Table 329: ElevationRequirementVariantType Union(s)

Type Name	Type Description
AltitudeAGLRequirementVariantType	The height above ground level.
AltitudeASFRequirementVariantType	The height above sea floor.
AltitudeGeodeticRequirementVariantType	The geodetic height above the ellipsoid.
AltitudeMSLRequirementVariantType	The orthometric height above the Geoid (Mean Sea Level).
AltitudeRateASFRequirementVariantType	The change in altitude as a function of time.
DepthRateRequirementVariantType	The change in depth as a function of time.
DepthRequirementVariantType	The depth below sea level.

6.2.87 ElevationVariantType

Namespace: UMAA::Common::Measurement::ElevationVariantType

Description: Union Type. The desired elevation used for the vehicle.

Table 330: ElevationVariantType Union(s)

Type Name	Type Description
AltitudeAGLVariantType	The height above ground level.
AltitudeASFVariantType	The height above sea floor.
AltitudeGeodeticVariantType	The geodetic height above the ellipsoid.
AltitudeMSLVariantType	The orthometric height above the Geoid (Mean Sea Level).
DepthVariantType	The depth below sea level.

6.2.88 EngineRPMSpeedRequirement

Namespace: UMAA::Common::Speed::EngineRPMSpeedRequirement

Description: Defines the engine rpm.

Table 331: EngineRPMSpeedRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	FrequencyRPM	Specifies speed via engine rpm.
speedTolerance†	EngineRPMSpeedTolerance	Specifies the tolerance for an engine rpm.

6.2.89 EngineRPMSpeedRequirementVariantType

Namespace: UMAA::Common::Speed::EngineRPMSpeedRequirementVariantType

Description: Defines the engine RPM.

Table 332: EngineRPMSpeedRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rpm	EngineRPMSpeedRequirement	Specifies engine rpm.

6.2.90 EngineRPMSpeedTolerance

Namespace: UMAA::Common::Speed::EngineRPMSpeedTolerance

Description: Defines the speed through engine rpm.

Table 333: EngineRPMSpeedTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	FrequencyRPM	Specifies the lower limit of allowable values for the engine rpm.
upperlimit	FrequencyRPM	Specifies the upper limit of allowable values for the engine rpm.

6.2.91 EngineRPMSpeedVariantType

Namespace: UMAA::Common::Speed::EngineRPMSpeedVariantType

Description: Defines the engine RPM.

Table 334: EngineRPMSpeedVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rpm	FrequencyRPM	Specifies speed via engine rpm.

6.2.92 ExpBinaryValueType

Namespace: UMAA::MM::BaseType::ExpBinaryValueType

Description: This structure is used to define the binary value in a key/value pair.

Table 335: ExpBinaryValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
binaryValue	BinaryValue	Defines binary data for the value.

6.2.93 ExpBooleanValueType

Namespace: UMAA::MM::BaseType::ExpBooleanValueType

Description: This structure is used to define the boolean value in a key/value pair.

Table 336: ExpBooleanValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
booleanValue	boolean	Defines boolean data for the value.

6.2.94 ExpByteValueType

Namespace: UMAA::MM::BaseType::ExpByteValueType

Description: This structure is used to define the byte value in a key/value pair.

Table 337: ExpByteValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
byteValue	ByteValue	Defines byte data for the value.

6.2.95 ExpCharValueType

Namespace: UMAA::MM::BaseType::ExpCharValueType

Description: This structure is used to define the char value in a key/value pair.

Table 338: ExpCharValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
charValue	CharValue	Defines char data for the value.

6.2.96 ExpDateTimeValueType

Namespace: UMAA::MM::BaseType::ExpDateTimeValueType

Description: This structure is used to define the DateTime value in a key/value pair.

Table 339: ExpDateTimeValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
dateTimeValue	DateTime	Defines DateTime data for the value.

6.2.97 ExpDoubleValueType

Namespace: UMAA::MM::BaseType::ExpDoubleValueType

Description: This structure is used to define the double value in a key/value pair.

Table 340: ExpDoubleValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
doubleValue	DoubleValue	Defines double data for the value.

6.2.98 ExpIntegerValueType

Namespace: UMAA::MM::BaseType::ExpIntegerValueType

Description: This structure is used to define the integer value in a key/value pair.

Table 341: ExpIntegerValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
integerValue	IntegerValue	Defines integer data for the value.

6.2.99 ExpLongLongValueType

Namespace: UMAA::MM::BaseType::ExpLongLongValueType

Description: This structure is used to define the long long value in a key/value pair.

Table 342: ExpLongLongValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
longlongValue	LargeCount	Defines long long data for the value.

6.2.100 ExpObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::ExpObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the experimental objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::ExpObjectiveDetailedStatusType

Table 343: ExpObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
expObjectiveStatus	sequence<KeyValueType> max size = 170	Defines a set of key/value pairs for the experimental objective status.
timeExpCompleted	DateTime	Defines the absolute time at which the experimental objective is estimated to be completed.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.101 ExpObjectiveType

Namespace: UMAA::MM::BaseType::ExpObjectiveType

Description: This structure is used to define the goal of an experimental objective by specifying key/value pairs. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::ExpObjectiveType

Table 344: ExpObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
expObjectiveDescription	StringShortDescription	Defines a short name for the experimental objective.
keyValues	sequence<KeyValueType> max size = 170	Defines a set of key/value pairs for the experimental objective.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.102 ExpStringValueType

Namespace: UMAA::MM::BaseType::ExpStringValueType

Description: This structure is used to define the string value in a key/value pair.

Table 345: ExpStringValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
stringValue	StringValue	Defines string data for the value.

6.2.103 ExpValueType

Namespace: UMAA::MM::BaseType::ExpValueType

Description: Union Type. This structure is used to define the value in a key/value pair.

Table 346: ExpValueType Union(s)

Type Name	Type Description
ExpBinaryValueType	This structure is used to define the binary value in a key/value pair.
ExpBooleanValueType	This structure is used to define the boolean value in a key/value pair.
ExpByteValueType	This structure is used to define the byte value in a key/value pair.
ExpCharValueType	This structure is used to define the char value in a key/value pair.
ExpDateTimeValueType	This structure is used to define the DateTime value in a key/value pair.
ExpDoubleValueType	This structure is used to define the double value in a key/value pair.
ExpIntegerValueType	This structure is used to define the integer value in a key/value pair.
ExpLongLongValueType	This structure is used to define the long long value in a key/value pair.
ExpStringValueType	This structure is used to define the string value in a key/value pair.

6.2.104 FLSAdditionalConfigType

Namespace: UMAA::SEM::BaseType::FLSAdditionalConfigType

Description: Union Type. The desired configuration mode for an FLS.

Table 347: FLSAdditionalConfigType Union(s)

Type Name	Type Description
FLSConfigSearchBottomType	Specifies search bottom configuration for the subsystem. Corresponds to FLSConfigModeEnumType SEARCH_BOTTOM.
FLSConfigSearchVolumeType	Specifies search volume configuration for the subsystem. Corresponds to FLSConfigModeEnumType SEARCH_VOLUME.
FLSConfigTestType	Specifies test configuration for the subsystem. Corresponds to FLSConfigModeEnumType TEST.

6.2.105 FLSConfigSearchBottomType

Namespace: UMAA::SEM::BaseType::FLSConfigSearchBottomType

Description: Specifies search bottom configuration for the subsystem. Corresponds to FLSConfigModeEnumType SEARCH_-BOTTOM.

Table 348: FLSConfigSearchBottomType Structure Definition

Attribute Name	Attribute Type	Attribute Description
goalVehicleAltitude	DistanceASF	Goal altitude for the vehicle during the bottom search.
goalVehicleDepth	DistanceBSL	Goal depth for the vehicle during the bottom search.
maxRange	Distance	Maximum range during the bottom search.
minRange†	Distance	Minimum range during the bottom search.

6.2.106 FLSConfigSearchVolumeType

Namespace: UMAA::SEM::BaseType::FLSConfigSearchVolumeType

Description: Specifies search volume configuration for the subsystem. Corresponds to FLSConfigModeEnumType SEARCH_-VOLUME.

Table 349: FLSConfigSearchVolumeType Structure Definition

Attribute Name	Attribute Type	Attribute Description
goalVehicleAltitude	DistanceASF	Goal altitude for the vehicle during the volume search.
goalVehicleDepth	DistanceBSL	Goal depth for the vehicle during the volume search.
maxDepth	DistanceBSL	Maximum depth during the volume search.
maxRange	Distance	Maximum range during the volume search.
minDepth	DistanceBSL	Minimum depth during the volume search.
minRange†	Distance	Minimum range during the volume search.

6.2.107 FLSConfigTestType

Namespace: UMAA::SEM::BaseType::FLSConfigTestType

Description: Specifies test configuration for the subsystem. Corresponds to FLSConfigModeEnumType TEST.

Table 350: FLSConfigTestType Structure Definition

Attribute Name	Attribute Type	Attribute Description
activeBandwidth	FrequencyHertz	Difference between the upper and lower frequencies for the receiver. Centered around the active frequency.
activeCenterFrequency	FrequencyHertz	Frequency used to center the active processing data stream.
attenuation	TransmitAttenuation	Attenuation required or power used by the transmitter.
beamwidth	FLSBeamwidthEnumType	The difference between the upper and lower edges of the sonar beam.
passiveBandwidth	FrequencyHertz	Difference between the upper and lower frequencies for the receiver. Centered around the passive frequency.
passiveCenterFrequency	FrequencyHertz	Frequency used to center the passive processing data stream.
range	Distance	Distance over which the sonar would like to make perceptions. Defines the maximum ping rate of the sonar.
steering	AngleHalf	Vertical steering Angle. Identical transmit and receive declination elevation.
upSweep	boolean	Indicates that the FLS will sweep from low to high frequencies.
waveformLength	FLSWaveformLengthEnumType	The length of the transmitted waveform in time. The difference between beginning and end of transmission.

6.2.108 Figure8ObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::Figure8ObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the figure 8 objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::Figure8ObjectiveDetailedStatusType

Table 351: Figure8ObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isCrossTrackLimitAchieved	boolean	Indicates that the cross track limit requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing the figure 8 maneuver (True = system is executing the figure 8 maneuver, False = system is in transit).
isSpeedAchieved	boolean	Indicates that the speed requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
referencePosition	GeoPosition2D	Defines the reference position for the figure 8 pattern. (Important if reference position was not defined in original objective).

Attribute Name	Attribute Type	Attribute Description
timePatternAchieved	DateTime	Defines the absolute time at which the figure 8 pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the figure 8 pattern is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.109 Figure8ObjectiveType

Namespace: UMAA::MM::BaseType::Figure8ObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for following the figure 8 pattern. The figure 8 objective is achieved by having the vehicle execute the figure 8 maneuver at a defined elevation (or current elevation if not defined) as specified by the reference position, length, radius, and orientation, with the defined speed, trackTolerance, and turnDirection. If a reference position is not specified, then the current vehicle position is used as the reference position for the figure 8 pattern. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::Figure8ObjectiveType

Table 352: Figure8ObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the remaining pattern portion of the objective. If not specified, duration is not used to determine when the figure 8 maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the figure 8 maneuver. If not specified, the maneuver is performed at the current elevation.
length	Distance	Defines the length between the semicircles at either end for the figure 8 pattern.
loops†	SizeReal	Defines the number of loops around the figure 8 pattern to execute. If not specified, the loops attribute is not used to determine when the figure 8 maneuver is complete.
orientation	DirectionVariantType	Defines the orientation of the figure 8 pattern, measured perpendicular to the length axis.

Attribute Name	Attribute Type	Attribute Description
position†	GeoPosition2D	Defines the reference position for the figure 8 pattern. If not specified, the reference position is the current vehicle position.
radius	Distance	Defines the radius of the semicircles for the figure 8 pattern.
speed	SpeedRequirementVariantType	Defines the vehicle speed to maintain while executing the figure 8 maneuver.
trackTolerance	DistanceRequirementType	Defines the maximum allowable cross track error while executing the figure 8 maneuver.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the figure 8 pattern location before transitioning to the figure 8 maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the figure 8 pattern location before transitioning to the figure 8 maneuver.
turnDirection	WaterTurnDirectionEnumType	Defines the turn direction while executing the figure 8 maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.110 FinCommandType

Namespace: UMAA::EO::FinsControl::FinCommandType

Description: Describes the desired fin deflection of an individual fin.

Table 353: FinCommandType Structure Definition

Attribute Name	Attribute Type	Attribute Description
deflection	Angle	The desired fin deflection with respect to the vehicle coordinate system.
deflectionRate†	AngleRate	The desired fin deflection rate. If not defined the nominal rate of the hardware will be used.

6.2.111 FinSpecsType

Namespace: UMAA::EO::FinsSpecs::FinSpecsType

Description: This structure is used to report the specifications of an individual fin.

Table 354: FinSpecsType Structure Definition

Attribute Name	Attribute Type	Attribute Description
maxDeflectionRate	AngleRate	The maximum rate of fin deflection.
maxNegativeDeflection	Angle	The maximum amount of fin deflection in the negative direction with respect to the vehicle coordinate system.
maxPositiveDeflection	Angle	The maximum amount of fin deflection in the positive direction with respect to the vehicle coordinate system.
minDeflectionRate	AngleRate	The minimum rate of fin deflection.
name	StringShortDescription	The name of the fin.
nominalDeflectionRate†	AngleRate	Nominal deflection rate of the fin.
orientation	Orientation3DPlatformType	The orientation of the fin away from the attachment point defined by position.
position	Position3DBodyXYZ	The position of the base of the fin with respect to the vehicle coordinate system.

6.2.112 FinStatusType

Namespace: UMAA::EO::FinsStatus::FinStatusType

Description: This structure is used to describe the current deflection of an individual fin.

Table 355: FinStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
deflection	Angle	The status of the fin deflection with respect to the vehicle coordinate system.
deflectionRate†	AngleRate	The status of deflection rate of a fin on a vehicle.

6.2.113 FreeFloatObjectiveType

Namespace: UMAA::MM::BaseType::FreeFloatObjectiveType

Description: The goal of the free float objective is to terminate all vehicle propulsion so that the vehicle is in a free float condition. The free float objective is achieved when all vehicle propulsion has been terminated. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::FreeFloatObjectiveType

Table 356: FreeFloatObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
duration†	DurationSeconds	Defines the desired duration to free float; if not specified, runs indefinitely until it is interrupted (e.g., another objective takes precedence, it is canceled, etc.).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.114 GammaAnglePropulsorRequirementType

Namespace: UMAA::Common::Angle::GammaAnglePropulsorRequirementType

Description: Describes the required value of the propulsor's gamma angle.

Table 357: GammaAnglePropulsorRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
gammaAnglePropulsor	GammaAnglePropulsor	Describes the required gamma value.
gammaAnglePropulsorToleranceType†	GammaAnglePropulsorToleranceType	Describes the required gamma value's tolerance.

6.2.115 GammaAnglePropulsorToleranceType

Namespace: UMAA::Common::Angle::GammaAnglePropulsorToleranceType

Description: Describes a tolerance range for the required gamma value.

Table 358: GammaAnglePropulsorToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	GammaAnglePropulsor	Specifies the lower limit of allowable values.
upperlimit	GammaAnglePropulsor	Specifies the upper limit of allowable values.

