Unmanned Maritime Autonomy Architecture (UMAA)
Situational Awareness (SA)
Interface Control Document (ICD)
(UMAA-SPEC-SAICD)

MDE Version 4.2.1 Commit 7c2b513

UMAA Spec Commit f2fa426

Version 3.0.1
25 February 2021

DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

UMAA Situational Awareness ICD

(UMAA-SPEC-SA-ICD)

Signature Page

Submitted by: Rothgeb.Mark.E.ORC3011008580.ID — Digitally signed by Rothgeb.Mark.E.ORC3011008580.ID Date: 2021.05.05 11:55:18 -04'00' **Date Signed:** _____

Mark Rothgeb
Unmanned Maritime Autonomy Architecture
Standards Board Chair

PMS 406 Approvals:

ANDERSON.JEREMIAH.P.1254266481 — Digitally signed by ANDERSON.JEREMIAH.P.1254266481 Date: 2021.05.10 14:52:11 -04'00' **Date Signed:** _____

CDR Jeremiah Anderson
PMS 406 Advanced Autonomous Capabilities,
Principal Assistant Program Manager

SMALL.PETER.D.1048778646 — Digitally signed by SMALL.PETER.D.1048778646 Date: 2021.06.03 17:18:02 -04'00' **Date Signed:** _____

CAPT Pete Small
Program Manager
PMS 406 Unmanned Maritime Systems,
PEO Unmanned and Small Combatants

Contents

List of Figures

List of Tables

1 Scope

1.1 Identification

This document defines a set of services and interfaces as part of the Unmanned Maritime Autonomy Architecture (UMAA). The services and corresponding interfaces covered in this ICD encompass the functionality to provide situational awareness for an Unmanned Maritime Vehicle (UMV) (surface or undersea). As such, it provides services that focuses on providing decision-level information, which includes self (ownship) state information (such as navigation data and self-health); environmental information (both a-priori and sensed) such as map/chart data, bathymetry, currents, and weather; and world model data, including contact information and classification information (from single or multiple data-fused sources). This document is generated automatically from data models that define its services and interfaces as part of the Unmanned Systems (UxS) Control Segment (UCS) Architecture as extended by UMAA to provide autonomy services for UMVs.

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.

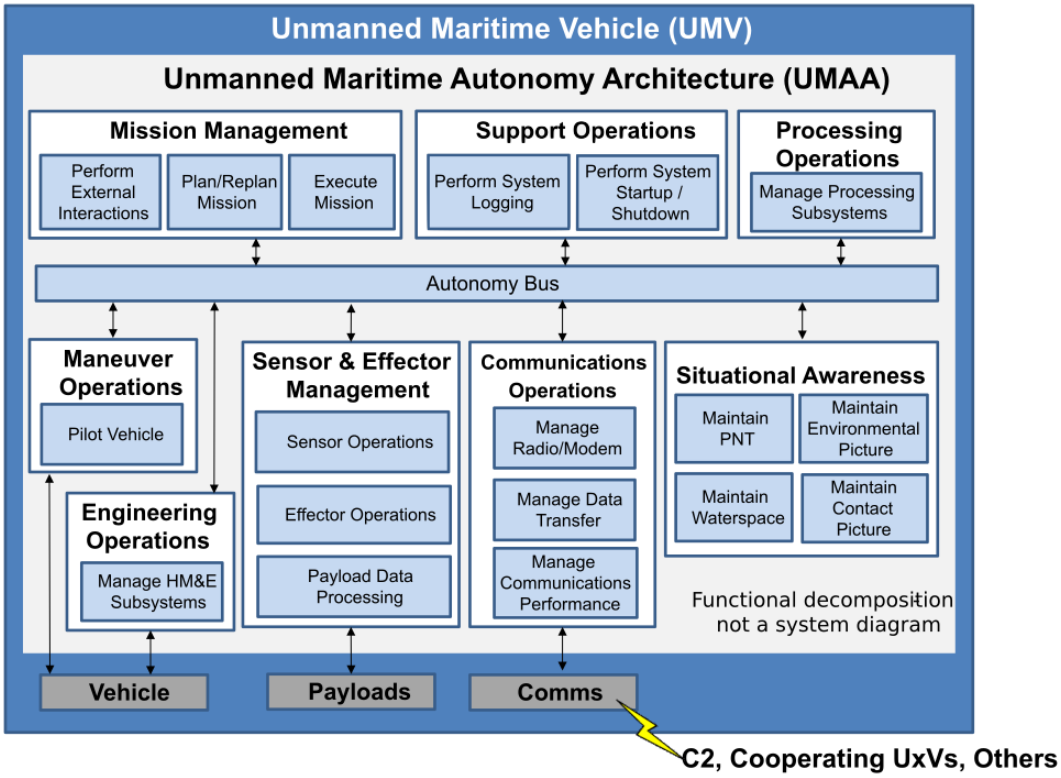

Figure 1: UMAA Functional Organization

1.2 Overview

The fundamental purpose of UMAA is to promote the development of common, modular, and scalable software for UMV's that is independent of a particular autonomy implementation. Unmanned Maritime Systems (UMSs) consist of Command and Control (C2), one or more UMVs, 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 UMV. This includes the autonomy for all classes of UMVs and must support varying levels of communication in mission (i.e., constant, intermittent, or none) with its C2 System. 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 platform interface.

Situational Awareness includes the services and interfaces required to enable autonomous decision making by providing information gleaned from the sensing system (e.g. radar or sonar), from a-priori data (e.g. bathymetry), and from any externally provided data (contact data from a collaborating UMV). Situational Awareness (often also referred to as world model) also includes such things as data fusion from multiple data sources (e.g. visual and radar), inferencing of higher level knowledge from raw data (e.g. contact reported by a radar is a merchant), registration of different data sources to common spatial references (e.g. contact is 570 yards away on bearing 27 degrees relative to ownship and is at 34.5 degrees latitude and 76.3048 degrees longitude). Standardization of these services enables them to be used across platforms and across programs based on available sensing capabilities on each individual platform.

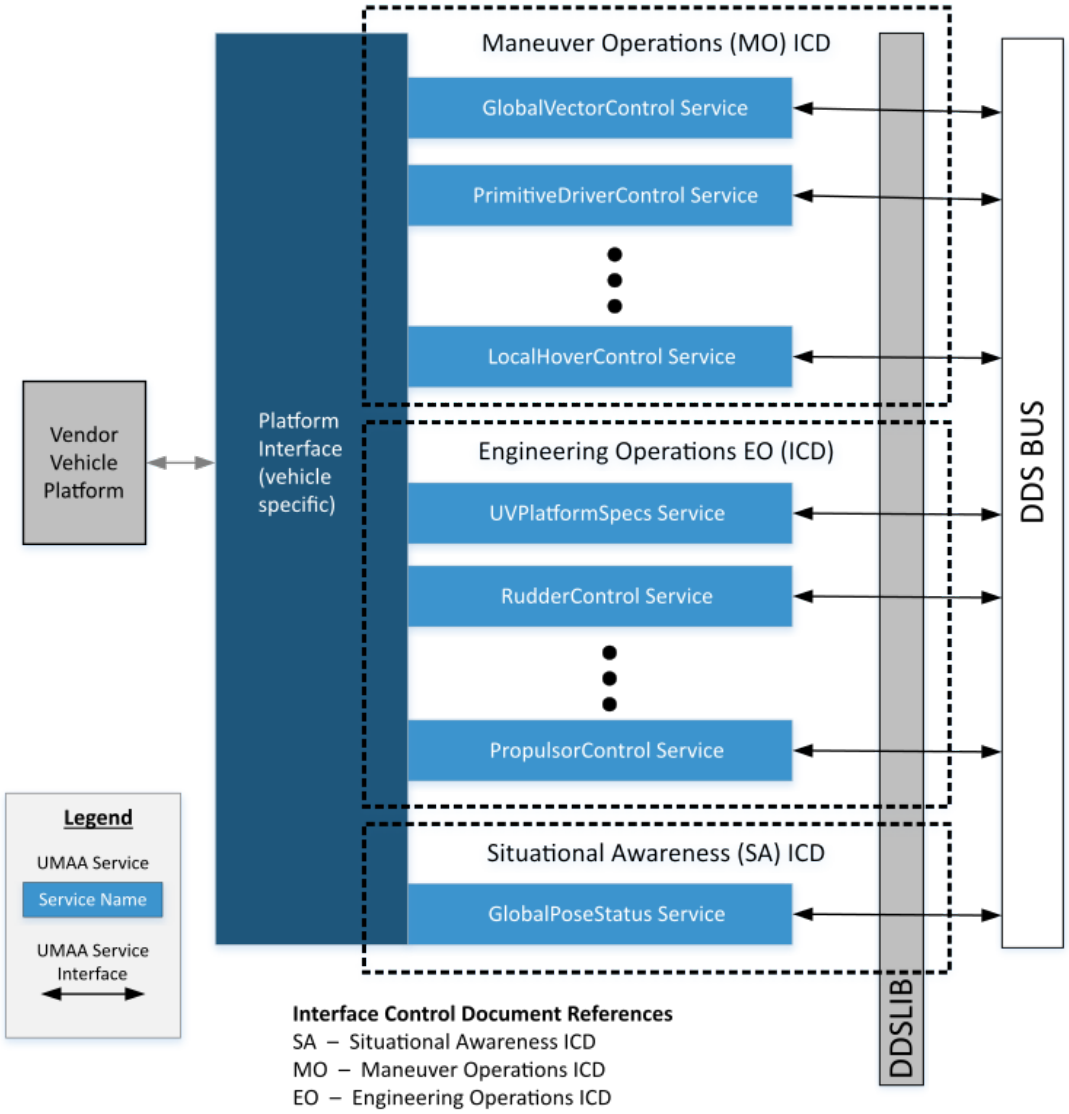

Figure 2: UMAA Services and Interfaces Example

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 – Situational Awareness (SA) 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 3: 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

Table 4: 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, 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 to assure they are consistent with the data model and to ensure delivered software matches its interface specification.

The data model is maintained as a maritime extension to the UCS Architecture and will be maintained under configuration control by the UMAA Board. 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 compliance with a UMAA ICD is to provide service and interface specification compliance. Components may group services and their 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 represent 1 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. Compliance is satisfied at the individual service level. Given this context, section 6 correspondingly defines services with their interfaces and not components.

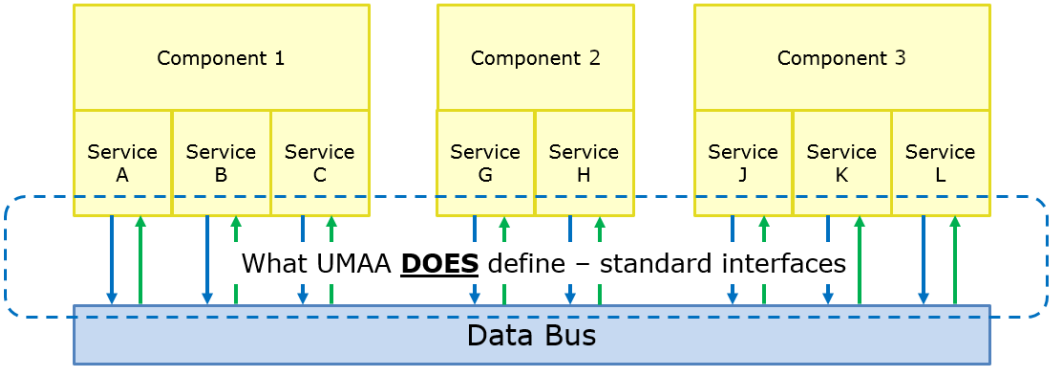

Figure 3: Services and Interfaces Exposed on the UMAA Data Bus

Services may use other services within this ICD or in other UMAA defined ICDs in order to provide their capability. Additionally, components for acquisition and development may span ICDs. An example of this would be a vehicle control system on a UMV. The control of the vehicle would be found in the Maneuver Operations ICD. However, an Inertial Navigation Unit (INU) that gives dynamic vehicle status is found in the Situational Awareness ICD. These are often organic to a vehicle and in that case are provided together with the vehicle as a component.

3.3 Data Distribution Service (DDS™)

The data bus supporting autonomy messaging as depicted 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 components of a system 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 a 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. (UCSArchitecture,ArchitectureTechnicalGovernance)

In order to promote commonality across service definitions, a common way of naming services and their set 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 will not be broken up as indicated below 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 in implementation. The conventions used to define UCS-MDE services are as follows:

Service Name
- <SBN>Config
- <SBN>Control
- <SBN>Specs
- <SBN>Status

where the SBN should be descriptive of the task or information provided by the service.

Table 5: Service Requests and Associated Responses

	Service Requests (Inputs)	**Service Responses (Outputs)**
Config	query<SBN><FN>Config	report<SBN><FN>Config
Control	set<SBN><FN> query<SBN><FN>CommandAck cancel<SBN><FN>Command query<SBN><FN>ExecutionStatus	report<SBN><FN>CommandStatus report<SBN><FN>CommandAck report<SBN><FN>CancelCommandStatus report<SBN><FN>ExecutionStatus
Specs	query<SBN><FN>Specs	report<SBN><FN>Specs
Status	query<SBN><FN>	report<SBN><FN>

Service Requests (operation:message)
- 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
- query<SBN><FN>ExecutionStatus:<SBN><FN>ExecutionStatusRequestType[1]
- query<SBN><FN>Specs:<SBN><FN>SpecsRequestType[1]
- query<SBN><FN>:<SBN><FN>RequestType [1] [2]

[1]These message types are required for compatibility with the UCS model but are not used by the UMAA specification.
[2]At this time there are no Requests in the specification but when they have been added, this will be the message format.

Service Responses (operation:message)
report<SBN><FN>Config:<SBN><FN>ConfigReportType
report<SBN><FN>CommandStatus:<SBN><FN>CommandStatusType
report<SBN><FN>CommandAck:<SBN><FN>CommandAckReportType
report<SBN><FN>CancelCommandStatus:<SBN><FN>CancelCommandStatusType
report<SBN><FN>ExecutionStatus:<SBN><FN>ExecutionStatusReportType
report<SBN><FN>Specs:<SBN><FN>SpecsReportType
report<SBN><FN>:<SBN><FN>ReportType

where,

- Config (Configuration) Report – 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, maximum allowable radio transmit power.
- Command Status – the current state of a particular command (either control or configuration)
- Command – the ability to influence or direct the behavior of a resource during operation of a particular task. Attributes are variable. Examples include a vehicleâ€™s speed, engine RPM, antenna raising/lowering, controlling a light or gong.
- Command Ack (Acknowledgement) Report – the command currently being executed.
- Cancel – the ability to cancel a particular command that has been issued.
- Execution Status Report – the status related to executing a particular command. Examples associated with a waypoint command include cross track error, time to achieve, distance remaining.
- Specs (Specifications) Report – 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 UMV, cycle time of a radar.
- Report – the current information provided by a resource. Examples include a vehicle speed, rudder angle, current waypoint, 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 service under Maneuver Operations:
UMAA::MO::PrimitiveDriver
Access the Feature Service under Situational Awareness:
UMAA::SA::Feature

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 Report Message Fields:
UMAA::UMAARpt
Access the common UMAA Position2D (i.e., latitude and longitude) structure:
UMAA::Measurement::Position2D

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 Sets

Some reports under the UMAA standard utilize Large Sets, which are unordered sets of related data. The purpose of a Large Set is to provide the ability to update one or more elements of the set without having to republish the entire set on the DDS bus and consuming more resources as a set is appended or updated. In a given DDS topic, each element of the set is tracked to its identifier through the use of the <service>SetID identifier (a key). Additionally, users will be able to trace an element in a set by its source attribute (a NumericGUID) to the Service Provider that is generating the report with this set.

When elements of the set are updated, the timestamp of the metadata must be updated as well to signal a change in the set. The element timestamp for the update must be later than the current metadata timestamp. Once the element is updated, the timestamp of the metadata must be updated to a time equal to or later than the timestamp of the individual element update. The set can be updated as a batch (multiple elements in a single "update cycle," as determined by the provider) provided the metadata timestamp is updated to a time that is no earlier than the the most recent timestamp of all element updates in the batch. This allows for a coarse synchronization: data elements with timestamps later than the current metadata timestamp 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 metadata timestamp is advanced beyond the element's timestamp to signal the complete update cycle has finished and consider the set as a whole.

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 orientation of the vehicle 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.

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 Platform Orientation

Determining the orientation of the vehicle (Figure 4) with respect to any reference frame is carried out via the following procedure (Figure 5).

1. Align the vehicle's Longitudinal or X axis with the reference frame X axis. In the global reference, this is the North direction.
2. Align the vehicle's down or 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 6).
5. Rotate the vehicle about the vehicle's newly oriented Y axis by the pitch angle (Figure 7).
6. Rotate the vehicle about the vehicle's newly oriented X axis by the roll angle (Figure 8).