6.2.116 GammaZPlatformType

Namespace: UMAA::Common::Orientation::GammaZPlatformType

Description: A requirement that specifies a gamma angle relative to the Platform coordinate system.

Table 359: GammaZPlatformType Structure Definition

Attribute Name	Attribute Type	Attribute Description
gamma	YawAngle	Defines a gamma angle relative to the Platform coordinate system.

6.2.117 GeoPosition2D

Namespace: UMAA::Common::Measurement::GeoPosition2D

Description: Specifies a location on the surface of the Earth.

Table 360: GeoPosition2D Structure Definition

Attribute Name	Attribute Type	Attribute Description
geodeticLatitude	GeodeticLatitude	Specifies the north-south coordinate of the position.
geodeticLongitude	GeodeticLongitude	Specifies the east-west coordinate of the position.

6.2.118 GeoPosition2DRequirement

Namespace: UMAA::Common::Position::GeoPosition2DRequirement

Description: Defines a position requirement.

Table 361: GeoPosition2DRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
tolerance†	GeoPosition2DTolerance	Specifies the required position tolerance.
value	GeoPosition2D	Specifies the required position.

6.2.119 GeoPosition2DTolerance

Namespace: UMAA::Common::Position::GeoPosition2DTolerance

Description: Defines a position tolerance.

Table 362: GeoPosition2DTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
limit	Distance	Specifies the limit of the tolerance.

6.2.120 GeoPosition3DWGS84

Namespace: UMAA::Common::Measurement::GeoPosition3DWGS84

Description: Specifies a location relative to the ellipsoid.

Table 363: GeoPosition3DWGS84 Structure Definition

Attribute Name	Attribute Type	Attribute Description
geodeticAltitude	GeodeticAltitude	Specifies the height above the ellipsoid.
geodeticPosition	GeoPosition2D	Specifies a location on the surface of the Earth.

6.2.121 GroundSpeedRequirement

Namespace: UMAA::Common::Speed::GroundSpeedRequirement

Description: Defines the speed over ground.

Table 364: GroundSpeedRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	GroundSpeed	Specifies speed over ground.
speedTolerance†	GroundSpeedTolerance	Specifies the tolerance for a speed over ground.

6.2.122 GroundSpeedRequirementVariantType

Namespace: UMAA::Common::Speed::GroundSpeedRequirementVariantType

Description: Defines the speed over ground.

Table 365: GroundSpeedRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	GroundSpeedRequirement	Specifies speed over ground.

6.2.123 GroundSpeedTolerance

Namespace: UMAA::Common::Speed::GroundSpeedTolerance

Description: Defines the speed over ground tolerance.

Table 366: GroundSpeedTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	GroundSpeed	Specifies the lower limit of allowable values for the ground speed.
upperlimit	GroundSpeed	Specifies the upper limit of allowable values for the ground speed.

6.2.124 GroundSpeedVariantType

Namespace: UMAA::Common::Speed::GroundSpeedVariantType

Description: Defines the speed over ground.

Table 367: GroundSpeedVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	GroundSpeed	Specifies speed over ground.

6.2.125 HoverObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::HoverObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the hover objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::HoverObjectiveDetailedStatusType

Table 368: HoverObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
hoverHeading	DirectionVariantType	Defines the heading for hover.
hoverPosition	GeoPosition2D	Defines the position for hover. (Important if reference position was not defined in original objective).
isHoverHeadingAchieved	boolean	Indicates that the hover heading is within the requested tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute
isHoverPositionAchieved	boolean	Indicates that the hover position requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing the hover maneuver (True = system is executing the hover maneuver, False = system is in transit).
timePatternAchieved	DateTime	Defines the absolute time at which the hover is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the hover is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.126 HoverObjectiveType

Namespace: UMAA::MM::BaseType::HoverObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for hovering. The hover objective is achieved by having the vehicle actively maintain its position at a defined elevation (or current elevation if not defined) within the circle as defined by the reference position and hoverRadius, and optionally maintain a specified heading. If a position is not specified, then the current vehicle position is used as the reference position for hovering. If a heading is not specified, then the system is allowed to determine the best heading for hovering. Elevation, heading and hoverRadius include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::HoverObjectiveType

Table 369: HoverObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
controlPriority	HoverKindEnumType	Defines the priority to hover at the specified reference position.
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the hovering portion of the objective. If not specified, duration is not used to determine when hovering is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while hovering. If not specified, the maneuver is performed at the current elevation.
heading†	DirectionRequirementVariantType	Defines the heading for the vehicle to maintain while hovering. If not specified, the system will determine the best heading (e.g. current heading, into the wind/current, etc.) for hovering.
hoverRadius	DistanceRequirementType	Defines the maximum distance the vehicle position is allowed to be from the hover position and still considered to be achieved.
position†	GeoPosition2D	Defines the reference position for hovering. If not specified, the reference position is the current vehicle position.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the hover location before transitioning to hovering. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the hover location before transitioning to hovering.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.127 IdentifierType

Namespace: UMAA::Common::IdentifierType

Description: This structure defines a two-level hierarchical identifier, where the parent is defined to be a group or collection of entities.

Table 370: IdentifierType Structure Definition

Attribute Name	Attribute Type	Attribute Description
id	NumericGUID	Provides the identifier of an entity.
parentID	NumericGUID	Provides the identifier of the parent, which is a group or collection of one or more entities. If the entity has no parent (it is the root of the tree), this value will be the Nil UUID.

6.2.128 KeyValueType

Namespace: UMAA::MM::BaseType::KeyValueType

Description: This structure is used to define a key/value pair.

Table 371: KeyValueType Structure Definition

Attribute Name	Attribute Type	Attribute Description
key	StringName	Defines an identifier for the data contained in key/value pair.
value	ExpValueType	Defines the data contained in key/value pair.

6.2.129 LevelType

Namespace: UMAA::Common::Measurement::LevelType

Description: Defines the desired percentage filled or empty level of the ballast tank.

Table 372: LevelType Structure Definition

Attribute Name	Attribute Type	Attribute Description
level	VolumePercent	Specifies the desired percentage filled or empty level of the ballast tank.

6.2.130 MassBallastFlowRateType

Namespace: UMAA::EO::BallastTank::MassBallastFlowRateType

Description: The desired flow rate to fill or empty the ballast pump measured by mass.

Table 373: MassBallastFlowRateType Structure Definition

Attribute Name	Attribute Type	Attribute Description
massBallastFlowRate	MassFlowRate	Specifies the desired flow rate to fill or empty the ballast pump measured by mass.

6.2.131 ObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::ObjectiveDetailedStatusType

Description: This structure is used to define the objective detailed status.

Table 374: ObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
childObjectiveIDs	sequence<NumericGUID> max size = 256	The current child objective IDs associated with this objective.
errors	StringShortDescription	Human details about why the operation is not currently meeting objectives.
feedback	StringShortDescription	The explanation for what is being performed and the 'why' of any changes.
isCurrentlyMeetingObjective	boolean	An indication whether the executor is currently meeting the objective.
objectiveID	NumericGUID	The current objective identification within the current task plan.
objectiveStatus	ObjectiveExecutorStateEnumType	The objective status enumeration value.
objectiveStatusReason	ObjectiveExecutorStateReasonEnumType	This field provides the reason for the objective status.
startTime	DateTime	Actual past time of objective start.
warnings	StringShortDescription	Any warnings the operation should be aware of when performing this objective.
specializationID	NumericGUID	ID to capture specializations of ObjectiveDetailedStatusType.
specializationTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization.
specializationTopic	StringShortDescription	Topic to capture specializations of ObjectiveDetailedStatusType.

Table 375: ObjectiveDetailedStatusType Specialization(s)

Type Name	Type Description
AreaRandomWalkObjectiveDetailedStatusType	This structure provides a detailed execution status for the area random walk objective.

Type Name	Type Description
CircleObjectiveDetailedStatusType	This structure provides a detailed execution status for the circle objective.
DriftObjectiveDetailedStatusType	This structure provides a detailed execution status for the drift objective.
ExpObjectiveDetailedStatusType	This structure provides a detailed execution status for the experimental objective.
Figure8ObjectiveDetailedStatusType	This structure provides a detailed execution status for the figure 8 objective.
HoverObjectiveDetailedStatusType	This structure provides a detailed execution status for the hover objective.
RacetrackObjectiveDetailedStatusType	This structure provides a detailed execution status for the racetrack objective.
RegularPolygonObjectiveDetailedStatusType	This structure provides a detailed execution status for the regular polygon objective.
RouteObjectiveDetailedStatusType	This structure provides a detailed execution status for the route objective.
ScreenRandomWalkObjectiveDetailedStatusType	This structure provides a detailed execution status for the screen random walk objective.
StationkeepObjectiveDetailedStatusType	This structure provides a detailed execution status for the stationkeep objective.
VectorObjectiveDetailedStatusType	This structure provides a detailed execution status for the vector objective.

6.2.132 ObjectiveType

Namespace: UMAA::MM::BaseType::ObjectiveType

Description: This is a base structure that all specialization objectives are inherited from. Each specialized objective structure shall be used to define or report its own specialized data.

Table 376: ObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
approvalRequired	boolean	An indication whether approval is required for the specified objective within a mission.
duringConditionID†	NumericGUID	A reference to a conditional that must be kept true during objective execution by completing/executing one or more actions. If not specified, then no duringCondition exists for the objective.
name	StringShortDescription	A short name for the objective.
objectiveDescription	StringShortDescription	A description of the objective.
objectiveID	NumericGUID	Unique identifier for the objective within a mission.
objectivePriority	Priority	Specifies the desired importance for completing the objective in order to handle the case where a plan cannot be generated to complete all objectives. For this case, objective priority is used to determine what objectives(s) to drop from the plan. Objectives with the lowest priority must be dropped before objectives with a higher priority. If multiple objectives have the lowest priority, then the order in which they are dropped is not defined by their priority and must be determined by some other mechanism.

Attribute Name	Attribute Type	Attribute Description
preconditionID†	NumericGUID	A reference to a conditional that must be made true prior to executing the objective by completing one or more actions. If not specified, then no precondition exists for the objective.
preferredResourceID	sequence<IdentifierType> max size = 16	If defined, specifies a list of preferred resource(s) to execute the objective in order of preference. A preferred resource must be used if a valid plan can be generated. Otherwise, if a valid plan cannot be generated then another resource may be used (i.e., the inability to use a preferred resource when alternatives are available is not a cause for failure).
stateTrigger	sequence<StateTriggerType> max size = 16	Specifies the conditional statement that when true attempts to change the current state. Each trigger is evaluated individually, meaning if multiple triggers are defined with the same state, then their conditional statements are treated as if they are logically OR'd.
specializationID	NumericGUID	ID to capture specializations of ObjectiveType.
specializationTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization.
specializationTopic	StringShortDescription	Topic to capture specializations of ObjectiveType.

Table 377: ObjectiveType Specialization(s)

Type Name	Type Description
FreeFloatObjectiveType	The goal of the free float objective is to terminate all vehicle propulsion so that the vehicle is in a free float condition. The free float objective is achieved when all vehicle propulsion has been terminated.
AreaRandomWalkObjectiveType	The goal of the area random walk objective is to execute a random walk maneuver within a given area. This structure is used to specify the area where the random walk must be conducted. The area random walk objective is achieved by having the vehicle execute random vectors at a specified elevation (or current elevation if not specified) while maintaining the vehicle location within a defined area. The area is defined by specifying the vertices of a polygon, and the random vectors within this area can be configured by specifying min/max speeds and min/max time on course. Area and elevation include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action.

Type Name	Type Description
CircleObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for following the circle pattern. The circle objective is achieved by having the vehicle execute the circle pattern maneuver at a specified elevation (or current elevation if not specified) as specified by the center position and radius, with the defined speed, trackTolerance, and turnDirection. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely.
DriftObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for drifting. The drift objective is achieved by having the vehicle, under a reduced power mode, maintain its position within the circle at a defined elevation (or current elevation if not defined) as specified by the reference position and driftRadius. If a position is not specified, then the current vehicle position is used as the reference position for drifting. DriftRadius and elevation include optional tolerances. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action.
ExpObjectiveType	This structure is used to define the goal of an experimental objective by specifying key/value pairs.
Figure8ObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for following the figure 8 pattern. The figure 8 objective is achieved by having the vehicle execute the figure 8 maneuver at a defined elevation (or current elevation if not defined) as specified by the reference position, length, radius, and orientation, with the defined speed, trackTolerance, and turnDirection. If a reference position is not specified, then the current vehicle position is used as the reference position for the figure 8 pattern. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely, or until interrupted by some other action.

Type Name	Type Description
HoverObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for hovering. The hover objective is achieved by having the vehicle actively maintain its position at a defined elevation (or current elevation if not defined) within the circle as defined by the reference position and hoverRadius, and optionally maintain a specified heading. If a position is not specified, then the current vehicle position is used as the reference position for hovering. If a heading is not specified, then the system is allowed to determine the best heading for hovering. Elevation, heading and hoverRadius include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action.
RacetrackObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for following the racetrack pattern. The racetrack objective is achieved by having the vehicle execute the racetrack maneuver at a defined elevation (or current elevation if not defined) as specified by the reference position, length, radius and orientation, with the defined speed, trackTolerance and turnDirection. If position is not specified, then the current vehicle position is used as the reference position for the racetrack pattern. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely, or until interrupted by some other action.
RegularPolygonObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for following the polygon pattern. The regular polygon objective is achieved by having the vehicle execute the regular polygon maneuver at a defined elevation (or current elevation if not defined) as specified by the reference position, diameter, and orientation, with the defined speed, trackTolerance, and turnDirection. If position is not specified, then the current vehicle position is used as the reference position for the regular polygon pattern. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely, or until interrupted by some other action.

Type Name	Type Description
RouteObjectiveType	This structure is used to report an element that describes a clearly defined goal specifying the action(s) required for following a route. The route objective is achieved by having the vehicle achieve each waypoint in order and is complete when the final waypoint is achieved. Each waypoint is specified by a position along with a captureRadius, and each waypoint optionally includes a specified attitude and elevation. If the attitude or elevation is not specified, then any attitude or elevation at the waypoint is acceptable, respectively. The desired effect when a waypoint is achieved is for the vehicle's position to be within a distance less than or equal to the captureRadius. The captureRadius includes an optional tolerance. If specified and the system is unable to achieve the captureRadius, then the waypoint can be considered completed if the vehicle's position is within a distance less than or equal to the specified tolerance. If the system is unable to complete the waypoint within the specified tolerance, then the objective is considered to have failed. If the captureRadius tolerance is not specified and the system is not able to achieve the captureRadius, then it is a "best effort" to achieve the waypoint, and is therefore not considered a cause for the objective to fail. If attitude or elevation is specified, then the desired effect when a waypoint is achieved includes the vehicle's attitude aligning with this specified attitude or obtaining the specified elevation, respectively, at the waypoint position along with the positional effect described above. Both attitude and elevation include an optional tolerance. If the system is unable to achieve the specified attitude or elevation within their tolerance, then the objective is considered to have failed. If a tolerance is not defined for attitude or elevation, then it is a "best effort" to achieve the specified attitude or elevation, respectively, and is therefore not considered a cause for the objective to fail. Additionally, there is an optional trackTolerance. If trackTolerance is specified, then the defined path between waypoints is specified to be the track line created by the waypoints with a cross track error less than or equal to the trackTolerance. The trackTolerance includes an optional tolerance. If specified, the vehicle is allowed to operate within this tolerance without having the objective fail. Otherwise, if the tolerance is violated, then the objective is considered to have failed. If the trackTolerance tolerance is not specified, then it is a "best effort" to maintain the specified trackTolerance, and is therefore not considered a cause for the objective to fail. If the trackTolerance is not specified, then path between waypoints is not specified and therefore left for the system to determine the best path.
ScreenRandomWalkObjectiveType	The goal of the screen random walk objective is to execute a random walk maneuver within a specified area that is relative to a guide (e.g., a high value unit (HVU)). This structure is used to specify the guide and the random walk area relative to the guide where the screening must be conducted. The screen random walk objective is achieved by having the vehicle execute random vectors at a defined elevation (or current elevation if not defined) while maintaining the vehicle position within a defined area relative to a guide. The area is defined by specifying a start and end bearing along with a min and max range. The random vectors can be configured by specifying min/max speeds and min/max time on course. Area and elevation include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action.