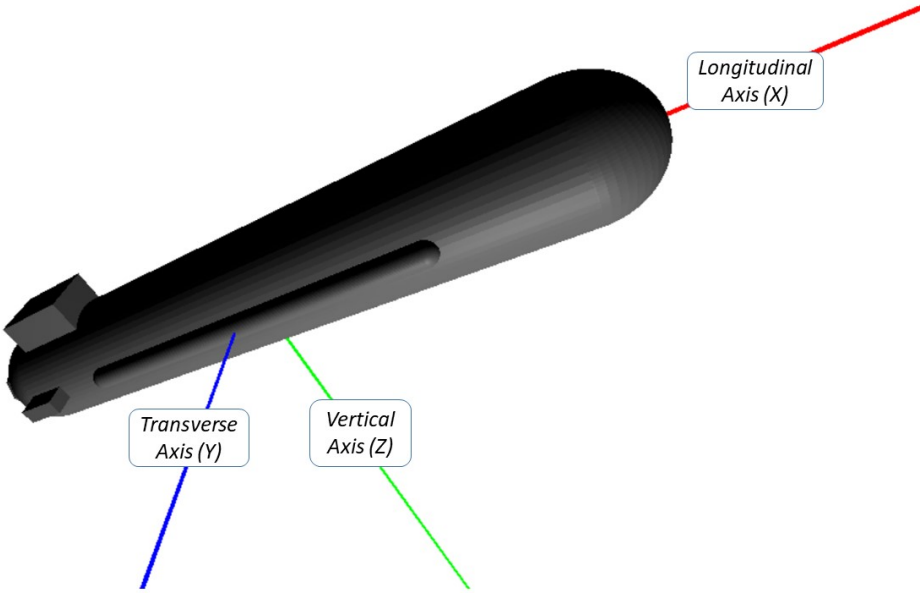

Figure 4: Given a vehicle in arbitrary orientation

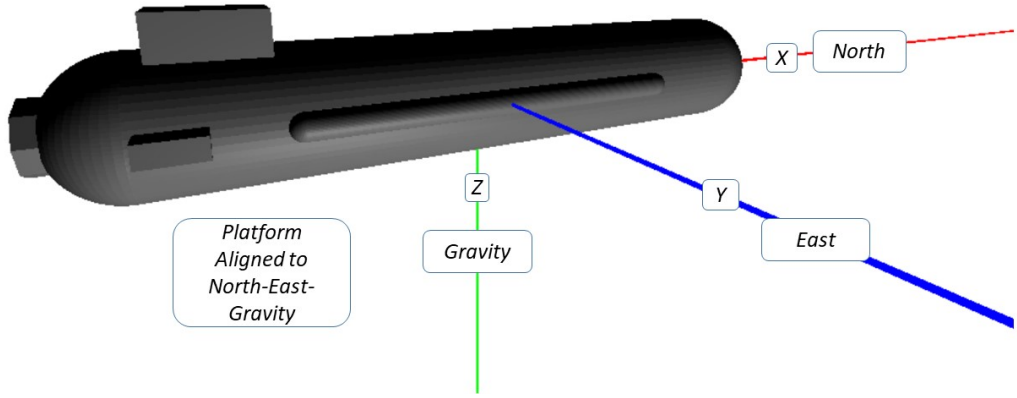

Figure 5: Align the vehicle with the reference frame axes

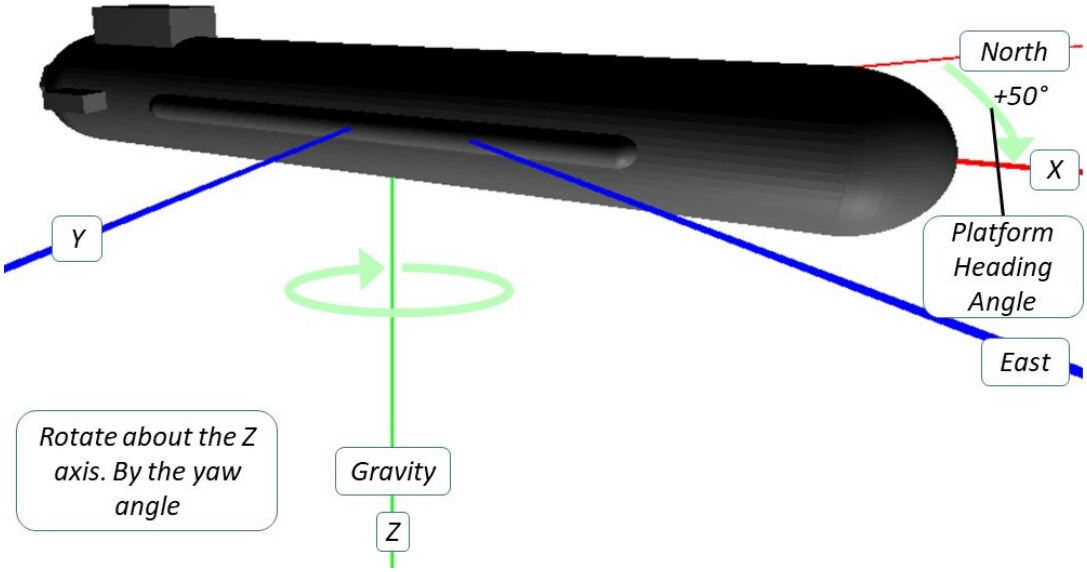

Figure 6: Rotate the vehicle by the Yaw angle

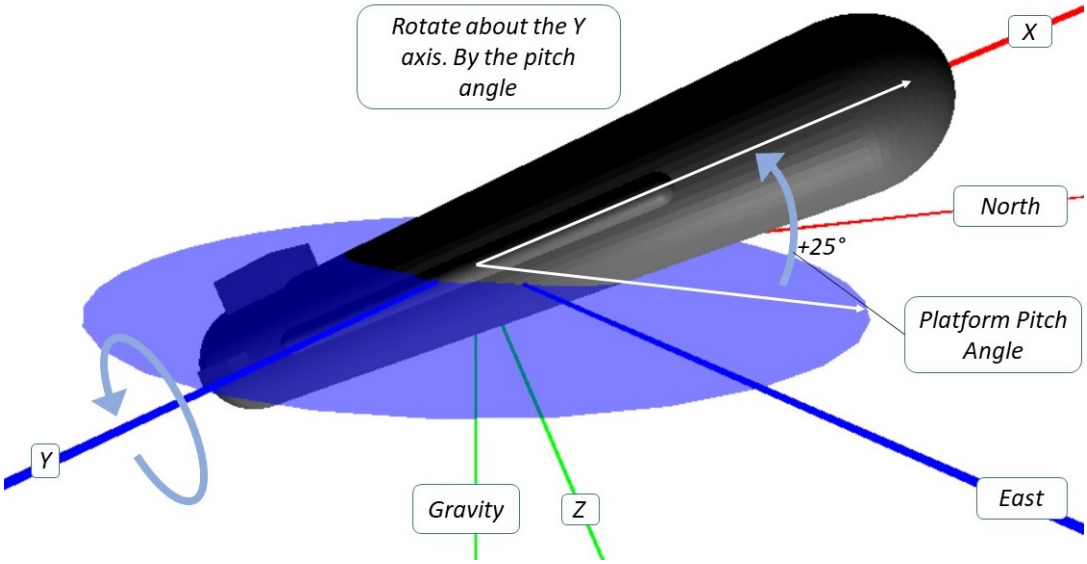

Figure 7: Rotate the vehicle by the Pitch angle

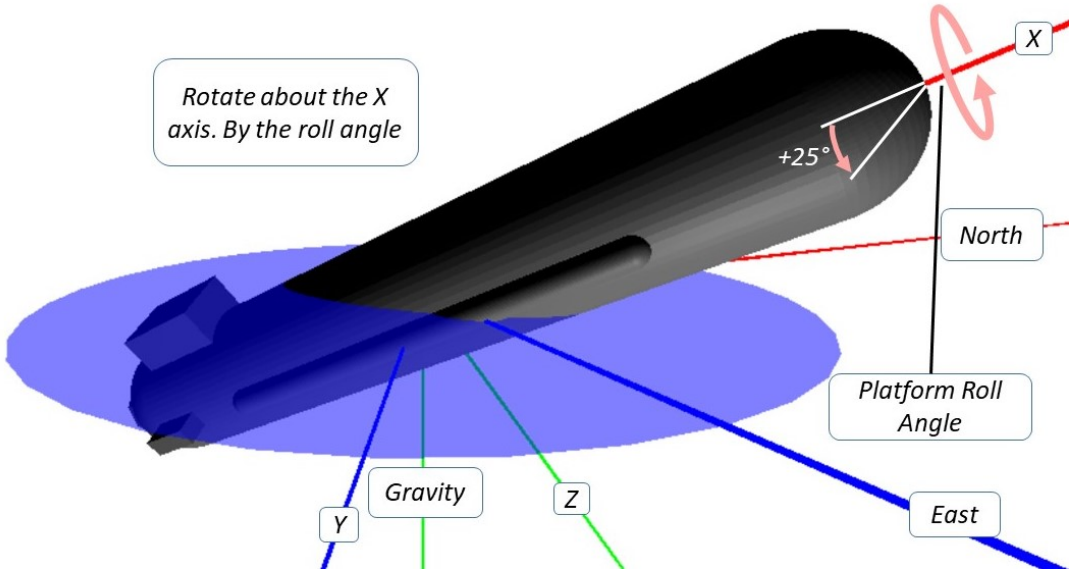

Figure 8: Rotate the vehicle by the Roll angle

4.3 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.

Definitions

- Design Waterline (DWL) - The line representing the waterline on the vehicle at designed load in summer temperature seawater.
- Centerline - The vertical plane passing fore and aft down the center of the ship.
- Aft Perpendicular (AP) - The vertical line passing through the rudder stock.
- Forward Perpendicular (FP) - The vertical line through the intersection of the forward side of the stern with the Design Waterline.
- Amidships - The midpoint between the Forward and Aft Perpendiculars.

Common practice puts the origin at the intersection of the Design Waterline, Centerline, and Amidships (Figure 9).

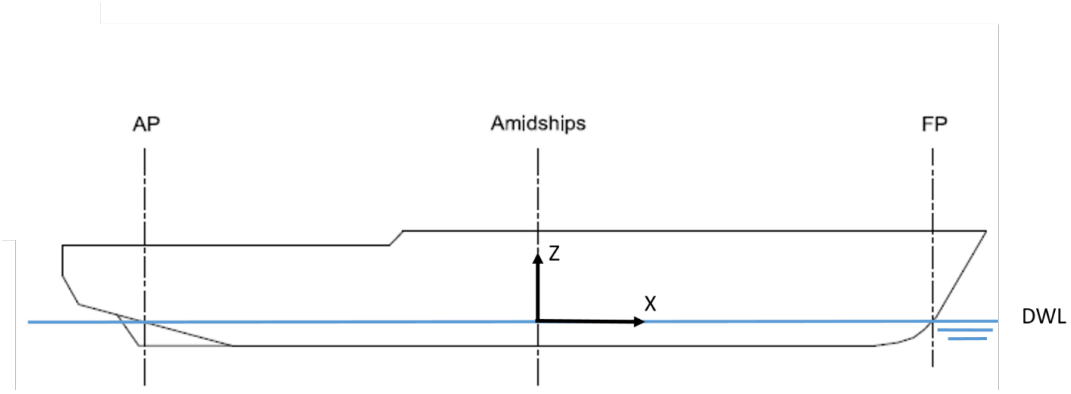

Figure 9: Origin location on a USV as example

For UUVs, common practice puts the origin as in Figure 10:

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

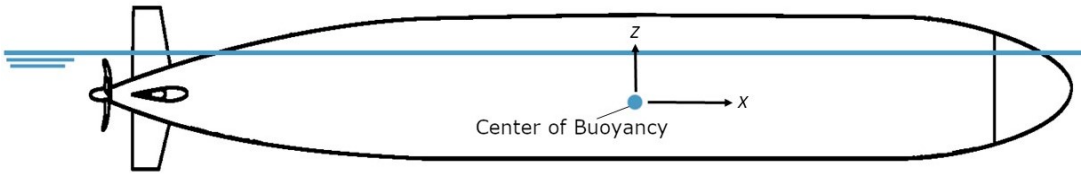

Figure 10: Origin location on a UUV as example

5 Flow Control

5.1 Command / Response

This section defines the flow of control for command/response over the DDS bus. A command/response is used to control 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 unique `sessionID` for the command when commanded. The `sessionID` is used to track the command execution as a key into other command-related messages. 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`. 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 11. 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.

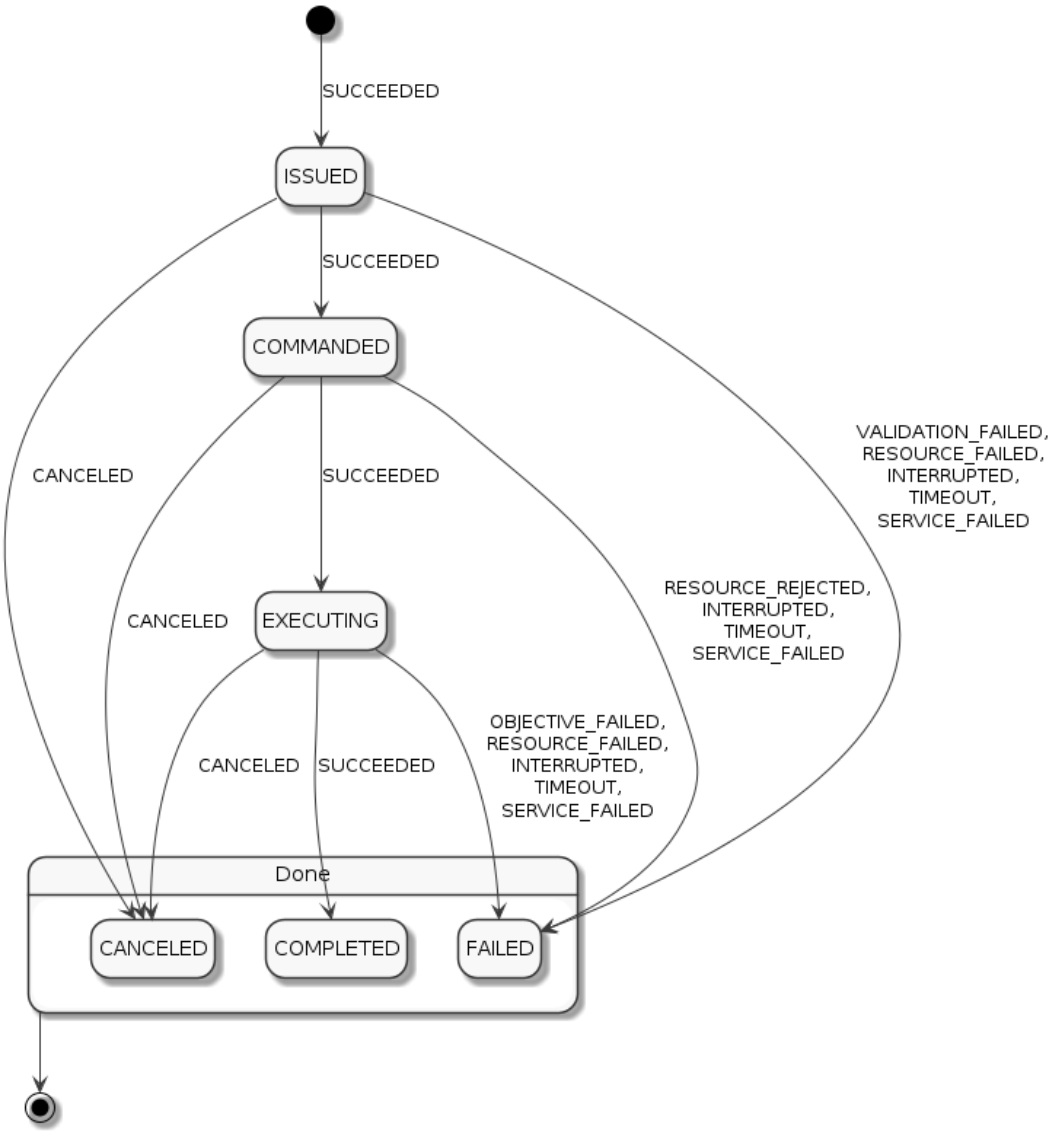

Figure 11: The state transitions of the `commandStatus` as commands are processed. Labels on the arrows represent valid `commandStatusReason` values for each transition.

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 similiar 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 12 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 my not be identical to the overall system execution duration. For example, providers may only be available to execute certain commands during specific phases of a misison or when certain hardware is in specific configurations. This Command Startup Sequence is not required to happend during a system startup phase. The only requirement is 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 anytime the `FunctionCommandType` will no longer be supported. There is no requirement the Command Shutdown Sequence only be performed during a system shutdown phase.

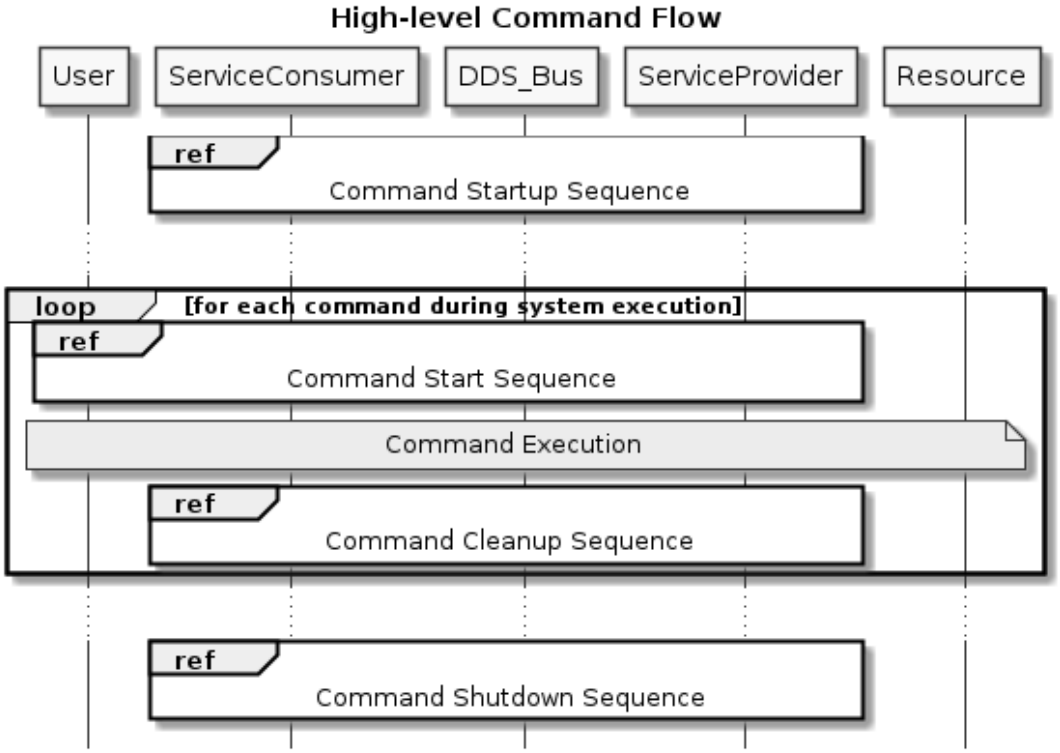

Figure 12: The sequence diagram for the high-level description of a command exeuction.

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 13.

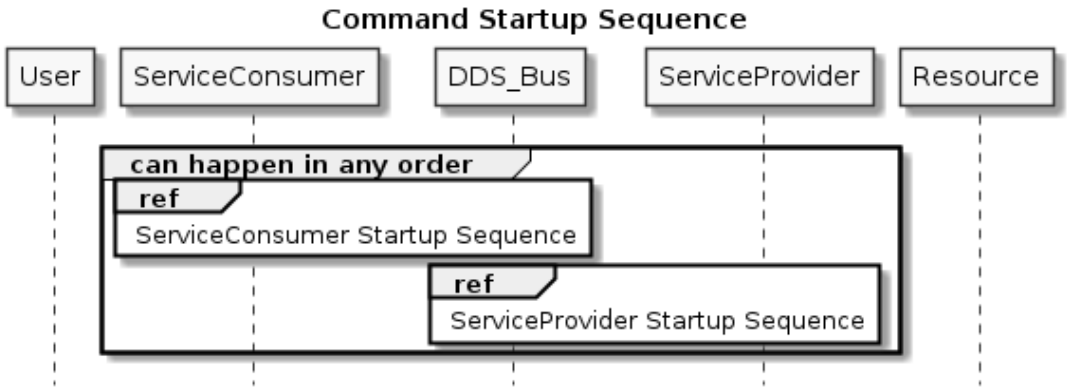

Figure 13: The sequence diagram for command startup.

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 `FunctionExecutionStatus` 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 choses 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 14.

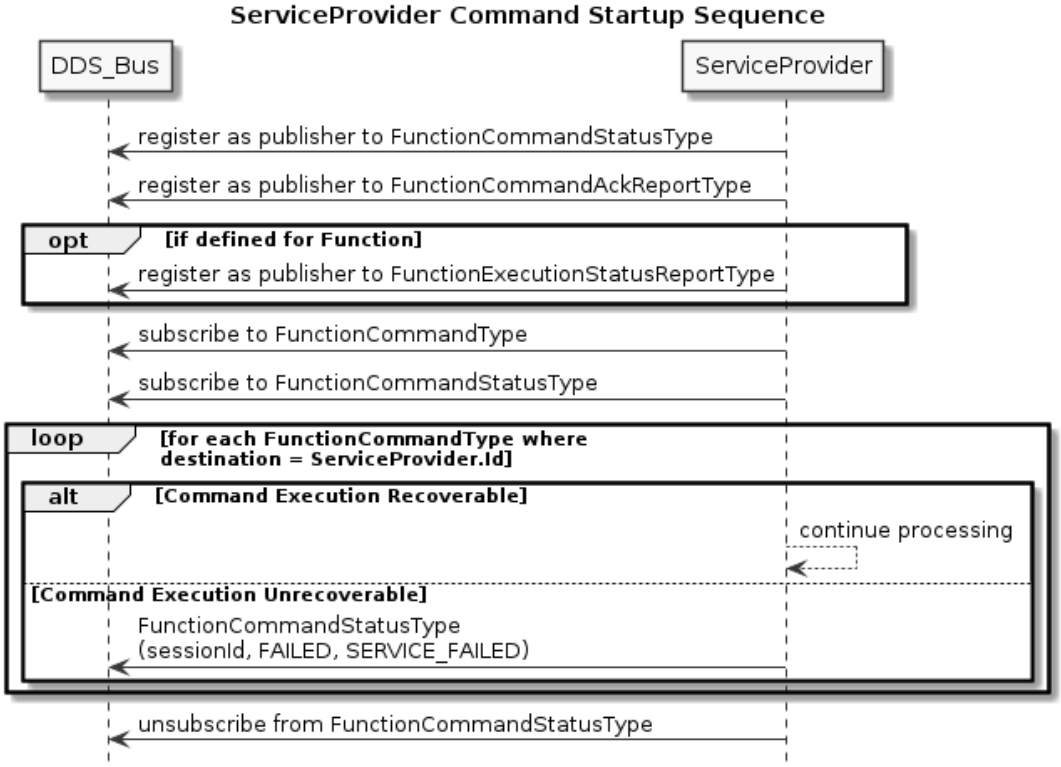

Figure 14: The sequence diagram for command startup for Service Providers.

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 choses 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 15.

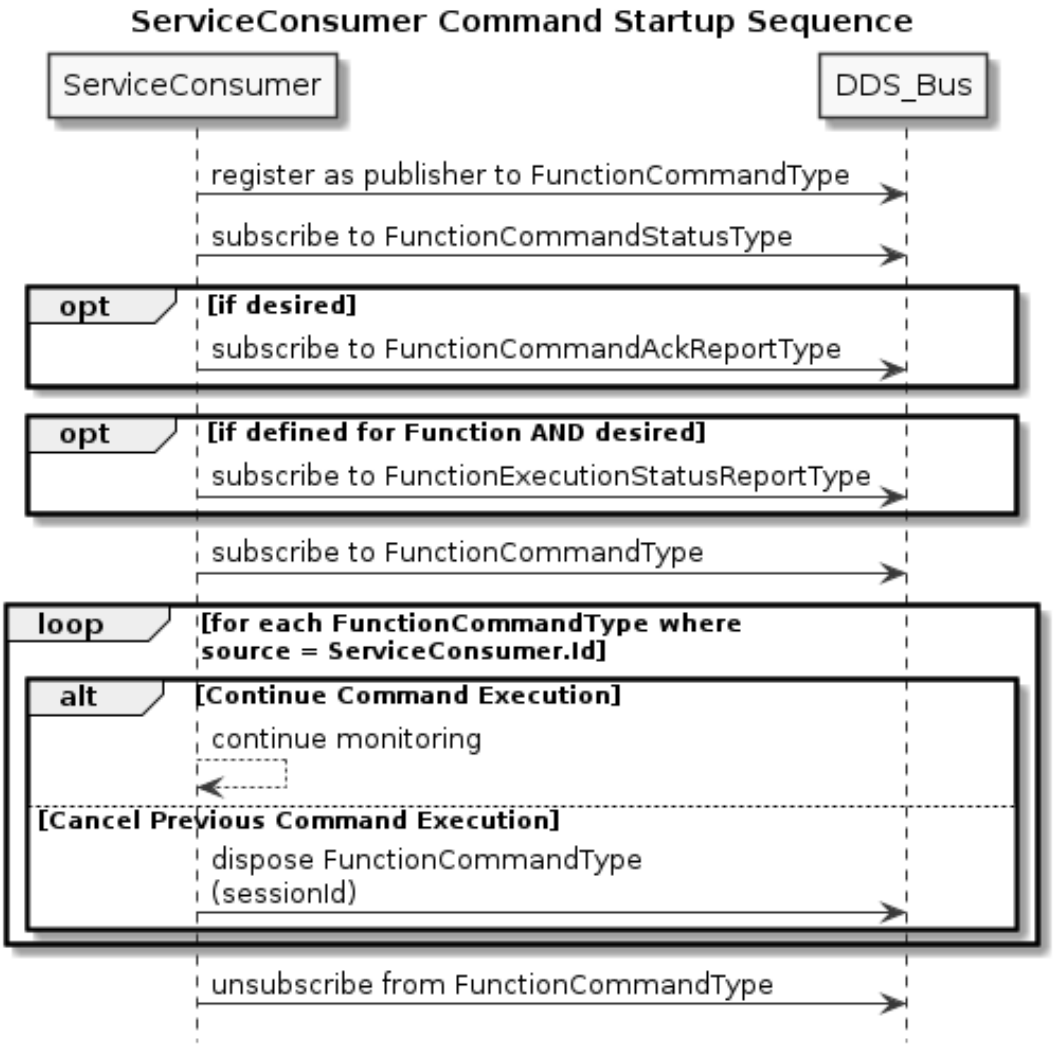

Figure 15: The sequence diagram for command startup for Service Consumers.

5.1.3 Command Execution Sequences

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

5.1.4 Command Start Sequence

The initial start sequence to execute a single 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 the same session ID as the new `FunctionCommandType` and the status of `ISSUED` and reason of `SUCCEEDED` to notify the Service Consumer it has received the new command.

The Command Start Sequence is shown in Figure 16. This pattern will be repeated each time a new command is requested. After the Command Start Sequence, the sequence can take different paths depending on the actual execution of the command. Some possible paths are detailed in the following sections, but they do not enumerate all of the possible execution paths. Other paths (e.g., an objective failing) will follow a similiar pattern to other failures; all are required to follow the state diagram shown in Figure 11 and eventually end with the Command Cleanup Sequence (as shown in Figure 22).

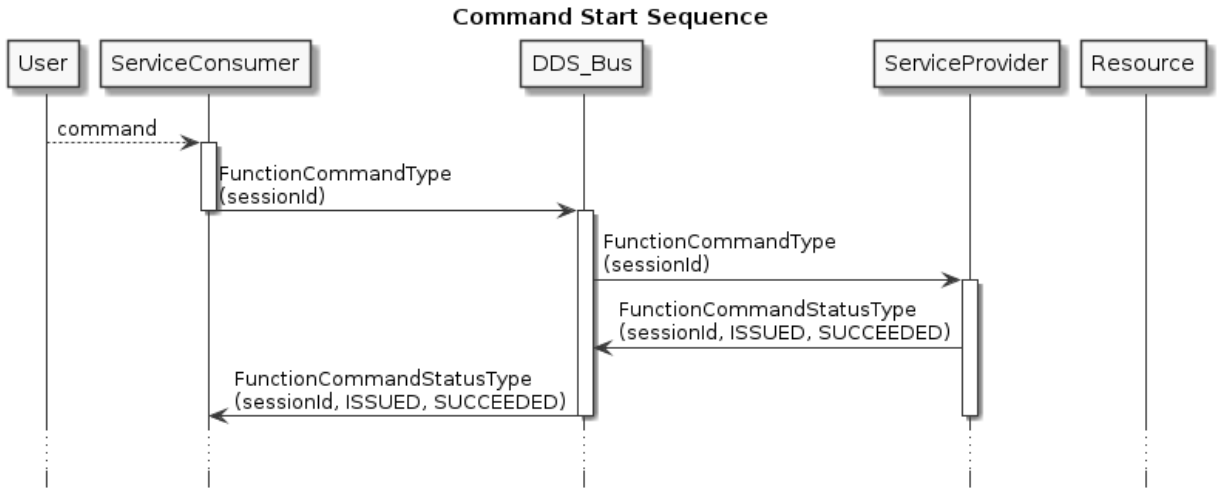

Figure 16: The sequence diagram for the start of a command execution.

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 `EXECUTED` 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 defintions of the command-specific attributes in the execution status report.

The command execution sequence is shown in Figure 17. This sequence holds until the command completes execution.

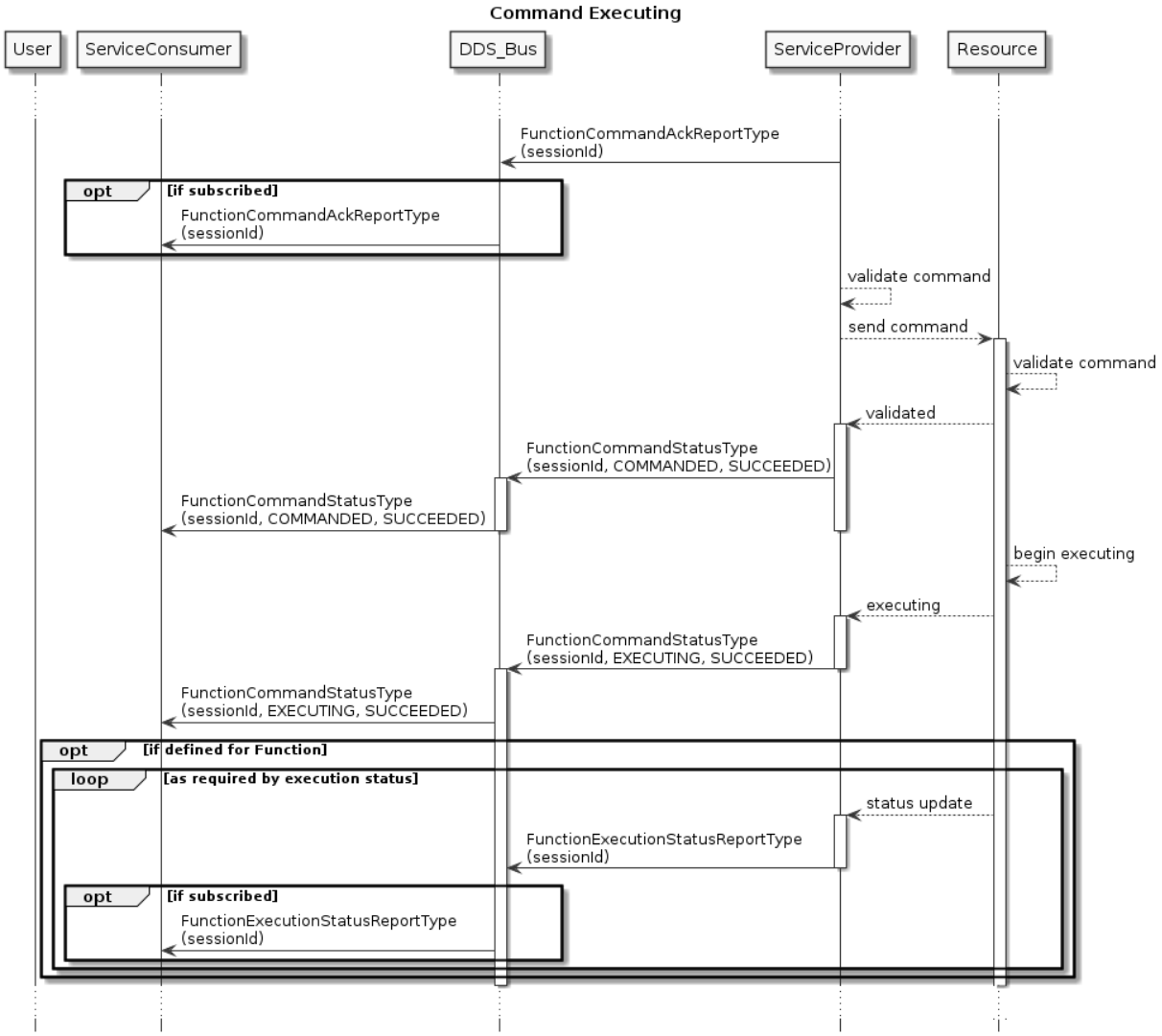

Figure 17: The beginning sequence diagram for a command execution.

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

The Command Execution Success sequence is shown in Figure 18.

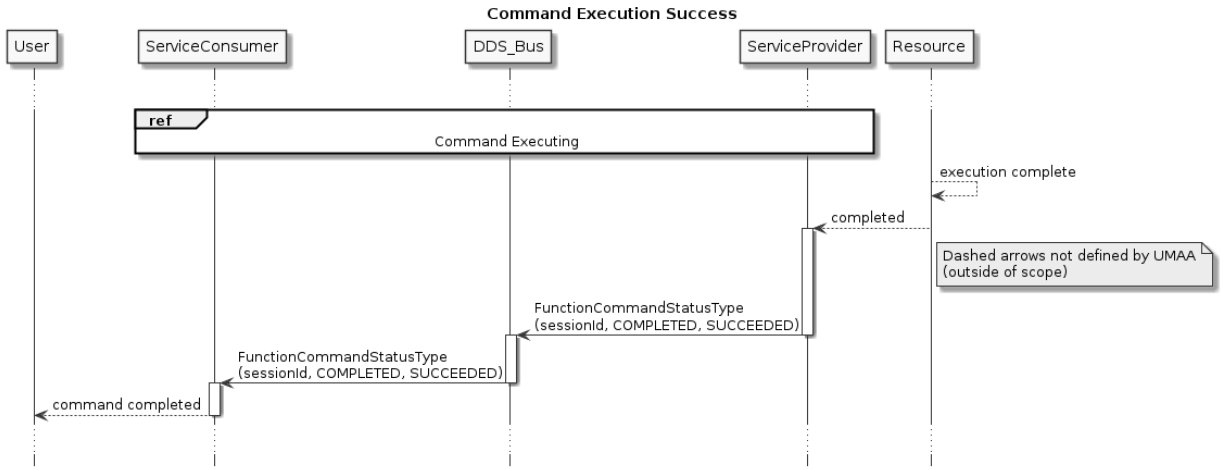

Figure 18: The sequence diagram for a command that completes successfully.

5.1.4.3 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).

The following figures provide examples of cases where a command has failed.

In the first example, the backing Resource has failed and the Service Provider in 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 19.

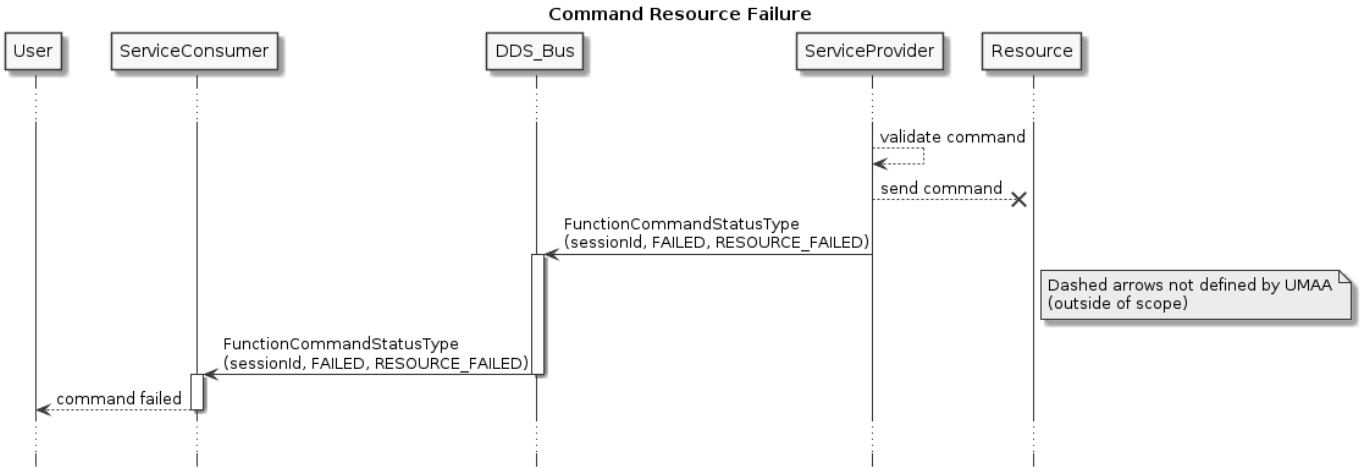

Figure 19: The sequence diagram for a command that fails due to Resource failure.

In the second example, the Resource takes too long to response, 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 20.

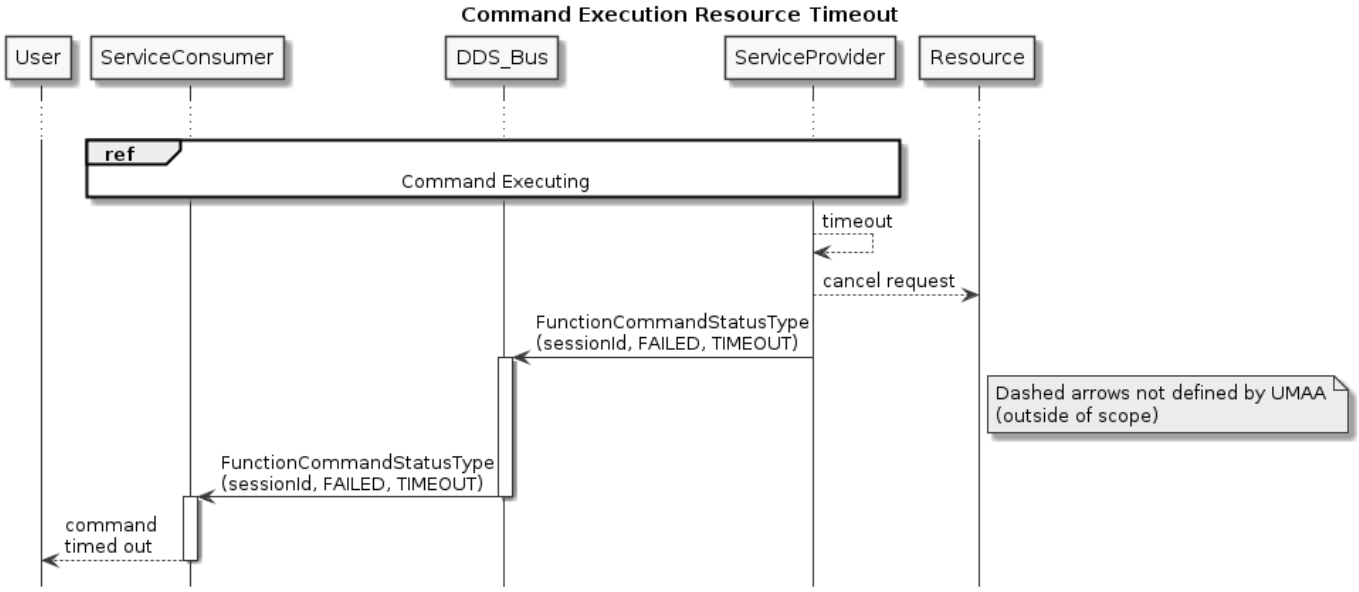

Figure 20: The sequence diagram for a command that times out before completing.

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.

5.1.4.4 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 the existing `FunctionCommandType` from the DDS bus before the execution is complete. When notified of the command disposal, 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` and then dispose the `FunctionCommandStatusType` and `FunctionCommandAckReportType` and, if defined for the Function service, the `FunctionExecutionStatusReportType` from the bus. This is shown in Figure 21. If the command cannot be canceled the Service Provider can continue to update the command status until the execution is completed, reporting `FunctionCommandStatusType` with a status of `COMPLETED` and a reason of `SUCCEEDED`, and then dispose the `FunctionCommandStatusType` and `FunctionCommandAckReportType` and, if defined for the Function service, the `FunctionExecutionStatusReportType` from the DDS bus.

There is no new unique 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.

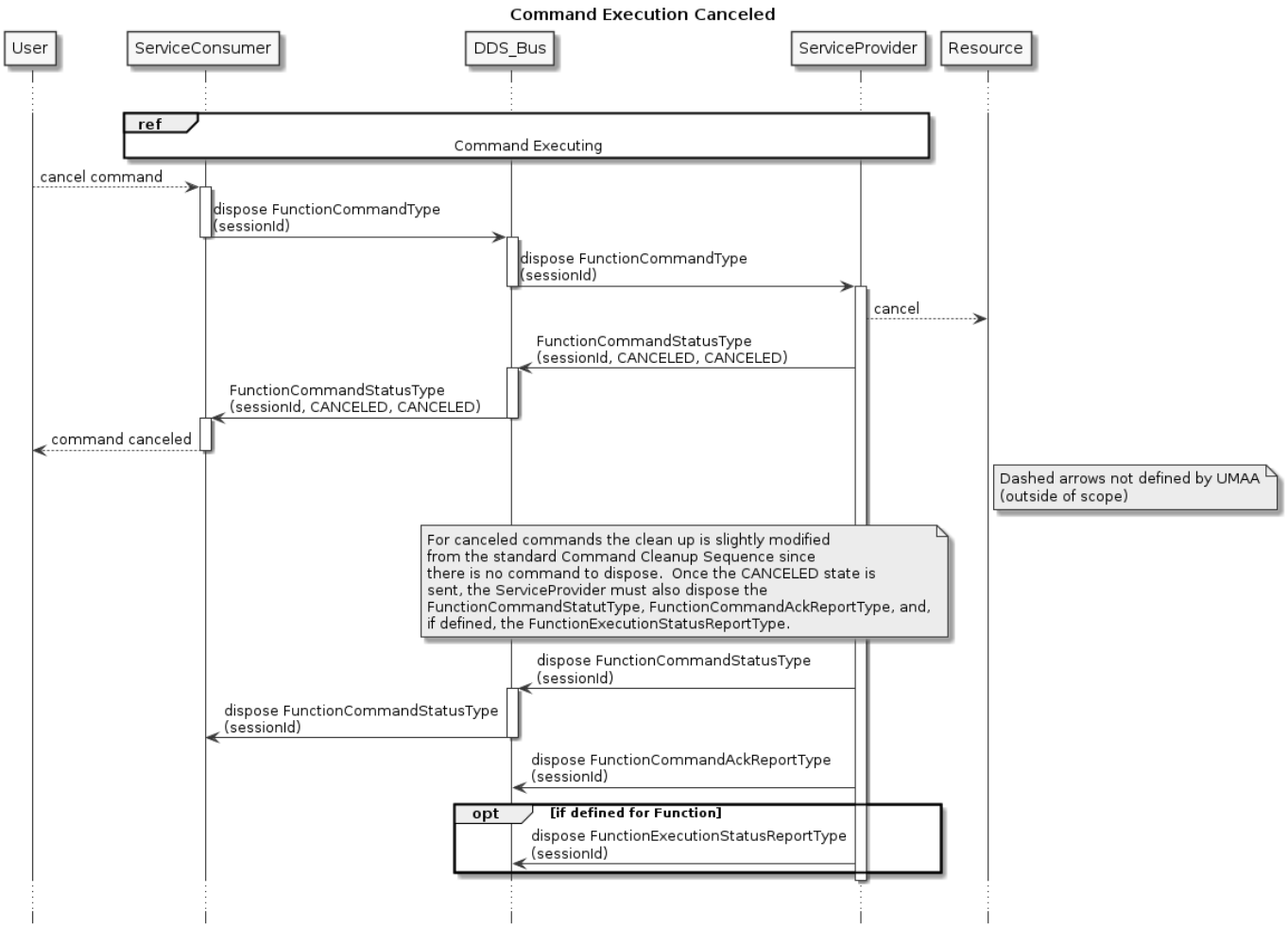

Figure 21: The sequence diagram for a command that is canceled by the Service Consumer before the Service Provider is able to complete it.

5.1.5 Command Cleanup

The Service Consumer and Service Provider are responsible for disposing corresponding data 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 22.

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

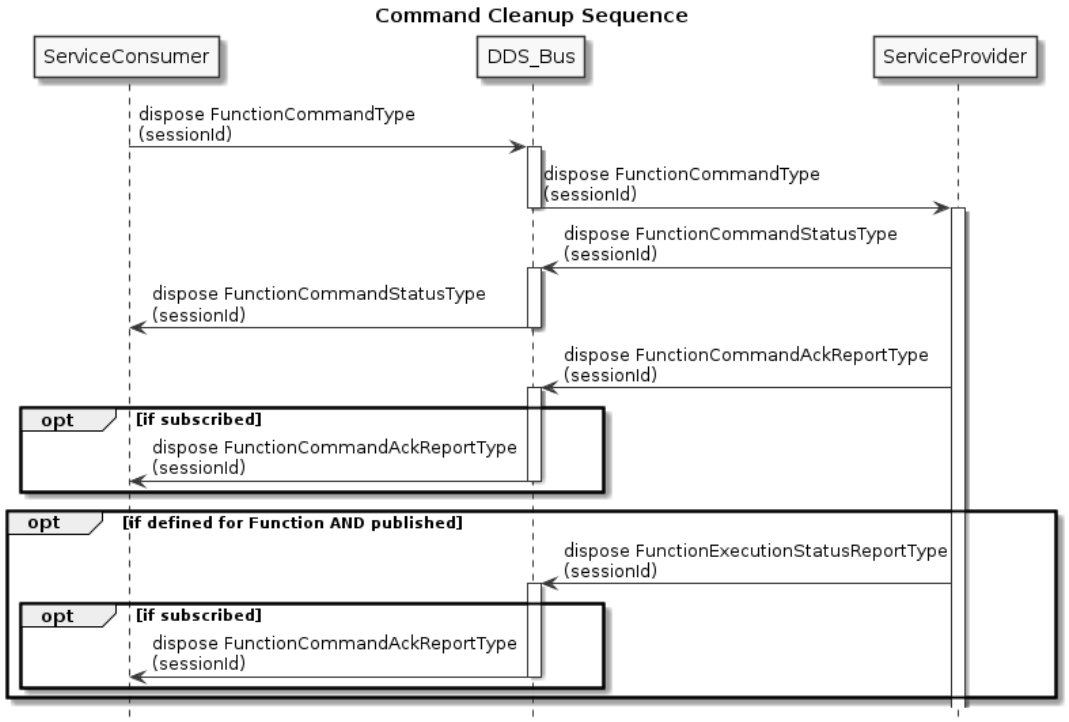

Figure 22: The sequence diagram showing cleanup of the bus when a command has been completed and the Service Consumer no longer wishes to maintain the commanded state.

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 shutdown in any order. The sequence diagram is shown in Figure 23.