Type Name	Type Description
StationkeepObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for station keeping. The station keep objective is achieved by maintaining the vehicle location at a defined elevation (or current elevation if not defined) within a defined area relative to a guide. The area is defined by specifying a start and end bearing along with a min and max range. Area and elevation include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action.
VectorObjectiveType	This structure is used to describe a clearly defined goal specifying the action(s) required for the vector objective to achieve the specified speed, direction of travel, and altitude or depth of the vehicle. The vector objective is achieved by having the vehicle execute the vector maneuver at a defined elevation (or current elevation if not defined) as specified by direction, directionMode, and speed. The vector can be configured by setting an optional depthChangePitch. Direction, elevation, and speed include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action.

6.2.133 Orientation3DNEDRequirement

Namespace: UMAA::Common::Orientation::Orientation3DNEDRequirement

Description: A requirement that describes a desired 3D orientation in a NED coordinate system.

Table 378: Orientation3DNEDRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchY†	PitchYNEDRequirement	Defines a pitch relative to the NED coordinate system.
rollX†	RollXNEDRequirement	Defines a roll relative to the NED coordinate system.
yawZ	YawZNEDRequirement	Defines a yaw relative to the NED coordinate system.

6.2.134 Orientation3DNEDType

Namespace: UMAA::Common::Orientation::Orientation3DNEDType

Description: A requirement that specifies an orientation relative to a North-East-Down coordinate system centered on the platform.

Table 379: Orientation3DNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitch	PitchYNEDType	Defines a pitch relative to a North-East-Down coordinate system centered on the platform.
roll	RollXNEDType	Defines a roll relative to a North-East-Down coordinate system centered on the platform.
yaw	YawZNEDType	Defines a yaw relative to a North-East-Down coordinate system centered on the platform.

6.2.135 Orientation3DPlatformType

Namespace: UMAA::Common::Orientation::Orientation3DPlatformType

Description: A requirement that specifies an orientation relative to the Platform coordinate system.

Table 380: Orientation3DPlatformType Structure Definition

Attribute Name	Attribute Type	Attribute Description
alpha	AlphaXPlatformType	Defines an alpha angle relative to the Platform coordinate system.
beta	BetaYPlatformType	Defines a beta angle relative to the Platform coordinate system.
gamma	GammaZPlatformType	Defines a gamma angle relative to the Platform coordinate system.

6.2.136 OrientationAcceleration3DPlatformXYZ

Namespace: UMAA::Common::Orientation::OrientationAcceleration3DPlatformXYZ

Description: Specifies the acceleration for each axis of an Orientation.

Table 381: OrientationAcceleration3DPlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
xAccel	AngleAcceleration	Specifies the acceleration of the platform's rotation about the longitudinal axis in the XYZ Platform coordinate system.
yAccel	AngleAcceleration	Specifies the acceleration of the platform's rotation about the lateral axis in the XYZ Platform coordinate system.
zAccel	AngleAcceleration	Specifies the acceleration of the platform's rotation about the vertical axis in the XYZ Platform coordinate system.

6.2.137 OrientationVel3D

Namespace: UMAA::Common::Measurement::OrientationVel3D

Description: Specifies the rate of change for each axis of an Orientation.

Table 382: OrientationVel3D Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchRate	PitchRate	Specifies the rate of change of the platform's pitch angle.
rollRate	RollRate	Specifies the rate of change of the platform's roll angle.
yawRate	YawRate	Specifies the rate of change of the platform's yaw angle.

6.2.138 PassiveContactType

Namespace: UMAA::SA::PassiveContactReport::PassiveContactType

Description: This structure is used to report the passive contact characteristics.

Table 383: PassiveContactType Structure Definition

Attribute Name	Attribute Type	Attribute Description
bearing	BearingAngle	True bearing to the center of the contact at the time of detection for the current report.
bearingRate	AngleRate	Rate of change in bearing to the center of the contact at the time of detection for the current report.
bearingRateUncertainty	ContactUncertainty	Uncertainty of the contact's bearing rate estimate.
bearingUncertainty	ContactUncertainty	Uncertainty of the contact's bearing estimate.
contactLevel	PeakSoundPressureLevel	The categorization of acoustic properties of the detected signal.
contactType†	PassiveContactFeatureEnumType	The current identity of the contact track. Used to indicate the threat level into which the contact was classified.
course†	CourseTrueNorth	True course to the center of the contact at the time of detection for the current report.
courseUncertainty†	ContactUncertainty	Uncertainty of the contact's course.
declination†	AngleHalf	Declination angle where up is positive and down is negative.
declinationUncertainty†	ContactUncertainty	Uncertainty of the contact's declination estimate.
narrowbandContactFrequency†	FrequencyHertz	Primary frequency of a Narrowband contact. This parameter is set to zero if no primary frequency is detected.
range†	Distance	Estimated distance to the contact based on tracking over time. Does not include depth ranges (flat range).
rangeUncertainty†	ContactUncertainty	Uncertainty of the distance to the contact.

Attribute Name	Attribute Type	Attribute Description
threatType†	TrackIdentityEnumType	The current identity of the contact track. Used to infer the threat level of the contact.

6.2.139 PathReporterType

Namespace: UMAA::SA::PathReporterSpecs::PathReporterType

Description: This structure is used to report the capabilities the service supports. This implementation may support one or more types of path, and specify further limitations on the constraints used in the message querying the path. For example, an implementation may specify that it only supports a certain maximum number of points, and/or a fixed target resolution.

Table 384: PathReporterType Structure Definition

Attribute Name	Attribute Type	Attribute Description
maxDistance	Distance	The maximum supported distance for the list of points to be returned.
maxPoints	Count	The maximum supported element count for the list of points to be returned.
maxTgtResolution	Distance	The maximum supported distance between reported path points. For implementations that do not support interpolation, the Min and Max should be identical.
maxTime	DurationSeconds	The maximum supported time for the list of points to be returned.
minTgtResolution	Distance	The minimum supported distance between reported path points. For implementations that do not support interpolation, the Min and Max should be identical.
pathType	PathWayEnumType	Path type.

6.2.140 PitchYNEDRequirement

Namespace: UMAA::Common::Orientation::PitchYNEDRequirement

Description: A requirement that specifies a pitch relative to the NED coordinate system.

Table 385: PitchYNEDRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitch	PitchYNEDType	Defines a pitch relative to the NED system.
pitchTolerance†	PitchYNEDTolerance	Describes the pitch bounding limits.

6.2.141 PitchYNEDTolerance

Namespace: UMAA::Common::Orientation::PitchYNEDTolerance

Description: A down or up angle tolerance.

Table 386: PitchYNEDTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	PitchYNEDType	Defines the steepest downangle allowed.
upperlimit	PitchYNEDType	Defines the steepest upangle allowed.

6.2.142 PitchYNEDType

Namespace: UMAA::Common::Orientation::PitchYNEDType

Description: Specifies a pitch relative to the NED coordinate system.

Table 387: PitchYNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitch	PitchHalfAngle	Defines a pitch relative to the NED coordinate system.

6.2.143 Polygon

Namespace: UMAA::Common::Measurement::Polygon

Description: Specifies an area defined by a polygon.

Table 388: Polygon Structure Definition

Attribute Name	Attribute Type	Attribute Description
lineKind	LineSegmentEnumType	Indicates the type of lines that form the polygon.
referencePoint	sequence<GeoPosition2D> max size = 128	Specifies the geospatial points defining the vertices of a polygon. Three or more points are needed to define a polygon.

6.2.144 PolygonAreaRequirementType

Namespace: UMAA::MM::BaseType::PolygonAreaRequirementType

Description: A requirement that specifies the area of a polygon.

Table 389: PolygonAreaRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
area	Polygon	Specifies the area enclosed by the simple polygon.
areaTolerance†	PolygonAreaToleranceType	Specifies the tolerance for the polygon area.

6.2.145 PolygonAreaToleranceType

Namespace: UMAA::MM::BaseType::PolygonAreaToleranceType

Description: Defines the polygon area tolerance.

Table 390: PolygonAreaToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
limit	Distance	Specifies the amount of error in position allowed from the polygon area.

6.2.146 PoseType

Namespace: UMAA::Common::Environment::PoseType

Description: Specifies current position and orientation in the global coordinate system.

Table 391: PoseType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude†	MSLAltitude	The current orthometric height above the Geoid (Mean Sea Level) of the vehicle.
altitudeAGL†	DistanceAGL	The current height above ground level of the vehicle.
altitudeASF†	DistanceASF	The current height above the sea floor of the vehicle.
altitudeGeodetic†	GeodeticAltitude	The current geodetic height above the ellipsoid of the vehicle.
attitude	Orientation3DNEDType	The current orientation (roll, pitch, yaw) of the vehicle.
course	CourseTrueNorth	The current course angle of the vehicle.
depth†	DistanceBSL	The current depth of the maritime vehicle.
navigationSolution	NavigationSolutionEnumType	The desired navigation solution (estimated or measured).
position	GeoPosition2D	The current position of the vehicle.
positionCovariance†	CovariancePositionNEDType	The current error covariance value of the position data.

6.2.147 Position3DBodyXYZ

Namespace: UMAA::Common::Measurement::Position3DBodyXYZ

Description: Specifies a three-dimensional location on a Cartesian coordinate system relative to the origin of the body.

Table 392: Position3DBodyXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
xAxis	XPosition	The position on the body x-axis, which extends out the front of the reference body.
yAxis	YPosition	The position on the body y-axis, which extends out the right (starboard) of the reference body.
zAxis	ZPosition	The position on the body z-axis, which is perpendicular to the x and y axes and is directed downward from the center of the body.

6.2.148 PropellerPitchAnglePropulsorRequirementType

Namespace: UMAA::Common::Angle::PropellerPitchAnglePropulsorRequirementType

Description: Describes the required pitch angle of the propulsor's propeller for propulsors with a variable angle propeller.

Table 393: PropellerPitchAnglePropulsorRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
propellerPitchAnglePropulsor	PropellerPitchAnglePropulsor	Describes the required value of the propeller pitch angle.
propellerPitchAnglePropulsorTolerance†	PropellerPitchAnglePropulsorToleranceType	Describes the required propeller pitch angle's tolerance.

6.2.149 PropellerPitchAnglePropulsorToleranceType

Namespace: UMAA::Common::Angle::PropellerPitchAnglePropulsorToleranceType

Description: Describes a tolerance range for the propeller pitch angle.

Table 394: PropellerPitchAnglePropulsorToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	PropellerPitchAnglePropulsor	Specifies the lower limit of allowable values.
upperlimit	PropellerPitchAnglePropulsor	Specifies the upper limit of allowable values.

6.2.150 PropulsorCommandType

Namespace: UMAA::EO::PropulsorsControl::PropulsorCommandType

Description: Describes the desired state of a propulsor on the vehicle.

Table 395: PropulsorCommandType Structure Definition

Attribute Name	Attribute Type	Attribute Description
gamma†	GammaAnglePropulsorRequirementType	Sets the angle of a propulsor about the y-axis with one or two articulations.
propellerPitch†	PropellerPitchAnglePropulsorRequirementType	Sets the angle of the propeller pitch for propulsors that have a variable pitch propeller.
propulsion	EngineRPMSpeedRequirement	Sets the speed of the propulsor.

Attribute Name	Attribute Type	Attribute Description
rho†	RhoAnglePropulsorRequirementType	Sets the angle of a propulsor about the z-axis with one or two articulations.

6.2.151 PropulsorSpecsType

Namespace: UMAA::EO::PropulsorsSpecs::PropulsorSpecsType

Description: This structure is used to report the specifications of a propulsor on the vehicle.

Table 396: PropulsorSpecsType Structure Definition

Attribute Name	Attribute Type	Attribute Description
counterRotator	boolean	If true, specifies that the propulsor is a counter rotator.
gamma†	GammaAnglePropulsorToleranceType	Specifies an upper and lower gamma angle limit for the propulsor; attribute is not defined when the propulsor angle is fixed.
name	StringShortDescription	The name of the propulsor unit.
orientation	Orientation3DPlatformType	Specifies the rotation offset of the propulsor coordinate reference frame with respect to the vehicle coordinate reference frame.
position	Position3DBodyXYZ	Specifies the position offset of the propulsor coordinate reference frame with respect to the vehicle coordinate reference frame.
propellerPitch†	PropellerPitchAnglePropulsorToleranceType	Specifies an upper and lower angle limit for the propeller angle; attribute is not defined for propellers that have static pitch angles.
propulsionLowerLimit†	FrequencyRPM	Specifies a lower speed limit for propulsors that are reversible; attribute is not defined for propulsors that are not reversible.
propulsionUpperLimit	FrequencyRPM	Specifies an upper speed limit for the propulsor.
rho†	RhoAnglePropulsorToleranceType	Specifies an upper and lower rho angle limit for the propulsor; attribute is not defined when the propulsor angle is fixed.

6.2.152 PropulsorStatusType

Namespace: UMAA::EO::PropulsorsStatus::PropulsorStatusType

Description: This structure is used to report the current (i.e. feedback) propulsor configuration status values.

Table 397: PropulsorStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
gamma†	GammaAnglePropulsor	Provides the current angle of a propulsor with one or two articulations.
propellerPitch†	PropellerPitchAnglePropulsor	Provides the current angle of the propeller for propulsors with a variable pitch propeller.
propulsion	FrequencyRPM	Provides the current speed of the propulsor.
rho†	RhoAnglePropulsor	Provides the current angle of a propulsor with one or two articulations.

6.2.153 RGBType

Namespace: UMAA::SEM::BaseType::RGBType

Description: An element that describes a color.

Table 398: RGBType Structure Definition

Attribute Name	Attribute Type	Attribute Description
blue	ColorComponent	The blue component of the color.
green	ColorComponent	The green component of the color.
red	ColorComponent	The red component of the color.

6.2.154 RacetrackObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::RacetrackObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the racetrack objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::RacetrackObjectiveDetailedStatusType

Table 399: RacetrackObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isCrossTrackLimitAchieved	boolean	Indicates that the cross track limit requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing the racetrack maneuver (True = system is executing the racetrack maneuver, False = system is in transit).