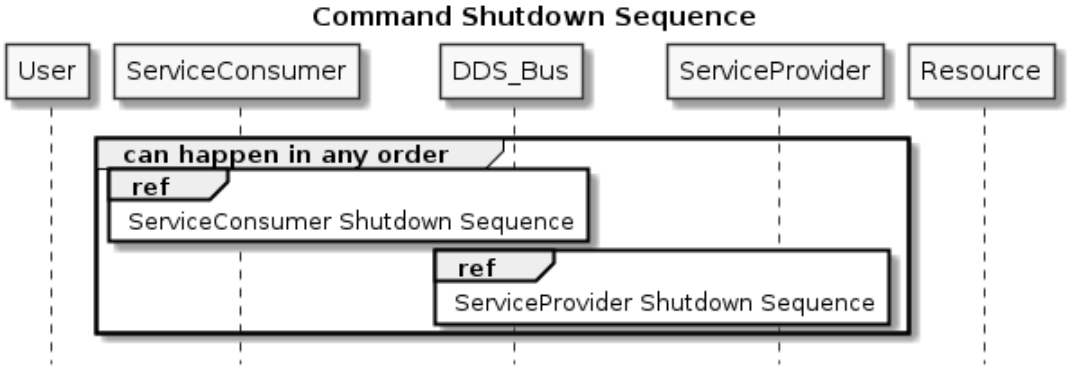

Figure 23: The sequence diagram for command shutdown.

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 24.

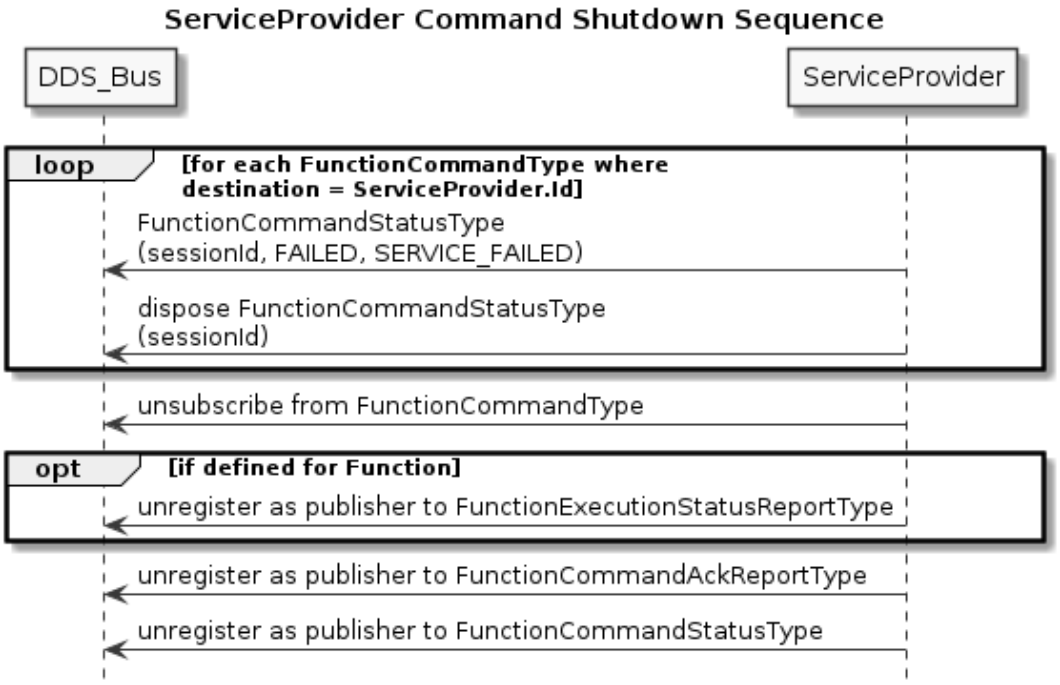

Figure 24: The sequence diagram for command shutdown for Service Providers.

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 25.

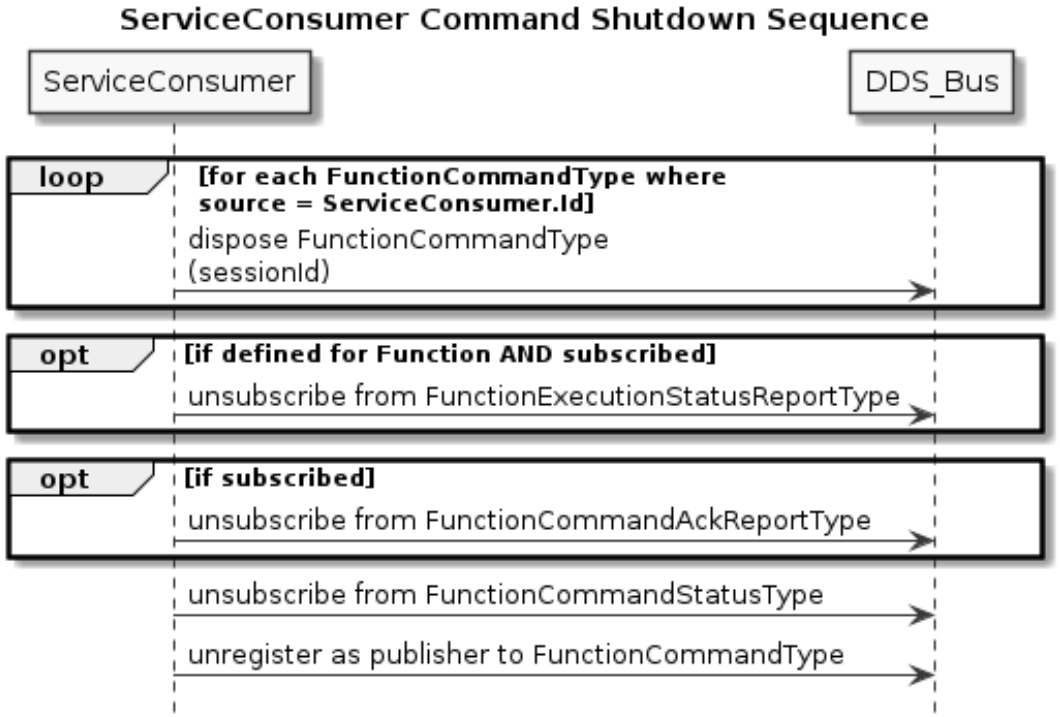

Figure 25: The sequence diagram for command shutdown for Service Consumers.

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.

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. Additionally, 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 external Resource). In all implementations, however, 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

In the case where there is no specific query data (i.e., the service is always just providing the current data to the bus) the sequence of exchanges is show in Figure 26.

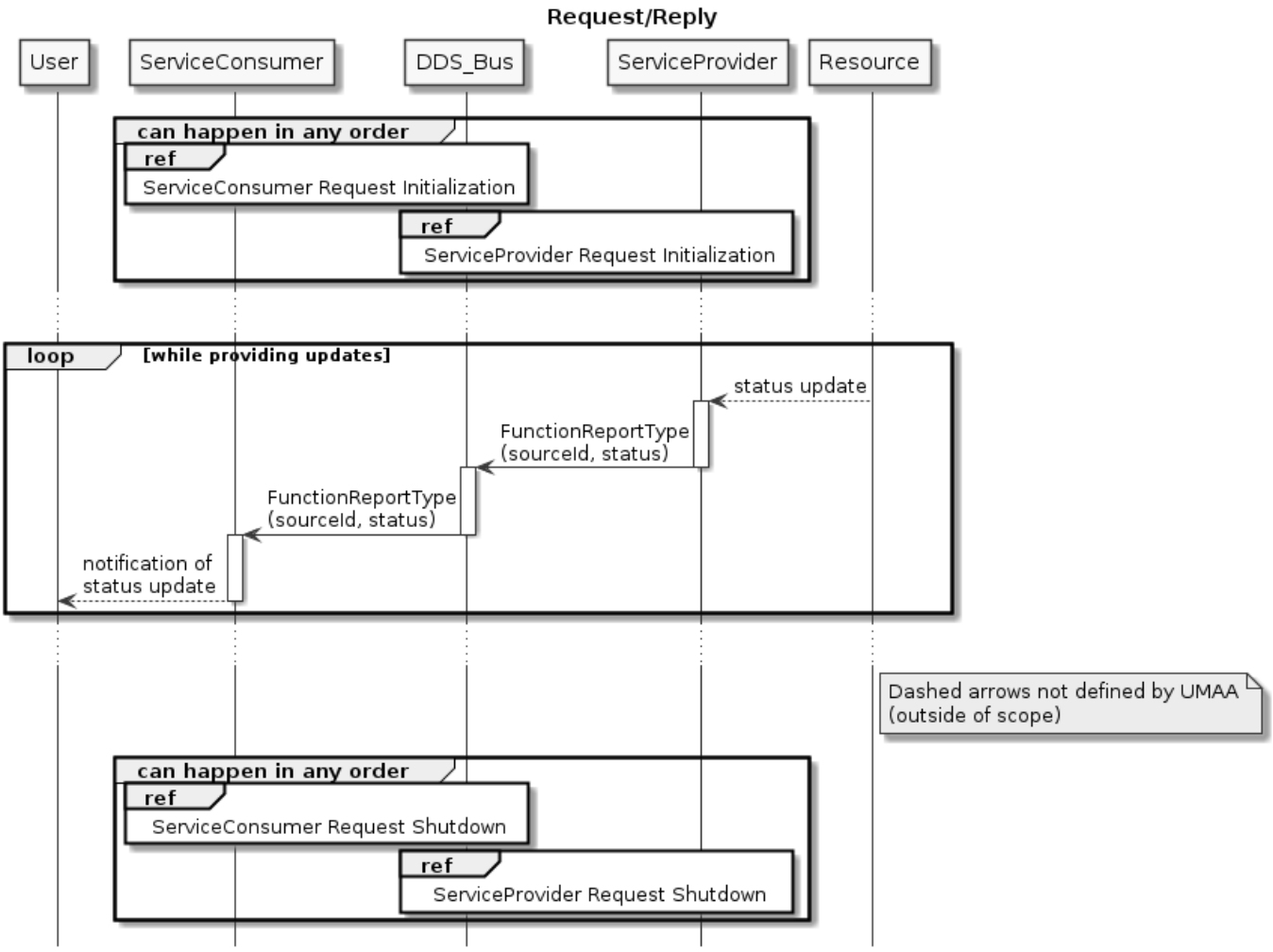

Figure 26: The sequence diagram for a request/reply for report data that does not require any specific query data.

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 27.

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.

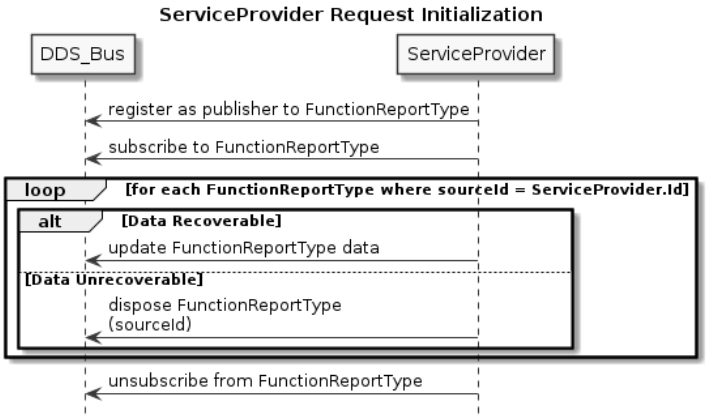

Figure 27: The sequence diagram for initialization of a Service Provider to provide FunctionReportTypes.

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 28.

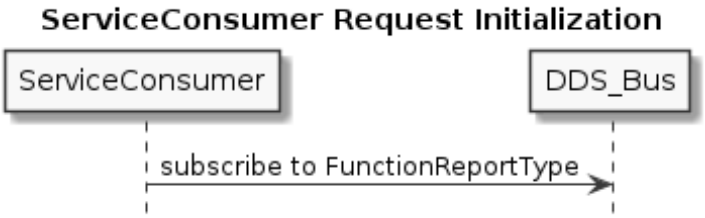

Figure 28: The sequence diagram for initialization of a Service Consumer to request FunctionReportTypes.

5.2.1.3 Service Provider Shutdown To no longer provide `FunctionReportTypes`, the Service Provider disposes the `FunctionReportType` and unregisters as a publisher of the data as shown in Figure 29.

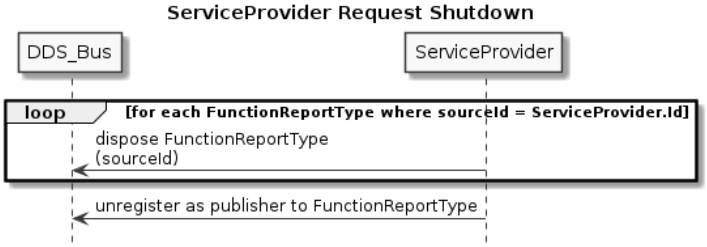

Figure 29: The sequence diagram for shutdown of a Service Provider.

5.2.1.4 Service Consumer Shutdown To no longer request `FunctionReportTypes`, the Service Consumer unsubscribes from `FunctionReportType` as shown in Figure 30.

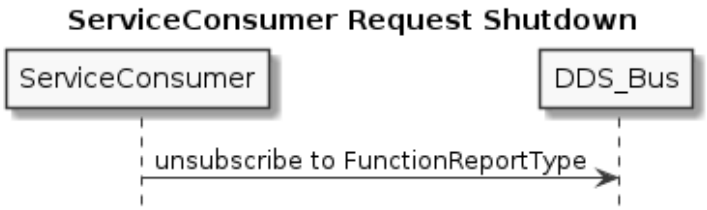

Figure 30: The sequence diagram for shutdown of a Service Consumer.

5.2.2 Request/Reply with Query Data

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

6 Situational Awareness (SA) 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

The following operations are all handled directly by DDS. They are marked in the operations tables with a ⊕.

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 unmanned platform.

Table 6: 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 rate of change in linear velocity and rotational rate of the unmanned platform.

Namespace: UMAA::SA::AccelerationStatus

Topic: AccelerationReport

Data Type: AccelerationReportType

Table 7: AccelerationReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
acceleration	Acceleration3D_PlatformXYZ	The current rate of change in linear velocity of the unmanned platform.
accelerationCovariance	CovarianceNEDAccelerationAccelerationType	The covariance matrix indicating the validity of the acceleration data.
attitudeAcceleration	OrientationAcceleration3D	The current rate of change in rotational rate of the unmanned platform.
rotationalAccelerationCovariance	CovarianceOrientationAccelerationAccelerationType	The covariance matrix of the rotational acceleration data.

6.1.2 CTDStatus

The purpose of this service is to report the current conductivity, temperature, and depth (CTD) in the water column around the unmanned platform.

Table 8: CTDStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryCTD⊕	reportCTD

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

6.1.2.1 reportCTD

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

Namespace: UMAA::SA::CTDStatus

Topic: CTDReport

Data Type: CTDReportType

Table 9: CTDReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
conductivity	Conductivity	The measured conductivity
density†	Density	The measured density
depth	Distance_BSL	The measured depth below sea level
salinity†	Salinity	The measured salinity
soundVelocity†	Speed	The measured velocity of sound

Attribute Name	Attribute Type	Attribute Description
temperature	Temperature	The measured temperature

6.1.3 CompartmentConfig

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

Table 10: 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 ⊕.

6.1.3.1 reportCompartmentConfig

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

Namespace: UMAA::SA::CompartmentConfig

Topic: CompartmentConfigReport

Data Type: CompartmentConfigReportType

Table 11: CompartmentConfigReportType Message Definition

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

6.1.4 CompartmentStatus

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

Table 12: 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 ⊕.

6.1.4.1 reportCompartment

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

Namespace: UMAA::SA::CompartmentStatus

Topic: CompartmentReport

Data Type: CompartmentReportType

Table 13: CompartmentReportType Message Definition

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

6.1.5 ContactCOLREGSClassificationStatus

The purpose of this service is to provide COLREGS classification to existing Contacts (manmade object) to the unmanned vehicles and/or the operator for situational awareness in the operational area. If a contact is dropped from the contact service, it should be removed from this service as well.

Table 14: ContactCOLREGSClassificationStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryContactCOLREGSClassificationStatus⊕	reportContactCOLREGSClassificationStatus

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

6.1.5.1 reportContactCOLREGSClassificationStatus

Description: This operation is a response to retrieve the current contact classification data.

Namespace: UMAA::SA::ContactCOLREGSClassificationStatus

Topic: ContactCOLREGSClassificationStatusReport

Data Type: ContactCOLREGSClassificationStatusReportType

Table 15: ContactCOLREGSClassificationStatusReportType Message Definition

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

Attribute Name	Attribute Type	Attribute Description
confidence	Percent	The confidence in the COLREGS classification of the contact.
colregsClassification*	COLREGSClassificationEnumType	The COLREGS classification of the contact.
contactID*	NumericGUID	an identifier of the contact

6.1.6 ContactStatus

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

Table 16: ContactStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryContact⊕	reportContact
queryContactRemoved⊕	reportContactRemoved

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

6.1.6.1 reportContact

Description: This operation is a response to retrieve the current contact data.

Namespace: UMAA::SA::ContactStatus

Topic: ContactReport

Data Type: ContactReportType

Table 17: ContactReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
contacts→contactsSetID	LargeSet<ContactType>	A list of manmade object structures. This attribute is implemented as a large set, see this section for an explanation. The associated topic is UMAA::SA::ContactStatus::ContactReport_ContactsSet.

6.1.6.2 reportContactRemoved

Description: This operation is used to report the ID of the Contact that was removed.

Namespace: UMAA::SA::ContactStatus

Topic: ContactRemovedReport

Data Type: ContactRemovedReportType

Table 18: ContactRemovedReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
contactID	sequence<NumericGUID>	An identifier of the contact

6.1.7 ContactVisualClassificationStatus

The purpose of this service is to provide visual classification to existing Contacts (manmade object) to the unmanned vehicles and/or the operator for situational awareness in the operational area. If a contact is dropped from the contact service, it should be removed from this service as well.

Table 19: ContactVisualClassificationStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryContactVisualClassificationStatus⊕	reportContactVisualClassificationStatus

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

6.1.7.1 reportContactVisualClassificationStatus

Description: This operation is a response to retrieve the current contact classification data.

Namespace: UMAA::SA::ContactVisualClassificationStatus

Topic: ContactVisualClassificationStatusReport

Data Type: ContactVisualClassificationStatusReportType

Table 20: ContactVisualClassificationStatusReportType 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.
visualClassification*	VisualClassificationEnumType	The visual classification of the contact.

6.1.8 DateTimeStatus

The purpose of this service is to allow participants to access the current vehicle time.

Table 21: DateTimeStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDateTime⊕	reportDateTime

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

6.1.8.1 reportDateTime

Description: This operation is used to report the current vehicle time.

Namespace: UMAA::SA::DateTimeStatus

Topic: DateTimeReport

Data Type: DateTimeReportType

Table 22: DateTimeReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
time	DateTime	The current vehicle time

6.1.9 ECEFPoseStatus

The purpose of this service is to report the current position and orientation of the unmanned platform in earth-centered, earth-fixed reference frame.

Table 23: ECEFPoseStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryECEFPose⊕	reportECEFPose

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

6.1.9.1 reportECEFPose

Description: This operation is used to report the current position and orientation of the unmanned platform in the earth-centered, earth-fixed reference frame.

Namespace: UMAA::SA::ECEFPoseStatus

Topic: ECEFPoseReport

Data Type: ECEFPoseReportType

Table 24: ECEFPoseReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
attitude	Orientation_PlatformECEF	The orientation (roll, pitch, yaw) of the unmanned platform.
xPosition	Distance	The x position of the unmanned platform.
yPosition	Distance	The y position of the unmanned platform.
zPosition	Distance	The z position of the unmanned platform.

6.1.10 FeatureImageStatus

The purpose of this service is to provide feature image related data to the unmanned vehicles and/or the operator for situational awareness in the operational area.

Table 25: FeatureImageStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFeatureImage⊕	reportFeatureImage
queryFeatureImageRemoved⊕	reportFeatureImageRemoved

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

6.1.10.1 reportFeatureImage

Description: This operation is a response that retrieves the current feature image.

Namespace: UMAA::SA::FeatureImageStatus

Topic: FeatureImageReport

Data Type: FeatureImageReportType

Table 26: FeatureImageReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
boundingBox†	BoundingBoxType	The bounding box of the feature image
featureID*	sequence<NumericGUID>	An identifier of the feature
imageID*	sequence<NumericGUID>	An identifier of the image

6.1.10.2 reportFeatureImageRemoved

Description: This operation is a response to handle notification of feature image data disposal.

Namespace: UMAA::SA::FeatureImageStatus

Topic: FeatureImageRemovedReport

Data Type: FeatureImageRemovedReportType

Table 27: FeatureImageRemovedReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
featureImagePair	sequence<FeatureImagePairType>	A pair consisting of a featureID and an imageID

6.1.11 GlobalDeadReckoning

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

Table 28: GlobalDeadReckoning Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalDeadReckoning	reportGlobalDeadReckoningCommandStatus
queryGlobalDeadReckoningCommand⊕	reportGlobalDeadReckoningCommand
cancelGlobalDeadReckoningCommand⊕	reportGlobalDeadReckoningCancelCommandStatus⊕

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

6.1.11.1 reportGlobalDeadReckoningCommand

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

Namespace: UMAA::SA::GlobalDeadReckoning

Topic: GlobalDeadReckoningCommandReport

Data Type: GlobalDeadReckoningCommandReportType

Table 29: GlobalDeadReckoningCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
attitude	Orientation3D_PlatformXYZ	The desired orientation (roll, pitch, yaw) of the unmanned platform.

Attribute Name	Attribute Type	Attribute Description
attitudeCovariance	CovarianceOrientationType	The desired covariance value of the validity of the orientation data.
elevation	ElevationType	Specifies the elevation of the vector.
position	Position2D	The desired position (latitude, longitude) of the unmanned platform
positionCovariance	CovariancePositionPositionType	The desired covariance value of the validity of the position data.

6.1.11.2 reportGlobalDeadReckoningCommandStatus

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

Namespace: UMAA::SA::GlobalDeadReckoning

Topic: GlobalDeadReckoningCommandStatus

Data Type: GlobalDeadReckoningCommandStatusType

Table 30: GlobalDeadReckoningCommandStatusType Message Definition

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

6.1.11.3 setGlobalDeadReckoning

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

Namespace: UMAA::SA::GlobalDeadReckoning

Topic: GlobalDeadReckoningCommand

Data Type: GlobalDeadReckoningCommandType

Table 31: GlobalDeadReckoningCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
attitude	Orientation3D_PlatformXYZ	The desired orientation (roll, pitch, yaw) of the unmanned platform.
attitudeCovariance	CovarianceOrientationType	The desired covariance value of the validity of the orientation data.
elevation	ElevationType	Specifies the elevation of the vector.
position	Position2D	The desired position (latitude, longitude) of the unmanned platform

Attribute Name	Attribute Type	Attribute Description
positionCovariance	CovariancePositionPosition Type	The desired covariance value of the validity of the position data.

6.1.12 GlobalPoseConfig

The purpose of this service is to set and report the current configuration for magnetic variation in the global coordinate system.

Table 32: GlobalPoseConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGlobalPoseConfig⊕	reportGlobalPoseConfig

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

6.1.12.1 reportGlobalPoseConfig

Description: This operation is used to report the geomagnetic variation being used by the unmanned platform.

Namespace: UMAA::SA::GlobalPoseConfig

Topic: GlobalPoseConfigReport

Data Type: GlobalPoseConfigReportType

Table 33: GlobalPoseConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
magneticVariation	MagneticVariation	The location specific magnetic variation.

6.1.13 GlobalPoseStatus

The purpose of this service is to report the current position and orientation of the unmanned platform in global coordinate system.

Table 34: GlobalPoseStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGlobalPose⊕	reportGlobalPose

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

6.1.13.1 reportGlobalPose

Description: This operation is used to report the current position and orientation of the unmanned platform in the global coordinate system.

Namespace: UMAA::SA::GlobalPoseStatus

Topic: GlobalPoseReport

Data Type: GlobalPoseReportType

Table 35: GlobalPoseReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
altitude†	Altitude_MSL	The current altitude mean-sea-level of the unmanned platform.
altitudeAGL†	Altitude_AGL	The current altitude above-ground-level of the unmanned platform.
altitudeASF†	Distance_ASF	The current altitude above-sea-floor of the unmanned platform.
attitude	Orientation3D	The current orientation (roll, pitch, yaw) of the unmanned platform.
attitudeCovariance	CovarianceOrientationType	The current covariance value of the orientation data.
course	Course_TrueNorth	The current optional course angle used for the unmanned platform.
depth†	Distance_BSL	The current depth of the maritime unmanned platform.
navigationSolution	NavigationSolutionEnumType	The desired navigation solution (estimated or measured)
position	Position2D	The current position (latitude, longitude) of the unmanned platform
positionCovariance	CovariancePositionPositionType	The current covariance value of the validity of the position data.

6.1.14 ImageStatus

The purpose of this service is to provide image related data to the unmanned vehicles and/or the operator for situational awareness in the operational area.

Table 36: ImageStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryImage⊕	reportImage
queryImageRemoved⊕	reportImageRemoved

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

6.1.14.1 reportImage

Description: This operation is a response that retrieves the current image.

Namespace: UMAA::SA::ImageStatus

Topic: ImageReport

Data Type: ImageReportType

Table 37: ImageReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
cameraPosition	Position3D_WGS84	The position of the camera, described using latitude, longitude, and elevation
imageName†	StringShortDescription	A description of the image
imageURI	StringShortDescription	A URI for images. At minimum, http, https, and file schemes are permitted. Other permissable schemes may include ftp or data.
timestamp	DateTime	The timestamp of the image
type	ImageFormatEnumType	The type of the image format
worldTransform†	WorldTransformType	An optional struct used to convert image pixel coordinates to real-world map coordinates
imageID*	NumericGUID	An identifier of the image

6.1.14.2 reportImageRemoved

Description: This operation is a response to handle notification of image data disposal.

Namespace: UMAA::SA::ImageStatus

Topic: ImageRemovedReport

Data Type: ImageRemovedReportType

Table 38: ImageRemovedReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
imageID	sequence<NumericGUID>	An identifier of the image

6.1.15 LocalDeadReckoning

The purpose of this service is to command the current position and orientation of the unmanned platform in local coordinate system.

Table 39: LocalDeadReckoning Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalDeadReckoning	reportLocalDeadReckoningCommandStatus
queryLocalDeadReckoningCommand⊕	reportLocalDeadReckoningCommand
cancelLocalDeadReckoningCommand⊕	reportLocalDeadReckoningCancelCommandStatus⊕

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

6.1.15.1 reportLocalDeadReckoningCommand

Description: This operation is used to report the current commanded configuration values of the position and orientation configuration of an unmanned platform in the local coordinate system.

Namespace: UMAA::SA::LocalDeadReckoning

Topic: LocalDeadReckoningCommandReport

Data Type: LocalDeadReckoningCommandReportType

Table 40: LocalDeadReckoningCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
attitude	Orientation3D_PlatformXYZ	The desire orientation (roll, pitch, yaw) of the unmanned platform.
attitudeCovariance	CovarianceOrientationType	The desired covariance value of the validity of the orientation data.
elevation	ElevationType	Specifies the elevation of the vector.
position	Position3D_PlatformXYZ	The desired position (latitude, longitude) of the unmanned platform
xyPositionCovariance	CovariancePositionVelocityType	The desired covariance value of the validity of the position data.
zPositionCovariance	CovariancePositionVelocityType	The desired covariance value of the validity of the position data.

6.1.15.2 reportLocalDeadReckoningCommandStatus

Description: This operation is used to report the status of the position and orientation configuration command.

Namespace: UMAA::SA::LocalDeadReckoning

Topic: LocalDeadReckoningCommandStatus

Data Type: LocalDeadReckoningCommandStatusType

Table 41: LocalDeadReckoningCommandStatusType Message Definition

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

6.1.15.3 setLocalDeadReckoning

Description: This operation is used to configure the position and orientation of an unmanned platform in the local coordinate system. The position is given in latitude, longitude and Z position with optional altitude or depth.

Namespace: UMAA::SA::LocalDeadReckoning

Topic: LocalDeadReckoningCommand

Data Type: LocalDeadReckoningCommandType

Table 42: LocalDeadReckoningCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
attitude	Orientation3D_PlatformXYZ	The desire orientation (roll, pitch, yaw) of the unmanned platform.
attitudeCovariance	CovarianceOrientationType	The desired covariance value of the validity of the orientation data.
elevation	ElevationType	Specifies the elevation of the vector.
position	Position3D_PlatformXYZ	The desired position (latitude, longitude) of the unmanned platform
xyPositionCovariance	CovariancePositionVelocityType	The desired covariance value of the validity of the position data.
zPositionCovariance	CovariancePositionVelocityType	The desired covariance value of the validity of the position data.

6.1.16 LocalPoseStatus

The purpose of this service is to report the current position and orientation of the unmanned platform in local coordinate system.

Table 43: LocalPoseStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryLocalPose⊕	reportLocalPose

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

6.1.16.1 reportLocalPose

Description: This operation is used to report the current position and orientation of the unmanned platform in the local coordinate system.

Namespace: UMAA::SA::LocalPoseStatus

Topic: LocalPoseReport

Data Type: LocalPoseReportType

Table 44: LocalPoseReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
altitude†	Altitude_MSL	The current altitude mean-sea-level of the unmanned platform.
altitudeAGL†	Altitude_AGL	The current altitude above-ground-level of the unmanned platform.
altitudeASF†	Distance_ASF	The current altitude above-sea-floor of the unmanned platform.
attitude	Orientation3D	The current orientation (roll, pitch, yaw) of the unmanned platform.
attitudeRMS	CovarianceOrientationType	The current RMS value of the validity of the orientation data.
depth†	Distance_BSL	The current depth of the unmanned maritime platform.
position†	Position3D_PlatformXYZ	The current position (latitude, longitude) of the unmanned platform
xyPositionCovariance	CovariancePositionVelocityType	The current covariance value of the validity of the position data.
zPositionCovariance	CovariancePositionVelocityType	The current covariance value of the validity of the position data.

6.1.17 NEDVelocityStatus

This service has the responsibility of reporting the instantaneous velocity of the platform in 3D in the North-East-Down reference frame.

Table 45: NEDVelocityStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryNEDVelocity⊕	reportNEDVelocity

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

6.1.17.1 reportNEDVelocity

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

This message is used to provide the receiver the linear velocity and rotational rate of the platform.

Namespace: UMAA::SA::NEDVelocityStatus

Topic: NEDVelocityReport

Data Type: NEDVelocityReportType

Table 46: NEDVelocityReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
attitudeRate†	OrientationVelocity3D	The current rotational rate of the unmanned platform.
attitudeRateRMS†	CovarianceOrientationType	An RMS value indicating the validity of the rotational velocity data.
velocity	Velocity3D_PlatformNED	The current linear velocity of the unmanned platform.
velocityRMS†	CovarianceVelocityVelocityType	An RMS value indicating the validity of the velocity data.

6.1.18 PathReporterSpecs

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

Table 47: 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 ⊕.

6.1.18.1 reportPathReporterSpecs

Description: This operation is used to report the capabilities.

Namespace: UMAA::SA::PathReporterSpecs

Topic: PathReporterSpecsReport

Data Type: PathReporterSpecsReportType

Table 48: PathReporterSpecsReportType Message Definition

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

6.1.19 PathReporterStatus

The purpose of this service is to provide a mechanism for reporting the past and/or future expected path of the unmanned platform. 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 49: 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 ⊕.

6.1.19.1 reportPathReporter

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

Namespace: UMAA::SA::PathReporterStatus

Topic: PathReporterReport

Data Type: PathReporterReportType

Table 50: PathReporterReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
historicalGlobalPaths	sequence<WaypointType>	Historical Global Path
historicalLocalPaths	sequence<WaypointType>	Historical Local Path
plannedGlobalPaths	sequence<WaypointType>	Planned Global Path
plannedLocalPaths	sequence<WaypointType>	Planned Local Path

6.1.20 RangeConfig

The purpose of this service is to configure the proximity sensors of the unmanned platform.

Table 51: RangeConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRangeConfig⊕	reportRangeConfig

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

6.1.20.1 reportRangeConfig

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

Namespace: UMAA::SA::RangeConfig

Topic: RangeConfigReport

Data Type: RangeConfigReportType

Table 52: RangeConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
horizontalFOVStartAngle	RelativeAngle	The current horizontal Field of View (FOV) starting angle in fixed coordinate system +Z axis
horizontalFOVStopAngle	RelativeAngle	The current horizontal Field of View (FOV) stop angle in fixed coordinate system +Z axis.
maxRange	Distance	The current maximum range of data points
minRange	Distance	The current minimum range of data points
rangeError	RangeErrorType	The range error code and description
status	PowerStatusEnumType	The current power status of the sensor
updateRate	Frequency_Hertz	The current update rate of the range (0 to 1,000)
verticalFOVStartAngle	RelativeAngle	The current vertical Field of View (FOV) starting angle in fixed coordinate system +Y axis
verticalFOVStopAngle	RelativeAngle	The current vertical Field of View (FOV) stop angle in fixed coordinate system +Y axis.

6.1.21 RangeSpecs

The purpose of this service is to provide the specifications of the proximity sensors on the unmanned platform.

Table 53: RangeSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRangeSpecs⊕	reportRangeSpecs

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

6.1.21.1 reportRangeSpecs

Description: This operation is used to report the capabilities of the range.

Namespace: UMAA::SA::RangeSpecs

Topic: RangeSpecsReport

Data Type: RangeSpecsReportType

Table 54: RangeSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
coordinateTransformation	BooleanEnumType	The unsupported/supported coordinate transformation
maxHorizontalFOVStopAngle	RelativeAngle	The maximum horizontal Field of View (FOV) stop angle in fixed coordinate system +Z axis
maxRange	Distance	The range of the data points (0 to 1,00,000)
maxUpdateRate	Frequency_Hertz	The data collection rate of the sensor
maxVerticalFOVStopAngle	RelativeAngle	The maximum vertical Field of View (FOV) stop angle in fixed coordinate system +Y axis
minHorizontalFOVStartAngle	RelativeAngle	The minimum horizontal Field of View (FOV) starting angle in fixed coordinate system +Z axis
minRange	Distance	The range of the data points (0 to 1,00,000)
minUpdateRate	Frequency_Hertz	The data collection rate of the sensor
minVerticalFOVStartAngle	RelativeAngle	The minimum vertical Field of View (FOV) starting angle in fixed coordinate system +Y axis
name	StringShortDescription	Name of the range
stateActive	BooleanEnumType	The unsupported/supported active state
stateOff	BooleanEnumType	The unsupported/supported off state
stateStandby	BooleanEnumType	The unsupported/supported standby state
supportedBzip2	BooleanEnumType	The unsupported/supported bzip2 compression
supportedDeflated	BooleanEnumType	The unsupported/supported deflate compression
supportedLZMA	BooleanEnumType	The unsupported/supported LZMA compression
supportedNoCompression	BooleanEnumType	The unsupported/supported data compression

6.1.22 RangeStatus

The purpose of this service is to provide information from the proximity sensors of the unmanned platform. Data can be reported in both a compressed and uncompressed format.

Table 55: RangeStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRange⊕	reportRange

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

6.1.22.1 reportRange

Description: This operation is used to report data of the range.

Namespace: UMAA::SA::RangeStatus

Topic: RangeReport

Data Type: RangeReportType

Table 56: RangeReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
coordinateSystem	CoordinateSystemEnumType	The coordinate system.
dataPoints	sequence<RangeDataPointType>	List of data points collected by range
errorMessage	StringShortDescription	The human-readable error message
rangeErrorCode	RangeErrorCodeEnumType	The error code.

6.1.23 SeaStatus

The purpose of this service is to report the current sea state at the operational area.

Table 57: SeaStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySeaStatus⊕	reportSeaStatus

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

6.1.23.1 reportSeaStatus

Description: This operation is used to report the current sea state from the sea state sensor on board of the unmanned platform.

Namespace: UMAA::SA::SeaStatus

Topic: SeaStatusReport

Data Type: SeaStatusReportType

Table 58: SeaStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	SeaStateEnumType	The WMO Sea State.

6.1.24 SpeedStatus

This service has the responsibility of reporting the instantaneous speed of the platform.

Table 59: SpeedStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySpeed⊕	reportSpeed

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

6.1.24.1 reportSpeed

Description: This operation is used to report the data parameters of the VehicleSpeed service.

This message is used to provide the receiver the speed of the platform.

Namespace: UMAA::SA::SpeedStatus

Topic: SpeedReport

Data Type: SpeedReportType

Table 60: SpeedReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
engineRPM†	EngineSpeed	The engine RPM of the unmanned platform.
speedOverGround†	GroundSpeed	The value of speed over the ground of the unmanned platform.
speedThroughAir†	IndicatedAirspeed	The value of speed through air of the unmanned platform.

Attribute Name	Attribute Type	Attribute Description
speedThroughWater†	Speed_LocalWaterMass	The value of speed through water of the unmanned platform.

6.1.25 StillImageConfig

The purpose of this service is to provide access to the configuration of a camera, allowing the controlling component to set the camera to a particular operational profile.

Table 61: StillImageConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryStillImageConfig⊕	reportStillImageConfig

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

6.1.25.1 reportStillImageConfig

Description: This operation is used to report the configuration of the still image.

Namespace: UMAA::SA::StillImageConfig

Topic: StillImageConfigReport

Data Type: StillImageConfigReportType

Table 62: StillImageConfigReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
frameSize	FrameSizeEnumType	Current frame size configuration of the still image
stillImageError	StillImageErrorType	Still image error codes and descriptions
supportedImageFormat	ImageFormatEnumType	Current image format configuration of the still image

6.1.26 StillImageSpecs

The purpose of this service is to provide access to the camera capabilities.

Table 63: StillImageSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryStillImageSpecs⊕	reportStillImageSpecs

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

6.1.26.1 reportStillImageSpecs

Description: This operation is used to report the capabilities of a still image.

Namespace: UMAA::SA::StillImageSpecs

Topic: StillImageSpecsReport

Data Type: StillImageSpecsReportType

Table 64: StillImageSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
supportedBMP	BooleanEnumType	The unsupported/supported image format
supportedCGA_320x200	BooleanEnumType	The unsupported/supported frame size
supportedCIF_1408x1152	BooleanEnumType	The unsupported/supported frame size
supportedCIF_352x288	BooleanEnumType	The unsupported/supported frame size
supportedCIF_704x576	BooleanEnumType	The unsupported/supported frame size
supportedCR2	BooleanEnumType	The unsupported/supported image format
supportedDNG	BooleanEnumType	The unsupported/supported image format
supportedEGA_640x350	BooleanEnumType	The unsupported/supported frame size
supportedGIF	BooleanEnumType	The unsupported/supported image format
supportedHD1080_1920x1080	BooleanEnumType	The unsupported/supported frame size
supportedHD480_852x480	BooleanEnumType	The unsupported/supported frame size
supportedHD720_1280x720	BooleanEnumType	The unsupported/supported frame size
supportedHSXGA_5120x4096	BooleanEnumType	The unsupported/supported frame size
supportedJPEG	BooleanEnumType	The unsupported/supported image format
supportedNEF	BooleanEnumType	The unsupported/supported image format
supportedPGM	BooleanEnumType	The unsupported/supported image format
supportedPNG	BooleanEnumType	The unsupported/supported image format
supportedPNM	BooleanEnumType	The unsupported/supported image format
supportedPPM	BooleanEnumType	The unsupported/supported image format
supportedQCIF_176x144	BooleanEnumType	The unsupported/supported frame size
supportedQQVGA_160x120	BooleanEnumType	The unsupported/supported frame size
supportedQSXGA_2560x2048	BooleanEnumType	The unsupported/supported frame size
supportedQVGA_320x240	BooleanEnumType	The unsupported/supported frame size
supportedQXGA_2048x1536	BooleanEnumType	The unsupported/supported frame size
supportedSQCIF_128x96	BooleanEnumType	The unsupported/supported frame size

Attribute Name	Attribute Type	Attribute Description
supportedSVGA_800x600	BooleanEnumType	The unsupported/supported frame size
supportedSXGA_1280x1024	BooleanEnumType	The unsupported/supported frame size
supportedTIFF	BooleanEnumType	The unsupported/supported image format
supportedUXGA_1600x1200	BooleanEnumType	The unsupported/supported frame size
supportedVGA_640x480	BooleanEnumType	The unsupported/supported frame size
supportedWHSXGA_6400x4096	BooleanEnumType	The unsupported/supported frame size
supportedWHUXGA_7680x4800	BooleanEnumType	The unsupported/supported frame size
supportedWOXGA_2560x1600	BooleanEnumType	The unsupported/supported frame size
supportedWQSXGA_3200x2048	BooleanEnumType	The unsupported/supported frame size
supportedWQUXGA_3840x2400	BooleanEnumType	The unsupported/supported frame size
supportedWSXGA_1600x1024	BooleanEnumType	The unsupported/supported frame size
supportedWUXGA_1920x1200	BooleanEnumType	The unsupported/supported frame size
supportedWVGA_852x480	BooleanEnumType	The unsupported/supported frame size
supportedWXGA_1366x768	BooleanEnumType	The unsupported/supported frame size
supportedXGA_1024x768	BooleanEnumType	The unsupported/supported frame size

6.1.27 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 65: 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 ⊕.

6.1.27.1 reportStillImage

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

Namespace: UMAA::SA::StillImageStatus

Topic: StillImageReport

Data Type: StillImageReportType

Table 66: StillImageReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
coordinateSystem	BooleanEnumType	Reference coordinate system for image.
imageFrame	ImageFormatEnumType	Current image format of the still image
URI	UniformResourceIdentifier	The current location of the still image

6.1.28 TranslationalShipMotionStatus

This service has the responsibility of reporting the translational motion of the platform.

Table 67: TranslationalShipMotionStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryTranslationalShipMotion⊕	reportTranslationalShipMotion

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

6.1.28.1 reportTranslationalShipMotion

Description: This operation is used to report the the current translational motion of the unmanned platform.

Namespace: UMAA::SA::TranslationalShipMotionStatus

Topic: TranslationalShipMotionReport

Data Type: TranslationalShipMotionReportType

Table 68: TranslationalShipMotionReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
heave†	Down	The linear vertical motion of the ship. Positive heave is down (toward the water bottom); a vessel increasing draft is moving in the positive heave direction.
surge†	Forward	The linear longitudinal motion of the ship. Positive surge is forward of the vessel.
sway†	Left	The linear transverse motion of the ship. Positive sway is to the port side of the vessel.

6.1.29 VehicleNavigationModeStatus

The purpose of this service is to report the actual speed, course, and navigation mode of the unmanned platform.

Table 69: VehicleNavigationModeStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryVehicleNavigationMode⊕	reportVehicleNavigationMode

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

6.1.29.1 reportVehicleNavigationMode

Description: This operation is used to report the current navigation mode of the unmanned platform.

Namespace: UMAA::SA::VehicleNavigationModeStatus

Topic: VehicleNavigationModeReport

Data Type: VehicleNavigationModeReportType

Table 70: VehicleNavigationModeReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
mode	StringShortDescription	The current navigation mode of the unmanned platform.