Attribute Name	Attribute Type	Attribute Description
isSpeedAchieved	boolean	Indicates that the speed requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
referencePosition	GeoPosition2D	Defines the reference position for the racetrack pattern. (Important if reference position was not defined in original objective).
timePatternAchieved	DateTime	Defines the absolute time at which the racetrack pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the racetrack pattern is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.155 RacetrackObjectiveType

Namespace: UMAA::MM::BaseType::RacetrackObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for following the racetrack pattern. The racetrack objective is achieved by having the vehicle execute the racetrack maneuver at a defined elevation (or current elevation if not defined) as specified by the reference position, length, radius and orientation, with the defined speed, trackTolerance and turnDirection. If position is not specified, then the current vehicle position is used as the reference position for the racetrack pattern. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::RacetrackObjectiveType

Table 400: RacetrackObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the remaining pattern portion of the objective. If not specified, duration is not used to determine when the racetrack maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the racetrack maneuver. If not specified, the maneuver is performed at the current elevation.

Attribute Name	Attribute Type	Attribute Description
length	Distance	Defines the length between the semicircles at either end for the racetrack pattern.
loops†	SizeReal	Defines the number of loops around the racetrack pattern to execute; if not specified, the loops attribute is not used to determine when the racetrack maneuver is complete.
orientation	DirectionVariantType	Defines the orientation of the racetrack, measured perpendicular to the length axis.
position†	GeoPosition2D	Defines the reference position for the racetrack pattern. If not specified, the reference position is the current vehicle position.
radius	Distance	Defines the radius of the semicircles for the racetrack pattern.
speed	SpeedRequirementVariantType	Defines the vehicle speed to maintain while executing the racetrack maneuver.
trackTolerance	DistanceRequirementType	Defines the maximum allowable cross track error while executing the racetrack maneuver.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the racetrack pattern location before transitioning to the racetrack maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the racetrack pattern location before transitioning to the racetrack maneuver.
turnDirection	WaterTurnDirectionEnumType	Defines the turn direction while executing the racetrack maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.156 RegularPolygonObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::RegularPolygonObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the regular polygon objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::RegularPolygonObjectiveDetailedStatusType

Table 401: RegularPolygonObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isCrossTrackLimitAchieved	boolean	Indicates that the cross track limit requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.

Attribute Name	Attribute Type	Attribute Description
isInPattern	boolean	Is system currently executing the regular polygon maneuver (True = system is executing the regular polygon maneuver, False = system is in transit).
isSpeedAchieved	boolean	Indicates that the speed requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
referencePosition	GeoPosition2D	Defines the reference position for the regular polygon pattern. (Important if reference position was not defined in original objective).
timePatternAchieved	DateTime	Defines the absolute time at which the regular polygon pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the regular polygon pattern is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.157 RegularPolygonObjectiveType

Namespace: UMAA::MM::BaseType::RegularPolygonObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for following the polygon pattern. The regular polygon objective is achieved by having the vehicle execute the regular polygon maneuver at a defined elevation (or current elevation if not defined) as specified by the reference position, diameter, and orientation, with the defined speed, trackTolerance, and turnDirection. If position is not specified, then the current vehicle position is used as the reference position for the regular polygon pattern. Elevation, speed, and trackTolerance include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If both the duration attribute and the loops attribute are defined, complete the action(s) at whichever attribute condition completes first. If neither the duration attribute nor the loops attribute is specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::RegularPolygonObjectiveType

Table 402: RegularPolygonObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
diameter	Distance	Defines the diameter of a circumscribed circle around the polygon for the regular polygon pattern.

Attribute Name	Attribute Type	Attribute Description
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the remaining pattern portion of the objective. If not specified, duration is not used to determine when the regular polygon maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the regular polygon maneuver. If not specified, the maneuver is performed at the current elevation.
loops†	SizeReal	Defines the number of loops around the regular polygon pattern to execute; if not specified, the loops attribute is not used to determine when the regular polygon maneuver is complete.
numberSides	SidesCount	Defines the number of sides for the regular polygon pattern.
orientation	DirectionVariantType	Defines the orientation of the regular polygon pattern, measured from the reference position of the polygon to one point on the polygon.
position†	GeoPosition2D	Defines the reference position for the regular polygon pattern. If not specified, the reference position is the current vehicle position.
speed	SpeedRequirementVariantType	Defines the vehicle speed to maintain while executing the regular polygon maneuver.
trackTolerance	DistanceRequirementType	Defines the maximum allowable cross track error while executing the regular polygon maneuver.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the regular polygon pattern location before transitioning to the regular polygon maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the regular polygon pattern location before transitioning to the regular polygon maneuver.
turnDirection	WaterTurnDirectionEnumType	Defines the turn direction while executing the regular polygon maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.158 ResourceAllocationControlInfo

Namespace: UMAA::SO::ResourceAllocation::ResourceAllocationControlInfo

Description: This structure is used to define attributes related to the controller of a resource.

Table 403: ResourceAllocationControlInfo Structure Definition

Attribute Name	Attribute Type	Attribute Description
controlSession†	ResourceAllocationControlSession	Information on the consumer currently controlling the resource. If empty, this resource is not currently allocated for control.
resourceID	IdentifierType	The identifier of the resource being controlled.

6.2.159 ResourceAllocationControlSession

Namespace: UMAA::SO::ResourceAllocation::ResourceAllocationControlSession

Description: This structure is used to define attributes related to the current controller of a resource.

Table 404: ResourceAllocationControlSession Structure Definition

Attribute Name	Attribute Type	Attribute Description
controllingConsumer	IdentifierType	The source identifier of the resource consumer in control of the resource.
endTime†	DateTime	The absolute end time of the consumer's control. After this time is reached, the resource is available to be controlled by another process. If this field is empty, then the duration is assumed to be infinite.

6.2.160 ResourceAllocationDefinitionType

Namespace: UMAA::SO::ResourceAllocation::ResourceAllocationDefinitionType

Description: This structure is used to define the attributes associated with a resource - that is, a collection of related service providers whose functionality cannot be executed simultaneously.

Table 405: ResourceAllocationDefinitionType Structure Definition

Attribute Name	Attribute Type	Attribute Description
relatedSources	sequence<IdentifierType> max size = 100	The source identifiers of each service that is logically part of this resource. For instance, this resource could represent driving-related services at large. This field would then contain the source of each driving-related service provider active in the system.
resourceID	IdentifierType	The identifier of the resource being controlled.

6.2.161 ResourceAllocationPriorityInfo

Namespace: UMAA::SO::ResourceAllocation::ResourceAllocationPriorityInfo

Description: This structure is used to define the configuration of resource control priority for a particular resource.

Table 406: ResourceAllocationPriorityInfo Structure Definition

Attribute Name	Attribute Type	Attribute Description
priorities	sequence<IdentifierType> max size = 64	The priority-ordered (low to high) sequence of client identifiers.
resourceID	IdentifierType	The identifier of the resource being controlled.

6.2.162 RhoAnglePropulsorRequirementType

Namespace: UMAA::Common::Angle::RhoAnglePropulsorRequirementType

Description: Describes the required value of the propulsor's rho angle.

Table 407: RhoAnglePropulsorRequirementType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rhoAnglePropulsor	RhoAnglePropulsor	Describes the required rho value.
rhoAnglePropulsorTolerance†	RhoAnglePropulsorToleranceType	Describes the required rho value's tolerance.

6.2.163 RhoAnglePropulsorToleranceType

Namespace: UMAA::Common::Angle::RhoAnglePropulsorToleranceType

Description: Describes a tolerance range for the required rho value.

Table 408: RhoAnglePropulsorToleranceType Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	RhoAnglePropulsor	Specifies the lower limit of allowable values.
upperlimit	RhoAnglePropulsor	Specifies the upper limit of allowable values.

6.2.164 RollXNEDRequirement

Namespace: UMAA::Common::Orientation::RollXNEDRequirement

Description: A requirement that specifies a roll relative to the NED coordinate system.

Table 409: RollXNEDRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
roll	RollXNEDType	Defines a roll relative to the NED system.
rollTolerance†	RollXNEDTolerance	Describes the roll bounding limits.

6.2.165 RollXNEDTolerance

Namespace: UMAA::Common::Orientation::RollXNEDTolerance

Description: A rotational tolerance.

Table 410: RollXNEDTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	RollXNEDType	Defines the lower bound.
upperlimit	RollXNEDType	Defines the upper bound.

6.2.166 RollXNEDType

Namespace: UMAA::Common::Orientation::RollXNEDType

Description: Specifies a roll relative to the NED coordinate system.

Table 411: RollXNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
roll	RollAngle	Defines a roll relative to the NED coordinate system.

6.2.167 RouteObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::RouteObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the route objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::RouteObjectiveDetailedStatusType

Table 412: RouteObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
crossTrackError†	Distance	The current cross track error. This value must be set if trackTolerance is defined and must not be set if trackTolerance is not defined.
currentWaypointID	NumericGUID	Identification of the current waypoint in the route.
distanceRemaining	Distance	The estimated distance remaining for the route.
distanceToWaypoint	Distance	The distance from the vehicle position to the current waypoint.
isCrossTrackLimitAchieved†	boolean	Indicates that the track limit requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedToWaypoint	SpeedVariantType	The commanded speed to the current waypoint.
waypointDetailedStatus→setID	LargeSet<WaypointDetailedStatusType>	The detailed status for each waypoint. This attribute is implemented as a large set, see subsection 3.8 for an explanation. The associated topic is UMAA::MM::BaseType::RouteObjectiveDetailedStatusTypeWaypointDetailedStatusSetElement.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.168 RouteObjectiveType

Namespace: UMAA::MM::BaseType::RouteObjectiveType

Description: This structure is used to report an element that describes a clearly defined goal specifying the action(s) required for following a route. The route objective is achieved by having the vehicle achieve each waypoint in order and is complete when the final waypoint is achieved. Each waypoint is specified by a position along with a captureRadius, and each waypoint optionally includes a specified attitude and elevation. If the attitude or elevation is not specified, then any attitude or elevation at the waypoint is acceptable, respectively. The desired effect when a waypoint is achieved is for the vehicle's position to be within a distance less than or equal to the captureRadius. The captureRadius includes an optional tolerance. If specified and the system is unable to achieve the captureRadius, then the waypoint can be considered completed if the vehicle's position is within a distance less than or equal to the specified tolerance. If the system is unable to complete the waypoint within the specified tolerance, then the objective is considered to have failed. If the captureRadius tolerance is not

specified and the system is not able to achieve the captureRadius, then it is a "best effort" to achieve the waypoint, and is therefore not considered a cause for the objective to fail. If attitude or elevation is specified, then the desired effect when a waypoint is achieved includes the vehicle's attitude aligning with this specified attitude or obtaining the specified elevation, respectively, at the waypoint position along with the positional effect described above. Both attitude and elevation include an optional tolerance. If the system is unable to achieve the specified attitude or elevation within their tolerance, then the objective is considered to have failed. If a tolerance is not defined for attitude or elevation, then it is a "best effort" to achieve the specified attitude or elevation, respectively, and is therefore not considered a cause for the objective to fail. Additionally, there is an optional trackTolerance. If trackTolerance is specified, then the defined path between waypoints is specified to be the track line created by the waypoints with a cross track error less than or equal to the trackTolerance. The trackTolerance includes an optional tolerance. If specified, the vehicle is allowed to operate within this tolerance without having the objective fail. Otherwise, if the tolerance is violated, then the objective is considered to have failed. If the trackTolerance tolerance is not specified, then it is a "best effort" to maintain the specified trackTolerance, and is therefore not considered a cause for the objective to fail. If the trackTolerance is not specified, then path between waypoints is not specified and therefore left for the system to determine the best path. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::RouteObjectiveType

Table 413: RouteObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
routeDescription	StringShortDescription	Description of a route.
waypoints→listID	LargeList<WaypointType>	Specifies the route the vehicle is to travel. This attribute is implemented as a large list, see subsection 3.8 for an explanation. The associated topic is UMAA::MM::BaseType::RouteObjectiveTypeWaypointsListElement.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.169 ScreenRandomWalkObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::ScreenRandomWalkObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the screen random walk objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::ScreenRandomWalkObjectiveDetailedStatusType

Table 414: ScreenRandomWalkObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isAreaAchieved	boolean	Indicates that the area requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.

Attribute Name	Attribute Type	Attribute Description
isInPattern	boolean	Is system currently executing screen random walk (True = system is executing screen random walk, False = system is in transit).
timePatternAchieved	DateTime	Defines the absolute time at which the screen random walk is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the screen random walk is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.170 ScreenRandomWalkObjectiveType

Namespace: UMAA::MM::BaseType::ScreenRandomWalkObjectiveType

Description: The goal of the screen random walk objective is to execute a random walk maneuver within a specified area that is relative to a guide (e.g., a high value unit (HVU)). This structure is used to specify the guide and the random walk area relative to the guide where the screening must be conducted. The screen random walk objective is achieved by having the vehicle execute random vectors at a defined elevation (or current elevation if not defined) while maintaining the vehicle position within a defined area relative to a guide. The area is defined by specifying a start and end bearing along with a min and max range. The random vectors can be configured by specifying min/max speeds and min/max time on course. Area and elevation include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::ScreenRandomWalkObjectiveType

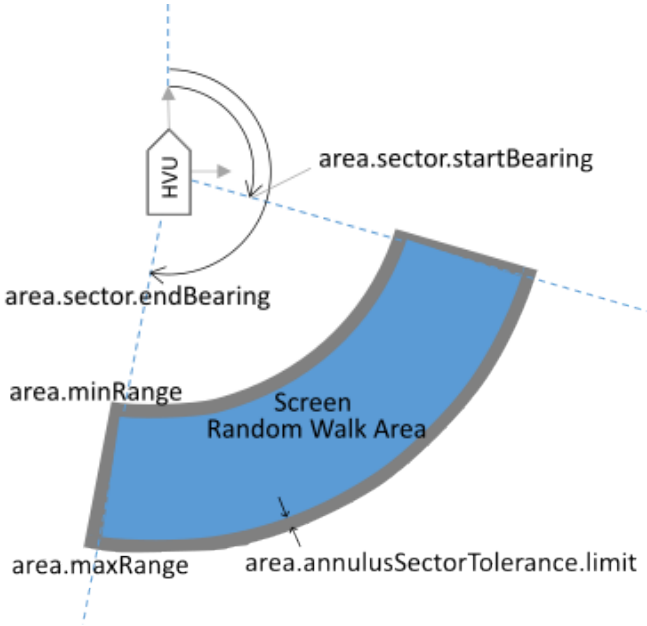

Figure 49: A Screen Random Walk

Table 415: ScreenRandomWalkObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
area	AnnulusSectorRequirementType	Defines the area the vehicle must stay in while executing the random walk maneuver.
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the random walk portion of the objective. If not specified, then duration is not used to determine when the random walk maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the random walk. If not specified, the maneuver is performed at the current elevation.
guideID	NumericGUID	Defines the ID of the guide relative to which the area for conducting the random walk is defined.
maxSpeed	SpeedVariantType	Defines the maximum vehicle speed on a given vector.
maxTimeOnCourse	DurationSeconds	Defines the maximum time spent on a given vector.
minSpeed	SpeedVariantType	Defines the minimum vehicle speed on a given vector.
minTimeOnCourse	DurationSeconds	Defines the minimum time spent on a given vector.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the random walk location before transitioning to the random walk maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the random walk location before transitioning to the random walk maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.171 SpeedRequirementVariantType

Namespace: UMAA::Common::Speed::SpeedRequirementVariantType

Description: Union Type. Speed of the vehicle.