6.1.30 VelocityStatus

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

Table 71: VelocityStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryVelocity⊕	reportVelocity

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

6.1.30.1 reportVelocity

Description: This operation is used to report the current linear velocity and rotational rate of the unmanned platform.

Namespace: UMAA::SA::VelocityStatus

Topic: VelocityReport

Data Type: VelocityReportType

Table 72: VelocityReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
attitudeRate	OrientationVelocity3D	The current rotational rate of the unmanned platform.
attitudeRateCovariance†	CovarianceOrientationType	The covariance value indicating the validity of the rotational velocity data.
mode†	VehicleSpeedModeEnumType	the speed mode.
velocity	Velocity3D_PlatformNED	The current linear velocity (velocity X, Y, Z) of the unmanned platform.
velocityCovariance†	CovarianceVelocityVelocityType	The covariance value indicating the validity of the velocity data.

6.1.31 WaterCurrentStatus

The purpose of this service is to provide information about the state of the water current

Table 73: WaterCurrentStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryWaterCurrentStatus⊕	reportWaterCurrentStatus

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

6.1.31.1 reportWaterCurrentStatus

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

Namespace: UMAA::SA::WaterCurrentStatus

Topic: WaterCurrentStatusReport

Data Type: WaterCurrentStatusReportType

Table 74: WaterCurrentStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
currentDrift	Speed	The speed of the water current.
currentSet	Heading_TrueNorth_Angle	The heading of the water current relative to true north.

6.1.32 WaterspaceStatus

Defines the current active waterspace zones.

Table 75: WaterspaceStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryWaterspace⊕	reportWaterspace

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

6.1.32.1 reportWaterspace

Description: This operation is used to report the current waterspace zones.

Namespace: UMAA::SA::WaterspaceStatus

Topic: WaterspaceReport

Data Type: WaterspaceReportType

Table 76: WaterspaceReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
zones	sequence<ZoneType>	The zone described by the waterspace.

6.1.33 WeatherStatus

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

Table 77: WeatherStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryWeatherStatus⊕	reportWeatherStatus

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

6.1.33.1 reportWeatherStatus

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

Namespace: UMAA::SA::WeatherStatus

Topic: WeatherStatusReport

Data Type: WeatherStatusReportType

Table 78: WeatherStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
airTemperature†	Temperature	The ambient air temperature (-100 to 100 Celsius)
barometricPressure†	BarometricPressure	A measure of air pressure (600 to 1200 millibars) that correlates with weather and altitude
cloudiness†	CloudCoverType_All	The current state of cloud cover
dewPoint†	Temperature	The temperature (-100 to 100 Celsius) at which water vapor condenses into water
icingSeverity†	WeatherSeverityType_All	The extent of icing present
precipitation†	PrecipitationType_All	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†	Temperature	The water temperature at surface (-5 to 40 Celsius)

6.1.34 WindStatus

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

Table 79: WindStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryWindStatus⊕	reportWindStatus

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

6.1.34.1 reportWindStatus

Description: This operation is used to report the wind state.

Namespace: UMAA::SA::WindStatus

Topic: WindStatusReport

Data Type: WindStatusReportType

Table 80: WindStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
relativeAverageDirection†	RelativeAngle	The average direction from which the wind blows, measured with respect to ship's centerline. The average is taken over some time window defined by the service provider
relativeAverageSpeed†	Speed	The average magnitude of the relative wind. The average is taken over some time window defined by the service provider
relativeInstantaneousDirection†	RelativeAngle	The instantaneous direction from which the wind blows, measured with respect to ship's centerline
relativeInstantaneousSpeed†	Speed	The instantaneous magnitude of the relative wind
relativeMaximumDirection†	RelativeAngle	The maximum direction from which the wind blows, measured with respect to ship's centerline. The maximum is taken over some time window defined by the service provider. See relativeMinimumDirection for additional details.
relativeMaximumSpeed†	Speed	The maximum magnitude of the relative wind. The maximum is taken over some time window defined by the service provider
relativeMinimumDirection†	RelativeAngle	The minimum direction from which the wind blows, measured with respect to ship's centerline. The minimum is taken over some time window defined by the service provider. The overall direction of the wind is contained between the relativeMinimumDirection and relativeMaximumDirection direction always traveling clockwise around the circle. Therefore if the relativeMinimumDirection value is greater than the relativeMaximumDirection value, it is assumed the angle crosses 0 degrees.
relativeMinimumSpeed†	Speed	The minimum magnitude of the relative wind. The minimum is taken over some time window defined by the service provider
straightDeckCrossSpeed†	Speed	The component of the wind normal to the ship. A negative value indicates a port crosswind and a positive value indicates a starboard crosswind
straightDeckHeadSpeed†	Speed	The component of the wind that is parallel to the ship. A negative value indicates a tailwind and a positive value indicates a headwind
trueAverageDirection†	Azimuth_TrueNorth_PosAngle	The average direction from which the wind blows, measured from true north. The average is taken over some time window defined by the service provide
trueAverageSpeed†	GroundSpeed	The average magnitude of the true wind. The average is taken over some time window defined by the service provider
trueInstantaneousDirection†	Azimuth_TrueNorth_PosAngle	The instantaneous direction from which the wind blows, measured from true north
trueInstantaneousSpeed†	GroundSpeed	The instantaneous magnitude of the true wind
trueMaximumDirection†	Azimuth_TrueNorth_PosAngle	The maximum direction from which the wind blows, measured from true north. The maximum is taken over some time window defined by the service provider. See trueMinimumDirection for additional details.

Attribute Name	Attribute Type	Attribute Description
trueMaximumSpeed†	GroundSpeed	The maximum magnitude of the true wind. The maximum is taken over some time window defined by the service provider
trueMinimumDirection†	Azimuth_TrueNorth_PosAngle	The minimum direction from which the wind blows, measured from true north. The minimum is taken over some time window defined by the service provider. The overall direction of the wind is contained between the trueMinimumDirection and trueMaximumDirection direction always traveling clockwise around the circle. Therefore if the trueMinimumDirection value is greater than the trueMaximumDirection value, it is assumed the angle crosses 0 degrees
trueMinimumSpeed†	GroundSpeed	The minimum magnitude of the true wind. The minimum is taken over some time window defined by the service provider

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 81: UCSMDEInterfaceSet Structure Definition

Attribute Name	Attribute Type	Attribute Description
timeStamp	DateTime	The time at which the data was derived.

6.2.2 UMAACommand

Namespace: UMAA::UMAACommand

Description: Defines the common UMAA Command Message Fields.

Table 82: UMAACommand Structure Definition

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

6.2.3 UMAAStatus

Namespace: UMAA::UMAAStatus

Description: Defines the common UMAA Status Message Fields.

Table 83: UMAAStatus Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UCSMDEInterfaceSet		
source*	NumericGUID	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 84: UMAACommandStatusBase Structure Definition

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

6.2.5 UMAACommandStatus

Namespace: UMAA::UMAACommandStatus

Description: Defines the common UMAA Command Status Message Fields.

Table 85: 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::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 86: 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 Acceleration3D_PlatformXYZ

Namespace: UMAA::Common::Measurement::Acceleration3D_PlatformXYZ

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

Table 87: Acceleration3D_PlatformXYZ Structure Definition

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

6.2.8 AltitudeAGLType

Namespace: UMAA::Common::Measurement::AltitudeAGLType

Description: Defines the altitude above ground level

Table 88: AltitudeAGLType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Measurement::ElevationType		
altitudeAGL	Altitude_AGL	specifies the distance above ground level

6.2.9 AltitudeASFType

Namespace: UMAA::Common::Measurement::AltitudeASFType

Description: Defines the altitude above sea floor

Table 89: AltitudeASFType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Measurement::ElevationType		
altitudeASF	Distance_ASF	specifies the distance above sea level

6.2.10 AltitudeMSLType

Namespace: UMAA::Common::Measurement::AltitudeMSLType

Description: Defines the altitude above mean sea level

Table 90: AltitudeMSLType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Measurement::ElevationType		
altitudeMSL	Altitude_MSL	specifies the distance above mean sea level

6.2.11 Altitude_AGL

Namespace: UMAA::Common::Measurement::Altitude_AGL

Description: Altitude_AGL specifies the entity's height above terrain, as reported by a radar system.

Table 91: Altitude_AGL Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	RadarHeight	Specifies the entity's height above terrain, as reported by a radar system.

6.2.12 Altitude_HAE

Namespace: UMAA::Common::Measurement::Altitude_HAE

Description: Altitude_HAE specifies the entity's height above the reference ellipsoid.

Table 92: Altitude_HAE Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	EllipsoidalHeight	Specifies the entityâ€™s height above the reference ellipsoid.

6.2.13 Altitude_MSL

Namespace: UMAA::Common::Measurement::Altitude_MSL

Description: Altitude_MSL specifies the entity's height above the geoid.

Table 93: Altitude_MSL Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude	MSLHeight	Specifies the entity's height above the geoid.

6.2.14 BoundingBoxType

Namespace: UMAA::SA::BoundingBox::BoundingBoxType

Description: This is a base structure used to report the bounds of an image in pixel coordinates.

Table 94: BoundingBoxType Structure Definition

Attribute Name	Attribute Type	Attribute Description
lowerRight_x	Order	The x-coordinate of the lower-right pixel in pixel coordinates
lowerRight_y	Order	The y-coordinate of the lower-right pixel in pixel coordinates
upperLeft_x	Order	The x-coordinate of the upper-left pixel in pixel coordinates
upperLeft_y	Order	The y-coordinate of the upper-left pixel in pixel coordinates

6.2.15 CloudCoverType_All

Namespace: UMAA::Common::Measurement::CloudCoverType_All

Description: Realizes CloudCoverType: the Selector for the CloudCoverEnumType.

Table 95: CloudCoverType_All Structure Definition

Attribute Name	Attribute Type	Attribute Description
domain	sequence<CloudCoverEnumType>	An EnumerationSet which defines the allowable values for the Selector.
setPoint	CloudCoverEnumType	An EnumerationSet which defines the allowable values for the Selector.
value	CloudCoverEnumType	An EnumerationSet which defines the allowable values for the Selector.

6.2.16 ContactType

Namespace: UMAA::SA::ContactStatus::ContactType

Description: This structure is used to report the contact characteristics.

Table 96: ContactType Structure Definition

Attribute Name	Attribute Type	Attribute Description
area	Polygon	area that defines the extent of the structure's surface
callSign†	StringShortDescription	Designator assigned by licensing authority, or name of vessel
contactName	StringShortDescription	The name of the contact
course	Course_TrueNorth	The course of the contact
CPA†	Position2D	An estimated point in which the distance between the platform and contact will reach the minimum value.
CPATime†	DateTime	Time of contact CPA.
diameter	Distance	The diameter of the contact
elevation	Altitude_MSL	elevation of the feature above the structure's surface

Attribute Name	Attribute Type	Attribute Description
elevationError	RMSDistanceErrorType	error in the elevation of the feature above the structure's surface
heading	Heading_TrueNorth_Angle	The heading of the contact
height†	Distance	Height of the contact, intended to be the largest dimension orthogonal to length in the vertical plane
kind	FeatureKindEnumType	the types of manmade object
length†	Distance	The length of the contact, intended to be the longest dimension of the shape
MMSINumber†	MMSI	Nine digit numbers used by maritime digital selective calling (DSC), automatic identification systems (AIS) and certain other equipment to uniquely identify a ship or a coast radio station
originator	StringShortDescription	The name of the source where contact is generated from
position	Position2D	a position of the manmade object
positionCovariance	CovariancePositionPositionType	the margin of error of the manmade object position
SIDC	StringShortDescription	An alphanumeric symbol identification code based on a database structure that provides the minimum elements required to construct the basic icon and/or a complete symbol
source	sequence<NumericGUID>	A list of source contacts.
sourceIndicator	SourceIndicatorEnumType	the data source indicator.
speedOverGround	GroundSpeed	The speed over the ground of the contact.
timeFirstAcquired	DateTime	The time the contact was acquired.
timeLost†	DateTime	The time the contact was lost; if this is not provided it means the track has not been lost.
width†	Distance	Width of the contact, intended to be the largest dimension orthogonal to length in the horizontal plane
featureID*	NumericGUID	an identifier of the manmade object

6.2.17 CovarianceNEDAccelerationAccelerationType

Namespace: UMAA::Common::Measurement::CovarianceNEDAccelerationAccelerationType

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 97: CovarianceNEDAccelerationAccelerationType Structure Definition

Attribute Name	Attribute Type	Attribute Description
adAd	AccelerationScalar	Down-Down acceleration covariance.
aeAd	AccelerationScalar	East-Down acceleration covariance.
aeAe	AccelerationScalar	East-East acceleration covariance.

Attribute Name	Attribute Type	Attribute Description
anAd	AccelerationScalar	North-Down acceleration covariance.
anAe	AccelerationScalar	North-East acceleration covariance.
anAn	AccelerationScalar	North-North acceleration covariance.

6.2.18 CovarianceOrientationAccelerationAccelerationType

Namespace: UMAA::Common::Measurement::CovarianceOrientationAccelerationAccelerationType

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 98: CovarianceOrientationAccelerationAccelerationType Structure Definition

Attribute Name	Attribute Type	Attribute Description
apAp	AngleAcceleration	Pitch-Pitch angle-angle covariance.
apAy	AngleAcceleration	Pitch-Yaw angle-angle covariance.
arAp	AngleAcceleration	Roll-Pitch angle-angle covariance.
arAr	AngleAcceleration	Roll-Roll angle-angle covariance.
arAy	AngleAcceleration	Roll-Yaw angle-angle covariance.
ayAy	AngleAcceleration	Yaw-Yaw angle-angle covariance.

6.2.19 CovarianceOrientationType

Namespace: UMAA::Common::Measurement::CovarianceOrientationType

Description: Realizes ObservableError: 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 99: CovarianceOrientationType Structure Definition

Attribute Name	Attribute Type	Attribute Description
rpRp	CovarOrientation	Row 2 Column 2 term of the result matrix. Pitch-Pitch angle-angle covariance. As radian is unitless result is unitless.
rpRy	CovarOrientation	Row 2 Column 3 term of the result matrix. Pitch-Yaw angle-angle covariance. As radian is unitless result is unitless.

Attribute Name	Attribute Type	Attribute Description
rrRp	CovarOrientation	Row 1 Column 2 term of the result matrix. Roll-Pitch angle-angle covariance. As radian is unitless result is unitless.
rrRr	CovarOrientation	Row 1 Column 1 term of the result matrix. Roll-Roll angle-angle covariance. As radian is unitless result is unitless.
rrRy	CovarOrientation	Row 1 Column 3 term of the result matrix. Roll Yaw angle-angle covariance. As radian is unitless result is unitless.
ryRy	CovarOrientation	Row 3 Column 3 term of the result matrix. Yaw Yaw angle-angle covariance. As radian is unitless result is unitless.

6.2.20 CovariancePolarType

Namespace: UMAA::Common::Measurement::CovariancePolarType

Description: An Entity that specifies a range of uncertainty for the measurement of a point in a polar coordinate system. A point is described by its inclination, bearing, and distance from a reference position.

Table 100: CovariancePolarType Structure Definition

Attribute Name	Attribute Type	Attribute Description
abAb	CovarAngleAngle	Bearing-Bearing angle covariance.
aiAb	CovarAngleAngle	Inclination-Bearing angle covariance.
aiAi	CovarAngleAngle	Inclination-Inclination angle covariance.
arAb	CovarDisAngle	Range-Bearing covariance.
arAi	CovarDisAngle	Range-Inclination covariance.
arAr	CovarDisDis	Range-Range distance covariance.

6.2.21 CovariancePositionPositionType

Namespace: UMAA::Common::Measurement::CovariancePositionPositionType

Description: Realizes ObservableError: 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 101: CovariancePositionPositionType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pdPd†	CovarPosPos	Down-Down position covariance with units of meters squared. Required when the Position includes a down component.

Attribute Name	Attribute Type	Attribute Description
pePd†	CovarPosPos	East-Down position covariance with units of meters squared. Required when the Position includes a down component.
pePe	CovarPosPos	East-East position covariance with units of meters squared.
pnPd†	CovarPosPos	North-Down position covariance with units of meters squared. Required when the Position includes a down component.
pnPe	CovarPosPos	North-East position covariance with units of meters squared.
pnPn	CovarPosPos	North-North position covariance with units of meters squared.

6.2.22 CovariancePositionVelocityType

Namespace: UMAA::Common::Measurement::CovariancePositionVelocityType

Description: Realizes ObservableError: 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 102: CovariancePositionVelocityType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pdVd†	CovarPosVel	Down-Down position-velocity covariance with units of meters squared per second. Required when the velocity includes a down component.
pdVe†	CovarPosVel	Down-East position-velocity covariance with units of meters squared per second.
pdVn†	CovarPosVel	Down-North position-velocity covariance with units of meters squared per second.
peVd†	CovarPosVel	East-Down position-velocity covariance with units of meters squared per second. Required when the Velocity includes a down component.
peVe	CovarPosVel	East-East position-velocity covariance with units of meters squared per second.
peVn	CovarPosVel	East-North position-velocity covariance with units of meters squared per second.
pnVd†	CovarPosVel	North-Down position-velocity covariance with units of meters squared per second. Required when the Velocity includes a down component.
pnVe	CovarPosVel	North-East position-velocity covariance with units of meters squared per second.

Attribute Name	Attribute Type	Attribute Description
pnVn	CovarPosVel	North-North position-velocity covariance with units of meters squared per second.

6.2.23 CovarianceVelocityVelocityType

Namespace: UMAA::Common::Measurement::CovarianceVelocityVelocityType

Description: Realizes ObservableError: 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 103: CovarianceVelocityVelocityType Structure Definition

Attribute Name	Attribute Type	Attribute Description
vdVd†	CovarVelVel	Down-Down velocity-velocity covariance with units of meters squared per second squared.
veVd†	CovarVelVel	East-Down velocity-velocity covariance with units of meters squared per second squared.
veVe	CovarVelVel	East-East velocity-velocity covariance with units of meters squared per second squared.
vnVd†	CovarVelVel	North-Down velocity-velocity covariance with units of meters squared per second squared.
vnVe	CovarVelVel	North-East velocity-velocity covariance with units of meters squared.
vnVn	CovarVelVel	North-North velocity-velocity covariance with units of meters squared per second squared.

6.2.24 DepthType

Namespace: UMAA::Common::Measurement::DepthType

Description: Defines the depth below sea level

Table 104: DepthType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Measurement::ElevationType		
depth	Distance_BSL	specifies the distance below sea level

6.2.25 ElevationType

Namespace: UMAA::Common::Measurement::ElevationType

Description: Union Type. Elevation in either altitude from sea floor or depth from surface (other altitude options support above ground and sea level for potential hybrid vehicles)

Table 105: ElevationType Union(s)

Type Name	Type Description
AltitudeAGLType	Defines the altitude above ground level
AltitudeASFType	Defines the altitude above sea floor
AltitudeMSLType	Defines the altitude above mean sea level
DepthType	Defines the depth below sea level

6.2.26 FeatureImagePairType

Namespace: UMAA::SA::FeatureImagePair::FeatureImagePairType

Description: This is a base structure used to report the feature/image pair being disposed of.

Table 106: FeatureImagePairType Structure Definition

Attribute Name	Attribute Type	Attribute Description
featureID	NumericGUID	An identifier of the feature
imageID	NumericGUID	An identifier of the image

6.2.27 GeodeticLatitude

Namespace: UMAA::Common::Measurement::GeodeticLatitude

Description: GeodeticLatitude specifies the angle between the normal and the equatorial plane of the ellipsoid. The Latitude specifies the north-south position of a point.

Table 107: GeodeticLatitude Structure Definition

Attribute Name	Attribute Type	Attribute Description
latitude	GeodeticLatitude	GeodeticLatitude specifies the angle between the normal and the equatorial plane of the ellipsoid. The Latitude specifies the north-south position of a point.

6.2.28 GeodeticLongitude

Namespace: UMAA::Common::Measurement::GeodeticLongitude

Description: GeodeticLongitude specifies the angular measurement of a location east or west of the prime meridian of the reference ellipsoid.

Table 108: GeodeticLongitude Structure Definition

Attribute Name	Attribute Type	Attribute Description
longitude	GeodeticLongitude	GeodeticLongitude specifies the angular measurement of a location east or west of the prime meridian of the reference ellipsoid.

6.2.29 Orientation3D

Namespace: UMAA::Common::Measurement::Orientation3D

Description: Orientation3D specifies the orientation of the platform in the order yaw, pitch, roll. The angles are given in a locally level, North-East-Down coordinate system centered on the platform.

Table 109: Orientation3D Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchY	Pitch_HalfAngle	pitchY 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.
rollX	Roll_Angle	rollX specifies the platform's rotation about the longitudinal axis (e.g. the axis through the body of an aircraft from tail to nose) in a locally level, North-East-Down coordinate system centered on the platform. Roll is zero when the platform is "wing-tip to wing-tip" level in the North-East plane. The measurement is stated in radians between -pi and pi.
yawZ	Yaw_PosAngle	yawZ specifies the platform's rotation about the vertical axis (e.g. the axis from top to bottom through an aircraft) in a locally level, North-East-Down coordinate system centered on the platform. By this definition, yaw is zero when the platform is oriented toward true North and is equivalent to true North referenced heading. The measurement is stated in radians between -pi and pi.

6.2.30 Orientation3D_PlatformXYZ

Namespace: UMAA::Common::Measurement::Orientation3D_PlatformXYZ

Description: Angular orientation of the platform's XYZ axes relative to the NED reference frame.

Table 110: Orientation3D_PlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchY	Angle	Rotational orientation of the vehicle around its y-axis.
rollX	Angle	Rotational orientation of the vehicle around its x-axis.
yawZ	Angle	Rotational orientation of the vehicle around its z-axis.

6.2.31 OrientationAcceleration3D

Namespace: UMAA::Common::Measurement::OrientationAcceleration3D

Description: OrientationAcceleration3D specifies the acceleration for each axis of an Orientation.

Table 111: OrientationAcceleration3D Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchAccelY	PitchAcceleration	pitchAccelY specifies the acceleration of the platform's rotation about the lateral axis (e.g. the axis parallel to the wings) in a locally level, XYZ coordinate system centered on the platform.
rollAccelX	RollAcceleration	rollAccelX specifies the acceleration of the platform's rotation about the longitudinal axis (e.g. the axis through the body of an aircraft from tail to nose) in a locally level, XYZ coordinate system centered on the platform.
yawAccelZ	YawAcceleration	yawAccelZ specifies the acceleration of the platform's rotation about the vertical axis (e.g. the axis from top to bottom through an aircraft) in a locally level, XYZ coordinate system centered on the platform.

6.2.32 OrientationVelocity3D

Namespace: UMAA::Common::Measurement::OrientationVelocity3D

Description: OrientationVelocity3D specifies the rate of change for each axis of an Orientation.

Table 112: OrientationVelocity3D Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchRateY	PitchRate	pitchRateY specifies the rate of change of the platform's rotation about the lateral axis (e.g. the axis parallel to the wings) in a locally level, XYZ coordinate system centered on the platform.
rollRateX	RollRate	rollRateX specifies the rate of change of the platform's rotation about the longitudinal axis (e.g. the axis through the body of an aircraft from tail to nose) in a locally level, XYZ coordinate system centered on the platform.
yawRateZ	YawRate	yawRateZ specifies the rate of change of the platform's rotation about the vertical axis (e.g. the axis from top to bottom through an aircraft) in a locally level, XYZ coordinate system centered on the platform.

6.2.33 Orientation_PlatformECEF

Namespace: UMAA::Common::Measurement::Orientation_PlatformECEF

Description: Angular orientation of the platform's XYZ axes relative to the ECEF reference frame.

Table 113: Orientation_PlatformECEF Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchY	Angle	Rotational orientation of the vehicle around its y-axis.
rollX	Angle	Rotational orientation of the vehicle around its x-axis.
yawZ	Angle	Rotational orientation of the vehicle around its z-axis.

6.2.34 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 114: 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	Duration_Seconds	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.

6.2.35 PlanningZoneType

Namespace: UMAA::MM::BaseType::PlanningZoneType

Description: This structure is used to describe the operational parameters of a zone.

Table 115: PlanningZoneType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MM::BaseType::ZoneType		
endTime†	DateTime	Defines the end time for the zone. If not defined means active indefinitely.
startTime†	DateTime	Defines the start time for the zone. If not defined means immediately active.
zoneName	StringShortDescription	Defines a name associated with each defined zone.

6.2.36 Polygon

Namespace: UMAA::Common::Measurement::Polygon

Description: Polygon specifies an area defined by a polygon.

Table 116: Polygon Structure Definition

Attribute Name	Attribute Type	Attribute Description
lineKind	LineSegmentEnumType	lineKind indicates the type of lines that form the polygon.
referencePoint	sequence<Position2DTime>	referencePoint specifies the geospatial points defining the vertices of a polygon. Three or more points are needed to define a polygon.

6.2.37 Polygon_Volume

Namespace: UMAA::Common::Measurement::Polygon_Volume

Description: Realizes VolumeType: an entity that describes a quantity of three-dimensional space enclosed by some closed boundary.

Table 117: Polygon_Volume Structure Definition

Attribute Name	Attribute Type	Attribute Description
ceiling	Altitude_MSL	Describes the plane relative to the mean sea level that intersects the highest point or plane of the polygon.
floor	Altitude_MSL	Describes the plane relative to the mean sea level that intersects the lowest point or plane of the polygon.
lineKind	LineSegmentEnumType	Describes the type of line used to represent the polygon.
referencePoints	sequence<Position2D>	Describes a reference point for the polygon.

6.2.38 Position2D

Namespace: UMAA::Common::Measurement::Position2D

Description: Position2D specifies a location on the surface of the Earth.

Table 118: Position2D Structure Definition

Attribute Name	Attribute Type	Attribute Description
geodeticLatitude	GeodeticLatitude	geodeticLatitude specifies the north-south coordinate of the position.
geodeticLongitude	GeodeticLongitude	geodeticLongitude specifies the east-west coordinate of the position.

6.2.39 Position2DTime

Namespace: UMAA::Common::Measurement::Position2DTime

Description: Position2DTime specifies a location on the surface of the Earth at a given point in time.

Table 119: Position2DTime Structure Definition

Attribute Name	Attribute Type	Attribute Description
geodeticPosition	Position2D	geodeticPosition specifies a location on the surface of the Earth.
timeAtPosition†	DateTime	timeAtPosition specifies the date and time when this position is considered valid or an associated measurement was made.

6.2.40 Position3D_PlatformXYZ

Namespace: UMAA::Common::Measurement::Position3D_PlatformXYZ

Description: Position3D_PlatformXYZ specifies a location on a Cartesian coordinate system relative to the origin of the platform.

Table 120: Position3D_PlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
xA	Forward	xA specifies the X-axis position which is in the forward (toward the nose) direction.
yA	Right	yA specifies the Y-axis position which is in the right (starboard) direction.
zA	Down	zA specifies the Z-axis position which is in the down (toward the center of the Earth) direction.

6.2.41 Position3D_WGS84

Namespace: UMAA::Common::Measurement::Position3D_WGS84

Description: Position3D_WGS84 specifies a location relative to the WGS-84 ellipsoid.

Table 121: Position3D_WGS84 Structure Definition

Attribute Name	Attribute Type	Attribute Description
geodeticPosition	Position2D	geodeticPosition specifies a location on the surface of the Earth.
heightAboveEllipsoid	Altitude_HAE	heightAboveEllipsoid specifies the height above the WGS-84 ellipsoid.

6.2.42 PrecipitationType_All

Namespace: UMAA::Common::Measurement::PrecipitationType_All

Description: Realizes PrecipitationType: the Selector for the PrecipitationEnumType.

Table 122: PrecipitationType_All Structure Definition

Attribute Name	Attribute Type	Attribute Description
domain	sequence<PrecipitationEnumType>	An EnumerationSet which defines the allowable values for the Selector.
setPoint	PrecipitationEnumType	An EnumerationSet which defines the allowable values for the Selector.
value	PrecipitationEnumType	An EnumerationSet which defines the allowable values for the Selector.

6.2.43 Quaternion

Namespace: BasicTypes::Quaternion

Description: Defines a four-element vector that can be used to encode any rotation in a 3D coordinate system.

Table 123: Quaternion Structure Definition

Attribute Name	Attribute Type	Attribute Description
a	double	Real number a.
b	double	Real number b.
c	double	Real number c.
d	double	Real number d.

6.2.44 RMSDistanceErrorType

Namespace: UMAA::Common::Measurement::RMSDistanceErrorType

Description: Describes a distance in meters.

Table 124: RMSDistanceErrorType Structure Definition

Attribute Name	Attribute Type	Attribute Description
distanceError	Distance	distance error in meters.

6.2.45 RangeDataPointType

Namespace: UMAA::SA::RangeStatus::RangeDataPointType

Description: This structure is used to report data points identified by a range.

Table 125: RangeDataPointType Structure Definition

Attribute Name	Attribute Type	Attribute Description
bearing	Angle	Bearing from sensor in radians
bearingValidity	BooleanEnumType	True if bearing is valid; false otherwise
dataPointCovariance	CovariancePolarType	Data point error covariance
inclination	Angle	Inclination from sensor in radians
inclinationValidity	BooleanEnumType	True if inclination is valid; false otherwise
range	Distance	Distance from sensor in meters
rangeValidity	BooleanEnumType	True if range is valid; false otherwise
pointID*	NumericGUID	Unique ID of this point

6.2.46 RangeErrorType

Namespace: UMAA::SEM::SensorManagement::RangeErrorType

Description: This structure is used to report error code when the specified range configuration is set to be invalid.

Table 126: RangeErrorType Structure Definition

Attribute Name	Attribute Type	Attribute Description
errorCode	RangeErrorCodeEnumType	Error code reports when the specified range configuration is set to be invalid.
errorMessage	StringShortDescription	A description of the invalid range configuration setting

6.2.47 StillImageErrorType

Namespace: UMAA::SEM::SensorManagement::StillImageErrorType

Description: This structure is used to report error code when the specified still image configuration is set to be invalid.

Table 127: StillImageErrorType Structure Definition

Attribute Name	Attribute Type	Attribute Description
errorCode	StillImageErrorCodeEnumType	Error code reports when the specified still image configuration is set to be invalid.
errorMessage	StringShortDescription	A description of the invalid still image configuration setting

6.2.48 Velocity3D_PlatformNED

Namespace: UMAA::Common::Measurement::Velocity3D_PlatformNED

Description: Velocity3D_PlatformNED specifies the velocity given by northing, easting and down vectors in a North-East-Down coordinate system centered on the platform.

Table 128: Velocity3D_PlatformNED Structure Definition

Attribute Name	Attribute Type	Attribute Description
downSpeed	DownSpeed	downSpeed specifies the down velocity vector in a North-East-Down coordinate system centered on the platform.
eastSpeed	EastSpeed	eastSpeed specifies the easting velocity vector in a North-East-Down coordinate system centered on the platform.
northSpeed	NorthSpeed	northSpeed specifies the northing velocity vector in a North-East-Down coordinate system centered on the platform.

6.2.49 WaypointType

Namespace: UMAA::MM::BaseType::WaypointType

Description: This structure is used to define attributes of a waypoint including position, depth, and speed.

Table 129: WaypointType Structure Definition

Attribute Name	Attribute Type	Attribute Description
altitude†	Altitude_MSL	Specifies the distance along the vehicle path to the waypoint
altitudeAGL†	Altitude_AGL	Specifies the distance above ground level
altitudeASF†	Distance_ASF	Specifies the distance above sea level
attitude†	Orientation3D	includes yaw, pitch, roll that vehicle should assume at the given waypoint
depth†	Distance_BSL	Specifies the distance of the waypoint below sea level
endTime†	DateTime	Specifies the end of the valid time period
globalPosition†	Position2D	Specifies the global location of the waypoint (in latitude/longitude)
localPosition†	Position3D_PlatformXYZ	Specifies the local position of the waypoint (in x, y, z)
mode†	HeightModeEnumType	describe whether the waypoint is an altitude or depth mode
pathTolerance†	Distance	the current tolerance of the path measured by distance
speed†	Speed_LocalWaterMass	Specifies the speed to be maintained traveling to the waypoint
startTime†	DateTime	Specifies the beginning of the valid time period
waypointTolerance†	Distance	the current tolerance of the waypoint measured by distance.
waypointType	WaypointKindEnumType	A type of waypoint
waypointID*	NumericGUID	An unique identification of the waypoint

6.2.50 WeatherSeverityType_All

Namespace: UMAA::Common::Measurement::WeatherSeverityType_All

Description: Realizes WeatherSeverityType: the Selector for the WeatherSeverityEnumType.

Table 130: WeatherSeverityType_All Structure Definition

Attribute Name	Attribute Type	Attribute Description
domain	sequence<WeatherSeverityEnumType>	An EnumerationSet which defines the allowable values for the Selector.
setPoint	WeatherSeverityEnumType	An EnumerationSet which defines the allowable values for the Selector.
value	WeatherSeverityEnumType	An EnumerationSet which defines the allowable values for the Selector.

6.2.51 WorldTransformType

Namespace: UMAA::SA::WorldTransform::WorldTransformType

Description: This is a base structure used to report the image-to-world transform according to the ESRI world file standard.

Table 131: WorldTransformType Structure Definition

Attribute Name	Attribute Type	Attribute Description
pixelSize_x	Distance	The pixel size in the x-direction in map units
pixelSize_y	Distance	The pixel size in the y-direction in map units
rotation_x	Angle	The rotation about the x-axis
rotation_y	Angle	The rotation about the y-axis
upperLeftCoordinate_x	Distance	The x-coordinate of the center of the upper-left pixel of the image, given in the map frame
upperLeftCoordinate_y	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

6.2.52 ZoneType

Namespace: UMAA::MM::BaseType::ZoneType

Description: This structure is used to describe a the operational parameters of a zone.

Table 132: ZoneType Structure Definition

Attribute Name	Attribute Type	Attribute Description
zone	Polygon_Volume	Defines the zone.
zoneKind	ZoneKindEnumType	Defines the type of zone, i.e., keep in, keep out, etc.
zoneID*	NumericGUID	Defines an identifier associated with each defined zone. This zoneID does not have to match the source zoneID from the WaterspacePlan.

Table 133: ZoneType Children

Type Name	Type Description
PlanningZoneType	This structure is used to describe the operational parameters of a zone.

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 CloudCoverEnumType

Namespace: UMAA::Common::OrderedEnumeration::CloudCoverEnumType

Description: CloudCoverEnumTypeLDM is a Realization of CloudCoverEnumType which is a mutually exclusive set of values that defines a discrete set of cloud cover conditions.

Table 134: CloudCoverEnumType Enumeration

Enumeration Value	Description
OVERCAST	Cloud cover is complete (eight-eighths of the sky).
BROKEN	Cloud cover is from five-eighths to seven-eighths of the sky.
FEW	Cloud cover is from one-eighth to two-eights of the sky.
SCATTERED	Cloud cover is from three-eighths to four-eighths of the sky.
CLEAR	There are no clouds below 12,000 ft (zero-eighths of the sky).

6.3.2 COLREGSClassificationEnumType

Namespace: UMAA::Common::MaritimeEnumeration::COLREGSClassificationEnumType

Description: This enumeration is designed to classify a vessel with accordance to COLREGS definitions.

Table 135: COLREGSClassificationEnumType Enumeration

Enumeration Value	Description
POWER_DRIVEN_UNDERWAY	The term power-driven vessel means any vessel propelled by machinery.
SAILING	The term sailing vessel means any vessel under sail provided that propelling machinery, if fitted, is not being used.
FISHING	The term vessel engaged in fishing means any vessel fishing with nets, lines, trawls, or other fishing apparatus which restricts maneuverability, but does not include a vessel fishing with trolling lines or other fishing apparatus which do not restrict maneuverability.
NOT_UNDER_COMMAND	The term vessel not under command means a vessel which, through some exceptional circumstance, is unable to maneuver as required by these Rules and is therefore unable to keep out of the way of another vessel.
RESTRICTED_IN_ABILITY_TO_MANUEVER	The term vessel restricted in her ability to maneuver means a vessel which, from the nature of her work, is restricted in her ability to maneuver as required by these Rules and is therefore unable to keep out of the way of another vessel.
ANCHORED	The vehicle is currently anchored.
CONSTRAINED_BY_DRAUGHT	The vehicle is limited in its navigational capabilities by its draught.
PUSHING	The vessel is engaged in a pushing operation such as severely restricts the pushing vessel and her push in their ability to deviate from their course.
TOWING	The vessel is engaged in a towing operation such as severely restricts the towing vessel and her tow in their ability to deviate from their course.
NON_VESSEL	This value refers to contacts that are determined to be not vessels (buoy, land, etc) where COLREGS does not apply.

6.3.3 CommandStatusReasonEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CommandStatusReasonEnumType

Description: Defines a mutually exclusive set of reasons why a command status state transition has occurred.

Table 136: CommandStatusReasonEnumType Enumeration

Enumeration Value	Description
CANCELED	Indicates a transition to the CANCELED state when the command is canceled successfully.
VALIDATION_FAILED	Indicates a transition to the FAILED state when the command contains missing, out-of-bounds, or otherwise invalid parameters.
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.
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.
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.
RESOURCE_REJECTED	Indicates a transition to the FAILED state when the commanded resource rejects the command for some reason.
INTERRUPTED	Indicates a transition to the FAILED state when the command has been interrupted by a higher priority process.
TIMEOUT	Indicates a transition to the FAILED state when the command is not acknowledged within some defined time bound.
SUCCEEDED	Indicates the conditions to proceed to this state have been met and a normal state transition has occurred.

6.3.4 CoordinateSystemEnumType

Namespace: UMAA::Common::MaritimeEnumeration::CoordinateSystemEnumType

Description: A mutually exclusive set of values that defines the coordinate system of the image data.

Table 137: CoordinateSystemEnumType Enumeration

Enumeration Value	Description
SENSOR_COORDINATE_SYSTEM	Sensor relative coordinate system
VEHICLE_COORDINATE_SYSTEM	Vehicle relative coordinate system

6.3.5 FrameSizeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::FrameSizeEnumType

Description: A mutually exclusive set of values that defines the resolution settings for the digital video frame that can be streamed down from the camera.

Table 138: FrameSizeEnumType Enumeration

Enumeration Value	Description
CGA_320x200	CGA_320x200
CIF_1408x1152	CIF_1408x1152
CIF_352x288	CIF_352x288
CIF_704x576	CIF_704x576
EGA_640x350	EGA_640x350
HD1080_1920x1080	HD1080_1920x1080
HD480_852x480	HD480_852x480
HD720_1280x720	HD720_1280x720
HSXGA_5120x4096	HSXGA_5120x4096
QCIF_176x144	QCIF_176x144
QQVGA_160x120	QQVGA_160x120
QSXGA_2560x2048	QSXGA_2560x2048
QVGA_320x240	QVGA_320x240
QXGA_2048x1536	QXGA_2048x1536
SQCIF_128x96	SQCIF_128x96
SVGA_800x600	SVGA_800x600
SXGA_1280x1024	SXGA_1280x1024
UXGA_1600x1200	UXGA_1600x1200
VGA_640x480	VGA_640x480
WHSXGA_6400x4096	WHSXGA_6400x4096
WHUXGA_7680x4800	WHUXGA_7680x4800
WOXGA_2560x1600	WOXGA_2560x1600
WQSXGA_3200x2048	WQSXGA_3200x2048
WQUXGA_3840x2400	WQUXGA_3840x2400
WSXGA_1600x1024	WSXGA_1600x1024
WUXGA_1920x1200	WUXGA_1920x1200
WVGA_852x480	WVGA_852x480
WXGA_1366x768	WXGA_1366x768
XGA_1024x768	XGA_1024x768

6.3.6 HeightModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::HeightModeEnumType

Description: An enumeration that is used to set which height mode the unmanned vehicle should be operated in under the specified route.

Table 139: HeightModeEnumType Enumeration

Enumeration Value	Description
ALTITUDE	Height value is distance above sea-floor.
DEPTH	Height value is distance below sea surface.

6.3.7 ImageFormatEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ImageFormatEnumType

Description: A mutually exclusive set of values that defines the image formats.

Table 140: ImageFormatEnumType Enumeration

Enumeration Value	Description
BMP	BMP
CR2_RAW	CR2 (Canon RAW)
DNG	DNG (Adobe RAW)
GIF	GIF
JPEG	JPEG (default)
NEF	NEF (Nikon RAW)
PGM	PGM
PNG	PNG
PNM	PNM
PPM	PPM
TIFF	TIFF

6.3.8 LineSegmentEnumType

Namespace: UMAA::Common::Enumeration::LineSegmentEnumType

Description: LineSegmentEnumTypeLDM is a Realization of LineSegmentEnumType which is a mutually exclusive set of values that defines the line segment types used for navigation.

Table 141: 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).

6.3.9 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 142: CommandStatusEnumType Enumeration

Enumeration Value	Description
FAILED	The command has been attempted, but was not successful.
COMPLETED	The command has been completed successfully.
ISSUED	The command has been issued to the resource (typically a sensor or streaming device), but processing has not yet commenced.
COMMANDED	The command has been placed in the resource's command queue but has not yet been accepted.
EXECUTING	The command is being performed by the resource and has not yet been completed.
CANCELED	The command was canceled by the requestor before the command completed successfully.

6.3.10 FeatureKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::FeatureKindEnumType

Description: A mutually exclusive set of values that defines the feature classification types in maritime environment.

Table 143: FeatureKindEnumType Enumeration

Enumeration Value	Description
CONTACT	Contact
OTHER	Other

6.3.11 WaypointKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::WaypointKindEnumType

Description: A mutually exclusive set of values that defines the types of waypoints in the route.

Table 144: WaypointKindEnumType Enumeration

Enumeration Value	Description
APPROACH_FINAL_POINT	Approach final point
APPROACH_INITIAL_POINT	Approach initial point
LAUNCH	Launch
LOITER	Loiter
NAV_ONLY	Navigation only
NAV_TARGET	Navigation target
OTHER	Other
RECOVERY	Recovery
RENDEZVOUS	Rendezvous

6.3.12 NavigationSolutionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::NavigationSolutionEnumType

Description: A mutually exclusive set of values that defines the navigation solution.

Table 145: NavigationSolutionEnumType Enumeration

Enumeration Value	Description
ESTIMATED	Estimated
GROUND_TRUTH	Ground Truth
MEASURED	Measured

6.3.13 PathWayEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PathWayEnumType

Description: A mutually exclusive set of values that defines the different types of path.

Table 146: PathWayEnumType Enumeration

Enumeration Value	Description
HISTORIAL_GLOBAL	Historical global path
HISTORICAL_LOCAL	Historical local path
PLANNED_GLOBAL	Planned global path
PLANNED_LOCAL	Planned local path

6.3.14 PowerStatusEnumType

Namespace: UMAA::Common::Enumeration::PowerStatusEnumType

Description: PowerStatusEnumTypeLDM is a Realization of PowerStatusEnumType which is a mutually exclusive set of values that defines the power state or status of device or subsystem.

Table 147: PowerStatusEnumType Enumeration

Enumeration Value	Description
POWER_OFF	Power for the device is requested/reported to be off.
POWER_ON	Power for the device is requested/reported to be on.
EMERGENCY_POWER	Power for the device is requested/reported to use emergency power.
POWER_STANDBY	Power for the device is requested/reported to use standby power.