Table 416: SpeedRequirementVariantType Union(s)

Type Name	Type Description
AirSpeedRequirementVariantType	Defines the speed through air.
EngineRPMSpeedRequirementVariantType	Defines the engine RPM.
GroundSpeedRequirementVariantType	Defines the speed over ground.
VehicleSpeedModeRequirementVariantType	Defines the speed mode.
WaterSpeedRequirementVariantType	Defines the speed through water.

6.2.172 SpeedVariantType

Namespace: UMAA::Common::Speed::SpeedVariantType

Description: Union Type. Speed of the vehicle.

Table 417: SpeedVariantType Union(s)

Type Name	Type Description
AirSpeedVariantType	Defines the speed through air.
EngineRPMSpeedVariantType	Defines the engine RPM.
GroundSpeedVariantType	Defines the speed over ground.
VehicleSpeedModeVariantType	Defines the speed mode.
WaterSpeedVariantType	Defines the speed through water.

6.2.173 StateTriggerType

Namespace: UMAA::MM::BaseType::StateTriggerType

Description: This structure is used to specify a mechanism that attempts to initiate a planned state for a Mission Plan, Task Plan, or Objective when its defined conditional expression is determined to transition to logically true.

Table 418: StateTriggerType Structure Definition

Attribute Name	Attribute Type	Attribute Description
conditionalID	NumericGUID	Uniquely identifies the conditional.
count†	Count	Specifies the number of times the trigger can be used. If not included, assumed to be infinite.
state	TriggerStateEnumType	Specifies the state of the trigger.

6.2.174 StationkeepObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::StationkeepObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the stationkeep objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::StationkeepObjectiveDetailedStatusType

Table 419: StationkeepObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
bearingGuide†	HeadingTarget	Specifies the current bearing to the guide.
bearingMagneticNorth†	HeadingMagneticNorth	Specifies the current bearing to the guide.
bearingTrueNorth†	HeadingTrueNorthAngle	Specifies the current bearing to the guide.
closingSpeed	GroundSpeed	Defines the current closing speed relative to the guide.
distanceFromTrack	Distance	Defines the current distance from the guide.
guideLost	boolean	The system is unable to determine the guide's location (True = system is not tracking the guide's location, False = system is currently tracking the guide's location).
isAreaAchieved	boolean	Indicates that the area requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isInPattern	boolean	Is system currently executing the stationkeep (True = system is executing stationkeep, False = system is in transit).
timePatternAchieved	DateTime	Defines the absolute time at which the stationkeep is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the stationkeep is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.175 StationkeepObjectiveType

Namespace: UMAA::MM::BaseType::StationkeepObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for station keeping. The station keep objective is achieved by maintaining the vehicle location at a defined elevation (or current elevation if not defined) within a defined area relative to a guide. The area is defined by specifying a start and end bearing along with a min and max range. Area and elevation include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::StationkeepObjectiveType

Table 420: StationkeepObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
area	AnnulusSectorRequirementType	Defines the area the vehicle must stay in while executing the station keep maneuver.
duration†	DurationSeconds	After the transit portion of the objective is complete, defines the duration to execute the station keep portion of the objective. If not specified, then duration is not used to determine when the station keep maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the station keep maneuver. If not specified, the maneuver is performed at the current elevation.
guideID	NumericGUID	Defines the ID of the guide relative to which the area for conducting station keep is defined.
guideLostFailureDelay†	DurationSeconds	After the station keep objective is achieved (either initially or after an update), defines the amount of time to delay transitioning to a failed state when the system is unable to determine the guide's location. This measured time is reset each time the guide is tracked. If not defined, then a system configured delay time is used.
transitElevation†	ElevationVariantType	Defines the elevation used while transiting to the station keep location before transitioning to the station keep maneuver. If not specified, transit at the current elevation.
transitSpeed	SpeedVariantType	Defines the speed used while transiting to the station keep location before transitioning to the station keep maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.176 VectorObjectiveDetailedStatusType

Namespace: UMAA::MM::BaseType::VectorObjectiveDetailedStatusType

Description: This structure provides a detailed execution status for the vector objective. This type is a specialization of ObjectiveDetailedStatusType.

Topic: UMAA::MM::BaseType::VectorObjectiveDetailedStatusType

Table 421: VectorObjectiveDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
isDirectionAchieved	boolean	Indicates that the direction requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
isSpeedAchieved	boolean	Indicates that the speed requested is currently achieved. Achievement may be lost and regained resulting in multiple changes to this attribute.
timePatternAchieved	DateTime	Defines the absolute time at which the vector is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	Defines the absolute time at which the vector is estimated to be completed (optional in case duration is not specified).
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveDetailedStatusType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveDetailedStatusType generalization.

6.2.177 VectorObjectiveType

Namespace: UMAA::MM::BaseType::VectorObjectiveType

Description: This structure is used to describe a clearly defined goal specifying the action(s) required for the vector objective to achieve the specified speed, direction of travel, and altitude or depth of the vehicle. The vector objective is achieved by having the vehicle execute the vector maneuver at a defined elevation (or current elevation if not defined) as specified by direction, directionMode, and speed. The vector can be configured by setting an optional depthChangePitch. Direction, elevation, and speed include optional tolerances for their values. If specified, the vehicle is allowed to operate within these tolerances without having the objective fail. Otherwise, if any tolerances are violated after the objective is initially achieved, then the objective is considered to have failed. If a tolerance is not specified for any of these attributes, then it is a "best effort" to maintain the specified value for that attribute, and is therefore not considered a cause for the objective to fail. If the duration attribute is not specified, the action(s) should continue indefinitely, or until interrupted by some other action. This type is a specialization of ObjectiveType.

Topic: UMAA::MM::BaseType::VectorObjectiveType

Table 422: VectorObjectiveType Structure Definition

Attribute Name	Attribute Type	Attribute Description
depthChangePitch†	PitchYNEDType	Defines the desired angle of the vehicle when traversing to the requested elevation for UUVs.
direction	DirectionRequirementVariantType	Defines the vehicle direction to maintain while executing the vector maneuver.
directionMode	DirectionModeEnumType	Defines the direction mode while executing the vector maneuver.
duration†	DurationSeconds	Defines the duration to execute the global vector. If not specified, then duration is not used to determine when the vector maneuver is complete.
elevation†	ElevationRequirementVariantType	Defines the vehicle elevation to maintain while executing the vector maneuver. If not specified, the maneuver is performed at the current elevation.
speed	SpeedRequirementVariantType	Defines the vehicle speed to maintain while executing the vector maneuver.
specializationReferenceTimestamp	DateTime	This field identifies the timestamp that signals the end of an atomic update to the instance of the specialization. NOTE: Ties this element back to the specializationID in ObjectiveType generalization.
specializationReferenceID*	NumericGUID	NOTE: Ties this element back to the ObjectiveType generalization.

6.2.178 VehicleSpeedModeRequirementVariantType

Namespace: UMAA::Common::Speed::VehicleSpeedModeRequirementVariantType

Description: Defines the speed mode.

Table 423: VehicleSpeedModeRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
mode	VehicleSpeedModeEnumType	Specifies the speed mode.

6.2.179 VehicleSpeedModeVariantType

Namespace: UMAA::Common::Speed::VehicleSpeedModeVariantType

Description: Defines the speed mode.

Table 424: VehicleSpeedModeVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
mode	VehicleSpeedModeEnumType	Specifies the speed mode.

6.2.180 VolumeBallastFlowRateType

Namespace: UMAA::EO::BallastTank::VolumeBallastFlowRateType

Description: The desired flow rate to fill or empty the ballast pump measured by volume.

Table 425: VolumeBallastFlowRateType Structure Definition

Attribute Name	Attribute Type	Attribute Description
volumeBallastFlowRate	VolumetricFlowRate	Specifies the desired flow rate to fill or empty the ballast pump measured by volume.

6.2.181 WaterSpeedRequirement

Namespace: UMAA::Common::Speed::WaterSpeedRequirement

Description: Defines the speed through water.

Table 426: WaterSpeedRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	SpeedLocalWaterMass	Specifies speed through water.
speedTolerance†	WaterSpeedTolerance	Specifies the tolerance for a speed through water.

6.2.182 WaterSpeedRequirementVariantType

Namespace: UMAA::Common::Speed::WaterSpeedRequirementVariantType

Description: Defines the speed through water.

Table 427: WaterSpeedRequirementVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	WaterSpeedRequirement	Specifies speed through water.

6.2.183 WaterSpeedTolerance

Namespace: UMAA::Common::Speed::WaterSpeedTolerance

Description: Defines the speed through water tolerance.

Table 428: WaterSpeedTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	SpeedLocalWaterMass	Specifies the lower limit of allowable values for the water speed.
upperlimit	SpeedLocalWaterMass	Specifies the upper limit of allowable values for the water speed.

6.2.184 WaterSpeedVariantType

Namespace: UMAA::Common::Speed::WaterSpeedVariantType

Description: Defines the speed through water.

Table 429: WaterSpeedVariantType Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	SpeedLocalWaterMass	Specifies speed through water.

6.2.185 WaypointDetailedStatusType

Namespace: UMAA::MM::BaseType::WaypointDetailedStatusType

Description: This structure provides a detailed execution status for the waypoint.

Table 430: WaypointDetailedStatusType Structure Definition

Attribute Name	Attribute Type	Attribute Description
avgCrossTrackError†	Distance	The average cross track error for this track; optional if queued or cross track limit not specified.
avgSpeed†	GroundSpeed	The average vehicle speed to the waypoint; optional if queued.
errors	StringShortDescription	Human details about why the operation is not currently meeting waypoint.
feedback	StringShortDescription	The explanation for what is being performed and the 'why' of any changes.
maxCrossTrackError†	Distance	The maximum cross track error for this track; optional if queued or cross track limit not specified.
maxSpeed†	GroundSpeed	The maximum vehicle speed to the waypoint; optional if queued.
state	WaypointStateEnumType	The current execution state of the waypoint.
warnings	StringShortDescription	Any warnings the operation should be aware of when performing this objective.
waypointID	NumericGUID	A unique identification of the waypoint.

6.2.186 WaypointType

Namespace: UMAA::MM::BaseType::WaypointType

Description: This structure is used to define attributes of a waypoint including position, depth, and speed.

Table 431: WaypointType Structure Definition

Attribute Name	Attribute Type	Attribute Description
attitude†	Orientation3DNEDRequirement	Defines the attitude at the waypoint the vehicle must achieve. If not included, then the vehicle's attitude at the waypoint is left for the system to determine.
captureRadius	DistanceRequirementType	Defines a capture radius for the waypoint, which is the maximum distance the vehicle position is allowed to be from the waypoint position for it to be considered achieved.
elevation†	ElevationRequirementVariantType	Defines the elevation at the waypoint the vehicle must achieve. If not included, then the vehicle's elevation at the waypoint is left for the system to determine.
name†	StringShortDescription	A short name for the waypoint.
position	GeoPosition2D	Defines the position of the waypoint the vehicle must achieve.
speed†	SpeedVariantType	Defines the vehicle speed to maintain while executing the route. If not included, then the vehicle speed is left for the system to determine (e.g., in order to meet defined time constraint).

Attribute Name	Attribute Type	Attribute Description
trackTolerance†	DistanceRequirementType	Defines the maximum allowable cross track error, where the previous waypoint (or the vehicle's current position when execution of the route objective begins for the first waypoint) is used to define a track line. If not included, then the path between waypoints is left for the system to determine.
waypointID	NumericGUID	A unique identification of the waypoint.

6.2.187 WorldTransformType

Namespace: UMAA::Common::Environment::WorldTransformType

Description: This is a base structure used to report the image-to-world transform according to the ESRI world file standard.

Table 432: WorldTransformType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pixelSizex	Distance	The pixel size in the x-direction in map units.
pixelSizey	Distance	The pixel size in the y-direction in map units.
rotationx	Angle	The rotation about the x-axis.
rotationy	Angle	The rotation about the y-axis.
upperLeftCoordinatex	Distance	The x-coordinate of the center of the upper-left pixel of the image, given in the map frame.
upperLeftCoordinatey	Distance	The y-coordinate of the center of the upper-left pixel of the image, given in the map frame and expressed as a negative.

6.2.188 YawZNEDRequirement

Namespace: UMAA::Common::Orientation::YawZNEDRequirement

Description: A requirement that specifies a yaw relative to the NED coordinate system.

Table 433: YawZNEDRequirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
yaw	YawZNEDType	Defines a yaw relative to the NED system.
yawTolerance†	YawZNEDTolerance	Describes the yaw bounding limits.

6.2.189 YawZNEDTolerance

Namespace: UMAA::Common::Orientation::YawZNEDTolerance

Description: A directional tolerance.

Table 434: YawZNEDTolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
failureDelay†	DurationSeconds	After the value with a specified tolerance is achieved (either initially or after an update), defines the amount of time to delay transitioning a command or objective to a failed state when the specified tolerance is violated. This measured time is reset each time the value is determined to be within the specified tolerance. If not defined, then a system configured delay time is used.
lowerlimit	YawZNEDType	Defines the lower bound.
upperlimit	YawZNEDType	Defines the upper bound.

6.2.190 YawZNEDType

Namespace: UMAA::Common::Orientation::YawZNEDType

Description: Specifies a yaw relative to the NED coordinate system.

Table 435: YawZNEDType Structure Definition

Attribute Name	Attribute Type	Attribute Description
yaw	YawAngle	Defines a yaw relative to the NED coordinate system.

6.3 Enumerations

Enumerations are used extensively throughout UMAA. This section lists the values associated with each enumeration defined in UCS-UMAA.

6.3.1 ActivationStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ActivationStateEnumType

Description: A mutually exclusive set of values that defines the activation state for the sensor.

Table 436: ActivationStateEnumType Enumeration

Enumeration Value	Description
ACTIVE	Indicates sensor is actively pinging and/or acquiring.
ERROR	Indicates sensor has detected an error that requires attention.
OFF	Indicates sensor is unpowered.
READY	Indicates sensor is configured and awaiting activation.
STANDBY	Indicates sensor is powered on but not configured.

6.3.2 ActivationStateTargetEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ActivationStateTargetEnumType

Description: A mutually exclusive set of values that defines the activation state.

Table 437: ActivationStateTargetEnumType Enumeration

Enumeration Value	Description
ACTIVE	Directs sensor to actively ping and/or acquire.
OFF	Directs sensor to shutdown.
READY	Directs sensor to process configuration and await activation.
STANDBY	Directs sensor to be powered on but not configured.

6.3.3 AutoOffModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AutoOffModeEnumType

Description: A mutually exclusive set of values that defines the auto off mode.

Table 438: AutoOffModeEnumType Enumeration

Enumeration Value	Description
DEACTIVATE	Indicates that active elements should be turned off when abnormal conditions are indicated.
SHUTDOWN	Indicates that the sensor should be shutdown when abnormal conditions are indicated.

6.3.4 BilgeStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BilgeStateEnumType

Description: A mutually exclusive set of values that defines the states of each bilge pump on the vehicle.

Table 439: BilgeStateEnumType Enumeration

Enumeration Value	Description
FAULT	Fault
OFF	Off
ON	On

6.3.5 CloudCoverEnumType

Namespace: UMAA::Common::OrderedEnumeration::CloudCoverEnumType

Description: A mutually exclusive set of values that defines a discrete set of cloud cover conditions.

Table 440: CloudCoverEnumType Enumeration

Enumeration Value	Description
BROKEN	Cloud cover is from five-eighths to seven-eighths of the sky.
CLEAR	There are no clouds below 12,000 ft (zero-eighths of the sky).
FEW	Cloud cover is from one-eighth to two-eights of the sky.
OVERCAST	Cloud cover is complete (eight-eighths of the sky).
SCATTERED	Cloud cover is from three-eighths to four-eighths of the sky.

6.3.6 CommandStatusReasonEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CommandStatusReasonEnumType

Description: Defines a mutually exclusive set of reasons why a command status state transition has occurred.

Table 441: CommandStatusReasonEnumType Enumeration

Enumeration Value	Description
CANCELED	Indicates a transition to the CANCELED state when the command is canceled successfully.
INTERRUPTED	Indicates a transition to the FAILED state when the command has been interrupted by a higher priority process.
OBJECTIVE_FAILED	Indicates a transition to the FAILED state when the commanded resource is unable to achieve the command's objective due to external factors.
RESOURCE_FAILED	Indicates a transition to the FAILED state when the commanded resource is unable to achieve the command's objective due to resource or platform failure.

Enumeration Value	Description
RESOURCE_REJECTED	Indicates a transition to the FAILED state when the commanded resource rejects the command for some reason.
SERVICE_FAILED	Indicates a transition to the FAILED state when the commanded resource is unable to achieve the command's objective due to processing failure.
SUCCEEDED	Indicates the conditions to proceed to this state have been met and a normal state transition has occurred.
TIMEOUT	Indicates a transition to the FAILED state when the command is not acknowledged within some defined time bound.
UPDATED	Indicates a transition back to the ISSUED state from a non-terminal state when the command has been updated.
VALIDATION_FAILED	Indicates a transition to the FAILED state when the command contains missing, out-of-bounds, or otherwise invalid parameters.

6.3.7 ContingencyBehaviorEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ContingencyBehaviorEnumType

Description: A mutually exclusive set of values that defines the behavior of the vehicle used in case of emergency during the mission.

Table 442: ContingencyBehaviorEnumType Enumeration

Enumeration Value	Description
CONTINUE	Continue the mission
FINISH	Finish the mission
HOME	Return to home
LOITER	Loiter
NONE	None
VEHICLE_SPECIFIC	None of the above (specific to the vehicle)

6.3.8 ContinuousTestEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ContinuousTestEnumType

Description: Defines a mutually exclusive set of continuous test options.

Table 443: ContinuousTestEnumType Enumeration

Enumeration Value	Description
DISABLED_NO_TEST	Indicates that the continuous test is disabled, and/or that there is no continuous test.
FULL_TEST	Indicates a full continuous test.
NON_INTRUSIVE_TESTS_ONLY	Indicates non-intrusive tests only.

6.3.9 DirectionModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::DirectionModeEnumType

Description: Specifies whether direction is a course or heading.

Table 444: DirectionModeEnumType Enumeration

Enumeration Value	Description
COURSE	Specifies that direction is the course of the vehicle, which is the direction of motion of the vehicle over the ground.
HEADING	Specifies that direction is the heading of the vehicle, which is the direction in which the vehicle's bow is pointing.

6.3.10 DomainEnumType

Namespace: UMAA::Common::MaritimeEnumeration::DomainEnumType

Description: A mutually exclusive set of values that defines the area or region in which a vehicle operates.

Table 445: DomainEnumType Enumeration

Enumeration Value	Description
AIR	Air
GROUND	Surface, ground
SURFACE	Surface, water
UNDERSEA	Undersea

6.3.11 EmitterStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::EmitterStateEnumType

Description: Defines a mutually exclusive set of values that defines the emitter state.

Table 446: EmitterStateEnumType Enumeration

Enumeration Value	Description
ALLOWED	Emissions are allowed.
SECURED	Emissions are secured, i.e. no emissions.

6.3.12 EngineKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::EngineKindEnumType

Description: Defines a mutually exclusive set of values that defines the engine kind.

Table 447: EngineKindEnumType Enumeration

Enumeration Value	Description
DIESEL	A diesel engine.
GAS	A gas engine.

6.3.13 FLSBeamwidthEnumType

Namespace: UMAA::Common::MaritimeEnumeration::FLSBeamwidthEnumType

Description: Defines a mutually exclusive set of values that defines the FLS beamwidth.

Table 448: FLSBeamwidthEnumType Enumeration

Enumeration Value	Description
MEDIUM	Beamwidth is medium.
NARROW	Beamwidth is narrow.
WIDE	Beamwidth is wide.

6.3.14 FLSConfigModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::FLSConfigModeEnumType

Description: Defines a mutually exclusive set of values that defines the FLS configuration mode.

Table 449: FLSConfigModeEnumType Enumeration

Enumeration Value	Description
DEV_TEST	Put the sonar into a mode where the configuration is fully determined by the sonar. This allows the sonar developers to test new configurations without hijacking another configuration.
DIVE	The vehicle is diving off the surface and needs to quickly get a look of the bottom to determine the terrains threat level.
PASSIVE_ONLY	The sonar is configured to never make noise or actively ping but still listen for interesting acoustic emissions.
SEARCH_BOTTOM	The sonar is configured to search the seafloor to find contacts and targets of interest. If this enum member is selected, additional configuration is required.
SEARCH_VOLUME	The sonar is configured to search the volume (area between the surface and the seafloor) to find contacts and targets of interest. If this enum member is selected, additional configuration is required.
SURFACE	The vehicle is coming to the surface. The sonar should investigate the surface to determine if it is clear of hazards.
TEST	Put the sonar into a dedicated mode defined by the operator. Disables the sonar's ability to optimize. If this enum member is selected, additional configuration is required.
TRANSIT	The vehicle is intended to be transiting and thus doesn't have to operate at full capacity to save power. Vehicle believes it is in a safe environment or not collecting data.

6.3.15 FLSWaveformLengthEnumType

Namespace: UMAA::Common::MaritimeEnumeration::FLSWaveformLengthEnumType

Description: Defines a mutually exclusive set of values that defines the FLS waveform length.

Table 450: FLSWaveformLengthEnumType Enumeration

Enumeration Value	Description
LONG	Waveform length is long.
MEDIUM	Waveform length is medium.
SHORT	Waveform length is short.
XSHORT	Waveform length is extra short.

6.3.16 HandoverResultEnumType

Namespace: UMAA::Common::MaritimeEnumeration::HandoverResultEnumType

Description: A mutually exclusive set of values that defines the status of the mode transition on a vehicle platform.

Table 451: HandoverResultEnumType Enumeration

Enumeration Value	Description
DEFERRED	Control handover deferred temporarily by current controller
DENIED	Control handover denied by current controller
GRANTED	Control handover granted by current controller
INSUFFICIENT_AUTHORITY	Control transfer requestor had insufficient authority to take control from current controller
NOT_AVAILABLE	Vehicle is unavailable for control handover
TIMEOUT	Control handover timed out because current controller did not respond in the allotted time

6.3.17 HoverKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::HoverKindEnumType

Description: A mutually exclusive set of values that defines the hover priority of the vehicle.

Table 452: HoverKindEnumType Enumeration

Enumeration Value	Description
LAT_LON_PRIORITY	Prioritize maintaining a latitude/longitude position
Z_PRIORITY	Prioritize maintaining an elevation

6.3.18 IlluminatorStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::IlluminatorStateEnumType

Description: A mutually exclusive set of values that defines the state of an illuminator.

Table 453: IlluminatorStateEnumType Enumeration

Enumeration Value	Description
FLASHING	Flashing
OFF	Off
ON	On

6.3.19 ImageFormatEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ImageFormatEnumType

Description: A mutually exclusive set of values that defines the image formats.

Table 454: ImageFormatEnumType Enumeration

Enumeration Value	Description
ARW	Sony Alpha RAW proprietary camera sensor output format
BMP	BitMaP image
CR2_RAW	Canon RAW 2 (version 2) proprietary camera sensor output format
DNG	Digital NeGative image file, a royalty free Adobe format
GEOJPEG	GEOreferenced JPEG
GEOTIFF	GEOreferenced TIFF
GIF	Graphics Interchange Format
GPR	Go-Pro RAW proprietary camera sensor output format
JPEG	Joint Photographic Experts Group (default if optionally specified)
NEF	Nikon Electric Format RAW proprietary camera sensor output format
PGM	Portable Gray Map
PNG	Portable Network Graphic
PNM	Portable aNy Map
PPM	Portable Pixmap Format
TIFF	Tag Image File Format

6.3.20 InitiatedTestEnumType

Namespace: UMAA::Common::MaritimeEnumeration::InitiatedTestEnumType

Description: Defines a mutually exclusive set of initiated test options.

Table 455: InitiatedTestEnumType Enumeration

Enumeration Value	Description
DESTRUCTIVE	Indicates a destructive test.
NON_DESTRUCTIVE	Indicates a non-destructive test.

6.3.21 InterferenceEnumType

Namespace: UMAA::Common::MaritimeEnumeration::InterferenceEnumType

Description: A mutually exclusive set of values that defines the type of interference.

Table 456: InterferenceEnumType Enumeration

Enumeration Value	Description
ACOUSTIC	Interference type is sound produced by a different sonar system.
NONACOUSTIC	Interference type is non-sonar based noise.
UNKNOWN	Interference type is from an undetermined source.

6.3.22 LandmarkEnumType

Namespace: UMAA::Common::MaritimeEnumeration::LandmarkEnumType

Description: A mutually exclusive set of values that defines the types of landmarks.

Table 457: LandmarkEnumType Enumeration

Enumeration Value	Description
CLUSTER_OBJECT	Landmark is a cluster object.
LARGE_OBJECT	Landmark is a large object.
MARKED	Landmark is marked by a geographical tag.
TERRAIN	Landmark is terrain.

6.3.23 LineSegmentEnumType

Namespace: UMAA::Common::Enumeration::LineSegmentEnumType

Description: A mutually exclusive set of values that defines the line segment types used for navigation.

Table 458: LineSegmentEnumType Enumeration

Enumeration Value	Description
GREAT_CIRCLE	The line segment should be traversed as one following a great circle. A great circle is the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere.
RHUMB	The line segment should be traversed as one following a rhumb line. A rhumb line represents an arc cross all meridians of longitude at the same angle (i.e. a path with constant bearing).

6.3.24 CommandStatusEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CommandStatusEnumType

Description: Defines a mutually exclusive set of values that defines the states of a command as it progresses towards completion.

Table 459: CommandStatusEnumType Enumeration

Enumeration Value	Description
CANCELED	The command was canceled by the requestor before the command completed successfully.
COMMANDED	The command has been placed in the resource's command queue but has not yet been accepted.
COMPLETED	The command has been completed successfully.
EXECUTING	The command is being performed by the resource and has not yet been completed.
FAILED	The command has been attempted, but was not successful.
ISSUED	The command has been issued to the resource (typically a sensor or streaming device), but processing has not yet commenced.

6.3.25 NavigationSolutionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::NavigationSolutionEnumType

Description: A mutually exclusive set of values that defines the navigation solution.

Table 460: NavigationSolutionEnumType Enumeration

Enumeration Value	Description
ESTIMATED	Interpolated, projected, and/or produced through computation.
GROUND_TRUTH	Navigation solution from an external source that is considered to be correct.
MEASURED	Provided by a sensing system.

6.3.26 ObjectiveExecutorControlEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ObjectiveExecutorControlEnumType

Description: Defines a mutually exclusive set of values for the objective executor control.

Table 461: ObjectiveExecutorControlEnumType Enumeration

Enumeration Value	Description
EXECUTE	Execute the objective.
PAUSE	Pause the execution of the objective.
RESUME	Resume the execution of a paused objective.

6.3.27 ObjectiveExecutorStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ObjectiveExecutorStateEnumType

Description: Defines a mutually exclusive set of values for the objective executor state.

Table 462: ObjectiveExecutorStateEnumType Enumeration

Enumeration Value	Description
CANCELED	The objective has been canceled.
CANCELING	The objective is in the process of being canceled.
COMPLETED	The objective been completed. Collection tasks are considered complete when the resulting product is processed and disseminated. All other tasks are complete once the vehicle transitions from the executing state (vehicle releases weapon, stops jamming, etc.).
EXECUTING	The objective has begun execution (slews sensor and begins collect, begins to prepare weapons for release, starts jamming, etc.).
FAILED	The objective has failed.
MODIFYING	The objective executing is being modified.
PAUSED	The objective's execution is paused.
PAUSING	The objective is in the process of being paused.
QUEUED	The system has queued the objective for future execution.
RESUMING	The objective is in the process of being resumed.

6.3.28 ObjectiveExecutorStateReasonEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ObjectiveExecutorStateReasonEnumType

Description: Defines a mutually exclusive set of values for the objective executor state reason.

Table 463: ObjectiveExecutorStateReasonEnumType Enumeration

Enumeration Value	Description
BUS_MSG_DISPOSE	Indicates transition to CANCELING state due to dispose message.
BUS_MSG_UPDATE	Indicates transition to MODIFYING state due to modification of bus message with same key as executing objective.
CANNOT_PERFORM_UNDER_CONSTRAINTS	Indicates a transition to FAILED due to current system constraints.
COMMAND_VALIDATION_FAILED	Indicates a transition to the FAILED state when the command contains missing, out-of-bounds, or otherwise invalid parameters.
COMMANDED	Indicates transition due to incoming command.
INTERNAL_FAILURE	Indicates transition to FAILED based on failure within the objective executor.
LOWER_SERVICE_FAILED	Indicates transition to FAILED based on failure within lower level service leveraged by this objective executor.
LOWER_SERVICE_INTERRUPTED	Indicates transition to FAILED or PAUSING based on lower level service being interrupted by higher priority request.
LOWER_SERVICE_REJECTED	Indicates transition to FAILED based on rejection by lower level service when it was commanded by this objective executor.
LOWER_SERVICE_TIMEOUT	Indicates transition to FAILED based on unresponsiveness of lower level service leveraged by this objective executor.
OBJECTIVE_REPLACED	Indicates transition to FAILED due to another objective arriving at the objective executor.
SUCCEEDED	Indicates the conditions to proceed to this state have been met and a normal state transition has occurred.

6.3.29 OperationalModeControlEnumType

Namespace: UMAA::Common::MaritimeEnumeration::OperationalModeControlEnumType

Description: Defines a mutually exclusive set of values for the vehicle operational control mode. Note that MANUAL mode is controlled outside of UMAA (e.g., a physical switch).

Table 464: OperationalModeControlEnumType Enumeration

Enumeration Value	Description
AUTONOMOUS	Sets the operational mode to an unmanned mode where the associated commands that are from the onboard autonomy are allowed to execute.
REMOTE	Sets the operational mode to an unmanned mode where only the associated commands that are from the remote operator are allowed to execute.
STANDBY	Sets the operational mode to an unmanned mode where none of the associated commands are allowed to execute.