6.3.15 PrecipitationEnumType

Namespace: UMAA::Common::Enumeration::PrecipitationEnumType

Description: PrecipitationEnumTypeLDM is a Realization of PrecipitationEnumType which is a mutually exclusive set of values that defines types of precipitation.

Table 148: PrecipitationEnumType Enumeration

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.
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.
SNOW	Precipitation in the form of ice crystals formed directly from the freezing [deposition] of the water vapor in the air.
DRIZZLE	Precipitation that consists of numerous minute droplets of water less than 0.5 mm in diameter that reach the Earth's surface.
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.

6.3.16 RangeErrorCodeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::RangeErrorCodeEnumType

Description: A mutually exclusive set of values that defines the types of error reporting from proximity sensors onboard or offboard of the unmanned platform.

Table 149: RangeErrorCodeEnumType Enumeration

Enumeration Value	Description
INVALID_HORIZONTAL_FOV	Invalid horizontal FOV
INVALID_SENSOR_RANGE	Invalid sensor range
INVALID_SENSOR_STATE	Invalid sensor state
INVALID_UPDATE_RATE	Invalid update rate
INVALID_VERTICAL_FOV	Invalid vertical FOV
MULTIPLE_INVALID_PARAMETERS	Multiple invalid parameters
UNKNOWN_ERROR_FAULT	Unknown error fault
UNKNOWN_SENSOR_ID	Unknown sensor ID

6.3.17 SeaStateEnumType

Namespace: UMAA::Common::OrderedEnumeration::SeaStateEnumType

Description: SeaStateEnumTypeLDM is a Realization of SeaStateEnumType which is a mutually exclusive set of values that defines sea state conditions.

Table 150: SeaStateEnumType Enumeration

Enumeration Value	Description
CALM_GLOSSY	The sea state is calm glassy, which is characterized by wave height of 0 meters and a World Meteorological Organization (WMO) sea state code of 0.
CALM_RIPPLED	The sea state is calm rippled, which is characterized by wave heights of 0 to 0.1 meters and a World Meteorological Organization (WMO) sea state code of 1.
HIGH	The sea state is high, which is characterized by wave heights of 6 to 9 meters and a World Meteorological Organization (WMO) sea state code of 7.
MODERATE	The sea state is moderate, which is characterized by wave heights of 1.25 to 2.5 meters and a World Meteorological Organization (WMO) sea state code of 4.
PHENOMENAL	The sea state is phenomenal, which is characterized by wave heights of over 14 meters and a World Meteorological Organization (WMO) sea state code of 9.
ROUGH	The sea state is rough, which is characterized by wave heights of 2.5 to 4 meters and a World Meteorological Organization (WMO) sea state code of 5.
SLIGHT	The sea state is slight, which is characterized by wave heights of 0.5 to 1.25 meters and a World Meteorological Organization (WMO) sea state code of 3.
SMOOTH	The sea state is smooth, which is characterized by wave heights of 0.1 to 0.5 meters and a World Meteorological Organization (WMO) sea state code of 2.
VERY_HIGH	The sea state is very high, which is characterized by wave heights of 9 to 14 meters and a World Meteorological Organization (WMO) sea state code of 8.
VERY_ROUGH	The sea state is very rough, which is characterized by wave heights of 4 to 6 meters and a World Meteorological Organization (WMO) sea state code of 6.

6.3.18 SourceIndicatorEnumType

Namespace: UMAA::Common::MaritimeEnumeration::SourceIndicatorEnumType

Description: Defines a mutually exclusive set of values for the data source.

Table 151: SourceIndicatorEnumType Enumeration

Enumeration Value	Description
ACTUAL	Source of data is actual sensor data.
GROUND_TRUTH	Source of data is ground truth.
SIMULATED	Source of data is simulated sensor data.
TENTATIVE	Source of data is tentative sensor data.

6.3.19 StillImageErrorCodeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::StillImageErrorCodeEnumType

Description: A mutually exclusive set of values that defines the types of error reporting from the camera.

Table 152: StillImageErrorCodeEnumType Enumeration

Enumeration Value	Description
INVALID_FORMAT	Invalid format
INVALID_FRAME_SIZE	Invalid frame size
MULTIPLE_INVALID_PARAMETERS	Multiple invalid parameters
UNKNOWN_ERROR_FAULT	Unknown error fault
UNKNOWN_SENSOR_ID	Unknown sensor ID

6.3.20 VehicleSpeedModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VehicleSpeedModeEnumType

Description: A mutually exclusive set of values that defines the type of performance speed of the unmanned platform.

Table 153: VehicleSpeedModeEnumType Enumeration

Enumeration Value	Description
LRC	Long-Range Cruise
MEC	Maximum Endurance Cruise
MRC	Maximum Range Cruise
SLOW	Slow speed
VEHICLE_SPECIFIC	Vehicle Specific

6.3.21 VisualClassificationEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VisualClassificationEnumType

Description: Defines a mutually exclusive set of values that defines the visual classification.

Table 154: VisualClassificationEnumType Enumeration

Enumeration Value	Description
AID_TO_NAVIGATION_CHANNEL_MARKER	A channel marker used as an aid to navigation.
MEDIUM_VESSEL_FISHING	A fishing vessel.
AID_TO_NAVIGATION_GENERAL	A general aid to navigation.
SMALL_VESSEL_JET_SKI	A jet ski.
AID_TO_NAVIGATION_LARGE_-BUOY	A large buoy used as an aid to navigation.
LARGE_VESSEL_CARGO	A large cargo vessel.
LARGE_GENERAL_OBSTACLE	A large general obstacle.
LARGE_VESSEL_GENERAL	A large general vessel.
LARGE_VESSEL_MILITARY	A large military vessel.

Enumeration Value	Description
LARGE_VESSEL_PASSENGER	A large passenger vessel.
LARGE_VESSEL_OTHER	A large vessel that does not fit into other LARGE_VESSEL categories.
AID_TO_NAVIGATION_LIGHTHOUSE	A lighthouse used as an aid to navigation.
MEDIUM_VESSEL_GENERAL	A medium general vessel.
MEDIUM_VESSEL_MILITARY	A medium military vessel.
MEDIUM_VESSEL_OTHER	A medium vessel that does not fit into other MEDIUM_VESSEL categories.
SMALL_VESSEL_POWER_BOAT	A power boat.
SAILBOAT	A sailboat.
AID_TO_NAVIGATION_SMALL_BUOY	A small buoy used as an aid to navigation.
SMALL_GENERAL_OBSTACLE	A small general obstacle.
SMALL_VESSEL_GENERAL	A small general vessel.
SMALL_VESSEL_MILITARY	A small military vessel.
SMALL_VESSEL_OTHER	A small vessel that does not fit into other SMALL_VESSEL categories
MEDIUM_VESSEL_TUG	A tug vessel.
MEDIUM_VESSEL_TUG_IN_TOW	A tug vessel towing another vessel.
MEDIUM_VESSEL_YACHT	A yacht.

6.3.22 WeatherSeverityEnumType

Namespace: UMAA::Common::OrderedEnumeration::WeatherSeverityEnumType

Description: WeatherSeverityEnumTypeLDM is a Realization of WeatherSeverityEnumType which is a mutually exclusive set of values that defines classification levels for severe weather conditions.

Table 155: 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.
SEVERE	The adverse weather conditions are severe.
NONE	There is no adverse weather.

6.3.23 ZoneKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ZoneKindEnumType

Description: Defines a mutually exclusive set of zone kinds.

Table 156: ZoneKindEnumType Enumeration

Enumeration Value	Description
KEEP_IN	Defines a zone that the vehicle is required to keep in.
KEEP_OUT	Defines a zone that the vehicle is required to keep out.

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 157: Type Definitions

Type Name	Primitive Type	Range of Values	Description
AccelerationScalar	double	units=MeterPerSecondSquared minInclusive=-1310.68 maxInclusive=1310.68 fractionDigits=3	This type stores acceleration in m/s/s.
Angle	double	fractionDigits=3 maxInclusive=3.141592653589793238462643383279 5 minInclusive=-3.141592653589793126462643383279 5 units=Radian referenceFrame=Counting	Angle 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	fractionDigits=6 maxInclusive=10pi minInclusive=0 units=RadiansPerSecondSquared referenceFrame=PlatformXYZ	Represents the rate of change of angular velocity measured in radians per second squared.
Azimuth_TrueNorth_PosAngle	double	fractionDigits=3 maxInclusive=6.283185307179586364925286766559 minInclusive=0 units=Radian referenceFrame=TrueNorth	Azimuth_TrueNorth_PosAngle specifies the horizontal angular vector from an observer to an entity, relative to true north. The measurement is stated in radians between 0 and 2 pi.
BarometricPressure	double	fractionDigits=3 units=Bar referenceFrame=STP	Realizes PressureType: an Entity used to describe a pressure, defined as follows. Pressure is the weight or force per unit area that is produced when something presses or pushes against something else.
BooleanEnumType	boolean	units=N/A minInclusive=N/A maxInclusive=N/A fractionDigits=N/A length=N/A	BooleanEnumTypeLDM is a Realization of BooleanEnumType which is a mutually exclusive set of values that defines the truth values of logical algebra.
Conductivity	double	fractionDigits=3 maxInclusive=100 minInclusive=0 units=SiemensPerMeter referenceFrame=LocalWaterMass	Represents an object property that describes how well the object conducts electricity.
Count	long	units=N/A minInclusive=-2147483648 maxInclusive=2147483647 fractionDigits=0	Represents a whole (non-fractional) number that can be positive, negative or zero.

Type Name	Primitive Type	Range of Values	Description
Course_TrueNorth	double	fractionDigits=3 maxInclusive=6.283185307179586364925286766559 minInclusive=0 units=Radian referenceFrame=TrueNorth	Course_TrueNorth specifies the direction of the platform's motion relative to true north. The measurement is stated in radians between 0 and 2 pi.
CovarAngleAngle	double	fractionDigits=6 maxInclusive=1000000000 minInclusive=-1000000000	Specifies a dimensionless measure of linear dependence that indicates angle-angle covariance in the Polar coordinate system.
CovarDisAngle	double	fractionDigits=6 maxInclusive=1000000000 minInclusive=-1000000000	Specifies a dimensionless measure of linear dependence that indicates distance-angle covariance in the Polar coordinate system.
CovarDisDis	double	fractionDigits=6 maxInclusive=1000000000 minInclusive=-1000000000	Specifies a dimensionless measure of linear dependence that indicates distance-distance covariance in the Polar coordinate system.
CovarOrientation	double	referenceFrame=Counting	CovarOrientation specifies a dimensionless measure of linear dependence that indicates the 1-sigma covariance of the orientation angle.
CovarPosPos	double	referenceFrame=Counting	CovarPosPos specifies a dimensionless measure of linear dependence that indicates position-position covariance in the NED coordinate system.
CovarPosVel	double	referenceFrame=Counting	CovarPosVel specifies a dimensionless measure of linear dependence that indicates position-velocity covariance in the NED coordinate system.
CovarVelVel	double	referenceFrame=Counting	CovarVelVel specifies a dimensionless measure of linear dependence that indicates velocity-velocity covariance in the NED coordinate system.
DateTimeNanoseconds	long	units=Nanoseconds minInclusive=0 maxInclusive=999999999 fractionDigits=0	number of nanoseconds elapsed within the current second.
DateTimeSeconds	longlong	units=Seconds minInclusive=0 maxInclusive=18446744073709500000 fractionDigits=0	seconds offset from the standard POSIX (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.
Density	double	fractionDigits=3 units=KilogramPerCubicMeter referenceFrame=Counting	Realizes DensityType: an Entity that describes the number of occurrences of a repeating event per unit volume.
Distance	double	units=Meter minInclusive=0 maxInclusive=401056000 fractionDigits=3	This type stores a distance in meters.

Type Name	Primitive Type	Range of Values	Description
Distance_ASF	double	units=Meter minInclusive=0 maxInclusive=401056000 fractionDigits=3	The altitude or distance above the sea floor in meters.
Distance_BSL	double	units=Meter minInclusive=0 maxInclusive=10000 fractionDigits=3	The distance below sea level in meters.
Down	double	axisAbbrev=Z axisDirection=down axisUnit=Meter maximumValue=50000 minimumValue=-50000 rangeMeaning=exact resolution=0.001	The Down axis is used for measuring position and increases in magnitude as values extend toward the center of the Earth. Down measurements are expressed in meters.
DownSpeed	double	axisDirection=down axisUnit=MeterPerSecond maximumValue=-10000 minimumValue=10000 rangeMeaning=exact resolution=0.001	The DownSpeed axis is used for measuring speed and increases in magnitude as speed toward the center of the Earth increases. DownSpeed measurements are expressed in meters per second.
Duration_Seconds	double	units=Seconds minInclusive=0 maxInclusive=37817280 fractionDigits=6	Represents a time duration in seconds.
EastSpeed	double	axisDirection=east axisUnit=MeterPerSecond maximumValue=-10000 minimumValue=10000 rangeMeaning=exact resolution=0.001	The EastSpeed axis is used for measuring speed and increases in magnitude as speed in the easterly direction increases. EastSpeed measurements are expressed in meters per second.
EllipsoidalHeight	double	axisAbbrev=h axisDirection=up axisUnit=Meter maximumValue=700000 minimumValue=-10000 rangeMeaning=exact resolution=0.001	The EllipsoidalHeight axis is used for measuring position and increases in magnitude as values extend away from the center of the Earth. EllipsoidalHeight measurements are expressed in meters.
EngineSpeed	double	units=RevolutionsPerMinute minInclusive=-100000 maxInclusive=100000 fractionDigits=0	This type stores number of occurrences in revolutions per minute (RPM). Negative number is used for reverse RPM.
Forward	double	axisAbbrev=X axisDirection=fore axisUnit=Meter maximumValue=20000000 minimumValue=-20000000 rangeMeaning=exact resolution=0.001	The Forward axis is used for measuring position and increases in magnitude as position extends out the "front" of the reference body. Forward measurements are expressed in meters.
Frequency_Hertz	double	units=Hertz minInclusive=0.0 maxInclusive=10000000000 fractionDigits=6	This type stores Frequency in Hz.

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	The Latitude axis is 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	The Longitude axis is 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	units=MeterPerSecond minInclusive=0 maxInclusive=200 fractionDigits=6	This type stores speed in meters/s.
Heading_TrueNorth_Angle	double	units=Radian referenceFrame=TrueNorth minInclusive=-3.1415926535897932 maxInclusive=3.1415926535897932 fractionDigits=3	Describes heading as a value between -pi and pi with respect to True North.
IndicatedAirspeed	double	fractionDigits=6 units=MeterPerSecond referenceFrame=LocalAirMass	IndicatedAirspeed 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.
Left	double	axisAbbrev=Y axisDirection=port axisUnit=Meter maximumValue=20000000 minimumValue=-20000000 rangeMeaning=exact resolution=0.001	The Left axis is used for measuring position and increases in magnitude as position extends out the "left" of the reference body. Left measurements are expressed in meters.
MagneticVariation	double	fractionDigits=3 maxInclusive=3.1415926535897932 minInclusive=-3.1415926535897932 units=Radian referenceFrame=TrueNorth	MagneticVariation specifies the angle on the horizontal plane between true north and magnetic north (the direction the north end of the compass needle points). The measurement is stated in radians between -pi and pi.
MMSI	string	length=9	Maritime Mobile Service Identity, a series of nine digits which uniquely identifies a station

Type Name	Primitive Type	Range of Values	Description
MSLHeight	double	axisDirection=up axisUnit=Meter maximumValue=700000 minimumValue=-10000 rangeMeaning=exact resolution=0.001	The MSLHeight axis is used for measuring position and increases in magnitude as values extend away from the center of the Earth. MSLHeight measurements are expressed in meters.
NorthSpeed	double	axisDirection=north axisUnit=MeterPerSecond maximumValue=-10000 minimumValue=10000 rangeMeaning=exact resolution=0.001	The NorthSpeed axis is used for measuring speed and increases in magnitude as speed in the northerly direction increases. NorthSpeed measurements are expressed in meters per second.
NumericGUID	octet[16]	units=N/A minInclusive=0 maxInclusive=(2^128)-1 fractionDigits=0	Represents a 128-bit number according to RFC 4122 variant 2
Order	long	units=N/A minInclusive=0 maxInclusive=2147483647 fractionDigits=0	Represents nonnegative integers.
Percent	double	units=Percent minInclusive=0 maxInclusive=1000 fractionDigits=3	Defines a percentage 100% = 100.0. Values greater than 100% are allowed.
Pitch_HalfAngle	double	fractionDigits=3 maxInclusive=1.570796326794896619231321691639 minInclusive=-1.570796326794896619231321691639 units=Radian referenceFrame=PlatformNED	Pitch_HalfAngle 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.
PitchAcceleration	double	fractionDigits=3 maxInclusive=10000 minInclusive=0 units=RadianPerSecondSquared referenceFrame=Counting	PitchAcceleration specifies the acceleration of 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.
PitchRate	double	units=RadianPerSecond minInclusive=0 maxInclusive=32.767 fractionDigits=3 referenceFrame=Counting	PitchRate specifies the rate of change of the platformâ€™s rotation about the lateral axis in a locally level, North-East-Down coordinate system centered on the platform.
RadarHeight	double	axisDirection=up axisUnit=Meter maximumValue=700000 minimumValue=-10000 rangeMeaning=exact resolution=0.001	The RadarHeight axis is used for measuring position and increases in magnitude as values extend away from the center of the Earth. RadarHeight measurements are expressed in meters.

Type Name	Primitive Type	Range of Values	Description
RelativeAngle	double	fractionDigits=3 maxInclusive=3.141592653589793238462643383279 5 minInclusive=-3.141592653589793126462643383279 5 units=Radian referenceFrame=Counting	RelativeAngle specifies the angle between two intersecting rays. The measurement is stated in radians between -pi and pi.
RelativeHumidity	double	units=Percent minInclusive=0 maxInclusive=1000 fractionDigits=3	Defines a percentage 100% = 100.0. Values greater than 100% are allowed.
Right	double	axisAbbrev=Y axisDirection=starboard axisUnit=Meter maximumValue=20000000 minimumValue=-20000000 rangeMeaning=exact resolution=0.001	The Right axis is used for measuring position and increases in magnitude as position extends out the "right" of the reference body. Right measurements are expressed in meters.
Roll_Angle	double	fractionDigits=3 maxInclusive=3.141592653589793238462643383279 5 minInclusive=-3.141592653589793126462643383279 5 units=Radian referenceFrame=PlatformNED	Roll_Angle 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. The measurement is stated in radians between -pi and pi.
RollAcceleration	double	fractionDigits=3 maxInclusive=10000 minInclusive=0 units=RadianPerSecondSquared referenceFrame=Counting	RollAcceleration specifies the acceleration of the 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 platform.
RollRate	double	fractionDigits=3 units=RadianPerSecond referenceFrame=Counting	RollRate specifies the rate of change of the platform's rotation about the longitudinal axis (e.g. the axis through the body of the platform from tail to nose) in a locally level, North-East-Down coordinate system centered on the platform.
Salinity	double	fractionDigits=3 maxInclusive=1,000 minInclusive=0 units=GramsPerKilogram referenceFrame=LocalWaterMass	Represents the concentration of dissolved salts in water etc. measured in grams per kilogram.
Speed	double	units=MeterPerSecond minInclusive=0 maxInclusive=299792458 fractionDigits=6	This type stores speed in meters/s.

Type Name	Primitive Type	Range of Values	Description
Speed_LocalWaterMass	double	units=MeterPerSecond minInclusive=0 maxInclusive=299792458 fractionDigits=6	This type stores speed in meters/s.
StringLongDescription	string	fractionDigits=N/A length=4095 maxExclusive=N/A maxInclusive=N/A minExclusive=N/A minInclusive=N/A units=N/A	Represents a long format description.
StringShortDescription	string	fractionDigits=N/A length=1023 maxExclusive=N/A maxInclusive=N/A minExclusive=N/A minInclusive=N/A units=N/A	Represents a short format description.
Temperature	double	units=Celsius minInclusive=-273 maxInclusive=1000 fractionDigits=3	Represents the degree or intensity of warmness or coldess presence in a substance. Measured in Celsius.
UniformResourceIdentifier	string	fractionDigits=N/A length=2047 maxExclusive=N/A maxInclusive=N/A minExclusive=N/A minInclusive=N/A units=N/A	Represents a Uniform Resource Identifier (URI).
Yaw_PosAngle	double	fractionDigits=3 maxInclusive=6.2831853071795864925286766559 minInclusive=0 units=Radian referenceFrame=PlatformNED	Yaw_PosAngle specifies the platform's rotation about the Z axis of its body axis system (PlatformXYZ) relative to its velocity vector in the X-Y plane of its body axis system. Yaw is positive in a clockwise direction. The measurement is stated in radians between 0 and 2 pi.
YawAcceleration	double	fractionDigits=3 maxInclusive=10000 minInclusive=0 units=RadianPerSecondSquared referenceFrame=Counting	YawAcceleration specifies the acceleration of the platform's rotation about the Z axis of its body axis system (PlatformXYZ) relative to its velocity vector in the X-Y plane of its body axis system. Yaw is positive in a clockwise direction. The measurement is stated in radians per second per second.

Type Name	Primitive Type	Range of Values	Description
YawRate	double	fractionDigits=3 units=RadianPerSecond referenceFrame=Counting	YawRate specifies the rate of change of the platform's rotation about the Z axis of its body axis system (PlatformXYZ) relative to its velocity vector in the X-Y plane of its body axis system. Yaw is positive in a clockwise direction.The measurement is stated in radians per second.

A Appendices

A.1 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
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
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
WMO	World Meteorological Organization