6.3.30 OperationalModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::OperationalModeEnumType

Description: Defines a mutually exclusive set of values for the vehicle operational mode.

Table 465: OperationalModeEnumType Enumeration

Enumeration Value	Description
AUTONOMOUS	Operating in an unmanned mode where operational commands from onboard autonomy are executed.
MANUAL	Operating in a manned mode where no operational commands can be executed.
REMOTE	Operating in an unmanned mode where only operational commands from a remote operator are allowed to execute.
STANDBY	Operating in an unmanned mode where no operational commands are allowed to execute.

6.3.31 PassiveContactFeatureEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PassiveContactFeatureEnumType

Description: A mutually exclusive set of values that defines the acoustic properties of the passive contact type.

Table 466: PassiveContactFeatureEnumType Enumeration

Enumeration Value	Description
BROADBAND	Contact has a wide frequence spectrum.
NARROWBAND	Contact has a narrow frequency spectrum.
TRANSIENT	Contact is transitory.

6.3.32 PathWayEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PathWayEnumType

Description: A mutually exclusive set of values that defines the different types of path.

Table 467: PathWayEnumType Enumeration

Enumeration Value	Description
HISTORICAL_GLOBAL	Historical global path
HISTORICAL_LOCAL	Historical local path
PLANNED_GLOBAL	Planned global path
PLANNED_LOCAL	Planned local path

6.3.33 PowerOnTestEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PowerOnTestEnumType

Description: Defines a mutually exclusive set of power on test options.

Table 468: PowerOnTestEnumType Enumeration

Enumeration Value	Description
DISABLED_NO_TEST	Indicates that the power on test is disabled, and/or that there is no power on test.
FULL_TEST	Indicates a full power on test.
QUICK_TEST	Indicates a quick power on test.

6.3.34 PowerStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PowerStateEnumType

Description: A mutually exclusive set of values that defines the power state or status of a resource.

Table 469: PowerStateEnumType Enumeration

Enumeration Value	Description
EMERGENCY_POWER	Power for the resource is requested/reported to use emergency power.
POWER_OFF	Power for the resource is requested/reported to be off.
POWER_ON	Power for the resource is requested/reported to be on.
POWER_STANDBY	Power for the resource is requested/reported to use/be in stand-by mode or not fully powered up: for example if power is provided only to enable communications to the resource.

6.3.35 PrecipitationEnumType

Namespace: UMAA::Common::Enumeration::PrecipitationEnumType

Description: A mutually exclusive set of values that defines types of precipitation.

Table 470: PrecipitationEnumType Enumeration

Enumeration Value	Description
DRIZZLE	Precipitation that consists of numerous minute droplets of water less than 0.5 mm in diameter that reach the Earth's surface.
FOG	Fog is water droplets suspended in the air at or above the Earth's surface.
HAZE	Haze is an aggregation in the atmosphere of very fine, widely dispersed, solid or liquid particles, or both, giving the air an opalescent appearance that subdues colors.
RAIN	Precipitation that falls to earth in droplets of water more than 0.5 mm in diameter.
SHOWERS	Rain that falls intermittently over a small area. The rain from an individual shower can be heavy or light, but doesn't cover a large area or last more than an hour or so.
SNOW	Precipitation in the form of ice crystals formed directly from the freezing [deposition] of the water vapor in the air.

Enumeration Value	Description
THUNDERSTORMS	A rain or snow shower in which there is lightning. Thunder is always caused by lightning. In general, the upward and downward winds, updrafts and downdrafts, in thunderstorms are more violent than those in ordinary showers.

6.3.36 ProcessingUnitEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ProcessingUnitEnumType

Description: A mutually exclusive set of values that defines the processing unit type.

Table 471: ProcessingUnitEnumType Enumeration

Enumeration Value	Description
CPU	Central Processing Unit
DSP	Digital Signal Processing Unit
FPGA	Field-Programmable Gate Array Unit
GPU	Graphics Processing Unit
NPU	Neural Processing Unit
PhPU	Photonic Processing Unit
PPU	Physics Processing Unit
QPU	Quantum Processing Unit
SPU	Synergistic Processing Unit
TPU	Tensor Processing Unit
VPU	Vision Processing Unit

6.3.37 PumpStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PumpStateEnumType

Description: A mutually exclusive set of values that defines the mode of operation of each pump on the vehicle.

Table 472: PumpStateEnumType Enumeration

Enumeration Value	Description
FAULT	Faulted
OFF	Off
ON_FORWARD	Running forward direction
ON_REVERSE	Running reverse direction

6.3.38 ResourceAllocationStatusEnumType

Namespace: UMAA::Common::Enumeration::ResourceAllocationStatusEnumType

Description: A mutually exclusive set of values that defines allocation status for domain resources.

Table 473: ResourceAllocationStatusEnumType Enumeration

Enumeration Value	Description
ALLOCATED	The resource is allocated.
ALLOCATED_W_LAUNCH_RECOVERY	The resource is allocated with launch and recovery.
AVAILABLE	The resource is available.
FAULT	There was a fault in the resource allocation.
FORCED_ALLOCATION	The resource allocation is forced.
FORCED_ALLOCATION_W_LAUNCH_RECOVERY	The resource allocation with launch and recovery is forced.
RELEASED	The resource is released.
TEMPORARILY_UNAVAILABLE	The resource is temporarily unavailable.
UNAVAILABLE	The resource is unavailable.

6.3.39 SpecificLOIEnumType

Namespace: UMAA::Common::Enumeration::SpecificLOIEnumType

Description: A mutually exclusive set of val-ues that defines the Level Of Interoperability (LOI) of a UCS system.

Table 474: SpecificLOIEnumType Enumeration

Enumeration Value	Description
LOI_1	The data link level of interoperability (LOI)-1 is indirect receipt of UAV related data.
LOI_2	The data link level of interoperability (LOI)-2 is direct receipt of ISR or other data where "direct" covers reception of UAV data by the UCS when it has direct communication with the UAV.
LOI_3	The data link level of interoperability (LOI)-3 is control and monitoring of the UAV payload in additional to direct receipt of ISR or other data.
LOI_4	The data link level of interoperability (LOI)-4 is control and monitoring of the UAV, less launch and recovery.
LOI_5	The data link level of interoperability (LOI)-5 is control and monitoring of the UAV (LOI-4) plus launch and recovery functions.

6.3.40 TamperDetectionStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::TamperDetectionStateEnumType

Description: A mutually exclusive set of values that defines the state of tamper detection.

Table 475: TamperDetectionStateEnumType Enumeration

Enumeration Value	Description
ALWAYS_ENABLED_OR_CLEAR	ALWAYS_ENABLED for reporting data; CLEAR for clearing the previous activities
DISABLED	Disabled
ENABLED	Enabled

6.3.41 TrackCategoryEnumType

Namespace: UMAA::Common::MaritimeEnumeration::TrackCategoryEnumType

Description: Defines a mutually exclusive set of values that indicates the type of track, by category

Table 476: TrackCategoryEnumType Enumeration

Enumeration Value	Description
ADS_B_DIRECTIONAL_AIR	Definition to be defined
ADS_B_DIRECTIONAL_SURFACE	Definition to be defined
ADS_B_NONDIRECTIONAL_AIR	Definition to be defined
ADS_B_NONDIRECTIONAL_SURFACE	Definition to be defined
AIR	Air
ASW	Definition to be defined
EMERGENCY	Emergency
EW	Definition to be defined
LAND_POINT	Land Point
LAND_TRACK	Land Track
MP_AREA	Definition to be defined
MP_LINE	Definition to be defined
NA	Definition to be defined
NO_STATEMENT	Definition to be defined
POINTER	Definition to be defined
REF_POINT	Definition to be defined
SP_AREA	Definition to be defined
SPACE	Space
SUB_SURFACE	Below the surface
SURFACE	Surface

6.3.42 TrackIdentityEnumType

Namespace: UMAA::Common::MaritimeEnumeration::TrackIdentityEnumType

Description: Defines a mutually exclusive set of values that defines the current identity of the contact track.

Table 477: TrackIdentityEnumType Enumeration

Enumeration Value	Description
ASSUMED_FRIEND	A track which is assumed to be a friend because of its characteristics, behavior, or origin
FAKER	A friendly track acting as a "suspect" track for exercise purposes only
FRIEND	A track belonging to a declared friendly nation
HOSTILE	A track which is eligible to be engaged
JOKER	A friendly track acting as a "hostile" track for exercise purposes only
NEUTRAL	A track whose characteristics, behavior, origin, or nationality indicate that it is neither supporting nor opposing friendly forces
PENDING	A track for which identification is to be determined
SUSPECT	A track which is potentially hostile because of its characteristics, behavior, origin or nationality
UNKNOWN	An evaluated track which has not been identified

6.3.43 TriggerStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::TriggerStateEnumType

Description: Defines a mutually exclusive set of values that defines the trigger state.

Table 478: TriggerStateEnumType Enumeration

Enumeration Value	Description
CANCEL	A canceling state.
PAUSE	A pausing state.
PLAN	A planning state.
QUEUE	A queueing state.
RESTART	A restarting state.
RESUME	A resuming state.

6.3.44 VehicleSpeedModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VehicleSpeedModeEnumType

Description: A mutually exclusive set of values that defines the type of performance speed of the vehicle.

Table 479: VehicleSpeedModeEnumType Enumeration

Enumeration Value	Description
LRC	Long Range Cruise. A speed that optimizes time, distance and fuel consumption for a vehicle (definition of "optimized" is subjective. Example: for a planing hull, this is usually the minimum planing speed, even though lower speeds can achieve longer endurance or range.)

Enumeration Value	Description
MEC	Maximum Endurance Cruise. The speed that maximizes the time a vehicle can travel.
MRC	Maximum Range Cruise. The speed that maximizes the distance a vehicle can travel.
SLOW	Slow speed. Minimum speed at which the vehicle can operate (definition of "operate" is subjective. Example: minimum speed to achieve maneuverability, engine idle speed/gear clutched in "idle ahead", etc.)
VEHICLE_SPECIFIC	Preset speed for the vehicle, that is in the range of speeds for the subject vehicle

6.3.45 WaterTurnDirectionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::WaterTurnDirectionEnumType

Description: A mutually exclusive set of values that define the types of turn directions applied by the vehicle during turns.

Table 480: WaterTurnDirectionEnumType Enumeration

Enumeration Value	Description
LEFT_TURN	The vehicle will make left turns.
RIGHT_TURN	The vehicle will make right turns.

6.3.46 WaypointStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::WaypointStateEnumType

Description: Defines a mutually exclusive set of values for the objective executor state.

Table 481: WaypointStateEnumType Enumeration

Enumeration Value	Description
ACHIEVED	Waypoint is complete and achieved within tolerance.
COMPLETED	Waypoint is complete, but not achieved within tolerance.
EXECUTING	Waypoint is executing.
FAILED	Waypoint failed to complete.
QUEUED	Waypoint is queued for execution.

6.3.47 WeatherSeverityEnumType

Namespace: UMAA::Common::OrderedEnumeration::WeatherSeverityEnumType

Description: A mutually exclusive set of values that defines classification levels for severe weather conditions.

Table 482: WeatherSeverityEnumType Enumeration

Enumeration Value	Description
EXTREME	The adverse weather conditions are extreme.
LIGHT	The adverse weather conditions are light.
MODERATE	The adverse weather conditions are moderate.
NONE	There is no adverse weather.
SEVERE	The adverse weather conditions are severe.

6.4 Type Definitions

This section describes the type definitions for UMAA. The table below lists how UMAA defined types are mapped to the DDS primitive types.

Table 483: Type Definitions

Type Name	Primitive Type	Range of Values	Description
AccelerationScalar	double	maxInclusive=1310.68 minInclusive=-1310.68 units=MeterPerSecondSquared referenceFrame=Counting	This type stores acceleration in m/s/s.
AirTemperature	double	maxInclusive=100 minInclusive=-100 units=Celsius	Specifies the air temperature.
Angle	double	maxInclusive=3.1415926535897932 minInclusive=-3.1415926535897932 units=Radian referenceFrame=Counting	Specifies the amount of turning necessary to bring one ray, line or plane into coincidence with or parallel to another. The measurement is stated in radians between -pi and pi.
AngleAcceleration	double	maxInclusive=10000 minInclusive=-10000 units=RadiansPerSecondSquared	Represents the rate of change of angular velocity.
AngleHalf	double	maxInclusive=1.5707963267948966 minInclusive=-1.5707963267948966 units=Radian	Represents a half angle.
AngleRate	double	maxInclusive=62.831 minInclusive=-62.831 units=RadianPerSecond referenceFrame=Counting	Represents the rate of change of angular displacement measured in radians per second.
BearingAngle	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian	The yaw angle relative to true north measured from the platform to the target.
BinaryValue	octet[256]		Describes a binary value type.
BooleanEnumType	boolean		A mutually exclusive set of values that defines the truth values of logical algebra.
ByteValue	octet	maxInclusive=255 minInclusive=0	Describes a byte value type.
CharValue	char	maxInclusive=255 minInclusive=0	Describes a char value type.
ColorComponent	long	maxInclusive=255 minInclusive=0	Represents a color component.
Conductivity	double	units=SiemensPerMeter referenceFrame=LocalWaterMass	Represents an object property that describes how well the object conducts electricity.

Type Name	Primitive Type	Range of Values	Description
ContactUncertainty	double		Specifies the uncertainty of a measurement.
Count	long	referenceFrame=Counting minInclusive=-2147483648 maxInclusive=2147483647	Represents a whole (non-fractional) number that can be positive, negative or zero.
CourseTrueNorth	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=TrueNorth	Specifies the direction of the platform's motion relative to true north.
CovarAccelPlatformXYZ	double	maxInclusive=10000 minInclusive=-10000 referenceFrame=PlatformXYZ units=MetersSquaredPerSecondsSquaredSquared	Describes a measure of dependence that indicates the acceleration-acceleration error covariance in the PlatformXYZ coordinate system.
CovarOrientationAccelPlatformXYZ	double	maxInclusive=100 minInclusive=-100 referenceFrame=PlatformXYZ units=RadiansSquaredPerSecondsSquaredSquared	Describes a measure of linear dependence that in indicates orientation acceleration-orientation acceleration covariance in the PlatformXYZ coordinate system.
CovarOrientationNED	double	maxInclusive=10000 minInclusive=-10000 referenceFrame=NED units=RadiansSquared	Describes a measure of linear dependence that indicates orientation-orientation covariance in the NED coordinate system.
CovarOrientationVelNED	double	maxInclusive=10000 minInclusive=-10000 referenceFrame=NED units=RadiansSquaredPerSecondSquared	Describes a measure of linear dependence that indicates orientation velocity-orientation velocity covariance in the NED coordinate system.
CovarPosNED	double	maxInclusive=10000 minInclusive=-10000 referenceFrame=NED units=MetersSquared	Describes a measure of linear dependence that indicates position-position covariance in the NED coordinate system.
DateTimeNanoseconds	long	units=Nanoseconds minInclusive=0 maxInclusive=999999999	The number of nanoseconds elapsed within the current second.
DateTimeSeconds	longlong	units=Seconds minInclusive=-9223372036854775807 maxInclusive=9223372036854775807	The seconds offset from the standard POSIX (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.
Density	double	maxInclusive=3e17 minInclusive=0 units=KilogramPerCubicMeter referenceFrame=Counting	Realizes DensityType: an Entity that describes the number of occurrences of a repeating event per unit volume.
DewPointTemperature	double	maxInclusive=100 minInclusive=-100 units=Celsius	Specifies the dew point temperature.
Distance	double	maxInclusive=401056000 minInclusive=0 units=Meter referenceFrame=Counting	This type stores a distance in meters.

Type Name	Primitive Type	Range of Values	Description
DistanceAGL	double	minInclusive=0.0 units=Meter referenceFrame=AGL	Describes the height above ground level of the vehicle.
DistanceASF	double	maxInclusive=401056000 minInclusive=0 units=Meter referenceFrame=ASF	The altitude or distance above the sea floor in meters.
DistanceBSL	double	maxInclusive=10000 minInclusive=0 units=Meter referenceFrame=BSL	The distance below sea level in meters.
DoubleValue	double		Describes a double value type.
DurationHours	double	maxInclusive=10505 minInclusive=0 units=Hour referenceFrame=Counting	Represents a time duration in hours.
DurationMilliseconds	double	minInclusive=0 units=Millisecond referenceFrame=Counting	Represents a time duration.
DurationSeconds	double	maxInclusive=37817280 minInclusive=0 units=Seconds referenceFrame=Counting	Represents a time duration in seconds.
ElectroMagneticFrequencyHertz	double	maxInclusive=1e25 minInclusive=0 units=Hertz referenceFrame=Hertz	Used for specifying the frequency on the electromagnetic spectrum.
EngineSpeed	double	referenceFrame=Counting units=RevolutionsPerMinute minInclusive=-100000 maxInclusive=100000	This type stores number of occurrences in revolutions per minute (RPM). Negative number is used for reverse RPM.
FrequencyHertz	double	maxInclusive=1e10 minInclusive=0.0 units=Hertz referenceFrame=Counting	This type stores Frequency in Hz.
FrequencyRPM	long	maxInclusive=100000 minInclusive=-100000 units=RevolutionsPerMinute referenceFrame=Counting	This type stores number of occurrences in revolutions per minute (RPM). Negative number is used for reverse RPM.
GammaAnglePropulsor	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=PropulsorXYZ	Specifies the angle of the propulsor about the Y-Axis of the vehicle reference frame.
GeodeticAltitude	double	maxInclusive=700000 minInclusive=-10000 units=Meter axisAbbrev=Altitude axisDirection=up axisUnit=Meter rangeMeaning=exact resolution=0.0000000001	Used for measuring position and increases in magnitude as position extends upward. Altitude measurements are expressed in meters.

Type Name	Primitive Type	Range of Values	Description
GeodeticLatitude	double	axisAbbrev=Latitude axisDirection=north/south axisUnit=Degrees maximumValue=90.0 minimumValue=-90.0 rangeMeaning=exact resolution=0.0000000001	Used for measuring position and increases in magnitude as position extends from the south pole to the north pole. Latitude measurements are expressed in degrees.
GeodeticLongitude	double	axisAbbrev=Longitude axisDirection=east axisUnit=Degrees maximumValue=180.0 minimumValue=-180.0 rangeMeaning=wraparound resolution=0.0000000001	Used for measuring position and increases in magnitude as position extends eastward. Longitude measurements are expressed in degrees. Longitude measurements are periodic and whose limits (min and max), while mathematically discontinuous, represent a continuous range.
GroundSpeed	double	maxInclusive=299792458 minInclusive=-299792458 units=MeterPerSecond referenceFrame=Ground	The magnitude of the horizontal velocity vector of a vehicle relative to the ground.
HeadingCurrentDirection	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=CurrentDirection	Describes heading with respect to the current direction.
HeadingMagneticNorth	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=MagneticNorth	Heading as an angle specified with respect to Magnetic North.
HeadingTarget	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=Target	Describes heading with respect to the target.
HeadingTrueNorthAngle	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=TrueNorth	Describes heading with respect to True North.
HeadingWindDirection	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=WindDirection	Describes heading with respect to the wind direction.
IlluminatorBeamWidth	double	maxInclusive=6.28318530718 minInclusive=0 units=Radian	Specifies the illuminator beam width.
IlluminatorIntensityLevel	double	maxInclusive=100 minInclusive=0	Specifies the illuminator intensity level.

Type Name	Primitive Type	Range of Values	Description
IndicatedAirspeed	double	maxInclusive=299792458 minInclusive=0 units=MeterPerSecond referenceFrame=LocalAirMass	This type specifies the magnitude of an aircraft's velocity (the rate of change of its position). Indicated airspeed (IAS) is the airspeed read directly from the airspeed indicator on an aircraft, driven by the pitot-static system.
IntegerValue	long		Describes an integer value type.
LargeCollectionSize	long	maxInclusive=2147483647 minInclusive=0	Specifies the size of a Large Collection.
LargeCount	uint64	maxInclusive=18446744073709551615 minInclusive=0 referenceFrame=Counting	Represents a count of elements.
Mass	double	maxInclusive=100000000 minInclusive=0 units=Kilogram referenceFrame=Counting	This type stores mass in kilograms.
MassFlowRate	double	units=KilogramsPerSecond referenceFrame=Counting	Represents the mass flow rate measured in kilogram per second.
MaxEngineOilPressure	double	maxInclusive=512 minInclusive=0 units=KiloPascal	Describes the maximum oil pressure for an engine.
MSLAltitude	double	minInclusive=0.0 units=Meter referenceFrame=Altitude	Describes the current orthometric height above the Geoid (Mean Sea Level).
NumericGUID	octet[16]	minInclusive=0 maxInclusive=(2^128)-1	Represents a 128-bit number according to RFC 4122 variant 2.
PeakSoundPressureLevel	double	maxInclusive=400 minInclusive=-400 units=Decibel	Describes the peak sound pressure level.
Percent	double	maxInclusive=1000 minInclusive=0 units=Percent referenceFrame=Counting	Defines a percentage where 100% = 100.0. Values greater than 100% are allowed.
PitchHalfAngle	double	maxInclusive=1.5707963267948966 minInclusive=-1.5707963267948966 units=Radian referenceFrame=PlatformNED	Specifies the platform's rotation about the lateral axis (e.g. the axis parallel to the wings) in a locally level, North-East-Down coordinate system centered on the platform. Pitch is zero when the platform is "nose to tail level" in the North-East plane. The measurement is stated in radians between -0.5 pi and 0.5 pi.
PitchRate	double	maxInclusive=32.767 minInclusive=-32.767 units=RadianPerSecond referenceFrame=Counting	Specifies the rate of change of the platform's pitch angle.

Type Name	Primitive Type	Range of Values	Description
PressureKiloPascals	double	maxInclusive=51200 minInclusive=0 units=KiloPascal referenceFrame=STP	Represents barometric pressure and is stored in KiloPascals.
PressurePascals	double	maxInclusive=107558000 minInclusive=0 units=Pascal referenceFrame=STP	Represents pressure and is measured in Pascals.
Priority	long	maxInclusive=255 minInclusive=0	Represents the priority as a positive integer. Low numbers represent low priority while higher numbers represent high priority.
PropellerPitchAnglePropulsor	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=PropulsorXYZ	Specifies the angle of the propulsor propeller for propulsors with a variable pitch propeller.
RelativeHumidity	double	maxInclusive=1000 minInclusive=0 units=Percent referenceFrame=LocalAirMass	Defines a percentage where 100% = 100.0. Values greater than 100% are allowed.
RhoAnglePropulsor	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=PropulsorXYZ	Specifies the angle of the propulsor about the Z-Axis of the vehicle reference frame.
RollAngle	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 units=Radian referenceFrame=PlatformNED	Specifies a platform's rotation about the longitudinal axis (e.g. the axis through the body of the vehicle from tail to nose) in a locally level, North-East-Down coordinate system centered on the vehicle. Roll is zero when the platform is "wing-tip to wing-tip" level in the North-East plane.
RollRate	double	maxInclusive=32.767 minInclusive=-32.767 units=RadianPerSecond referenceFrame=Counting	Specifies the rate of change of the platform's roll angle.
Salinity	double	units=GramsPerKilogram referenceFrame=LocalWaterMass	Represents the concentration of dissolved salts in water etc. measured in grams per kilogram.
SegmentID	long	maxInclusive=100000 minInclusive=0 referenceFrame=Counting	Describes the index number of a map segment.
SidesCount	long	maxInclusive=255 minInclusive=3	Represents the number of sides a polygon has using a positive integer.
SizeBytes	long	maxInclusive=1000000000 minInclusive=0 units=Byte referenceFrame=Counting	Represents an amount of data and is stored in bytes.

Type Name	Primitive Type	Range of Values	Description
SizeLargeBytes	uint64	maxInclusive=18446744073709551616 minInclusive=0 units=Byte referenceFrame=Counting	Represents an amount of data and is stored in bytes.
SizeReal	double	units=None referenceFrame=Counting	Realizes SizeType: an entity that describes the magnitude or number of a measurable or countable entity.
Speed	double	maxInclusive=299792458 minInclusive=0 units=MeterPerSecond referenceFrame=Counting	This type stores speed in meters/s.
SpeedASF	double	maxInclusive=299792458 minInclusive=-299792458 units=MeterPerSecond referenceFrame=ASF	This type stores speed in meters/s in an above sea floor reference frame.
SpeedBSL	double	maxInclusive=299792458 minInclusive=-299792458 units=MeterPerSecond referenceFrame=BSL	This type stores speed in meters/s in a below sea level reference frame.
SpeedLocalWaterMass	double	maxInclusive=299792458 minInclusive=0 units=MeterPerSecond referenceFrame=LocalWaterMass	This type stores speed in meters/s.
StringLongDescription	string	length=4095	Represents a long format description.
StringName	string	length=64	Describes a 64 char string.
StringShortDescription	string	length=1023	Represents a short format description.
StringValue	string	length=256	Describes a 256 char string.
Temperature	double	maxInclusive=1000 minInclusive=-273 units=Celsius referenceFrame=Counting	Represents the degree or intensity of warmness or coldess presence in a substance. Measured in Celsius.
TransmitAttenuation	long	maxInclusive=18 minInclusive=0 units=Decibel referenceFrame=Counting	Describes the transmit attenuation.
Turbidity	double	minInclusive=0.0 units=NephelometricTurbidityUnits(NTU) referenceFrame=Environment	The measured cloudiness of the water.
TurnRate	double	maxInclusive=32.767 minInclusive=-32.767 units=RadianPerSecond referenceFrame=Counting	Specifies the rate of change of the heading angle of a platform.

Type Name	Primitive Type	Range of Values	Description
UniformResourceIdentifier	string	length=2047	Represents a Uniform Resource Identifier (URI).
VolumeCubicMeter	double	maxInclusive=1000 minInclusive=0 units=VolumeCubicMeter referenceFrame=Counting	Represents the quantity of three-dimensional space enclosed by some closed boundary
VolumePercent	double	maxInclusive=1000 minInclusive=0 units=Percent referenceFrame=Counting	Defines a percentage where 100% = 100.0. Values greater than 100% are allowed.
VolumetricFlowRate	double	maxInclusive=100000000 minInclusive=-100000000 units=CubicMeterPerSecond referenceFrame=Counting	Specifies the amount of fluid moving through a pipe or channel per unit time.
WaterTemperature	double	maxInclusive=100 minInclusive=-22 units=Celsius	Specifies the water temperature.
WeatherBarometricPressure	double	maxInclusive=1200 minInclusive=600 units=Millibar	Specifies the barometric pressure.
XPosition	double	units=Meter	Represents the x axis position.
YawAngle	double	maxInclusive=6.28318530718 minInclusive=-6.28318530718 referenceFrame=PlatformNED units=Radian	The yaw angle relative to the NED coordinate system centered at the platform location.
YawRate	double	maxInclusive=32.767 minInclusive=-32.767 units=RadianPerSecond referenceFrame=Counting	Specifies the rate of change of the platform's yaw angle.
YPosition	double	units=Meter	Represents the y axis position.
ZPosition	double	maxInclusive=100000 minInclusive=-100000 units=Meter	Represents the z axis position.

A Appendices

A.1 Glossary

Note: This glossary aims to define terms that are uncommon, or have a special meaning in the context of UMAA and/or the DoD. This glossary covers the complete UMAA specification. Not every word defined here appears in every ICD.

Almanac Data (GPS)	A navigation message that contains information about the time and status of the entire satellite constellation.
Coulomb	The SI unit of electric charge, equal to the quantity of electricity conveyed in one second by a current of one ampere.
Ephemeris Data (GPS)	A navigation message used to calculate the position of each satellite in orbit.
Glowplug or Glow Plug	A heating device used to aid in starting diesel engines.
Interoperability	1) The ability to act together coherently, effectively, and efficiently to achieve tactical, operational, and strategic objectives. 2) The condition achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users.
Mean Sea Level	The average height of the surface of the sea for all stages of the tide; used as a reference for elevations.
Middleware	A type of computer software that provides services to software applications beyond those available from the operating system. Middleware makes it easier for software developers to implement communication and input/output, so they can focus on the specific purpose of their application.
SoaML	The Service oriented architecture Modeling Language (SoaML) specification that provides a metamodel and a UML profile for the specification and design of services within a service-oriented architecture. The specification is managed by the Object Management Group (OMG).

A.2 Acronyms

Note: This acronym list is included in every ICD and covers the complete UMAA specification. Not every acronym appears in every ICD.

ADD	Architecture Design Description
AGL	Above Sea Level
ASF	Above Sea Floor
BSL	Below Sea Level
BWL	Beam at Waterline
C2	Command and Control
CMD	Command
CO	Comms Operations
CPA	Closest Point of Approach
CTD	Conductivity, Temperature and Depth
DDS	Data Distribution Service
DTED	Digital Terrain Elevation Data
EGM	Earth Gravity Model
EO	Engineering Operations
FB	Feedback
GUID	Globally Unique Identifier
HM&E	Hull, Mechanical, & Electrical

ICD	Interface Control Document
ID	Identifier
IDL	Interface Definition Language Specification
IMO	International Maritime Organization
INU	Inertial Navigation Unit
LDM	Logical Data Model
LOA	Length Over All
LRC	Long Range Cruise
LWL	Length at Waterline
MDE	Maritime Domain Extensions
MEC	Maximum Endurance Cruise
MM	Mission Management
MMSI	Maritime Mobile Service Identity
MO	Maneuver Operations
MRC	Maximum Range Cruise
MSL	Mean Sea Level
OMG	Object Management Group
PIM	Platform Independent Model
PMC	Primary Mission Control
PNT	Precision Navigation and Timing
PO	Processing Operations
PSM	Platform Specific Model
RMS	Root-Mean-Square
ROC	Risk of Collision
RPM	Revolutions per minute
RTPS	Real Time Publish Subscribe
RTSP	Real Time Streaming Protocol
SA	Situational Awareness
SEM	Sensor and Effector Management
SO	Support Operations
SoaML	Service-oriented architecture Modeling Language
STP	Standard Temperature and Pressure
UCS	Unmanned Systems Control Segment
UMAA	Unmanned Maritime Autonomy Architecture
UML	Unified Modeling Language
UMS	Unmanned Maritime System
UMV	Unmanned Maritime Vehicle
UxS	Unmanned System
WGS84	Global Coordinate System
WMM	World Magnetic Model
WMO	World Meteorological Organization